BIODIVERSITY BUILDING BLOCKS FOR POLICY

D5.1 Report of the workflow of the implemented biodiversity indicators

29/08/2025

Author(s): Shawn Dove

Funded by the European Union

Prepared under contract from the European Commission
Grant agreement No. 101059592
EU Horizon Europe Research and Innovation Action

Project acronym:	**B3**
Project full title:	**Biodiversity Building Blocks for policy**
Project duration:	01.03.2023 – 31.08.2026 (42 months)
Project coordinator:	Dr. Quentin Groom, Agentschap Plantentuin Meise (MeiseBG)
Call:	HORIZON-CL6-2021-GOVERNANCE-01
Deliverable title:	Report of the workflow of the implemented biodiversity indicators
Deliverable n°:	D5.1
WP responsible:	WP5
Nature of the deliverable:	Report
Dissemination level:	Public
Lead partner:	JLUG
Recommended citation:	Dove, S. (2025). ***Report of the workflow of the implemented biodiversity indicators***. B3 project deliverable D5.1.
Due date of deliverable:	Month 30
Actual submission date:	Month 30

Deliverable status:

Version	Status	Date	Author(s)
1.0	Final	29 August 2025	Shawn Dove, JLU

Table of contents

Key takeaway messages

- The `b3gbi` R package provides standardised, automated, reproducible workflows for biodiversity indicator calculation.
- It computes a range of spatial and temporal biodiversity indicators from GBIF data cubes.
- Robust uncertainty estimation is integrated for many indicators.
- Indicator results are visualised in customisable maps and time series.

Executive summary

This report details the implementation workflow of key biodiversity indicators within the **`b3gbi`** R package, developed as part of the EU-funded **B3 project**. The B3 project aims to improve biodiversity monitoring by integrating diverse biodiversity data, including from citizen science initiatives, into standardised **data cubes**. This innovative approach underpins the development of automated workflows and tools, such as `b3gbi`, which calculate robust biodiversity indicators and models to support evidence-based policy making and address critical data gaps for global and European conservation efforts.

Biodiversity indicators are fundamental to effective conservation, policy assessment, and understanding ecological trends, serving various stakeholders from governments and NGOs to researchers and consultancies. Despite their critical role, their calculation is often hindered by the fragmented, incomplete, and biased nature of available biodiversity data.

The B3 project directly addresses this challenge by providing standardised, publicly accessible, and reproducible workflows for calculating general biodiversity indicators and models from data cubes. The `b3gbi` R package generalizes these methodologies to encompass a diverse set of common biodiversity indicators applicable across any taxon and/or region.

This document outlines the operational workflow of the implemented indicators, detailing the processes from GBIF occurrence cube ingestion to the generation of robust indicator values and output of maps and time series. It describes the data preparation, specific calculation logic for each indicator, and output formats.

The `b3gbi` package facilitates the transformation of complex biodiversity data into actionable information, thereby strengthening the science-policy interface and contributing to a more comprehensive understanding and effective management of global biodiversity.

Non-technical summary

The `b3gbi` R package is a computational tool developed to provide robust and consistent insights into global biodiversity trends. It addresses the challenge of analysing vast amounts of species occurrence data, such as those compiled by the Global Biodiversity Information Facility (GBIF), in an efficient and consistent manner.

The package's workflow begins by utilizing structured, pre-processed biodiversity data, often organized as 'data cubes' that efficiently store information across different locations and time periods. From these data cubes, `b3gbi` automatically computes various biodiversity indicators – these are standardised measures that describe aspects like species richness, rarity, or how diversity changes over time. A critical part of its output is the quantification of uncertainty, which helps users understand the reliability of these calculated measures. All results are then presented through clear and customizable maps and time series visualisations.

Developed through the EU-funded B3 project, `b3gbi` aims to enhance the accessibility, reproducibility, and trustworthiness of biodiversity data analysis. This, in turn, supports informed decision-making in conservation, environmental monitoring, and ecological research, contributing to more effective strategies for protecting biodiversity.

List of abbreviations

B3	Biodiversity Building Blocks for Policy
b3gbi	B3 General Biodiversity Indicators
CBD	Convention on Biological Diversity
CI	Confidence Interval
CRS	Coordinate Reference System
EEA	European Environment Agency
EQDGC	Extended Quarter-Degree Grid Cells
EU	European Union
GBIF	Global Biodiversity Information Facility
MGRS	Military Grid Reference System
RAM	Random Access Memory
ts	Time Series

1. Introduction

Biodiversity indicators play a crucial role in monitoring biodiversity trends, assessing progress towards conservation goals, and informing policy decisions (Buckland et al., 2012; Jones et al., 2011; Nicholson et al., 2012). International frameworks, such as the Convention on Biological Diversity (CBD) Global Biodiversity Framework and the EU Biodiversity Strategy for 2030, rely on such metrics to measure progress (Buchanan et al., 2020; Viti et al., 2024). The vast availability of biodiversity occurrence data, particularly from platforms like the Global Biodiversity Information Facility (GBIF), offers immense potential for deriving these critical indicators. However, transforming this extensive and often heterogeneous data into standardised, reproducible, and robust indicators presents significant challenges due to missing or inadequate data, inherent biases, and a lack of standardisation (Chandler et al., 2017; Proença et al., 2017; Troia and McManamay, 2016; Troudet et al., 2017; Turak et al., 2017). Developing efficient, scalable, standardised, accessible, and reproducible workflows is therefore paramount.

The `b3gbi` R package was developed to directly address these multifaceted challenges. As part of the EU-funded B3 project (Biodiversity Building Blocks for Policy), its core objective is to provide automated, standardised, and reproducible open-source workflows for biodiversity indicator calculation. `b3gbi` operates on pre-processed, aggregated biodiversity data cubes (freely available from GBIF), an efficient and structured format for storing biodiversity information (Desmet et al., 2023). `b3gbi` implements automated workflows to compute a diverse range of spatial and temporal biodiversity indicators, including measures like species richness, rarity, and turnover. It includes integrated uncertainty estimation using robust bootstrapping methods. Complementing its analytical capabilities, `b3gbi` also includes flexible visualisation tools that generate customisable maps and time series plots, making complex biodiversity trends accessible and interpretable for diverse audiences.

This report details the operational workflow of the biodiversity indicators (Fig. 1) implemented within the `b3gbi` R package. It outlines the step-by-step process, including the initial ingestion of GBIF occurrence cubes, the subsequent data transformation, the integration of map data for geographical contextualization, the computational logic applied to calculate indicators and confidence intervals, and the ultimate generation of maps or time series. By elucidating these workflows, this document serves to clarify the methodology underpinning `b3gbi`, contributing to its transparency and reproducibility as a tool for robust biodiversity assessment.

2. Technical Environment and Setup

The `b3gbi` R package is designed to operate within a standard **R version 3.5.0 or later** environment. It relies on a comprehensive set of dependencies (see Tables A1 and A2 in Annex for a complete list), automatically managed upon installation, to ensure its functionality for processing GBIF occurrence cubes and calculating biodiversity indicators.

Computational resource needs naturally vary with data size and indicator complexity. If using `b3gbi` to process large occurrence cubes, a powerful system with a lot of RAM may help to make processing times manageable, especially for the calculation of bootstrapped confidence intervals.

The `b3gbi` package can be easily installed from the dedicated B3 R-universe repository by executing the following command in an R console:

```
install.packages("b3gbi", repos = c('<https://b-cubed-eu.r-universe.dev>',
'<https://cloud.r-project.org>'))
```

For users interested in accessing the latest development features, the package can also be installed directly from its GitHub repository using the remotes package:

```
# install.packages("remotes")
remotes::install_github("b-cubed-eu/b3gbi")
```

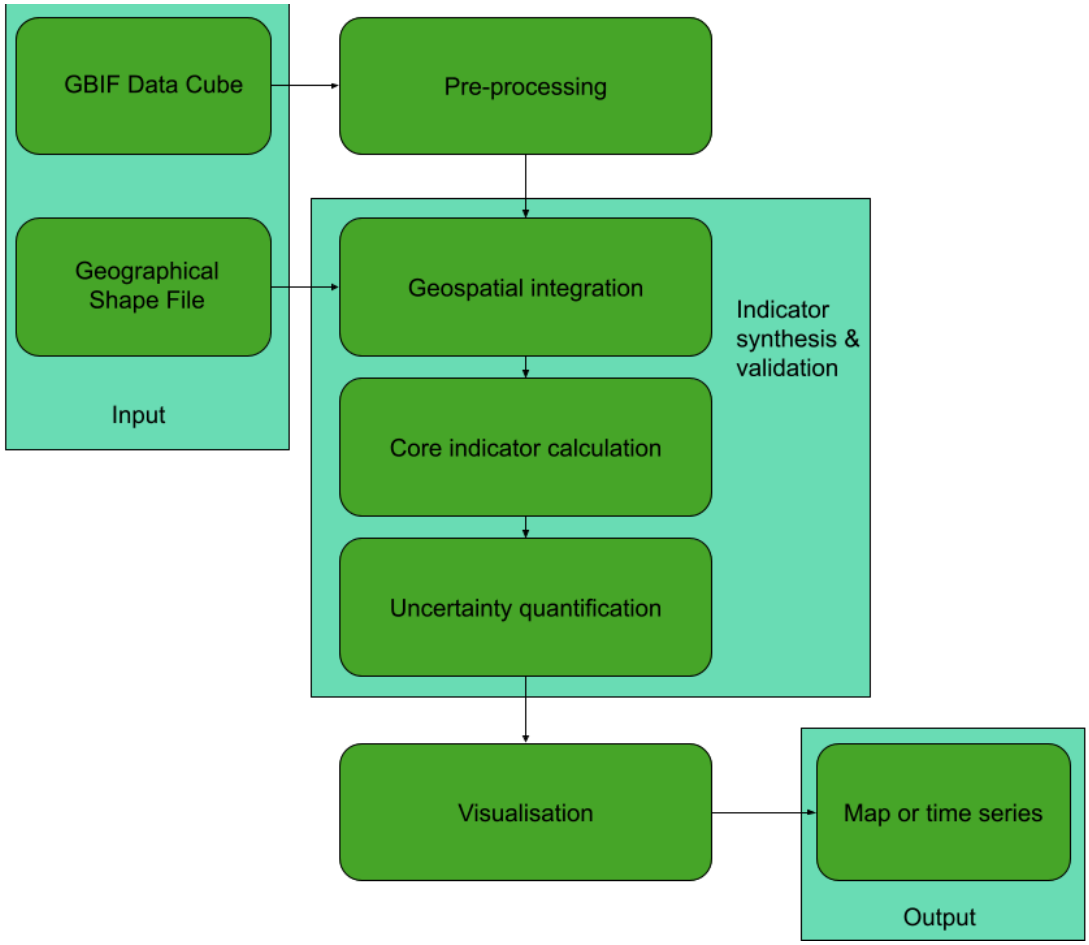

Figure 1. Overview of `b3gbi` workflow

3. Overview

From a user's perspective, the `b3gbi` workflow simplifies complex biodiversity data analysis into three main steps: **Pre-processing**, **Indicator Synthesis and Validation**, and **Visualisation** (Fig. 1).

