Biodiversity Informatics, 20, 2026, pp. 219-235

BiOS: AN OPEN-SOURCE FRAMEWORK FOR THE INTEGRATION OF HETEROGENEOUS BIODIVERSITY DATA

Alejandro Roldán[1#], Tomás Golomb Durán[1#], Antoni Josep Far[1], Maria Capa[1], Enrique Arboleda[1] and Tommaso Cancellario[1]*

[#] *Equally shared first authorship*
[1]*Centre Balear de Biodiversitat, Universitat de les Illes Balears, Palma, Spain*

Abstract. The era of Big Data has reshaped biodiversity and ecological research. Nevertheless, much of the potential held within these data is still constrained by their heterogeneity, database-incompatible schemas, and fragmentation of resources. Whilst standards such as Darwin Core have provided guidelines for standardizing biodiversity data, significant barriers persist in harmonizing heterogeneous biological datasets, including taxonomic mismatches (e.g., discordances in accepted taxa classifications and nomenclature), variability in genetic marker nomenclature (e.g., lack of standardized genetic marker names), inconsistencies in ecological or biological trait definitions (e.g., variation in trait terminology and measurement units), and the absence of a unified coordinate reference system for species distribution information (e.g., use of both WGS84 and UTM). Collectively, these challenges hinder straightforward research workflows, forcing researchers to make compromises that potentially reduce operational efficiency and may introduce biases or distortions in the results. Here, we present the Biodiversity Observatory System (BiOS), a comprehensive, open-source software designed to address these impediments through a modular and community-driven architecture. BiOS departs from monolithic database designs by decoupling the back-end data management from the front-end presentation layer. This separation supports a dual-access model tailored to diverse stakeholder needs. For researchers and developers, the system offers a comprehensive Application Programming Interface that exposes all back-end functionalities, enabling seamless programmatic access, automated data retrieval, and integration with external analytical workflows. At the same time, this software offers a user-friendly web interface designed to support non-technical users in accessing and exploring the data. Beyond access and usability, BiOS follows strict validation rules that ensure consistent data storage (e.g., taxonomic tree construction) and structures inputs using shared schemas and data formats that improve interoperability across biodiversity datasets. In addition, the system includes a synonymy management framework for both species and molecular marker names, helping to resolve nomenclatural redundancies and improve data consistency across sources. In this context, BiOS acts as a relational engine capable of integrating heterogeneous data streams while maintaining compliance with FAIR-oriented design principles. By providing a flexible, interoperable core that supports the 'seven shortfalls' framework of biodiversity knowledge, BiOS offers a turnkey solution to overcome data fragmentation and enhance collaborative conservation efforts.

Key words: Biodiversity tools, Data Integration, Data Standardization, Database Systems, FAIR.

* Corresponding author: Tommaso Cancellario t.cancellario@uib.eu - tommaso.cancellario@gmail.com

Introduction

Biodiversity research has fully entered the era of Big Data, providing access to vast amounts of information on an unprecedented temporal and spatial scale. Nowadays, a wide range of biodiversity records is available through online public databases (Hobern et al., 2019), reflecting the considerable efforts and financial resources that researchers and initiatives continue to invest in the collection and digitization of primary biodiversity data. These initiatives support many research activities and contribute to conservation efforts. Such resources are often developed with a specific thematic focus, although some cover multiple domains while remaining specialized within particular areas of biodiversity research. For example, Catalogue of Life (COL; Bánki et al., 2025) is mainly dedicated to storing nomenclatural and taxonomic information, though it occasionally includes supplementary details on species distribution and habitat. In contrast, other databases are specifically designed to store genetic information, such as the Barcode of Life Data System (BOLD; Ratnasingham & Hebert, 2007), which combines DNA barcoding sequences with taxonomic and distributional data. Whereas other platforms are designed to embrace a broader scope and include several kinds of data. For instance, the Global Biodiversity Information Facility (GBIF; GBIF.org, 2025) can be considered a 'data aggregator' since it integrates multiple types of taxon-related information from diverse sources. Nevertheless, its primary focus remains species occurrence and taxonomy, although genetic sequence data have recently begun to be incorporated into its records. In parallel, recent years have seen the development of increasingly specialized databases dedicated to specific aspects of biodiversity, such as species traits in the TRY Plant Trait Database (TRY; Kattge et al., 2020) or species interactions in Global Biotic Interactions (GLOBI; Poelen et al., 2014). Altogether, these infrastructures form a diverse ecosystem of applications that plays a fundamental role in advancing modern ecological and biodiversity research.

To date, biodiversity digital platforms host billions of records (e.g., 2.2 million accepted species in COL; 15 million plant trait records in the TRY; over 3.5 billion occurrence records in GBIF; more than 1 billion genetic records in BOLD; all accessed 22 December 2025), opening the door to the investigation of ecological questions that were previously intractable due to limitations in data quality, quantity, and availability. This rapid expansion in data accessibility has been closely linked to major technological advances achieved over the last two decades.

In particular, the development of more efficient computational infrastructures has made it possible to rapidly store, manage, and share massive volumes of information, and the introduction of new data formats (e.g., JSON) has enabled more flexible data storage and exchange. At the same time, biodiversity science has undergone a profound transformation driven by the transition from paper-based records to globally interconnected digital infrastructures. This process has been supported by the large-scale digitization of natural history collections, the development of standardized data frameworks and interoperable infrastructures, the establishment of global biodiversity repositories, the mobilization of information from scientific and grey literature, the rapid expansion of citizen science initiatives, and the integration of molecular and genomic data. As a result, biodiversity knowledge has evolved from fragmented and locally stored records into globally accessible digital resources that encompass multiple data types, broad geographic coverage, and extended temporal series. These characteristics now enable comprehensive analyses of ecological and biogeographical patterns across large spatial scales, as well as the investigation of long-term temporal trends, species dynamics, and environmental change. Collectively, these advances have not only accelerated biodiversity research, creating a unique opportunity for a more systematic assessment of the 'seven shortfalls' described by Hortal et al. (2015), but also strengthened the capacity for efficient conservation planning and environmental management.

Despite the exponential growth and increasing availability of biodiversity information, significant barriers still hinder the effective integration and full exploitation of these data, making the creation of fully interoperable and harmonized biodiversity infrastructures still, to some extent, a largely aspirational goal. Among these impediments are the limited awareness of existing databases and their contents, and the heterogeneity of data and internal schemas (see Feng et al., 2025 for a detailed review).

A first major challenge concerns taxonomy and nomenclature. Incompatibilities among backbone taxonomic systems represent a long-recognized issue that substantially limits interoperability between biodiversity databases (Grenié et al., 2023; Sandall et al., 2023; Feng et al., 2022). Indeed, biological databases may rely on different taxonomic concepts [e.g., morphological - Cronquist (Cronquist, 1981) *vs* molecular identification - APG (Angiosperm Phylogeny Group, 1998)] or adopt alternative classification reference versions [e.g., APG III (Angiosperm Phylogeny Group, 2009) *vs* APG IV (Angiosperm Phylogeny Group et al., 2016)]. Consequently, the same biological entities may be assigned to different classifications across data sources. Moreover, the lack of regular taxonomic updates further contributes to inconsistencies, as outdated nomenclature may persist and compromise

large-scale data synthesis (Grenié et al., 2023).

A second challenge concerns data quality and sampling methodologies. Biodiversity records are often generated through diverse monitoring programs and sampling protocols, which may differ substantially in sampling design, effort, spatial and temporal coverage, and methods for estimating or reporting abundance (e.g., exact counts *vs* qualitative or quantitative estimates). Another long-standing issue concerns the interpretation of species distributions, particularly the distinction between presence and absence records. The absence of a species record does not necessarily imply a true biological absence, but may instead reflect imperfect detection, insufficient sampling effort, or methodological limitations during data collection. Such uncertainties can introduce significant biases into ecological analyses, species distribution models, and biodiversity assessments, especially when integrating heterogeneous datasets across broad spatial and temporal scales.

A third challenge involves database structure and interoperability. Biodiversity databases are often developed using heterogeneous schemas and data standards (e.g., Atlas of Living Australia *vs* Global Biodiversity Information Facility; Mesibov, 2018), which frequently reflect traditional methodologies, discipline-specific terminology, or institution-specific conventions. These issues are particularly evident and less explored in the management of molecular and genomic data. For example, there is often no standardized nomenclature for genetic markers, with the same marker being reported under different names (e.g., COI, CO1, COX1, or COX-1 for *Cytochrome c oxidase subunit I*). In addition, genetic markers are not always consistently or explicitly annotated within database metadata. This problem is especially common in Next Generation Sequencing projects, such as whole-genome sequencing, mitogenomics, or genome-skimming approaches, where sequences of commonly used markers may be embedded within larger datasets without being specifically referenced in the associated metadata. As a consequence, retrieving, identifying, and systematically downloading all relevant records becomes difficult, thereby limiting data integration, reproducibility, and downstream comparative analyses.