In the initial **Pre-processing** step, the user invokes the `process_cube()` function (Table 1) on a data cube (provided as either a `.csv` or `data.table`). This function thoroughly validates the input, checks for data integrity, transforms column structures (e.g., renaming, grid code translation), and outputs a robust, package-specific data cube object. This refined object includes essential metadata and is consistently structured, ready for subsequent `b3gbi` operations.

The second step, **Indicator Synthesis and Validation**, begins when the user calls a specific indicator wrapper function, such as `obs_richness_map()` or `obs_richness_ts()`. These wrappers pass user preferences to `compute_indicator_workflow()` (Table 1), which acts as the central orchestrator for the entire indicator calculation process. This robust function manages all necessary sub-processes: **Geospatial Integration** (e.g., temporal filtering to user-chosen years, spatial selection from the cube or integration of geographical shapefiles), **Core Indicator Calculation** (dispatching to functions like `calc_map()` or `calc_ts()` (Table 1) to compute metrics), and Uncertainty Quantification (using `calc_ci()` (Table 1) to estimate confidence intervals). The output is a comprehensive, package-specific indicator object containing the calculated metric, aggregated as requested (by grid cell for mapping or by year for time series), complete with associated uncertainty values, metadata, and all information required for subsequent visualisation.

Finally, the third step, **Visualisation**, allows users to easily interpret their results. By calling the generic `plot()` function, `b3gbi` leverages R's S3 dispatch system to automatically direct the indicator object to the appropriate internal plotting method based on its type (e.g., `plot_map()` or `plot_ts()` (Table 1)). This function performs all necessary post-processing to generate a publication-ready map or time series plot (see Figs. 2-4 in section 14.4 in the Annex for examples). Users can further customize these plots using `ggplot2` functionalities and export them as desired.

Table 1: Overview of `b3gbi` main functions

Function Name	Role in Workflow	Description
process_cube()	Pre-processing	Data cube import and validation, temporal filtering, column renaming, grid code translation, structuring of object for package use
compute_indicator_workflow()	Indicator synthesis and validation	Orchestrates indicator calculation workflow, managing metadata, temporal filtering, geospatial integration, indicator calculation, and uncertainty quantification
calc_map(), calc_ts()	Core indicator calculation	Calculates specified indicator values across spatial grid cells or over time, leveraging S3 dispatch for different indicator types

calc_ci()	Uncertainty quantification	Calculates confidence intervals for specified indicators, leveraging S3 dispatch for different indicator types
plot_map(), plot_ts(), plot_species_map(), plot_species_ts()	Visualisation	Generate maps or time series plots of calculated indicators

4. Data Acquisition: GBIF Occurrence Cubes

The `b3gbi` R package is specifically designed to operate on **GBIF occurrence cubes**, which serve as its primary input data structure. These cubes represent temporally and spatially aggregated biodiversity occurrence records, assigned to a user-selected international grid system at a specified resolution, thus providing a standardised format for downstream analytical workflows. `b3gbi` currently supports three of the four available grid systems (Table 2) and all available resolutions. Note, however, that processing will be significantly slower at high resolutions.

Table 2: Supported grid systems for GBIF data cubes

Grid System	Supported	Available Resolutions
European Environment Agency (EEA) Reference Grid	Yes	100 km 10 km 1 km 250 m 100 m 25 m
Extended Quarter Degree Grid Cells (EQDGC)	Yes	Level 0: 1 degree Level 1: 0.5 degrees Level 2: 0.25 degrees Level 3: 0.125 degrees Level 4: 0.0625 degrees Level 5: 0.03125 degrees Level 6: 0.015625 degrees
Military Grid Reference System (MGRS)	Yes	6° x 8° (only grid-zone designation) 100 km 10 km 1 km 100 m 10 m 1 m
Inverse Snyder Equal-Area Aperture 3 Hexagon (ISEA3H) Discrete Global Grid System	No	n/a

4.1. Input Data Structure and Formats

`b3gbi` expects data cubes as either:

- a **file path** to a delimited text file (.csv)
- an **existing R** `data.frame` **or** `tibble` **object**.

Regardless of the input source, the data cube must contain a set of **essential columns** (occurrences, dates, scientific names, species keys) for indicator calculation. `b3gbi` internally standardises column names and validates that required fields are present to ensure consistency across its functions.

4.2. Data Ingestion and Pre-processing Workflow

The central function for ingesting and preparing GBIF occurrence cubes is `process_cube()`. This function performs a series of critical, automated pre-processing steps to ensure the data is correctly formatted, validated, and ready for indicator calculation.

Its workflow involves:

- **Input Validation and Standardization:** Ensuring data is in a usable format, standardizing column names, and validating the presence and data types of essential fields.
- **Temporal Data Handling:** Automating year extraction from date fields and allowing for flexible temporal filtering.
- **Grid System Detection and Spatial Coordinate Extraction:** Interpreting supported grid reference systems to extract spatial coordinates, crucial for spatial analysis and visualisation. Custom or no-grid options are also supported (e.g., for temporal analysis of simulated outputs from `gcube` (Langeraert, 2025)).
- **Data Cleaning and Structuring:** Removing invalid or duplicate records and organizing the data for consistent downstream processing.
- **Output Object Creation:** Encapsulating the processed and validated data into a custom S3 object (`sim_cube` or `processed_cube`) tailored for `b3gbi`'s subsequent workflow steps.

5. Workflow for Biodiversity Indicator Calculation

The calculation of biodiversity indicators within `b3gbi` is orchestrated through a streamlined and flexible workflow, primarily managed by the core function `compute_indicator_workflow()`. This function acts as a central hub, handling data preparation, spatial processing, indicator computation, and the final structuring of the output. It intelligently adapts its processing steps based on whether a spatial (map-based) or temporal (time-series) indicator is requested, and

whether the input is observed data from a `processed_cube` object or simulated data from a `sim_cube` object.

The workflow can be broken down into the following key stages:

5.1. Data Ingestion and Initial Filtering

The process begins by receiving a `processed_cube` (for observed data) or `sim_cube` (for simulated data) object.

- **Time-based Filtering:** The input data is initially filtered to include only occurrences within the specified `first_year` and `last_year` ranges, allowing users to focus analyses on specific temporal periods (the fields are optional, and temporal filtering can alternatively be done when importing the data with `process_cube()`).
- **Input Validation:** Robust checks ensure the input data cube is of the correct class, contains actual data, and includes the necessary '`obs`' (occurrences) column, preventing downstream errors.
- **Metadata Collection:** Key characteristics of the input data, such as the total number of species, years covered, and unique species names, are collected and stored for inclusion in the final indicator object.

5.2. Coordinate Reference System (CRS) Management

`b3gbi` automates crucial CRS determination and management. The function intelligently detects the input CRS (e.g., from "eea", "eqdgc", "mgrs" grid types) and allows for user-specified output CRSs. Robust checks ensure unit compatibility, preventing errors and advising users on necessary conversions.

5.3. Spatial and Temporal Processing Pathways

The workflow intelligently adapts based on the requested indicator dimension:

- **Spatial Pathway (`dim_type = "map"`):** For map-based indicators, b3gbi performs comprehensive spatial data preparation. This includes transforming raw coordinates into spatial geometries, generating a uniform analysis grid at a user-specified cell size (occurrences can be aggregated to a lower resolution than the input data cube), and **contextualizing data within user-defined geographic boundaries.** This contextualization can leverage geographic data from the `rnaturalearth` package (e.g., country borders, lakes, etc.) and allows the integration of **external shapefiles**. This enables highly targeted spatial analysis within specific areas of interest, such as **protected areas or RAMSAR sites**, by filtering indicator calculations to these defined regions. This process ultimately links occurrence records to grid cells for spatial aggregation.
- **Temporal Pathway (`dim_type = "ts"`):** For time-series indicators, the focus shifts to direct temporal and spatial filtering of occurrence points within specified geographic areas, without grid creation. Similar to the spatial pathway, **external shapefiles can be**

utilized here to restrict the time-series analysis to specific areas of interest, allowing for trend analysis within precise boundaries.

- For `sim_cube` objects (which contain simulated data lacking inherent spatial grid information), the workflow bypasses grid creation and detailed spatial intersection. `sim_cube` inputs are only supported via the temporal pathway.

5.4. Indicator Calculation

This is the core computational step where the actual biodiversity metric is derived. `b3gbi` leverages R's **S3 object-oriented system** to ensure modularity and extensibility. Based on the requested indicator and dimension (spatial or temporal), the system dynamically dispatches to the appropriate method.

5.5. Confidence Interval Generation

For many indicators, `b3gbi` provides robust uncertainty estimates. Where supported, **bootstrapped confidence intervals** are generated. Users can configure the number of bootstrap iterations and the specific type of interval, ensuring flexibility in uncertainty quantification. The system issues warnings if confidence intervals are requested for unsupported indicators.

5.6. Output Object Construction

The final stage packages the calculated results into structured and informative S3 objects: `indicator_map` for spatial outputs and `indicator_ts` for time-series results. These objects are also assigned classes specific to the indicator and encapsulate the indicator values, relevant metadata, and spatial geometries (for maps), designed for easy subsequent analysis and visualisation. This comprehensive workflow ensures consistent and reliable calculation of diverse biodiversity indicators with appropriate spatial and temporal contexts.

6. Output Structure and Classes

The `compute_indicator_workflow()` function in `b3gbi` is designed to produce highly structured and self-describing output objects, implemented as R's S3 classes. These objects encapsulate both the calculated indicator values and crucial metadata about the analysis, ensuring results are readily interpretable, easily accessible for further programmatic manipulation, and prepared for direct visualisation.

`b3gbi` generates two primary S3 output classes, corresponding to the two main types of indicator analysis:

- **`indicator_map`** Class: Used when `dim_type = "map"`, this class is designed for spatial biodiversity indicators. Its core data container is an `sf` (Simple Features) data frame, providing geometric information for spatial grid cells alongside computed indicator values and associated confidence intervals (if calculated).
- **`indicator_ts`** Class: Used when `dim_type = "ts"`, this class is designed for time-series biodiversity indicators. Its primary data is a `tibble` (or `data.frame`)

containing temporal identifiers (e.g., `year`) and the computed indicator values, including confidence intervals where applicable.