These difficulties are further compounded by differences in the underlying informatics infrastructures, including incompatible back-end architectures and data management frameworks across platforms. This is particularly problematic when databases are built on different paradigms, such as relational *versus* non-relational databases, which structure, store, and retrieve information in different ways. Moreover, inconsistencies in data formats (e.g., date standards or geographic reference systems), application programming interfaces, metadata standards, and indexing strategies can further preclude seamless data exchange. As a result, even when data are publicly available, technical incompatibilities may prevent efficient integration and large-scale comparative analyses. Together, these limitations make it difficult to create harmonized and comprehensive datasets, forcing researchers to devote considerable time and effort to searching, organizing, homogenizing taxonomies, and standardizing data, rather than focusing on ecological questions, scientific interpretation, and hypothesis testing.

To address some of the issues described, standards and protocols such as Darwin Core (Wieczorek et al., 2012; see glossary) and the Veg-X standard (Wiser et al., 2011; see glossary) have been developed and have demonstrated that their adoption and structured schemas can significantly enhance the usability and interoperability of biodiversity data. These frameworks provide essential guidelines and standardized vocabularies for recording taxonomic, spatial, temporal, and ecological information, enabling more effective data sharing and reuse across platforms and research communities. In practice, their implementation has supported large-scale biodiversity assessments and improved reproducibility in ecological research, by facilitating the aggregation of datasets from multiple sources (e.g., datasets originating from museums worldwide, each with different recording protocols, that within GBIF follow Darwin Core's standards). Despite these advances, however, only a few initiatives have adopted such standards and achieved high levels of standardization, and substantial heterogeneity persists across databases and resources. Differences in data formats, metadata completeness, and terminologies continue to limit the seamless integration of biodiversity datasets, hindering large-scale analyses and cross-institutional collaborations. Moreover, many existing tools focus on specific data types or taxonomic groups, making it challenging to adopt a unified approach for multi-typology biodiversity data. This ongoing fragmentation underscores the need to develop more flexible, interoperable, and community-driven platforms capable of harmonizing heterogeneous data whilst remaining extensible and broadly accessible.

Here, we present the foundations of a new bioinformatics tool designed to integrate and harmonize multiple types of biodiversity data, facilitating their sharing and offering potential benefits to both the scientific community and stakeholders engaged in environmental conservation. Biodiversity Observatory System (BiOS) is a relational database specifically designed to store, manage, and distribute biological and ecological data in accordance with

Table 1. Comparison of data availability and infrastructure across major biodiversity databases. The table evaluates the presence of specific biological/ecological data categories and technical features. Data accessibility is classified into three states: Yes (green; data is structured and directly accessible); No (light gray; data is absent); Lat. for Latent (light green; data is present but unstructured, such as within free-text fields, and requires extraction). All platforms accessed 13 February 2026. Databases specifically accessed and verified for the 27 reference species are indicated by underlined acronyms: Atlas of the Living Australia (ALA; Belbin et al., 2021); BoldSystems (BOLD; Ratnasingham et al., 2007); Catalogue Of Life (COL; Bánki et al., 2025); Global Biodiversity Information Facility (GBIF; GBIF.org, 2025); INaturalist (iNat; iNaturalist, 2026); International Union for Conservation of Nature (IUCN; IUCN, 2026); National Center for Biotechnology Information (NCBI; Sayers et al., 2025); Ocean Biodiversity Information System (OBIS; OBIS, 2026); Plants Of the Word Online (POWO; POWO, 2026); World Register of Marine Species (WoRMS; Ahyong et al., 2026).

		Database										
Data category		ALA	BOLD	COL	GBIF	iNat.	IUCN	NCBI	OBIS	POWO	WoRMS	BiOS
Taxonomy		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Georeferenced distribution		Yes	Yes	Lat.	Yes	Yes	Lat.	Lat.	Yes	Lat.	Lat.	Yes
Genetics		No	Yes	No	Yes	No	No	Yes	Yes	No	No	Yes
Biological traits (e.g. body length)		Yes	Yes	No	Lat.	Yes	Yes	Lat.	No	Lat.	Yes	No
Tags	Legislative (e.g., CITES)	Yes	No	No	Lat.	Yes	Yes	No	No	No	Yes	Yes
	Conservation status	Yes	No	No	Yes	Yes	Yes	No	Yes	Lat.	Yes	Yes
	System (e.g., marine)	Lat.	Yes	Yes	Yes	No	Yes	Lat.	No	No	Yes	Yes
	Habitat	Lat.	No	No	Yes	No	Yes	Lat.	No	Lat.	Yes	Yes
API		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Open source (infrastructure)		Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	No	Yes

* Caldwell et al. (2024) species list: *Canis familiaris*; *Canis lupus*; *Ursus arctos*; *Ovis aries*; *Picea abies*; *Panthera tigris*; *Oryctolagus cuniculus*; *Caretta caretta*; *Felis catus*; *Loxodonta africana*; *Panthera pardus*; *Panthera leo*; *Chelonia mydas*; *Vulpes vulpes*; *Sus scrofa*; *Calluna vulgaris*; *Rangifer tarandus*; *Capra hircus*; *Dermochelys coriacea*; *Gallus gallus*; *Lutra lutra*; *Eretmochelys imbricata*; *Molothrus ater*; *Trichechus manatus*; *Pteridium aquilinum*; *Aythya ferina*; *Pelecanus occidentalis*.

the FAIR principles (Findable, Accessible, Interoperable, Reusable; Wilkinson et al., 2016). We aimed to develop an easily deployable and modular tool built upon a standardized core structure that allows individual database instances to be seamlessly linked. By establishing a solid structural backbone for both the back-end and the front-end architectures, this system facilitates the integration of biodiversity data from multiple sources and typologies (e.g., taxonomy, genetics, and distribution), thereby enabling more comprehensive data sharing and expanding analysis possibilities. We developed BiOS following an open-source philosophy to support a collaborative, community-driven, and modular system applicable across diverse institutional stakeholders, spanning academia, government, the non-profit sector, and industry. Rather than creating a single monolithic database, we designed BiOS to be freely deployable by users, embracing a flexible and extensible architecture. Each BiOS module is independent, and communication among modules is mediated through a shared taxonomic backbone. This design allows users to modify or update individual modules without affecting the rest of the system or disrupting overall functionality. To ensure modularity and ease of maintenance, we implemented a simplified design that separates back-end components (database and API) from front-end components (web user interface). This separation allows the system to be dynamic whilst keeping all functions fully accessible through the API, enabling seamless integration with external tools and workflows. We designed the user interface to provide intuitive access for users, supporting efficient data exploration and management. By combining modular design, standardization, and interoperability, BiOS effectively addresses current challenges in biodiversity data integration and management, facilitating comprehensive ecological analyses and promoting collaborative, community-driven research.

An overview of contemporary biodiversity databases

To contextualize the need for a new integrative and modular biodiversity infrastructure, we conducted a systematic review of the major public biodiversity databases. We focused on those most widely used in biodiversity research and analyses across multiple dimensions, including taxonomy, species distribution, functional traits, genomics, and conservation status, assessing both data availability and accessibility across core biological data dimensions (Table 1). Our overview focused on evaluating whether each database is capable of storing, managing, and sharing specific categories of biodiversity data, rather than assessing the degree of interoperability or direct integration among different platforms. Thus, our objective was to determine the breadth and accessibility of the biological information available within each infrastructure, independently from the technical capacity of databases to communicate or exchange data with one another. Our assessment considered eight major biological and ecological data categories that are essential for integrative biodiversity research, together with an evaluation of the key infrastructural features of each database. We intentionally selected broad and general criteria in order to encompass the widest possible range of biodiversity databases and avoid focusing on highly specialized or domain-specific features. This approach allowed us to provide a neutral and comparative overview of the main capabilities shared across existing biodiversity infrastructures. Specifically, we assessed whether the systems are open-source, provide access through a public API, and support the storage and dissemination of information related to taxonomy, georeferenced species distribution, genetic data, biological and functional traits, conservation status, legislative frameworks, ecological systems, and habitat associations. We employed a hierarchical approach to perform this review. First, we assessed whether each platform provided API documentation. If it was available and supported any of the target data categories, these categories were classified as 'Yes' indicating that the data were structured and directly accessible. In cases where no API documentation was found, we queried the platform's website directly. For many widely studied species, data coverage in such databases is often comprehensive which helps reduce the likelihood of false negatives. Therefore, to standardize our assessment and minimize bias due to taxonomic relationships or preferences driven by the expertise of the authors, we used the species list proposed by Caldwell et al., (2024). The list comprises 27 taxa, representing the following classes: Mammalia, Reptilia, and Aves (Kingdom: Animalia) and Pinopsida, Magnoliopsida, and Polypodiopsida (Kingdom: Plantae). During the website-visiting, we determined whether the web page retrieved information via a third-party service (e.g., Wikipedia). When third-party sources were used, only English-language content was considered. If such external requests were detected, data availability was categorized as either 'Yes' (i.e. data were provided in a structured format) or 'Latent' (i.e. data were unstructured free-text), depending on the nature of the integration. Conversely, if no relevant data could be located through these steps, the platform was recorded as 'No'. Finally, to minimize potential omissions in any category, the results were reviewed and validated by biodiversity experts.