Both `indicator_map` and `indicator_ts` objects are enriched with comprehensive metadata attributes that provide essential context about the indicator, spatial/temporal scope, and original data characteristics.

6.1. Common Features and Benefits

These S3 objects offer significant benefits due to their design:

- **S3 Object System:** Their S3 nature allows for the definition of generic methods (e.g., `print()` and `plot()`), ensuring consistent behaviour and ease of interaction for users.
- **Self-Describing:** By bundling both the data and comprehensive metadata, these objects are self-contained and provide all necessary information about how the indicator was calculated and what it represents, without requiring external lookups.
- **Ready for Analysis:** The underlying `sf` data frame and `tibble` are standard R data structures, making them fully accessible and interoperable with the rich ecosystem of R packages for data analysis and visualisation.

This robust output structure ensures that the results of `b3gbi` calculations are not just numbers, but actionable data products ready for scientific interpretation and communication.

7. General Principles of Indicator Calculation

While `b3gbi` is capable of computing a diverse array of biodiversity indicators, a set of fundamental principles underpins how all these metrics are derived from the structured data within `processed_cube` (or `sim_cube`) objects. These principles ensure consistency, efficiency, and adaptability across various analytical scenarios, whether spatial or temporal.

7.1. Operation on Pre-Aggregated, Filtered and Prepared Data

A fundamental aspect of `b3gbi`'s workflow is that indicator calculations operate on data that has already been aggregated to specific spatial and temporal units (e.g., GBIF data cubes or simulated outputs from the `gcube` package (Langeraert, 2025)). The `compute_indicator_workflow()` function primarily performs filtering and preparation steps on this pre-aggregated input (as detailed in Section 5, Workflow for Biodiversity Indicator Calculation). The subsequent indicator calculations then involve performing computations on these already prepared units, deriving values for each grid cell in spatial indicators or for each year in temporal indicators.

7.2. Handling of Data Sparsity and Missing Values

During calculations on the pre-aggregated data, it is common for some spatial units (grid cells) or temporal units (years) to have few or no associated records. The calculation methods inherently manage such scenarios:

- Units with sufficient data will yield a calculated indicator value.

- Units with no applicable data (e.g., an empty grid cell within the filtered extent) or insufficient data for a valid calculation will typically result in a NA (Not Available) value for that indicator in that unit, or may be implicitly excluded, depending on the indicator's definition. This allows users to easily identify areas or periods with insufficient sampling.

7.3. Separation of Core Value and Uncertainty Estimation

The primary value of each indicator is computed as a distinct step. Estimation of uncertainty using bootstrapped confidence intervals is handled as a subsequent, optional process by the `calc_ci()` function. This modularity allows for the core indicator to be calculated efficiently, with uncertainty analysis added only when specifically requested and supported for that indicator.

8. Specific Biodiversity Indicators and Variables

This section provides information on the individual biodiversity indicators and variables that can be calculated using `b3gbi` (Table 3). A brief definition of what each indicator or variable measures is provided, along with some context for its interpretation. For more detailed technical information on how each indicator or variable is calculated, see section 14 (Annex).

Table 3: Overview of implemented biodiversity indicators and variables

Indicator or variable name	Type	Source	Reference
Total occurrences	Variable	GBIF	
Density of occurrences (occ/km^2)	Variable	GBIF	
Mean year of occurrences (newness)	Variable	GBIF	
Species richness	Indicator	GBIF	Magurran, 1988
Evenness	Indicator	GBIF	Pielou, 1966; Kvålseth, 2015
Rarity	Indicator	GBIF	Rabinowitz, 1981
Hill-Shannon diversity	Indicator	GBIF	Hill, 1973
Hill-Simpson diversity	Indicator	GBIF	Hill, 1973
Species occurrences	EBV	GBIF	Pereira et al., 2013
Species range	Variable	GBIF	
Taxonomic distinctness	EBV	GBIF	Clarke & Warwick, 1999
Species turnover	Indicator	GBIF	Jaccard, 1901

8.1. Observed Species Richness

8.1.1. Definition & Application

This fundamental indicator quantifies the total number of unique species detected within a given spatial unit (e.g., grid cell, for `indicator_map`) or temporal period (e.g., `year`, for `indicator_ts`). It provides a straightforward measure of detected biodiversity.

8.1.2. Key Interpretation Note

As with any richness metric, interpretation must carefully consider **sampling effort and completeness**. Higher observed richness can reflect increased survey effort rather than true ecological differences, or limitations in species detectability. It is best understood as "recorded richness."

8.2. Total Occurrences

8.2.1. Definition & Application

This variable quantifies the overall number of species occurrence records within a spatial unit or temporal period. While not a biodiversity indicator itself, it is crucial for understanding data comprehensiveness and distribution. It is presented as a map showing data density or a time series of annual record counts.

8.2.2. Key Interpretation Note

Total occurrences serve as a vital **contextual tool for interpreting other biodiversity indicators**. For instance, a high observed richness coupled with low total occurrences might suggest sparse but diverse sampling, highlighting potential data biases influenced by sampling effort rather than ecological reality.

8.3. Evenness

8.3.1. Definition & Application

Evenness measures how **uniformly individuals (or observations in GBIF data) are distributed among species** within a given area or over time. It complements species richness by providing insight into community structure. `b3gbi` supports **Pielou's Evenness** and **Williams' Evenness**, displaying their values spatially on maps (`indicator_map`) or as trends over time (`indicator_ts`).

8.3.2. Key Interpretation Note

High evenness suggests species are more equally represented in the dataset, while low evenness indicates dominance by a few species. It's crucial to remember that b3gbi's calculation is based on **proportions of observations**, which may not perfectly reflect true ecological evenness or individual abundance.

8.4. Rarity

8.4.1. Definition & Application

Rarity quantifies the scarcity or infrequency of species, and when summed over multiple species, serves as a crucial biodiversity indicator for conservation. `b3gbi` offers two distinct measures: Abundance-Based Rarity (based on species' proportional occurrences) and Area-Based Rarity (based on species' spatial occupancy). These can be mapped (`indicator_map`) to identify areas with a higher presence of rare species, or tracked over time (`indicator_ts`) to observe changes in overall rarity.

8.4.2. Key Interpretation Note

High rarity values highlight areas or periods with a greater presence of species that are either locally scarce or geographically restricted. This indicator is invaluable for pinpointing vulnerable

communities or species in conservation efforts, and changes in rarity can signal environmental shifts or population declines.

8.5. Estimated Hill Diversity

8.5.1. Definition & Application

Hill Diversity provides a **unified framework for various diversity measures**, allowing for different emphases on rare versus common species through a single parameter, q. `b3gbi` calculates three common forms: **q=0 (approximates Species Richness, weighing all species equally), q=1 (emphasizes common species, like Hill-Shannon), and q=2 (emphasizes very common species, like Hill-Simpson)**. These indicators represent the "effective number of species" and can be mapped (`indicator_map`) or tracked over time (`indicator_ts`) to provide multi-faceted insights into biodiversity patterns.

8.5.2. Key Interpretation Note

By varying `q`, users can gain different perspectives on diversity, from emphasizing rare species (lower `q`) to common ones (higher `q`). `b3gbi` utilizes **coverage-based estimation** to mitigate the effects of sample size and sampling biases, providing more robust comparisons across samples with varying completeness.

8.6. Cumulative Species Richness

8.6.1. Definition & Application

This indicator tracks the **total number of unique species observed from the beginning of a specified time period up to a given year**. It provides an estimation of how many new species are still being recorded over time within a region, helping to evaluate sampling effort and assess the overall recorded biodiversity over the study duration. This is an **inherently temporal indicator**, presented as a time series (`indicator_ts`).

8.6.2. Key Interpretation Note

A steadily increasing curve suggests ongoing species discovery or improved sampling; a flattening curve might indicate that most species have already been recorded. This indicator is highly dependent on sampling effort and **does not account for species loss**.

8.7. Mean Year of Occurrence (Newness)

8.7.1. Definition & Application

This variable calculates the **average year of occurrence for all records** within a given spatial unit (e.g., grid cell, for `indicator_map`) or temporal unit (e.g., year, for `indicator_ts`), providing an estimation of the relative recency of observations. Maps can highlight areas with more recent average records, while time series show the average observation date over cumulative data.

8.7.2. Key Interpretation Note

A recent mean year does not automatically imply higher data quality or ecological change; it primarily reflects **temporal bias or shifts in data collection activity** (e.g., new citizen science initiatives). It's valuable for understanding the temporal context of available data.

8.8. Occurrence Density

8.8.1. Definition & Application

Occurrence Density measures the **spatial concentration of records by calculating the total number of occurrences per square kilometre**. This allows for more meaningful comparisons between spatial units of different sizes. It can be displayed as a map (`indicator_map`) illustrating areas with higher concentrations of records, or as a time series (`indicator_ts`) showing changes in the rate of recording over time for the study area.

8.8.2. Key Interpretation Note

Higher occurrence density values suggest **more intensive or thorough survey efforts**, making this indicator particularly useful for assessing sampling intensity across a landscape and identifying well-sampled versus under sampled areas for future data collection planning.

8.9. Species Occurrences

8.9.1. Definition & Application

This variable provides the **number of occurrences for individual species**, focusing on their observed frequency and distribution. It can serve as a proxy for relative abundance for species with similar detectability. It visualizes the observed geographical distribution of a specific species on a map (`indicator_map`) or tracks its annual record counts over time (`indicator_ts`).

8.9.2. Key Interpretation Note

As an Essential Biodiversity Variable (EBV), this variable is fundamental for species-specific conservation assessments. It directly shows where and when a species has been recorded, offering insights into distribution patterns and potential changes in observed range or population proxy, but always representing the *observed* rather than true distribution.

8.10. Species Range

8.10.1. Definition & Application

Species Range refers to the **observed geographical extent of individual species**, depicted by the grid cells they occupy. This indicator primarily visualizes the distribution of specific species based on occurrence data, showing occupied cells on a map (`indicator_map`). Temporally (`indicator_ts`), it tracks how the **number of occupied grid cells for a species changes over time**, indicating range expansion, contraction, or stability.

8.10.2. Key Interpretation Note

This variable provides a direct depiction of observed presence, crucial for species distribution modelling and conservation planning. However, it represents the *observed* range, which may be a subset of the true ecological range due to incomplete sampling.

8.11. Taxonomic Distinctness

8.11.1. Definition & Application

Taxonomic Distinctness measures the **average taxonomic distance between any two randomly chosen species** in a sample, incorporating their phylogenetic or taxonomic relatedness. A higher value indicates a broader representation of evolutionary history within the

community. It can highlight areas on a map (`indicator_map`) or show trends over time (`indicator_ts`) where species are, on average, more distantly related.

8.11.2. Key Interpretation Note

This EBV complements richness and evenness by reflecting the evolutionary breadth of a community, which is important for ecosystem resilience. Its calculation relies on accurate and complete taxonomic information, and interpretation should consider the quality of the underlying data.

8.12. Species Turnover

8.12.1. Definition

Species Turnover measures the **rate at which species composition changes over time** within a community, quantifying the balance between species "gains" and "losses" between consecutive time intervals. High turnover indicates a dynamic community with frequent species replacement, while low turnover suggests stability. This is an **exclusively temporal indicator**, presented as a time series (`indicator_ts`).

8.12.2. Key Interpretation Note

High turnover can signal environmental instability or rapid ecological changes. A key limitation is its reliance on complete species lists for each time step; incomplete sampling can lead to misinterpretations. Interpretation should consider factors like sampling effort and habitat dynamics.

9. Bootstrapping and Uncertainty Estimation

Robust biodiversity indicators require not only point estimates but also measures of their uncertainty. The `b3gbi` package incorporates bootstrapping to calculate confidence intervals, providing a range within which the true indicator value is likely to fall. It is intended to outsource this functionality to the `dubicube` package (another product of B3; Langeraert & Van Daele, 2025) in the near future, but for now this process is managed by the internal `calc_ci()` generic function and its methods.

9.1. General Approach to Confidence Interval Calculation

The `calc_ci()` function acts as a dispatcher, calling specific methods tailored to each biodiversity indicator. It is automatically invoked when calculating a biodiversity indicator over time unless explicitly disabled.

The confidence interval logic performs several key steps:

1. **Calculates Confidence Intervals:** It derives lower and upper confidence limits from the bootstrap results, with the type of interval being user-configurable.
2. **Handles Non-Negative Values:** For indicators that cannot be less than zero (e.g., rarity), any calculated lower confidence limits that fall below zero are automatically converted to zero.

3. **Merges Results:** The calculated confidence intervals are integrated with the original indicator values by year.
4. **Error Handling:** Warnings are issued if there's insufficient data to calculate confidence intervals for a given unit.

9.2. Bootstrapping Parameters

Two key parameters control the bootstrapping process:

- **`num_bootstrap`**: This integer specifies the number of bootstrap replicates to generate (default: 1000). A higher number generally leads to more stable confidence intervals but increases computation time.
- **`ci_type`**: Specifies the type of bootstrap confidence interval to calculate (e.g., "`norm`" for normal approximation). Setting it to "`none`" will skip CI calculation.

9.3. Indicator-Specific Confidence Interval Considerations

While the general confidence interval calculation framework is consistent, the specific process adapts to the nature of each indicator:

- **Indicators with CIs calculated via iNEXT:** For **Estimated Hill Diversity** (including its forms representing Hill-derived Species Richness (q=0), Shannon-Hill (q=1), and Simpson-Hill (q=2)), confidence intervals are determined externally by the iNEXT package (Hsieh et al., 2016) during their initial calculation, leveraging its robust coverage-based estimation methods.
- **Indicators without Bootstrapped CIs:** Confidence intervals are **not calculated** for the following indicators within `b3gbi`'s bootstrapping framework, as their underlying metrics or the discrete nature of species presence/absence data make standard resampling of individual observations unsuitable for reliable interval estimation:
    - **Observed Species Richness:** This simple count of unique species does not utilize `b3gbi`'s internal bootstrapping for uncertainty estimation. This is because bootstrapping by resampling individual species can lead to replicates with fewer unique species than originally observed. This inherent limitation means the bootstrap cannot produce a richness value higher than the observed count, causing the **upper confidence intervals to be systematically underestimated.**
    - **Cumulative Species Richness:** This is an inherently temporal and sequential indicator for which CIs are not generated, as its nature as an accumulating count over time is not well-suited to standard bootstrapping methods for uncertainty.
    - **Species Turnover:** Confidence intervals are **not calculated** for this indicator. The metric's reliance on **comparing the unique species lists between consecutive time steps** means that bootstrapping individual observations within each time step would introduce artificial variability in these lists. This makes the resulting gains and losses highly unstable and would lead to **unreliable or biased confidence intervals** for the turnover metric.

    - **Taxonomic Distinctness:** Confidence intervals are **not calculated** for this indicator. Due to its high sensitivity to the exact species composition and taxonomic relationships within a sample, bootstrapping individual occurrences can introduce significant artificial variability, leading to unreliable or biased confidence intervals.

- **Other Indicators with Bootstrapped CIs:** For the following indicators, `b3gbi` applies its internal bootstrapping techniques, which involve resampling individual occurrence records (or their associated properties) for each unit (year or cell) and then applying a specific statistical function to each bootstrap replicate. The aggregated bootstrap results are then used to derive the confidence intervals:
    - **Total Occurrences**
    - **Occurrence Density**
    - **Mean Year of Occurrence (Newness)**
    - **Species Occurrences**
    - **Species Range**
    - **Evenness & Rarity:** While `b3gbi` calculates confidence intervals for these indicators, users should interpret them with caution. These metrics are particularly sensitive to the observed species list and their relative proportions. Bootstrapping individual occurrences may introduce variability that doesn't fully reflect the true uncertainty, potentially leading to less reliable confidence intervals, especially in communities with many rare species or with imperfect sampling.

10. Mapping and Visualisation

`b3gbi` provides robust and flexible visualisation capabilities, enabling users to easily interpret and communicate the calculated biodiversity indicators. Leveraging the `ggplot2` package for plot generation, `b3gbi` ensures high-quality graphics that can be further customized. The plotting system is built around four primary functions, each designed for a specific visualisation task, and supports both general biodiversity indicators and species-specific metrics.

10.1. Core Plotting Functions

The `b3gbi` plotting workflow is managed by four main functions, which are called by indicator-specific `plot()` S3 methods (e.g., `plot.obs_richness_map()`) dispatched by the generic `plot()` function.

10.1.1. `plot_map()`: Mapping General Biodiversity Indicators

This function is designed to create geographical maps of `indicator_map` objects, displaying the spatial distribution of biodiversity metrics.

- **Input:** An `indicator_map` object.
- **Key Features:**

- **Data Visualisation:** Grid cells are plotted, with fill colour mapped to indicator values.
- **Coordinate System Handling:** Automatically uses the CRS from the `indicator_map` object and sets plot limits (`xlim, ylim`) based either on the data's coordinate range or optional user input.
- **Geographical Context:** Can optionally include surrounding land areas for better geographical context. Background colours for land and oceans are customizable.
- **Map Cropping:** Offers options to crop maps, including `Europe_crop_EEA` (to exclude far-lying islands from the European continent when using the EPSG:3035 projection) and `crop_to_grid` (to strictly limit the map extent to the calculated grid).
- **Scale Transformations:** Supports various transformations (e.g. '`log`', '`boxcox`', '`modulus`', '`yj`') for the colour fill gradient, useful for visualizing skewed data.
- **Customization:** Provides extensive parameters for customizing titles, legend labels, breaks, colours, and axis limits.

- **Output:** A `ggplot` object, which can be further modified if desired using standard `ggplot2` functions.

10.1.2. `plot_ts()`: Plotting General Biodiversity Indicator Trends

This function is used to create time series plots for `indicator_ts` objects, visualizing how a biodiversity metric changes over time.

- **Input:** An `indicator_ts` object.
- **Key Features:**
  - **Data Visualisation:** Plots indicator values against `year` using either points or lines.
  - **Temporal Filtering:** Allows users to define minimum and maximum years to focus on specific time ranges.
  - **Smoothed Trendlines:** Can overlay a smoothed trendline to highlight overall temporal patterns.
  - **Uncertainty Visualisation:** If confidence intervals are present, they can be displayed as either error bars or a filled ribbon around the main trend. It also supports smoothed confidence envelopes around the loess trendline.
  - **Aesthetic Controls:** Offers broad control over line colours, transparencies (`alpha`), point sizes, line widths, and general plot themes.
  - **Dynamic Error Bar Width:** Error bar width is automatically scaled based on the number of years plotted, ensuring consistent visual appearance regardless of the time range.

- **Output:** A `ggplot` object for flexible customization.

10.1.3. `plot_species_map()`: Mapping Individual Species Occurrences/Ranges

This function is specifically designed for visualizing the spatial patterns of one or more individual species' occurrences or ranges.

- **Input:** An `indicator_map` object that contains species-specific data (e.g., derived from `spec_occ_map()` or `spec_range_map()`). Users must specify which species (by `taxonKey` or `scientificName`) they wish to plot.
- **Key Features:**
    - **Species Filtering:** Filters the input `indicator_map` data to include only the specified species, handling both numeric `taxonKeys` and partial `scientificName` matching.
    - **Multi-Species Plotting:** If multiple species are selected, their individual maps are combined into a single multi-panel plot by default. Users can also opt to plot each species separately.
    - **Contextual Background:** Plots the full `indicator_map` grid (excluding species-specific data) as a grey background layer, providing geographical context for the species' distribution.
    - **Customization:** Inherits many customization options from `plot_map()`, including title, legend, axis limits, and surrounding land features. It also allows suppressing the legend.
- **Output:** A single `ggplot` object (if combined) or a list of ggplot objects (if plotted separately).

10.1.4. `plot_species_ts()`: Plotting Individual Species Trendlines

This function is specialized for generating time series plots of individual species' occurrences or range sizes over time.

- **Input:** An `indicator_ts` object that contains species-specific time series data (e.g., from `spec_occ_ts()` or `spec_range_ts()`). Users must specify which species to plot.
- **Key Features:**
    - **Species Filtering:** Filters the input `indicator_ts` data to focus on the selected species.
    - **Multi-Species Plotting:** Similar to `plot_species_map()`, it combines multiple species' time series into a single multi-panel by default, or can plot them separately if specified.
    - **Trendlines and Uncertainty:** Incorporates options for smoothed trendlines and visualizing confidence intervals (error bars or ribbons), similar to `plot_ts()`.

    - **Aesthetic Customization:** Provides extensive control over plot aesthetics, including colours, line/point styles, axis labels, and title wrapping.

- **Output:** A single `ggplot` object (if combined) or a list of `ggplot` objects (if plotted separately).