Some considerations on the comparative assessment should be acknowledged. The evaluation was conducted using widely studied organisms to minimize taxonomic bias (Caldwell et al., 2024); however, platform functionalities evolve continuously, and the analysis reflects the state of each system at the time of access and could eventually change in time. Furthermore, the comparison focused on the presence and structural accessibility of data categories rather than on the completeness or depth of information contained within each platform. Future work could expand the assessment to additional taxa, functional use cases, and interoperability benchmarks.

Software Tool Description

BiOS is a software platform designed to store, manage, and visualize biodiversity data, providing a centralized environment for organizing and accessing taxonomic, genetic, and distributional information.

We developed BiOS using an architecture composed of three interconnected layers: i) the database, ii) the RESTful Application Programming Interface (RESTful API; thereafter API), which together compose the backend, and iii) the web interface, which constitutes the frontend (Figure 1). We designed the architecture paying particular attention to both scalability and interoperability. These two properties are fundamental requirements for building a complex database intended to store large and heterogeneous records, as they facilitate both the integration of data from multiple sources and the exchange of information among different systems. For example, interoperability allows BiOS to communicate with external biodiversity platforms, while scalability enables the system to efficiently accommodate increasing volumes of records without compromising performance. For a quick installation guide read *Supplementary Materials S1 & S5*.

Minimum system requirements

We tested the system using minimal hardware requirements (8 GB of RAM and 32 GB of storage) running

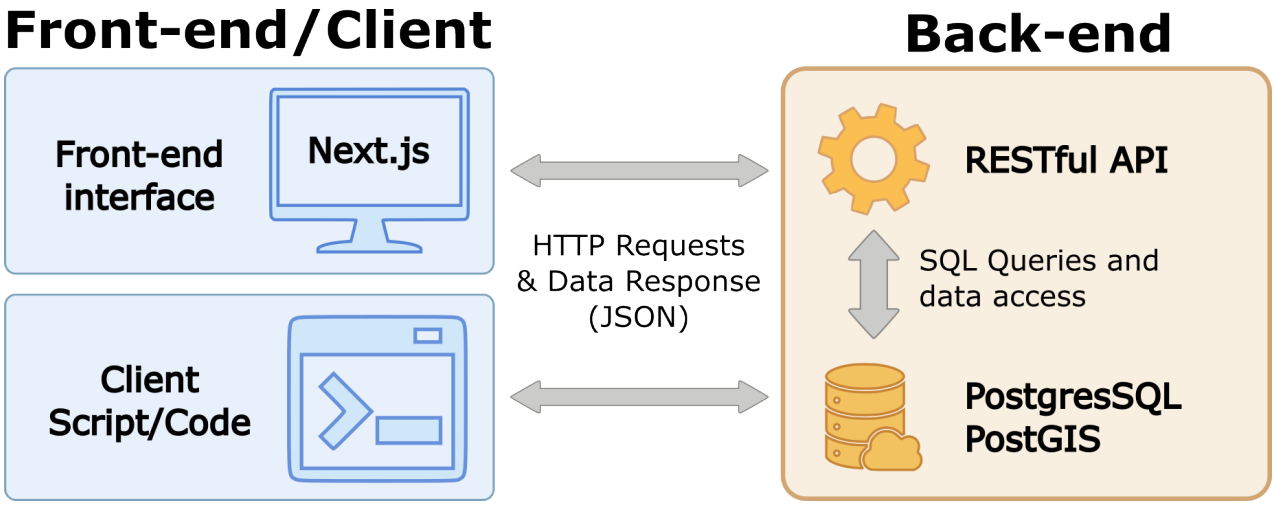

Figure 1. Schematic BiOS software architecture. The orange box includes the two main components of the back-end: the PostgreSQL-PostGIS database and the RESTful API service. The blue boxes represent the front-end, and the client query system. The vertical bidirectional arrow represents the interaction between the database and the API service (in the orange box), whereas the horizontal bidirectional arrows represent the HTTP request and response between the back-end and the front-end/client systems.

on a virtual machine. However, for moderate use cases, such as those that could be adopted by municipal or regional public administrations, we estimate, that a system equipped with 64 GB of RAM, approximately 120 GB of storage, and a standard modern processor (e.g., Intel i5/i7 or AMD Ryzen 5/7 class CPU; 4–8 core) would be sufficient to ensure stable and reliable operation of BiOS under routine usage conditions.

Database

The back-end relies on a PostgreSQL database system (PostgreSQL Global Development Group, 2025), and database interactions were implemented using *psycopg* (psycopg2-binary, 2022), a dedicated PostgreSQL adapter for Python (Van Rossum & Drake, 2009). We selected PostgreSQL because it is an open-source solution with strong integration capabilities and efficient spatial data management, as well as a long-standing reputation as a robust and reliable object-relational database management system. Moreover, it is widely adopted in scientific infrastructures, including major biodiversity platforms such as GBIF (https://www.gbif.org/article/5FlXBKbirSiq0a-scKYiA8q/gbif-infrastructure-registry). Together with PostgreSQL, we used the PostGIS extension (PostGIS Developers, 2023), which provides support for Geographic Information System data (e.g., occurrence) and functions (e.g., intersection). To facilitate the database development and ensure consistent relations between tables, we implemented the service using Django (Django Software Foundation, 2023), a high-level Python web framework. Django is a state-of-the-art framework for developing database applications, and it offers several advantages, including a robust Object–Relational Mapping system, built-in security features, an automatically generated administrative interface for immediate data management and a clear modular structure that accelerates development workflow whilst maintaining code consistency. In addition, it offers a built-in administration platform to manage the database with a user system.

To query and interact with the database, we created a RESTful API system built using the Django REST Framework extension (Django REST Framework, 2024). The API system serves a dual purpose. First, it acts as an interface between the back-end and its users (handling both web and local client requests), allowing the web interface to present information in a user-friendly format. Second, it is essential for filtering and executing operations based on the specific parameters included by the user in their server request.

We organized the information contained in the database into three modules: i) thematic modules, ii) geography, and iii) versioning. This modular structure facilitates the management of diverse data types (e.g., from spatial to genetic data) whilst preserving their traceability and efficiency. It is important to note that the schema presented here provides a simplified representation of the tables and their interactions and is intended to illustrate the core concepts underlying the database design. A comprehensive and detailed description of the framework's internal architecture, data models, and relationships (e.g., Foreign Keys, Many-to-Many fields) is reported in *Supplementary Materials S2*.

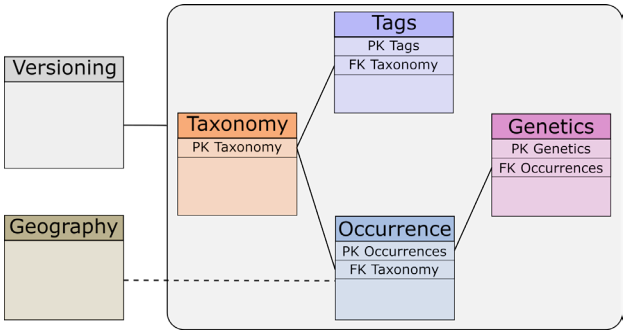

Figure 2. Simplified database schema showing its six thematic modules. Solid lines indicate direct relationships between modules, whereas dashed lines indicate indirect relationships. PK: primary key; FK: foreign key. The full schema is available in the *Supplementary Material S2 (Figure 1)* or can be downloaded from https://osf.io/jxp6h/files/osfstorage.

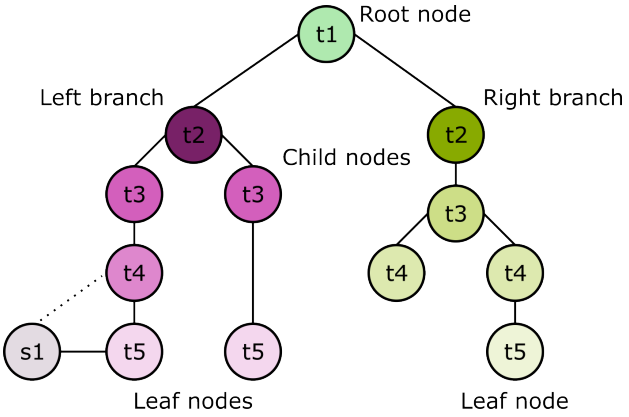

Figure 3. Taxonomic hierarchy scheme. Levels range from 't1', representing the highest taxonomic rank (the common ancestor or root node), to 't5', representing the lowest (leaf nodes). 's1' indicates an example of synonym. Circles represent tree nodes. Solid lines indicate the links between nodes. The dashed line represents the connection between the synonym and the parent node.

Thematic modules

We define a 'thematic module' (structurally as a Django App), a self-contained unit responsible for a specific category of biodiversity data. Each application has a group of primary database tables designed to store homogeneous attributes (e.g., taxonomy or genetics), supported by linked auxiliary tables that provide granular detail. This design ensures a separation of concerns, maintaining a clean and organized data schema. Moreover, this modular organization facilitates data management, offering several advantages. For example, when a record within a thematic module is added or modified, the change is automatically propagated throughout the database, avoiding the need to manually review other linked modules. Furthermore, it simplifies data maintenance and scalability, allowing new datasets or functional components to be integrated without disrupting the existing architecture (for further implementation details see *Supplementary Material S2.1*). Finally, the database is designed to handle incomplete records for individual taxa, ensuring flexibility in data management. Information from different thematic modules can still be stored, queried, and analyzed even when some data types are missing, provided that the taxonomic reference is available. For example, distributional records may be fully accessible despite the absence of corresponding genetic data, allowing the system to support progressive data integration as new information becomes available while maintaining taxonomic consistency across the database.
The database is composed of six thematic modules: Taxonomy, Occurrences, Genetics, Tags, Geography, and Versioning (Figure 2).