10.2. Generic `plot()` S3 Methods

`b3gbi` implements S3 generic `plot()` methods for each calculated indicator. These methods act as convenient wrappers, allowing users to simply call `plot()` on an indicator object (e.g., `plot(my_obs_richness_map)`) without needing to manually select the correct underlying plotting function or define standard labels.

Each S3 `plot()` method performs the following:

1. **Object Class Validation:** Verifies that the input `x` belongs to the expected indicator class (e.g., `obs_richness, tax_distinct`).
2. **Object Type Dispatch:** Checks whether `x` is an `indicator_ts` or `indicator_map` object.
3. **Default Parameter Setting:** Based on the indicator's class and whether it's a time series or map, it sets appropriate default values for:
    - The default label for the y-axis in time series plots.
    - The default label for the colour legend in map plots.
    - The default main title for the plot.
4. **Function Call:** Dispatches the plotting request to the appropriate core plotting function (`plot_map()`, `plot_ts()`, `plot_species_map()`, or `plot_species_ts()`), passing the `x` object, the determined default labels and title, and any additional arguments provided by the user. This allows users to easily override defaults.

This architecture ensures a user-friendly plotting interface, where indicator-specific details are handled automatically, while still providing full control over plot aesthetics.

11. Conclusion

The `b3gbi` R package has been developed to address the increasing need for standardised, reproducible, and robust methods for calculating biodiversity indicators from large datasets, such as those made available through the Global Biodiversity Information Facility (GBIF). `b3gbi` provides a comprehensive and user-friendly automated workflow for calculating a diverse suite of spatial and temporal biodiversity indicators and visualising the results.

Through this workflow, `b3gbi` offers users a flexible approach to analyse a variety of general biodiversity metrics, including both diversity and occurrence-based measures. The package supports the exploration of biodiversity trends over time and spatial variations, with integrated capabilities for uncertainty estimation and smoothed trend analysis. By leveraging the inherent

structure of these data cubes, the `b3gbi` workflow ensures an efficient and scalable method for assessing biodiversity metrics across different spatial and temporal resolutions.

As the final step in the workflow, `b3gbi` integrates powerful visualisation tools, built on `ggplot2`, to generate customizable maps and time series plots. Originating from the B3 project, `b3gbi` aims to contribute to more accessible and transparent biodiversity data analysis, facilitating reproducibility of assessments and reporting obligations and supporting critical efforts in conservation science and policy.

In summary, `b3gbi` provides a versatile and robust workflow for researchers, conservation practitioners, and policymakers to derive meaningful insights from biodiversity occurrence data. By offering standardised processes, a broad range of indicators, and flexible visualisation options, the package supports a more comprehensive understanding of biodiversity trends, thereby aiding ecological research, environmental monitoring, and the development of effective conservation strategies.

12. Acknowledgements

B^3 (Biodiversity Building Blocks for policy) receives funding from the European Union's Horizon Europe Research and Innovation Programme (ID No 101059592). This document was produced with the help of the generative AI Google Gemini Flash 2.5. I am grateful to Heliana Teixeira and Hanno Seebens for providing helpful feedback and suggestions to improve the manuscript.

13. References

1. Buchanan, G. M., Butchart, S. H., Chandler, G., & Gregory, R. D. (2020). Assessment of national-level progress towards elements of the Aichi Biodiversity Targets. *Ecological Indicators*, *116*, 106497. https://doi.org/10.1016/j.ecolind.2020.106497

2. Buckland, S. T., Baillie, S. R., Dick, J. M. P., Elston, D. A., Magurran, A. E., Scott, E. M., Smith, R. I., Somerfield, P. J., Studeny, A. C., & Watt, A. (2012). How should regional biodiversity be monitored? *Environmental and Ecological Statistics*, *19*(4), 601–626.

3. Chamberlain, S. A., & Szöcs, E. (2013). taxize: taxonomic search and retrieval in R. *F1000Research*, 2.

4. Chamberlain, S., Szoecs, E., Foster, Z., Boettiger, C., Ram, K., Bartomeus, I., Baumgartner, J., O'Donnell, J., Oksanen, J., Tzovaras, B. G., Marchand, P., Tran, V., Salmon, M., Li, G., & Grenié, M. (2020). taxize: Taxonomic Information from Around the Web. R package version 0.9.98. https://github.com/ropensci/taxize.

5. Chandler, M., See, L., Copas, K., Bonde, A. M., López, B. C., Danielsen, F., Kristoffer, J., Legind, J. K., Masinde, S., Miller-Rushing, A. J., Newman, G., Rosemartin, A., & Turak, E. (2017). Contribution of citizen science towards international biodiversity monitoring. *Biological conservation*, 213, 280-294.

6. Chao, A., Gotelli, N. J., Hsieh, T. C., Sander, E. L., Ma, K. H., Colwell, R. K., & Ellison, A. M. (2014). Rarefaction and extrapolation with Hill numbers: a framework for sampling and estimation in species diversity studies. *Ecological monographs*, *84*(1), 45-67.

7. Clarke, K. R., & Warwick, R. M. (1999). The taxonomic distinctness measure of biodiversity: weighting of step lengths between hierarchical levels. Marine Ecology Progress Series, 184, 21-29.

8. Desmet, P., Oldoni, D., Blissett, M., Robertson, T. (2023). Specification for species occurrence cubes and their production. B3 Project Deliverable D2.1.

9. Hill, M. O. (1973). Diversity and evenness: a unifying notation and its consequences. *Ecology*, *54*(2), 427-432.

10. Hsieh, T. C., Ma, K. H., & Chao, A. (2016). iNEXT: an R package for rarefaction and extrapolation of species diversity (Hill numbers). *Methods in Ecology and Evolution*, *7*(12), 1451-1456.

11. Jaccard, P. (1901). Étude de la distribution florale dans une portion des Alpes et des Jura. *Bulletin de la Societe Vaudoise des Science Naturelles*, *37*(142), 547-579.

12. Jones, J. P. G., Collen, B., Atkinson, G., Baxter, P. W. J., Bubb, P., Illian, J. B., Katzner, T. E., Keane, A., Loh, J., Mcdonald-Madden, E., Nicholson, E., Pereira, H. M., Possingham, H. P., Pullin, A. S., Rodrigues, A. S. L., Ruiz-Gutierrez, V., Sommerville, M., & MilnerGulland, E. J. (2011). The Why, What, and How of Global Biodiversity Indicators Beyond the 2010 Target. *Conservation Biology*, *25*(3), 450–457. https://doi.org/10.1111/j.1523- 1739.2010.01605.x

13. Kvålseth, T. O. (2015). Evenness indices once again: critical analysis of properties. *SpringerPlus*, *4*, 1-12.

14. Langeraert, W. (2025). *gcube: Simulating Biodiversity Data Cubes* [Computer software].

15. Langeraert, W., & Van Daele, T. (2025). *dubicube: Calculation and Interpretation of Data Cube Indicator Uncertainty* [Computer software]. https://github.com/b-cubed-eu/dubicube

16. Magurran, A. E. (1998). *Ecological diversity and its measurement*. Princeton University Press.

17. Nicholson, E., Collen, B., Barausse, A., Blanchard, J. L., Costelloe, B. T., Sullivan, K. M. E., Underwood, F. M., Burn, R. W., Fritz, S., Jones, J. P. G., McRae, L., Possingham, H. P., & Milner-Gulland, E. J. (2012). Making robust policy decisions using global biodiversity indicators. *PLoS ONE*, *7*(7).

18. Pereira, H. M., Ferrier, S., Walters, M., Geller, G. N., Jongman, R. H. G., Scholes, R. J., Bruford, M. W., Brummitt, N., Butchart, S. H. M., Cardoso, A. C., Coops, N. C., Dulloo, E., Faith, D. P., Freyhof, J., Gregory, R. D., Heip, C., Höft, R., Hurtt, G., Jetz, W., … Wegmann, M. (2013). Essential biodiversity variables. *Science*, *339*(6117), 277–278.

19. Pielou, E. C. (1966). The measurement of diversity in different types of biological collections. *Journal of theoretical biology*, *13*, 131-144.

20. Proença, V., Martin, L. J., Pereira, H. M., Fernandez, M., McRae, L., Belnap, J., Böhm, M., Brummitt, N., García-Moreno, J., Gregory, R. D., Honrado, J. P., Jürgens, N., Opige, M., Schmeller, D. S., Tiago, P., & van Swaay, C. A. M. (2017). Global biodiversity monitoring: From data sources to Essential Biodiversity Variables. *Biological Conservation*, *213*, 256–263.

21. Rabinowitz, D. (1981). Seven forms of rarity. *Biological aspects of rare plant conservation*.

22. Troia, M. J., & McManamay, R. A. (2016). Filling in the GAPS: evaluating completeness and coverage of open‐access biodiversity databases in the United States. *Ecology and evolution*, 6(14), 4654-4669.

23. Troudet, J., Grandcolas, P., Blin, A., Vignes-Lebbe, R., & Legendre, F. (2017). Taxonomic bias in biodiversity data and societal preferences. *Scientific reports*, 7(1), 9132.

24. Turak, E., Harrison, I., Dudgeon, D., Abell, R., Bush, A., Darwall, W., Finlayson, C. M., Ferrier, S., Freyhof, J., Hermoso, V., Juffe-Bignoli, D., Linke, S., Nel, J., Patricio, H. C., Pittock, J., Raghavan, R., Revenga, C., Simaika, J. P., & de Wever, A. (2017). Essential Biodiversity Variables for measuring change in global freshwater biodiversity. *Biological Conservation*, *213*, 272–279.

25. Viti, M. M., Gkimtsas, G., Liquete, C., Dubois, G., Borg, J., Dalla Costa, S., Teller, A., Hauser, R., & Robuchon, M. (2024). Introducing the progress monitoring tools of the EU Biodiversity Strategy for 2030. *Ecological Indicators*, *164*, 112147.

14. Annex

This annex provides additional technical details that were excluded from the main text to avoid overwhelming the reader, but may be of interest to some.

14.1. Dependencies

`b3gbi` relies on functions from a number of useful R packages, which are briefly described in Tables A1 and A2.