Taxonomy module

The *Taxonomy* module is the core of our database. This module stores the scientific nomenclature for every taxon included in the database and is organized according with standardized rules of biological nomenclature. We organized the *Taxonomy* module in a hierarchical tree structure where each node represents a taxon that inherits from a common ancestor (the root node; Figure 3). This parent–child organization ensures that every taxon is connected to its higher taxonomy, enabling precise and consistent relationships across the entire taxonomic classification. By following this strict hierarchical organization, we can efficiently navigate and query the taxonomy tree, supporting searches both downward (from higher taxonomic level to lower) and upward (from lower to higher). Additionally, this recursive design removes the need to store the complete higher taxonomic chain for each node. This significantly reduces storage requirements and simplifies maintenance, allowing the system to easily adapt to the dynamic nature of biological classification and its frequent revisions. The taxonomic tree begins with a top ancestor (e.g., Biota, rank Life), followed by eight nested taxonomic levels: kingdom, phylum, class, order, family, genus, species, and subspecies or variety. We chose to include these levels to keep the taxonomy simple and aligned with our objectives. It is important to note that the primary purpose of BiOS is to work as a tool to support biodiversity management, conservation efforts, and the identification of knowledge gaps in biodiversity, rather than for detailed taxonomic purposes. Nevertheless, due to the open-source code, users can add, modify, or remove taxonomic levels in the tree as needed to accommodate specific requirements.

The *Taxonomy* module also implements features designed to handle taxonomic synonyms. Specifically, a taxonomic node can be assigned the 'synonym' attribute and linked to a corresponding 'accepted' node. This ensures

that all data associated with a synonym are also connected to the branch of the accepted name, allowing access to the metadata connected for both taxon names (Figure 3, synonym node s1).

Linked to the *Taxonomy* module are the *Occurrence*, *Genetics*, and *Tags* modules, which store complementary metadata associated with a taxon. We implemented a normalized schema to strictly minimize data redundancy by avoiding storing repetitive nomenclature by referencing the *Taxonomy* module through a foreign key. The *Occurrence* module includes information about georeferenced organism records at a specific time. It contains details on where and when a taxon was observed or collected, including geographic coordinates, locality name, dates, and the person or entity that recorded the information. We stored and standardized the geographic coordinates using the WGS84 reference system, ensuring consistency and compatibility with many GIS tools and facilitating data exchange with global biodiversity databases (see *Supplementary Material S2.2* for more information on the relationships between the fields in the different tables).

Genetics module

The *Genetics* module contains metadata associated with the genetic information of a given taxon occurrence. In the database, we intentionally do not create specific fields to store full genetic sequences to prevent excessive storage demands. Instead, we included fields to store information such as the sequence marker type (e.g., COI, 16S, 18S, etc.), accession identifiers (e.g., GenBank accession number), and other relevant genetic descriptors. In designing this module, we assumed that genetic material is derived from a specimen sampled at a specific location, thus associated with a specific occurrence. Consequently, the *Genetics* module is not directly linked to the *Taxonomy* module, instead, the connection is established indirectly through the *Occurrence* module. We standardized genetic marker names to preserve consistency across the entire database using the same synonym approach as in the *Taxonomy* module.

Tags module

The *Tags* module was developed to store several types of metadata that complement the ecological and legislative information associated with a taxon. This module supports multiple-region conservation statuses of species (e.g., following the IUCN Red List evaluation such as Least Concern, Near Threatened, etc.), its degree of establishment (e.g., Darwin Core standard), its specific habitats and ecological systems (e.g., as defined by the IUCN Habitats Classification Scheme), as well as its inclusion within national or international directives. Similar to the *Occurrence* module, tags are linked to the *Taxonomy* module.

Geography module

We prepared the *Geography* module to store geographic geometries, such as spatial polygons provided in shapefile format. This module allows the execution of spatial intersection analyses and supports a variety of geospatial operations. It is also queried by the API to filter occurrence records based on the attributes and spatial relationships of the multilayer polygon features (*Supplementary Material S4.3*).

Versioning module

Finally, we developed the *Versioning* module that implements a dual-layer tracking strategy that facilitates flexible data lifecycle management. The *Batch* model encapsulates a data import event, allowing operations on full datasets, such as the removal of a complete upload history. Whereas, the *Source* model provides granular control over data provenance. This architecture ensures that specific pieces of data or repositories can be isolated and managed independently, without compromising the integrity of the entire batch (*Supplementary Material S3.3*).

API System

We developed an optimized API system to establish a solid communication infrastructure between the database and the web interface, as well as the end users (*Supplementary Material S4*). Following the thematic modules organization, we implemented six API blocks: Taxonomy, Occurrence, Genetics, Tags, Geography and Versioning. Each block is responsible for managing specific types of data and enabling complex queries (e.g., the Taxonomy endpoint provides access to different taxonomy-related data and performs defined filter operations). To ensure intuitive architecture and rapid data access, API blocks are named to mirror the database's thematic modules. Endpoints are organized using a strict, versioned routing structure: */api/{version}/{module}/{function}* (e.g., */api/v1/occurrence/list*). The API framework is built around two main types of endpoints: data retrieval, which handle queries to the database and return specific types of information (e.g., taxonomy of a species), and count, which provide a numerical summary of the requested data (e.g., how many occurrences of a species are stored in the database). The entire framework was developed using Django REST Framework and documented with *drf-yasg* (Figure 4, drf-yasg, 2023). Documentation on the API endpoints will be available at */api/docs* once the Django environment has been set up (*Supplementary Materials S1*).

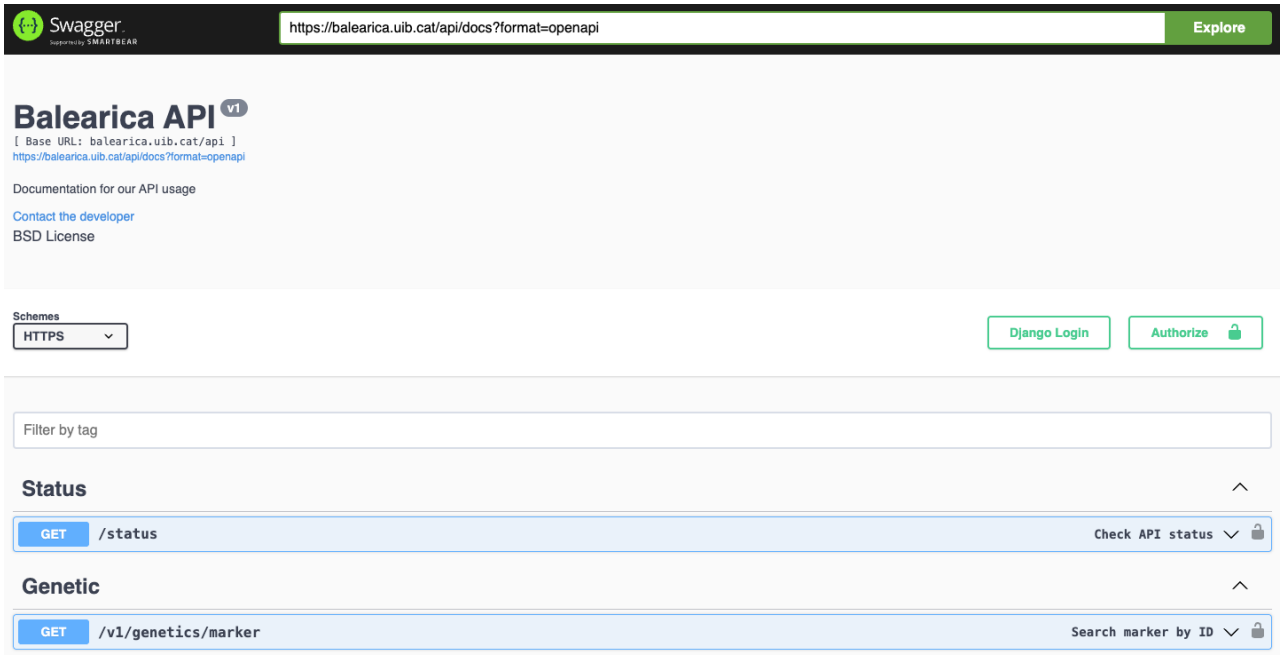

Figure 4. Screenshot of the interactive API documentation interface for the BiOS system, generated using Swagger. This interface exposes the back-end architecture, allowing users to visualize and interact with the API's resources.

To facilitate the configuration of the database and API system, we included it in a Docker container (Docker Inc., 2025). This approach creates an isolated system for either the development or production environment, ensuring efficient management of dependencies (i.e., PostGIS and Django) and increasing the reproducibility of the setup.

User Interface

To create an intuitive, responsive, and easy-to-use web tool connected with our back-end, we developed the graphical user interface using Next.js 15 (Vercel Inc., 2025; https://nextjs.org/). This web development framework provides a powerful solution for building fast and scalable online applications. Among the features that motivated our choice of Next.js (e.g., built-in routing, incremental static regeneration, and strong performance optimization) is that it notably enables the implementation of server-side-rendered web components. This relatively recent technology improves performance by rendering the initial HTML page on the server rather than in the user's browser. This feature is particularly useful especially when handling large amounts of data that could otherwise overload the user's browser. Additionally, Next.js is a widely adopted framework that benefits from a large developer community, providing extensive third-party packages for web development, and fast support for troubleshooting. For further details on how to deploy the front-end see *Supplementary Material S5*.

This framework also incorporates a powerful caching system. By caching rendered pages and API responses, Next.js reduces the need to repeatedly generate content for frequently accessed resources, minimizing server load and accelerating page delivery to users. Therefore, this significantly enhances the performance and responsiveness of the web applications, improving the user experience.

The front-end is organized by adopting a routing strategy based on the */[lang]* directory pattern (e.g., */en/about* or */es/about* both will render the same content in different languages). This architecture facilitates native internationalization, ensuring multi-language accessibility across all components through a localized routing system. The platform is structured into four primary dynamic views *Home*, *Taxon*, *Map*, and *Sources* and a static view *About* which provides general information about the platform's vision and objectives. To navigate between views and change the preferred language, a navigation bar is displayed at the top of the interface and a site map component at the bottom across all pages.

The *Home* view serves as the primary portal for user engagement, featuring a centralized search interface and a summary dashboard that aggregates global database metrics, including total species richness, occurrence counts, and genetic record availability. Upon executing a search for a scientific taxon name, the system redirects to the taxonomic view for the selected taxon.

The interface is partitioned into a hierarchical layout, with the lateral panel facilitating nomenclatural navigation through a visual component of the taxonomic tree and associated synonyms, whilst the central header provides useful metadata such as authorship, nomenclatural status, an image of the organism, and unique persistent identifiers. In-depth taxon data view is organized into three thematic modules: the *Overview* tab synthesizes ecological attributes including degree of establishment, habitat preferences, and regional conservation status; the *Distribution* tab offers an optimized cartographic distribution of known species occurrences and aggregated data for seasonality, history, and sources of the records; and the *Genetics* tab provides a comprehensive index of related genetic information associated with the taxon.

The *Map* view provides a robust environment for the interactive exploration of species distributions, allowing users to select and visualize multiple taxa simultaneously. Integrated into the front-end, the viewer utilizes the 'react-map-gl' wrapper with MapLibre (MapLibre, 2025; https://maplibre.org/) to efficiently render high-density datasets. The visualization engine is built on a multi-layer architecture prepared also for 3D environments, supporting the integration of diverse raster sources and utilizing raster-dem tiles for dynamic terrain and hillshading (Figure 5). The component communicates with the RESTful API, querying the specific */occurrences/map* routes, which performs advanced spatial queries on the PostGIS back-end to serve coordinates and metadata accurately. This design ensures that heavy geospatial processing is offloaded directly to the database. Users can interactively query specific records by clicking on individual occurrences to request its metadata and utilize the left-hand panel to filter results based on spatial uncertainty or localities. When locality filters are applied, the database executes spatial intersections between the georeferenced occurrences and the selected geographic polygons defined in the *Geography* module. To support data dissemination, the view includes a utility to export the current geospatial visualizations as a high-resolution image (.png format).

The figures below illustrate a practical example of an instance of BiOS used to create *Balearica* (see 'Example application: Balearica' section), a regional biodiversity platform (Figures 5-7).

We managed the data transparency through the *Sources* view, which provides a comprehensive registry of all datasets integrated into the platform (Figure 6). This view is designed to offer an accessible method for identifying and referencing the constituent datasets. To support academic referencing and data validation, each source entry can be expanded to reveal descriptive metadata and summary statistics regarding the specific volume and nature of the records contributed, ensuring that users can accurately identify and cite the primary data providers.

Example Application: Balearica

To evidence of the system's capacity, we employed the BiOS framework to develop Balearica, a regional infrastructure that continues to provide information about Balearic biodiversity. Through the integration of heterogeneous data sources, the platform provides a comprehensive and unified view of Balearic biodiversity, supporting

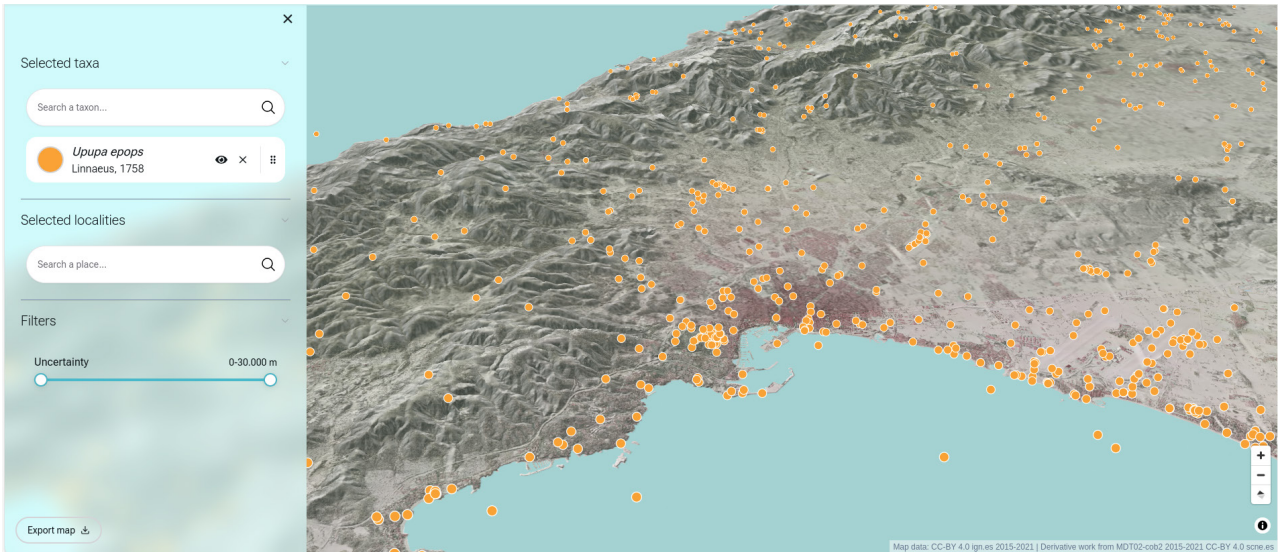

Figure 5. The interactive geospatial visualization interface view. It allows users to map species occurrences on a 3D terrain model. In this example, the distribution of the Hoopoe (*Upupa epops*) is visualized across the landscape with orange markers. The left-hand panel provides controls for searching specific taxa and localities, as well as a slider to filter records based on spatial uncertainty.

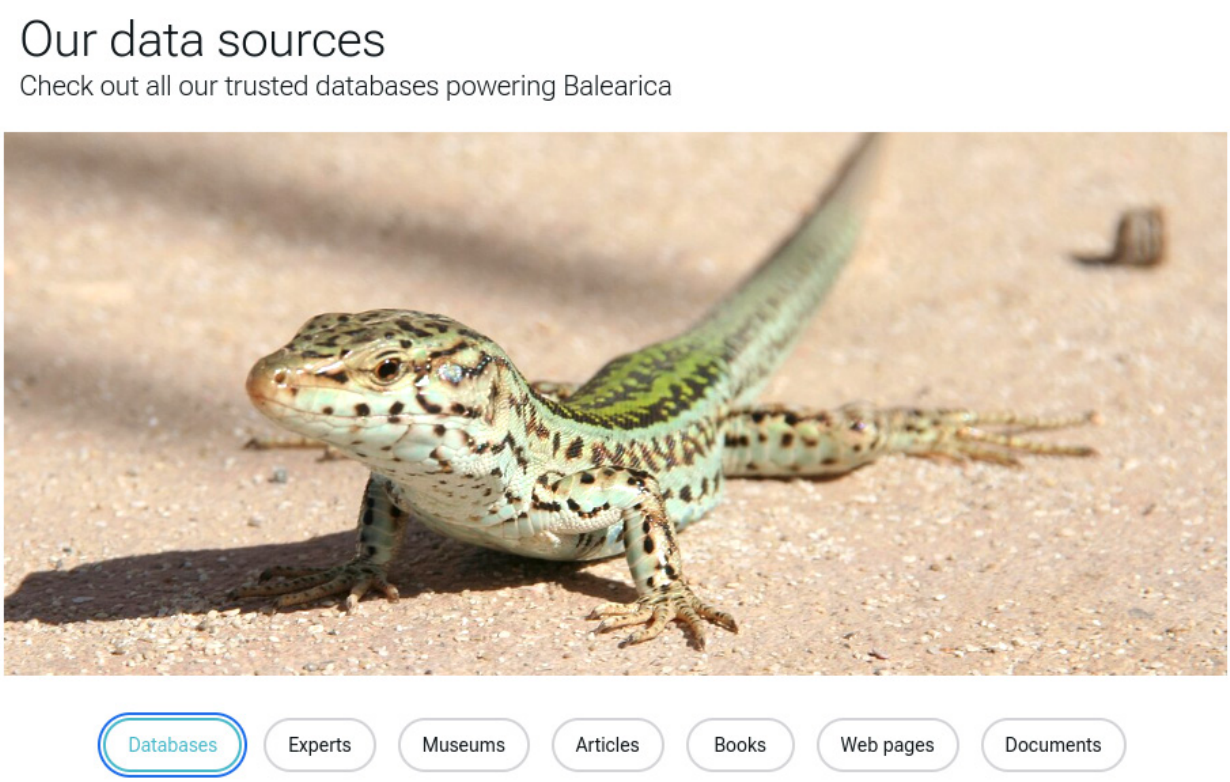

Figure 6. This section lists the external repositories integrated into the portal. As shown in the 'Databases' tab, the system aggregates biological records from major global infrastructures. By selecting a repository, the user is redirected to a specific source profile containing the aggregated data and citation information. Lizard photo: Wikipedia; Albert Masats, public domain (PD).