Table A1. Core R package dependencies (required)

Package	Primary Role/Functionality
`boot`	Provides functions for bootstrapping and related statistical methods
`dplyr`	Offers a consistent grammar for data manipulation and transformation
`ggplot2`	Used for generating high-quality data visualisations
`iNEXT`	Facilitates interpolation and extrapolation of species richness and diversity
`labeling`	Supports label placement in plots (often via ggplot2)
`magrittr`	Provides the pipe operator (%>%) for readable code
`mgrs`	Handles Military Grid Reference System (MGRS) conversions
`patchwork`	Simplifies combining multiple ggplot2 plots into a single layout
`permute`	Tools for permutations of data for statistical analyses
`purrr`	Iterates functions over lists and vectors in a consistent way
`readr`	Provides fast and friendly functions for reading rectangular data
`rlang`	Offers tools for building R packages and working with expressions
`rnaturalearth`	Facilitates access to natural earth vector and raster map data
`scales`	Tools for scaling, training, and mapping data values to visual properties
`sf`	Essential for handling and manipulating simple features spatial data
`stringr`	Simplifies common string operations
`tibble`	Modern reimplementation of data frames, for clearer data structures
`tidyr`	Tools for tidying data, helping to reshape and organize
`units`	Handles, converts, and manipulates physical units

Table A2. Suggested dependencies (required only for specific functionality)

Package	Primary Role/Functionality
`bold (>= 1.3.0)`	Provides functions to make text bold in the R console output

knitr	Weaves R code and output into dynamic reports
mockr	Tools for mocking and stubbing functions, primarily for unit testing in R
rmarkdown	Generates dynamic reports and documents from R code and Markdown
rnaturalearthdata	Provides vector map data from Natural Earth for R
taxize (>= 0.9.99)	Retrieves and standardizes taxonomic data from online sources
testthat (>= 3.0.0)	Provides a framework for unit testing code in R

14.2. Indicator calculation methodologies

14.2.1.Observed Species Richness

Within `b3gbi`, the calculation of Observed Species Richness leverages the structured nature of the `processed_cube` input, using `taxonKey` to identify species.

- **Spatial Analysis (`dim_type = "map"`):** When `obs_richness_map()` is called, it triggers the `calc_map.obs_richness()` method. This method directly counts the number of unique `taxonKey` values for each `cellid` using a `dplyr::summarize` operation. The `diversity_val` column then holds the observed species richness for each grid cell.
- **Temporal Analysis (`dim_type = "ts"`):** When `obs_richness_ts()` is called, it invokes the `calc_ts.obs_richness()` method. This method is analogous to its spatial counterpart but groups the distinct `taxonKey` counts by `year` instead of `cellid`. The `diversity_val` column in this case represents the observed species richness for each year.

14.2.2. Total Occurrences

`b3gbi` calculates total occurrences by summing all records (indicated by the `obs` column, representing the number of occurrences).

- **Spatial Analysis (`dim_type = "map"`):** Calling `total_occ_map()` invokes the `calc_map.total_occ()` method. This method sums the `obs` values for all records within each `cellid` using `dplyr::summarize`. The `diversity_val` column stores the total number of observed occurrences for each grid cell.
- **Temporal Analysis (`dim_type = "ts"`):** When `total_occ_ts()` is called, it triggers `calc_ts.total_occ()`. This method calculates the total number of occurrences by summing the `obs` values for all records, grouped by year. The `diversity_val` column indicates the total number of observed occurrences across the entire study area for each respective year.

14.2.3.Evenness

Both Pielou's and Williams' evenness measures rely on the proportion of occurrences for each species (p_i) and the total number of species (S) within each aggregation unit (cell or year).

- **Core Spatial Calculation (`calc_map.evenness_core()`):** When `pielou_evenness_map()` or `williams_evenness_map()` are called, they dispatch to `calc_map.evenness_core()`. This internal function calculates the total number of occurrences (`num_occ`) for each unique `taxonKey` within each `cellid`. It then reshapes this data into a matrix-like structure (cells as columns, species as rows), replaces NAs with 0, and applies `compute_evenness_formula()` to calculate evenness for each cell.
- **Core Temporal Calculation (`calc_ts.evenness_core()`):** When `pielou_evenness_ts()` or `williams_evenness_ts()` are called, they dispatch to `calc_ts.evenness_core()`. This function follows a very similar logic to its spatial counterpart, but the aggregation and pivoting are performed on `year` rather than `cellid`, applying `compute_evenness_formula()` to each `year`.
- **Evenness Formulas (`compute_evenness_formula()`):** This internal function takes the observation counts for species within a given aggregation unit (cell or year) and the specified type of evenness.
    - **Pielou's Evenness** (from Pielou, 1966)**:** Calculated as a transformation of the Simpson's index. The formula implemented is:

      $$\frac{-\sum_{i=1}^{S} p_i \times log(p_i)}{log(S)}$$

      where *S* is the number of species and p_i is the proportion of occurrences represented by species *i*. If S=1 (only one species), the denominator becomes 0, leading to `NaN`. In this case, the indicator value is explicitly set to `NA` as evenness is undefined.
    - **Williams' Evenness** (from Kvålseth, 2015)**:** Derived from the Gini-Simpson index. The formula implemented is:

      $$1 - \left[\frac{\left(S\sum_{i=1}^{S} p_i^2 - 1\right)}{S-1}\right]^{\frac{1}{2}}$$

      where *S* is the number of species and p_i is the proportion of occurrences represented by species *i*. If S=1, the denominator (S−1) becomes 0, leading to `NaN`. In this case, the indicator value is explicitly set to `NA`.

14.2.4. Rarity

`b3gbi` calculates the total summed rarity for each grid cell or year by aggregating the rarity values of each species present in that unit.

- **Abundance-Based Rarity (Spatial - `calc_map.ab_rarity()`):**
    1. For each `cellid` and `taxonKey`, it calculates `obs_taxon` (total occurrences of that species in that cell) and `obs_cell` (total occurrences in that cell).
    2. Then, for each species within a cell, its rarity is `1/(obs_taxon/obs_cell)`.

3. Finally, it sums these rarity values for all species within each `cellid`.

- **Abundance-Based Rarity (Temporal - `calc_ts.ab_rarity()`):**
    1. It first calculates `records_taxon` (total occurrences of each `taxonKey` across *all* years and cells) and then `rarity = 1 / (records_taxon / sum(obs))` for each occurrence, where `sum(obs)` is the total occurrences in the entire dataset. This implies a global rarity for each species based on its overall abundance.
    2. These rarity values are summed per `year` and `cellid`.
    3. Finally, these cell-level sums are aggregated by summing them again for each year to yield the `diversity_val` for that year.
- **Area-Based Rarity (Spatial - `calc_map.area_rarity()`):**
    1. It calculates `occ_by_taxa` (the number of unique `cellids` each `taxonKey` occurs in across the entire dataset) and `total_cells` (the total unique `cellids`).
    2. Then, for each species, its rarity is `1/(occ_by_taxa/total_cells)`.
    3. Finally, for each `cellid`, it sums the rarity values of all species present in that cell.
- **Area-Based Rarity (Temporal - `calc_ts.area_rarity()`):**
    1. It calculates `rec_tax_cell` (the number of unique `cellids` each `taxonKey` occurs in across *all* years and cells) and `sum(dplyr::n_distinct(cellid))` (the total number of unique cells in the dataset).
    2. Then, for each species, its rarity is `1/(rec_tax_cell/sum(dplyr::n_distinct(cellid)))`, implying a global rarity for each species based on its overall spatial occupancy.
    3. These rarity values are summed per `year` and `cellid`.
    4. Finally, the `diversity_val` for each year is computed as the *mean* of these cell-level rarity sums within that year.

14.2.5. Estimated Hill Diversity

Hill diversity (from Hill, 1973) is calculated as:

$$D = \left(\sum_{i=1}^{S} p_i \left(r_i \right)^{l} \right)^{1/l}$$

where D is diversity, S is the number of species, p_i is the proportion of occurrences represented by species i, and ℓ determines the rarity scale for the mean.

`b3gbi` uses **coverage-based estimation** to calculate Hill diversity values, aiming to mitigate the effects of sample size and sampling biases. This involves standardising by coverage, where coverage is the proportion of individuals in the theoretical community belonging to the species

that were observed in the sample. The `iNEXT` package (Chao et al., 2014; Hsieh et al., 2016) for R is used to estimate species richness at an equal level of coverage (e.g., 0.95) for each cell or year in the biodiversity data cube.

- **Required Parameters:** For Hill diversity calculations, users must specify:
    - `coverage`: The target sample coverage value for the estimator (default: 0.95).
    - `cutoff_length`: The minimum number of data points for each grid cell or year. Units with fewer data points than this threshold will be removed before calculations to avoid errors (default: 5).
- **Method Dispatch:**
    - `hill0_map()` and `hill0_ts()` set `type = "hill0"`.
    - `hill1_map()` and `hill1_ts()` set `type = "hill1"`.
    - `hill2_map()` and `hill2_ts()` set `type = "hill2"`.

Hill diversity calculations in `b3gbi` utilize coverage-based estimation via a wrapper function `my_estimateD()` which interfaces with the `iNEXT` package (Hsieh et al., 2016).

- **Core Spatial Calculation (`calc_map.hill_core()`):** When `hill0_map()`, `hill1_map()`, or `hill2_map()` are called, they dispatch to `calc_map.hill_core()`. This internal function extracts the `qval` (0, 1, or 2), `cutoff_length`, and `coverage`. It then processes data grouped by `cellid` to create presence-absence incidence matrices, filters cells with insufficient species (`cutoff_length`), and finally calls `my_estimateD()` (from `iNEXT`) with `datatype = "incidence_raw"`, `base = "coverage"`, `level = coverage`, and `q = qval`.
- **Core Temporal Calculation (`calc_ts.hill_core()`):** When `hill0_ts()`, `hill1_ts()`, or `hill2_ts()` are called, they dispatch to `calc_ts.hill_core()`. This function follows a similar logic but aggregates data by year.
    1. It extracts the `qval` and parameters like `cutoff_length` and `coverage`.
    2. It groups the input data (`x`) by year and for each year creates an occurrence matrix where rows are internal row identifiers and columns are `scientificNames`, containing `obs` values (converted to 1 for presence if greater than 1). This involves multiple `dplyr` and `tidyr` operations to transform the data.
    3. Years with fewer unique species than `cutoff_length` are removed to prevent `iNEXT` errors.
    4. The processed occurrence matrices for each year are then passed to `my_estimateD()` (a wrapper for `iNEXT::estimateD()`), using `datatype = "incidence_raw"`, `base = "coverage"`, `level = coverage`, and the appropriate `q = qval`.

The output indicator table includes the `qD` (estimated diversity), `t` (sample size estimate), `SC` (coverage), `Order.q` (diversity type), and confidence intervals (`qD.LCL`, `qD.UCL`) for each year.

14.2.6. Cumulative Species Richness

The calculation of cumulative species richness involves tracking all distinct species identified in the dataset up to a particular point in time.