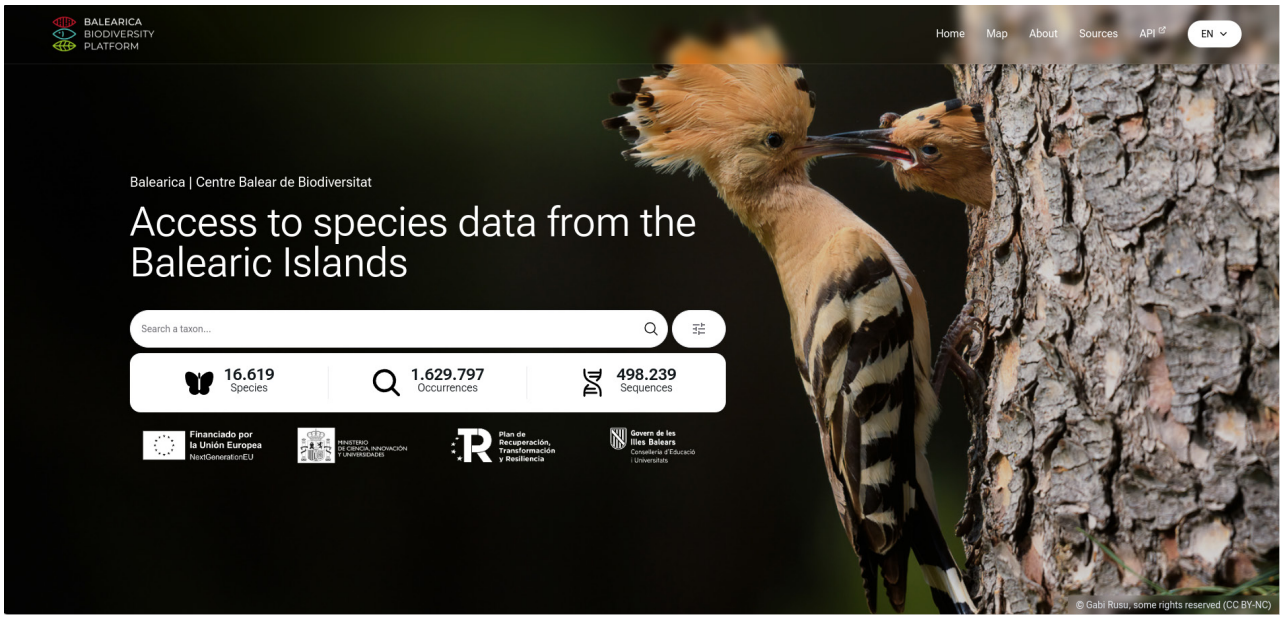

Figure 7. The main landing page of the *Balearica* web portal, serving as the primary user interface for the BiOS system. Background photo: © Gabi Rusu, some rights reserved (CC BY-NC).

scientific research and enabling the study of biological diversity across multiple biological dimensions from taxonomic to genetic. To date (accessed 22 December 2025), *Balearica* has catalogued more than 16,500 species and integrated approximately 1,600,000 occurrences. In addition, the platform links nearly half a million genetic sequences. Further details on the platform's architecture, data sources, and methodological approach are available in its official documentation (https://balearica.uib.cat/method/index.html), and the platform can be accessed directly at https://balearica.uib.cat (Figure 7).

Discussion

Here, we present the BiOS, an open-source and modular software designed to store, manage, and visualize multiple types of biodiversity data, ranging from taxonomy and genetics to species distribution and ecological information.

The review carried out in Table 1 highlights a persistent structural fragmentation within current biodiversity data infrastructures (Feng et al., 2025; Güntsch et al., 2025). Several existing platforms have evolved as highly specialized systems, each optimized for specific data domains such as taxonomy, species occurrences, genetic information, or conservation status. While this specialization has enabled the development of authoritative repositories within particular fields, it has also resulted in limited cross-domain integration as pointed out by Beyvers et al., 2026. This tendency towards specializations is partly rooted in historical technological constraints. In particular, until the early 2000s, limited computational capacity, high infrastructure costs, immature database technologies, and restricted digital connectivity represented major barriers to large-scale biodiversity information systems. These constraints also hindered international collaboration and coordinated biodiversity data management. Nowadays, these limitations are being progressively addressed through improved interoperability among databases and the adoption of standardized data models and formats. However, substantial interoperability and harmonization challenges still remain. Indeed, researchers and biodiversity managers frequently need to aggregate and reconcile data across multiple independent platforms in order to obtain a comprehensive taxon profile. When these dimensions are distributed across distinct infrastructures and repositories, data harmonization becomes a technically demanding process involving format transformation, taxonomic reconciliation, and repeated API interactions. Such workflows may introduce inconsistencies and reduce analytical reproducibility (Poisot et al., 2019; Powers & Hampton 2019; Borregaard & Hart 2016). In this context, we developed BiOS as an open-source framework designed to overcome the limitations of traditional biodiversity databases. Designed according to FAIR principles, BiOS addresses the critical challenge of data heterogeneity identified by Feng et al., (2025). Unlike static repositories, BiOS operates as a dynamic relational engine, harmonizing heterogeneous data types such as taxonomic, genomic, and geospatial data into a unified framework.

The successful deployment of BiOS through Balearica (https://balearica.uib.cat) demonstrates the system's capacity to aggregate and manage high-volume data (over 1.6 million occurrences), thereby validating its stability and scalability for regional biodiversity observatories. This implementation can be better contextualized within the broader biodiversity informatics landscape, which is anchored by large-scale infrastructures such as the ALA (Belbin et al., 2021) and the GBIF (GBIF.org, 2025), both of which represent foundational infrastructures for global biodiversity data management and mobilization. There are several key differences that distinguish ALA and GBIF from BiOS. Although ALA and GBIF share the overarching goal of improving access to biodiversity data, they differ substantially in scope, informatic architecture, and functional role, which makes them largely complementary. In particular, GBIF primarily operates as a global biodiversity data aggregation and standardization infrastructure (using Darwin Core standard to harmonize its data), focused on mobilizing and harmonizing species occurrence records at a worldwide scale, whereas ALA provides a more comprehensive and modular software ecosystem designed to support the development of national or regional biodiversity information systems. The latter has therefore been widely adopted by several national nodes (e.g., Spain, France, and Brazil). However, the highly integrated and feature-rich architecture of ALA requires specialized technical expertise for deployment and long-term maintenance, together with substantial operational and infrastructural resources that must be carefully considered (Atlas of Living Australia, 2016). In contrast, BiOS is designed as a lightweight and modular solution that prioritizes ease of deployment and operational efficiency while maintaining compatibility with biodiversity data standards. Rather than replicating the global aggregation role of GBIF or the full-featured national infrastructure model of ALA, BiOS targets a complementary niche by enabling institutions to rapidly establish standards-compliant biodiversity observatories without the substantial IT overhead typically associated with global-tier infrastructures.

The design of BiOS is a step forward in addressing the 'seven shortfalls' of biodiversity knowledge described by Hortal et al., (2015), particularly the taxonomic (*Linnean*)

and distributional (*Wallacean*) shortfalls, by improving the integration, accessibility, and usability of existing biodiversity data. The platform's taxonomic backbone enables experts to identify potential nomenclatural gaps, while advanced geospatial filtering tools may help reveal spatial biases in sampling effort. Furthermore, the integration of a dedicated genetic module enables BiOS to determine which genetic information is available for the taxa stored in the database and to assess whether these data originate from the study region or obtained worldwide. Such integration remains uncommon in regional databases, where genetic data are typically maintained separately from ecological records. This structured framework enables users to identify existing knowledge deficiencies and to support the development of targeted research projects, thereby facilitating the allocation of funding to fill these gaps. In practical applications, BiOS is currently being used in the context of the Balearic Islands through the *Balearica* platform (see 'Example application: Balearica' section) to support biodiversity assessments, including the monitoring of invasive species across the territory, the analysis of biodiversity patterns and species distributions, and the integration of genetic sampling data collected throughout the Balearic Islands.

Concerning more technical aspects, a foundational design paradigm of the BiOS architecture is the deliberate decoupling of the front-end from the back-end *via* an exposed API. Traditional biodiversity portals are often tight coupling between data logic and presentation layers, making them difficult for external institutions to adapt or maintain (Mesibov, 2018). By separating these concerns, BiOS offers two main advantages. First, the RESTful API democratizes access to the data, allowing researchers to bypass the web interface and interact programmatically with the database for reproducible research and automated meta-analyses. Second, this modularity ensures the longevity of the software as, for example, the front-end can be completely redesigned to suit emerging web technologies without disrupting the underlying data integrity.

Beyond academic research, BiOS could serve as a versatile toolbox that bridges the gap between raw scientific data and relevant stakeholders including public administrations, private environmental agencies, and NGOs, by facilitating data acquisition, integration, and visualizations. This, in turn, can support and accelerate complex decision-making processes, such as the designation or expansion of protected areas, spatial and territorial planning, and land-use management. The inclusion of a legislative labelling module allows the system to cross-reference biological records with local, national, and international protection statutes. This feature is particularly valuable for public administrations, to use the platform as an active management tool, in order to optimize and accelerate their routine environmental protection and management activities. For example, in the current implementation of BiOS in a regional context such as the Balearic Islands (i.e., *Balearica*), we have integrated additional information on national and regional species protection status as well as levels of endemism, which are often not readily available in more general biodiversity platforms. This capability underscores the potential of BiOS to serve not just as a research repository, but also as a piece of critical digital infrastructure for natural resource management.

Whilst BiOS offers a comprehensive solution for data management and interoperability, challenges persist regarding data ingestion at scale, particularly concerning the unstructured free text contained in the scrutinized databases (referred to as 'latent data' in this article). The results of our platform assessment (Table 1), reveal a contrast between biodiversity literature data and actual machine-readability. While taxonomy is generally structured and accessible across all assessed databases, other biodiversity data and more complex ecological or biological descriptors frequently suffer from structural bottlenecks. Critical information such as traits and even georeferenced distributions remain deeply embedded within unstructured text across several major repositories (Branco et al., 2025; Le Guillarme & Thuiller, 2022; Nelson & Ellis, 2019). This reliance on free-text formatting slows down large-scale automated ecological analyses, as data retrieval is often time-consuming and labor-intensive. Consequently, while these databases hold high biological value, leveraging their full scope currently requires substantial secondary extraction efforts, underscoring the need for advanced text-mining solutions or stricter data-entry standardizations within the biodiversity informatics community. However, the rapid development of Artificial Intelligence (AI) and Natural Language Processing technologies is progressively transforming this landscape by making text extraction more efficient and accessible. In particular, recent tools and software packages, such as *ARETE* (Branco et al., 2025) and *Specifind* (Golomb Durán et al., 2025), are designed to leverage AI-based models to automatically extract relevant ecological information from unstructured text, including species names, geographic references, and functional traits. Although there is still considerable work to be done, these advances suggest that AI-assisted workflows may significantly reduce the current reliance on manual text extraction and improve the speed of biodiversity data digitalization in the near future.

Future development iterations will focus on semi-automating these workflows by integrating *biodumpy* (Cancellario et al., 2026) to automatically update and retrieve already existing data from large-scale public databases under instance-specific constraints (e.g. the territorial scope). A first step towards mitigating the barrier associated with biodiversity data embedded in scientific and grey literature would be the integration of *Specifind* (Golomb Durán et al., 2025) to extract species occurrences into a structured machine-readable format. Furthermore, we aim to develop a dedicated module to include biological and ecological traits of species, increasing the multidimensional view of biodiversity. Collectively, these features will form a comprehensive, integrated suite of tools designed to improve biodiversity data management, accelerate analyses, and facilitate a multidimensional understanding of species distributions, ecological traits, and ecosystem dynamics.

Within this context, BiOS does not aim to replace established global infrastructures, nor to replicate the specialized functions of domain-specific repositories. Instead, it represents a shift towards modular, community-driven biodiversity informatics. By providing a free, open-source, and full-stack solution, we aim to lower the technical barrier for institutions worldwide to establish their own biodiversity portals. Whether applied to an archipelago ecosystem like the Balearic Islands or scaled to continental datasets, BiOS offers the structural foundation necessary to turn fragmented records into actionable scientific knowledge.

Acknowledgments

This study has been partially funded by GOIB/Conselleria d'Educació i Universitats through the project "SINCO2022/18146" and co-funded by the European Union.

This work has been partially funded and promoted by the Comunitat Autònoma de les Illes Balears through the Conselleria d'Educació i Universitats and by the European Union- Next Generation EU/PRTR-C17. I1 (SINCO2022/6717). Nevertheless, the views and opinions expressed are solely those of the author or authors, and do not necessarily reflect those of the Conselleria d'Educació i Universitats, the European Union or the European Commission. Therefore, none of these organisations shall be held liable.

We would like to thank the staff at the Centre Balear de Biodiversitat (CBB-UIB), Laura Bonacho Noguera, and Joan Pons Pons, for their contributions during the development of BiOS. We also thank everyone who has supported the initiative by offering insightful feedback, helpful suggestions, and continued support.

Authors' Contributions

TC, MC, and EA recognised the need of a modular and multidata biodiversity platform and conceived the conceptual framework to develop BiOS. TC and TGD conceived the back-end architecture, while AR and TGD developed the first version of the back-end. AR and TGD also developed the RESTful API system. TGD developed the front-end, with input and suggestions from all other authors. AJF and TC tested the software. AR, TC, and TGD wrote the original draft and led the manuscript writing. All authors contributed to discussions on the methodology and tested the software. They reviewed and edited the first draft of the manuscript and provided critical input for the final version.

Resource Availability

The code is freely available at https://github.com/centrebalearbiodiversitat/BiOS_backend for the back-end and https://github.com/centrebalearbiodiversitat/BiOS_frontend for the front-end. Testing data and Docker configuration files are available at https://osf.io/jxp6h/files/osfstorage.

Competing Interests

The authors declare no conflicts of interest.

Literature Cited

Ahyong, S., Boyko, C.B., Bernot, J., Brandão, S.N., Daly, M., De Grave, S., ... & Zullini, A. (2026). World Register of Marine Species (WoRMS). https://www.marinespecies.org

Angiosperm Phylogeny Group. (1998). An ordinal classification for the families of flowering plants. *Annals of the Missouri Botanical Garden*, 531-553. https://doi.org/10.2307/2992015

Angiosperm Phylogeny Group. (2009). An update of the Angiosperm Phylogeny Group classification for the orders and families of flowering plants: APG III. *Botanical Journal of the Linnean Society*, 161(2), 105-121. https://doi.org/10.1111/j.1095-8339.2009.00996.x

Angiosperm Phylogeny Group, Chase, M. W., Christenhusz, M. J., Fay, M. F., Byng, J. W., Judd, W. S., ... & Stevens, P. F. (2016). An update of the Angiosperm Phylogeny Group classification for the orders and families of flowering plants: APG IV. *Botanical Journal of the Linnean Society*, 181(1), 1-20. https://doi.org/10.1111/boj.12385

Atlas of Living Australia. (2016). ALA infrastructure implementation: Overview. https://www.ala.org.au/wp-content/uploads/2017/01/ALA-Infrastructure-Implementation-overview-October-2016-final.pdf

Bánki, O., Roskov, Y., Döring, M., Ower, G., Robles, D. R. H., Corredor, C. A. P., ... & Ferm, J. (2025). Catalogue of Life. https://doi.org/10.48580/dgw32

Belbin, L., Wallis, E., Hobern, D., & Zerger, A. (2021). The Atlas of Living Australia: history, current state and future directions. *Biodiversity Data Journal*, 9, e65023. https://doi.org/10.3897/BDJ.9.e65023

Beyvers, S., Schlegel, J., Brehm, L., Hansen, M., Goesmann, A., & Förster, F. (2026). Towards FAIR and federated data ecosystems for interdisciplinary research. *PLOS Computational Biology*, 22(2), e1013806. https://doi.org/10.1371/journal.pcbi.1013806

Borregaard, M. K., & Hart, E. M. (2016). Towards a more reproducible ecology. *Ecography*, 39(4). https://doi.org/10.1111/ecog.02493

Branco, V. V., Benedek, J., Pivovarova, L., Correia, L., & Cardoso, P. (2025). ARETE: an R package for Automated REtrieval from TExt with large language models. *arXiv* preprint arXiv:2511.04573. https://doi.org/10.48550/arXiv.2511.04573

Caldwell, I. R., Hobbs, J. P. A., Bowen, B. W., Cowman, P. F., DiBattista, J. D., Whitney, J. L., ... & Láruson, Á. J. (2024). Global trends and biases in biodiversity conservation research. *Cell Reports Sustainability*, 1(5). https://doi.org/10.1016/j.crsus.2024.100082