- **Temporal Analysis (`dim_type = "ts"`):** When `cum_richness_ts()` is called, it dispatches to `calc_ts.cum_richness()`. This method performs the following steps. The `diversity_val` column represents the total count of distinct species observed from the first year up to the respective year.
    1. It selects `year` and `taxonKey` columns and arranges the data by `year`.
    2. It then identifies all *unique* `taxonKey` values for each `year`, effectively listing new species observed in that `year`, and counts them as `unique_by_year`.
    3. Finally, it calculates the cumulative sum of `unique_by_year` values. This ensures that a species, once observed, contributes to the cumulative count for all subsequent years.

Spatial Application (`dim_type = "map"`): This indicator is inherently temporal, and a direct spatial map analogue is not provided, as cumulative richness is typically applied to trends over time.

14.2.7. Mean Year of Occurrence (Newness)

The mean year of occurrence indicates the "newness" of observations.

- **Spatial Analysis (`dim_type = "map"`):** When `newness_map()` is called, it dispatches to `calc_map.newness()`. This method computes the arithmetic mean of the year values for all occurrences within each `cellid` using `dplyr::summarize`, rounded to the nearest integer. An optional `newness_min_year` parameter can filter results below a threshold.
- **Temporal Analysis (`dim_type = "ts"`):** When `newness_ts()` is called, it triggers `calc_ts.newness()`. This method calculates the *cumulative mean year* of occurrence. For each year in the time series, it calculates the mean of all year values from the beginning of the dataset up to and including that specific year. This approach provides a smooth trend of the average observation date over the entire cumulative dataset, rather than a snapshot mean for each individual year.

14.2.8. Occurrence Density

Occurrence Density is calculated by summing the total number of occurrences (`obs`) per unit area (square kilometre).

- **Spatial Analysis (`dim_type = "map"`):** Calling `occ_density_map()` triggers `calc_map.occ_density()`. This method sums all `obs` values within each `cellid` and then divides this sum by the area (in km^2) of that cell. A check ensures a projected CRS is used.

- **Temporal Analysis (`dim_type = "ts"`):** When `occ_density_ts()` is called, it triggers `calc_ts.occ_density()`. This method first calculates the occurrence density for each individual `cellid` within each year (sum of `obs` divided by area). Subsequently, for each year, it calculates the *mean* of these cell-level densities to provide a single annual occurrence density value for the entire study area.

14.2.9. Species Occurrences

The calculation involves summing the total number of occurrences (`obs`) for a given species (`taxonKey`) within each aggregation unit.

- **Spatial Analysis (`dim_type = "map"`):** When `spec_occ_map()` is called, it dispatches to `calc_map.spec_occ()`. This method calculates the total `obs` for each unique combination of `taxonKey` and `cellid`. It then selects distinct combinations of `cellid` and `scientificName`, reporting the total occurrences for each species within each cell where it was observed.
- **Temporal Analysis (`dim_type = "ts"`):** When `spec_occ_ts()` is called, it dispatches to `calc_ts.spec_occ()`. This method calculates the total `obs` for each unique combination of `taxonKey` and `year`. It then selects distinct combinations of `year` and `scientificName`, reporting the total occurrences for each species within each year it was observed. This function explicitly retains `taxonKey` and `scientificName` in the output, indicating its species-specific nature for time series analysis.

14.2.10. Species Range

- **Spatial Analysis (`dim_type = "map"`):** When `spec_range_map()` is called, it dispatches to `calc_map.spec_range()`. This method effectively flattens all occurrences of a species within a cell to a simple presence (`diversity_val = 1`). It then ensures that only one entry per `cellid` and `scientificName` combination is retained, indicating that the species was observed in that cell.
- **Temporal Analysis (`dim_type = "ts"`):** When `spec_range_ts()` is called, it dispatches to `calc_ts.spec_range()`. This method calculates the number of *distinct grid cells* a species was observed in *for each year*. This provides a measure of the observed spatial extent of a species' range within each year. The `diversity_val` in this case represents the count of grid cells occupied by a given species in a given year.

14.2.11. Taxonomic Distinctness

Calculating taxonomic distinctness requires an external taxonomic hierarchy. `b3gbi` uses the `taxize` package (Chamberlain & Szöcs, 2013; Chamberlain et al., 2020) to retrieve this information and then computes the distinctness for each aggregation unit.

- **Core Spatial Calculation (`calc_map.tax_distinct()`):** It retrieves the taxonomic hierarchy for all unique `scientificNames` from GBIF using `my_classification()`. This hierarchy is then used by `compute_tax_distinct_formula()` to calculate taxonomic distinctness for each `cellid`.

- **Core Temporal Calculation (`calc_ts.tax_distinct()`):** When `tax_distinct_ts()` is called, it dispatches to `calc_ts.tax_distinct()`. This function performs similar steps to its spatial counterpart but groups the analysis by year, applying `compute_tax_distinct_formula()` to the species present in each year.
- **Taxonomic Distinctness Formula (`compute_tax_distinct_formula()`):** This internal function calculates the average taxonomic distance between species within a sample.
    1. It first filters the complete taxonomic hierarchy (`y`) to include only the species (`x$scientificName`) present in the current aggregation unit (`cell` or `year`).
    2. It checks if the number of species (`n_spec`) in the aggregation unit is less than 3. If so, taxonomic distinctness cannot be meaningfully calculated, and the function returns `NA`.
    3. If `n_spec` is 3 or more, it uses `taxize::class2tree()` to construct a taxonomic tree from the subsetted hierarchy.
    4. From this tree, `tax_tree$distmat` provides a pairwise distance matrix between all species based on their shared taxonomic ranks. The distance is a measure of how far apart two species are on the taxonomic tree (e.g., sharing a genus but not a family would have a smaller distance than sharing only a phylum).
    5. The taxonomic distinctness is then calculated as the Taxonomic Distinctness Index (TDI; from Clarke & Warwick, 1999), the sum of all unique pairwise distances (`sum(tax_distance)`) divided by the total number of unique pairwise comparisons possible for `n_spec` species, which is given by the formula

$$\left(\sum\sum_{i<j}\frac{\left(\left|R_i-R_j\right|\right)}{L}\right)/\left(\frac{S(S-1)}{2}\right)$$

    where S is the number of species, R_i and R_j are the taxonomic ranks of species i and j, and L is the maximum number of taxonomic ranks.

14.2.12. Species Turnover

Species turnover is a temporal indicator that compares the species lists between consecutive years.

- **Temporal Analysis (`dim_type = "ts"`):** When `occ_turnover_ts()` is called, it dispatches to `calc_ts.occ_turnover()`. This method calculates turnover for each year compared to the preceding year (the first year's turnover is undefined/NA). The `diversity_val` for each year (except the first) represents the species turnover rate. The process involves:
    1. Organizing the species (identified by `taxonKey`) present in each year into a list (`ind_list`).
    2. For each year `y` (starting from the second year):

- `tax_added`: Identifies species present in year `y` but *not* in year `y-1` (`setdiff(ind_list[[y]], ind_list[[y-1]])`).
- `tax_lost`: Identifies species present in year `y-1` but *not* in year `y` (`setdiff(ind_list[[y-1]], ind_list[[y]])`).
- `tax_present`: Identifies species present in *both* year `y-1` and year `y` (`intersect(ind_list[[y-1]], ind_list[[y]])`).

3. The **species turnover** for year `y` is then calculated using a modified Jaccard-like dissimilarity index:

$$\frac{b+c}{a+b+c}$$

where *a* is the species present in both years, *b* is the species present in year `y` but not `y-1` and *c* is the species present in year `y-1` but not `y`.

- **Spatial Analysis (`dim_type = "map"`):** This indicator is inherently temporal and not applicable for a single spatial snapshot.

14.3. Indicator-Specific Confidence Interval Calculation Details

14.3.1. Hill Diversity (Richness, Shannon-Hill, Simpson-Hill)

For **Hill Richness (`hill0`)**, **Shannon-Hill Diversity (`hill1`)**, and **Simpson-Hill Diversity (`hill2`)**, the confidence intervals are **not** calculated directly by the `calc_ci` methods using the boot package. Instead, their confidence intervals are determined externally by the iNEXT package (Chao et al., 2014; Hsieh et al., 2016), which is integrated into the `calc_ts.hill_core()` function during the initial indicator calculation. The `calc_ci.hill_core()` function primarily serves as a check to ensure these confidence intervals have already been computed.

14.3.2. Total Occurrences (`total_occ`)

- **Bootstrapping:** The raw observation counts (`obs`) for each year are collected. The `boot::boot()` function is then applied to these annual lists of observations.
- **Statistic:** The `boot_statistic_sum` function is used, which calculates the sum of observations in each bootstrap replicate.
- **Integration:** The resulting bootstraps are passed to `calc_ci.core()` to derive and merge the confidence intervals.

14.3.3. Occurrence Density (`occ_density`)

- **Bootstrapping:** For each year, the occurrence density per cell (sum of observations per cell divided by cell area) is calculated. These cell-based density values are then grouped by year, and `boot::boot()` is applied.

- **Statistic:** The `boot_statistic_mean` function is used, calculating the mean of the density values across cells in each bootstrap replicate.
- **Integration:** The bootstraps are then processed by `calc_ci.core()`.

14.3.4. Newness (newness)

- **Bootstrapping:** For each year, the data includes all years up to the current year. The `boot::boot()` function is applied to lists of these years.
- **Statistic:** A specific `boot_statistic_newness` function is utilized. It calculates the "newness" value for each bootstrap sample.
- **Integration:** The resulting bootstraps are processed by `calc_ci.core()`.

14.3.5. Evenness (Pielou's and Williams')

- **Bootstrapping:** For each year, the number of occurrences (`num_occ`) for each species (`taxonKey`) is summarized and pivoted into a wide format. `boot::boot()` is then applied to these species occurrence vectors for each year.
- **Statistic:** A custom `boot_statistic_evenness` function is used. This function takes the bootstrapped species counts and recalculates either Pielou's or Williams' evenness based on the type argument ("`pielou_evenness`" or "`williams_evenness`").
- **Special Handling:** An internal `ci_error_prevent()` function is applied to the bootstrap results before `calc_ci.core()` to handle potential `NA` values that might arise if bootstrap samples result in undefined evenness (e.g., only one species).
- **Integration:** The adjusted bootstraps are then passed to `calc_ci.core()`.

14.3.6. Abundance-based Rarity (`ab_rarity`) and Area-based Rarity (`area_rarity`)

- **Bootstrapping:**
  - For **abundance-based rarity**, the rarity of each observation is calculated based on total occurrences per taxon, and then these rarity values are grouped by year for bootstrapping.
  - For **area-based rarity**, species rarity per cell (based on the number of cells a species occupies) is aggregated by year, and these aggregated rarity values are bootstrapped.
- **Statistic:**
  - `boot_statistic_sum` is used for abundance-based rarity.
  - `boot_statistic_mean` is used for area-based rarity.

- **Integration:** The bootstraps are passed to `calc_ci.core()`. The `calc_ci.core()` function ensures that the lower confidence interval for rarity does not fall below zero.

14.3.7. Species Occurrences (`spec_occ`) and Species Range (`spec_range`)

- **Bootstrapping:** For these species-specific indicators, the bootstrapping is performed **per species** across years and grid cells. The observations (or presence/absence for range) are summarized by `taxonKey`, `year`, and `cellCode`, then split by `taxonKey`. For each species, the data is pivoted to have years as columns, and `boot::boot()` is applied to the time series of observations/range values.
- **Statistic:** The `boot_statistic_sum` function is used for both species occurrences and species range.
- **Integration:** The confidence intervals are calculated for each species individually using `get_bootstrap_ci()`, then combined into a single data frame before being joined with the main indicator data frame. The lower confidence interval is adjusted to zero if negative.

14.4. Example plots

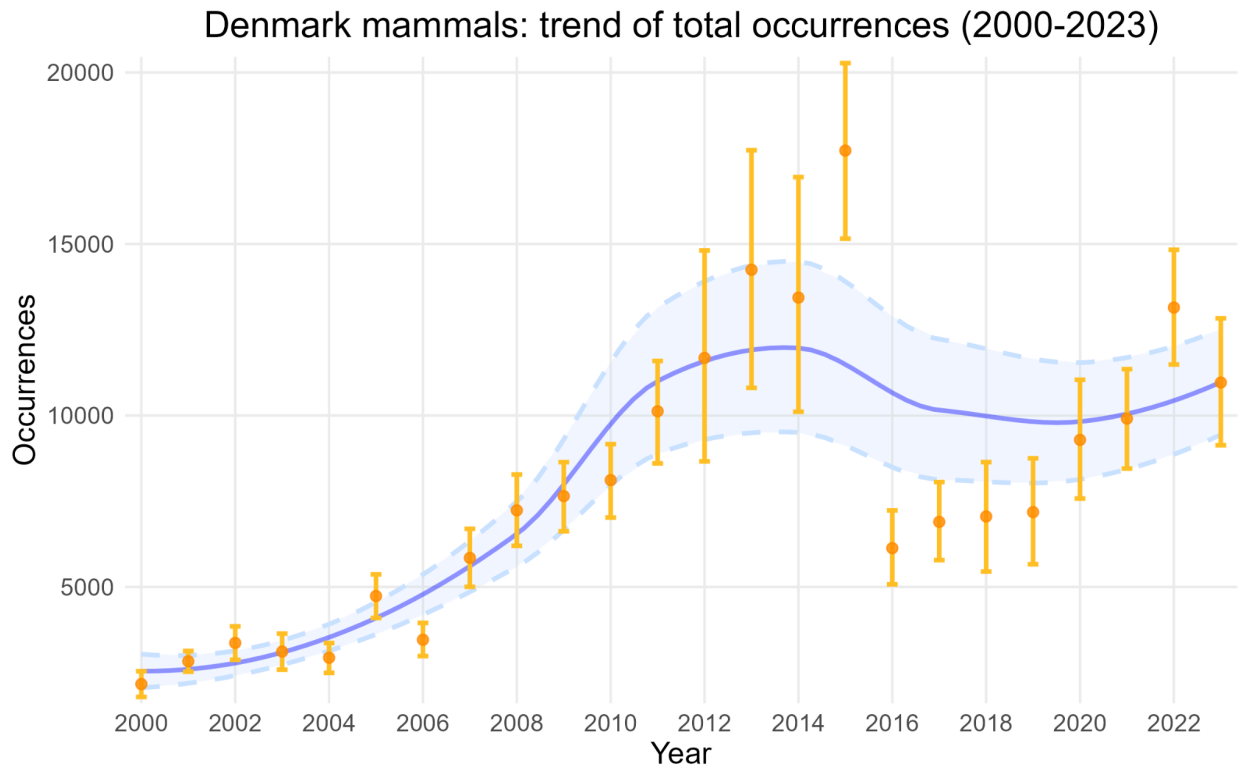

Figure 2. Time series of the total occurrences of mammals in Denmark from 2000-2023. The cube was downloaded from GBIF using the EQDGC grid. The orange dots show the actual values with the vertical orange lines representing the uncertainty. The blue line

represents the smoothed values, with the light blue envelope representing the smoothed uncertainty.

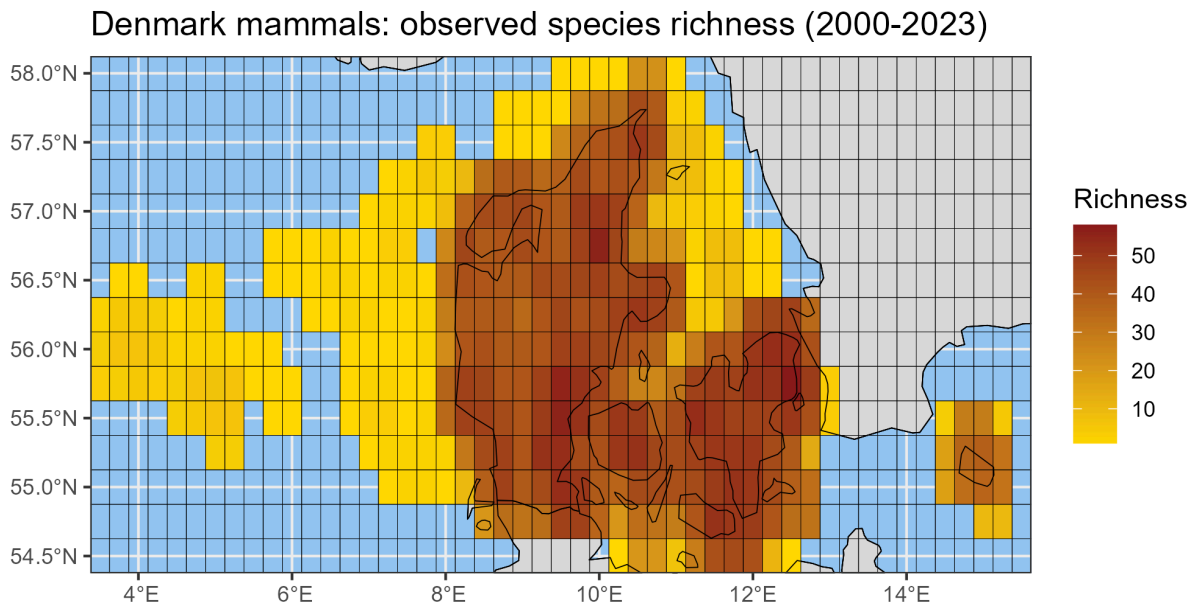

Figure 3. Map of the observed species richness of mammals in Denmark from 2000-2023. The cube was downloaded from GBIF using the EQDGC grid. The black grid lines show the outlines of individual cells.

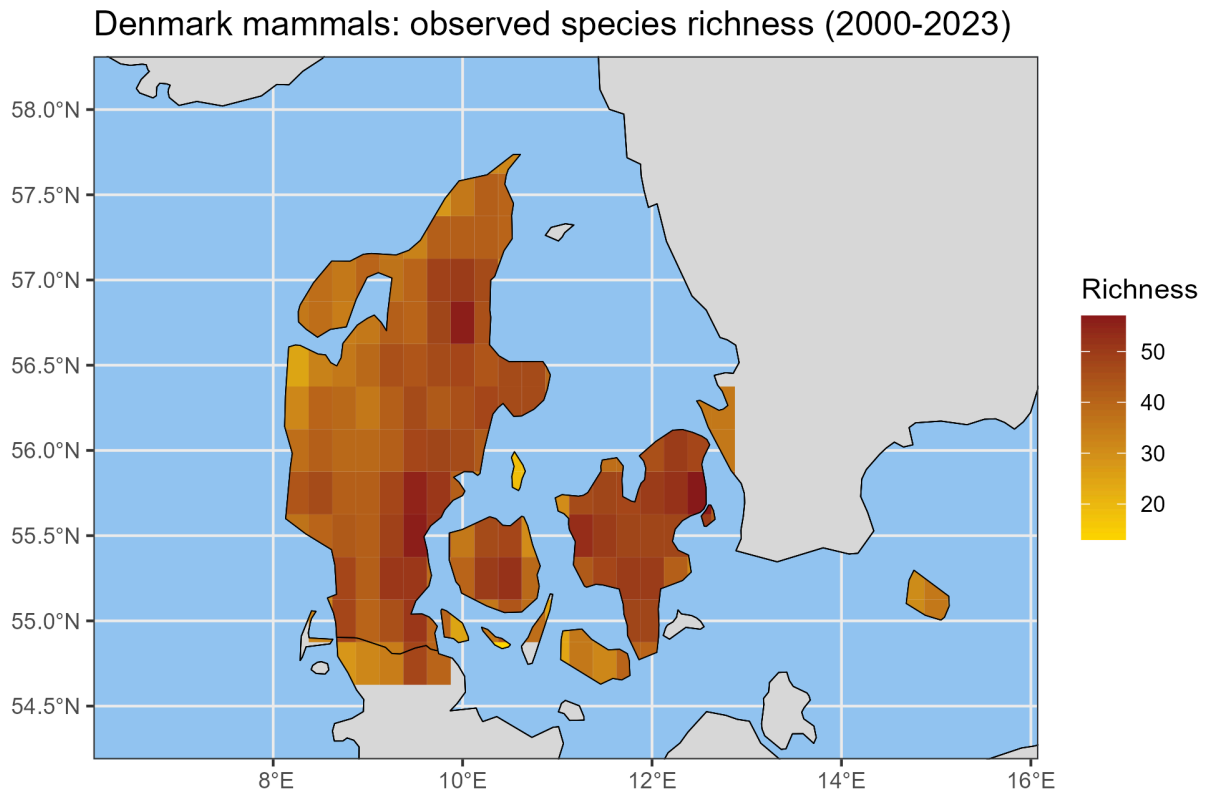

Figure 4. Map of the observed species richness of mammals in Denmark from 2000-2023. The cube was downloaded from GBIF using the EQDGC grid. This plot uses the same data as Fig. 3 but the parameter include_ocean was set to FALSE when calculating the indicator and visible_gridlines was set to FALSE when creating the plot. Cell borders are still visible but less obvious.