Cancellario, T., Golomb Durán, T., Far, A. J., Roldán, A., & Capa, M. (2026). Biodumpy: a comprehensive biological data downloader. *Biodiversity Informatics*, 20(1). https://doi.org/10.17161/bi.v20i1.24725

Cronquist, A. (1981). An integrated system of classification of flowering plants. *Columbia University Press*. https://doi.org/10.2307/2806386

Django REST Framework (Version 3.15.1.). [Software]. (2024) https://www.django-rest-framework.org

Django Software Foundation. Django (Version 4.2.5.). [Software] (2023). https://www.djangoproject.com

Docker Inc. Docker (Version 27.5.0.). [Software] (2025). https://www.docker.com

drf-yasg (Version 1.21.7.). [Software] (2023). https://github.com/axnsan12/drf-yasg

Feng, X., Smith, A. B., Boyle, B., Chen, X., Enquist, B. J., Gallagher, R., ... & Park, D. S. (2025). The next stage of biodiversity informatics: community-driven synthesis and integration of biodiversity databases. *BioScience*, 75(11), 913-925. https://doi.org/10.1093/biosci/biaf112

Feng, X., Enquist, B. J., Park, D. S., Boyle, B., Breshears, D. D., Gallagher, R. V., ... & López-Hoffman, L. (2022). A review of the heterogeneous landscape of biodiversity databases: opportunities and challenges for a synthesized biodiversity knowledge base. *Global Ecology and Biogeography*, 31(7), 1242-1260. https://doi.org/10.1111/geb.13497

GBIF.org. The Global Biodiversity Information Facility (2025). What is GBIF? Available from https://www.gbif.org/what-is-gbif [2025-11-03]

Golomb Durán, T., Far, J. A., Diaz, A., Barroso, M., Roldan, A., Colinas, N., & Cancellario, T. (2025). Specifind: A natural language processing tool for automating species occurrence (re-) discovery from scientific literature. *bioRxiv*, 2025-12. https://doi.org/10.64898/2025.12.24.696373

Grenié, M., Berti, E., Carvajal-Quintero, J., Dädlow, G. M. L., Sagouis, A., & Winter, M. (2023). Harmonizing taxon names in biodiversity data: a review of tools, databases and best practices. *Methods in Ecology and Evolution*, 14(1), 12-25. https://doi.org/10.1111/2041-210X.13802

Güntsch, A., Overmann, J., Ebert, B., Bonn, A., Le Bras, Y., Engel, T., ... & Luther, K. (2025). National biodiversity data infrastructures: ten essential functions for science, policy, and practice. *BioScience*, 75(2), 139-151. https://doi.org/10.1093/biosci/biae109

Hobern, D., Baptiste, B., Copas, K., Guralnick, R., Hahn, A., Van Huis, E., ... & Wieczorek, J. (2019). Connecting data and expertise: a new alliance for biodiversity knowledge. *Biodiversity Data Journal*, 7, e33679. https://doi.org/10.3897/BDJ.7.e33679

Hortal, J., De Bello, F., Diniz-Filho, J. A. F., Lewinsohn, T. M., Lobo, J. M., & Ladle, R. J. (2015). Seven shortfalls that beset large-scale knowledge of biodiversity. *Annual Review of Ecology, Evolution, and Systematics*, 46(1), 523-549. https://doi.org/10.1146/annurev-ecolsys-112414-054400

iNaturalist. (2026). Available from https://www.inaturalist.org. [2026-02-13]

IUCN. (2026). IUCN red list of threatened species. https://www.iucnredlist.org

Kattge, J., Bönisch, G., Díaz, S., Lavorel, S., Prentice, I. C., Leadley, P., ... & Wirth, C. (2020). TRY plant trait database–enhanced coverage and open access. *Global Change Biology*, 26, 119-188. https://doi.org/10.1111/gcb.14904

Le Guillarme, N., & Thuiller, W. (2022). TaxoNERD: deep neural models for the recognition of taxonomic entities in the ecological and evolutionary literature. *Methods in Ecology and Evolution*, 13(3), 625-641. https://doi.org/10.1111/2041-210X.13778

MapLibre. (Version 5.9.0). [Software] (2025). https://maplibre.org/

Mesibov, R. (2018). An audit of some processing effects in aggregated occurrence records. *ZooKeys*, (751), 129. https://doi.org/10.3897/zookeys.751.24791

Nelson, G., & Ellis, S. (2019). The history and impact of digitization and digital data mobilization on biodiversity research. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 374(1763). https://doi.org/10.1098/rstb.2017.0391

Poelen, J. H., Simons, J. D., & Mungall, C. J. (2014). Global biotic interactions: an open infrastructure to share and analyze species-interaction datasets. *Ecological Informatics*, *24*, 148-159. https://doi.org/10.1016/j.ecoinf.2014.08.005

Poisot, T., Bruneau, A., Gonzalez, A., Gravel, D., & Peres-Neto, P. (2019). Ecological data should not be so hard to find and reuse. *Trends in ecology & evolution*, 34(6), 494-496. https://doi.org/10.1016/j.tree.2019.04.005

Powers, S. M., & Hampton, S. E. (2019). Open science, reproducibility, and transparency in ecology. *Ecological Applications*, 29(1), e01822. https://doi.org/10.1002/eap.1822

OBIS. (2026). Ocean Biodiversity Information System. Intergovernmental Oceanographic Commission of UNESCO. https://obis.org

PostGIS Developers. PostGIS: Spatial and Geographic Objects for PostgreSQL (Version 3.3). [Software] (2023). https://postgis.net

PostgreSQL Global Development Group. PostgreSQL [Software] (2025). https://www.postgresql.org/

POWO. (2026). Plants of the World Online. Facilitated by the Royal Botanic Gardens, Kew. Published on the Internet; https://powo.science.kew.org/ Retrieved 20 February 2026

psycopg2-binary (Version 2.9.5.). [Software] (2022). https://pypi.org/project/psycopg2-binary/

Ratnasingham, S., & Hebert, P. D. (2007). BOLD: The Barcode of Life Data System (http://www. barcodinglife. org). *Molecular Ecology Notes*, 7(3), 355-364. https://doi.org/10.1111/j.1471-8286.2007.01678.x

Sandall, E. L., Maureaud, A. A., Guralnick, R., McGeoch, M. A., Sica, Y. V., Rogan, M. S., ... & Jetz, W. (2023). A globally integrated structure of taxonomy to support biodiversity science and conservation. *Trends in Ecology & Evolution*, 38(12), 1143-1153. https://doi.org/10.1016/j.tree.2023.08.004

Sayers, E. W., Beck, J., Bolton, E. E., Brister, J. R., Chan, J., Connor, R., & … Pruitt, K. D. (2025). Database resources of the National Center for Biotechnology Information in 2025. *Nucleic Acids Research*, 53(D1), D20–D29. https://doi.org/10.1093/nar/gkae979

Van Rossum, Guido, and Fred L. Drake. 2009. Python 3 Reference Manual. Scotts Valley, CA: CreateSpace

Vercel Inc. Next.js. Framework (Version 15.5.09) [Software] (2025). https://nextjs.org

Wieczorek, J., Bloom, D., Guralnick, R., Blum, S., Döring, M., Giovanni, R., ... & Vieglais, D. (2012). Darwin Core: an evolving community-developed biodiversity data standard. *PLoS One*, 7(1), e29715. https://doi.org/10.1371/journal.pone.0029715

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., ... & Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3(1), 1-9. https://doi.org/10.1038/sdata.2016.18

Wiser, S. K., Spencer, N., De Caceres, M., Kleikamp, M., Boyle, B., & Peet, R. K. (2011). Veg-X–an exchange standard for plot-based vegetation data. *Journal of Vegetation Science*, 22(4), 598-609. https://doi.org/10.1111/j.1654-1103.2010.01245.x

Glossary

Accessible: It should be clear how metadata and data can be accessed, including any required authentication and authorization (https://www.go-fair.org/fair-principles).

Back-end: Server-side part of a software (often a database and RESTful API) that manages the data storage, processing, and application logic.

Darwin Core: It is a standard maintained by the Darwin Core Maintenance Group. It provides a controlled vocabulary designed to facilitate the sharing of information on biological diversity. The standard aims to improve data interoperability by offering consistent identifiers, labels, and definitions (https://dwc.tdwg.org/; Wieczorek et al., 2012).

Findable: Metadata and data should be easily accessible to both humans and computers (https://www.go-fair.org/fair-principles/).

Front-end: Client-facing part of a software that users interact through a graphical user interface such as a web application. It handles the interaction with the back-end via APIs.

Interoperable: Data need to be integrated with other datasets or metadata, and their format should be compatible with applications or workflows for analysis, storage, and processing (https://www.go-fair.org/fair-principles).

Interoperability: Ability of distinct databases or systems to exchange information with minimal manual effort and effectively use the information exchanged.

Linnean shortfall: The gap between existing species diversity and the species that have been formally described and named. This shortfall also applies to extinct species, which are often underrepresented in the taxonomic record.

Non-relational database: A database that stores data in flexible, non-tabular formats (e.g., documents or key–value pairs), without requiring fixed schemas.

Relational database: A database that stores data in structured tables (rows and columns) and links them using defined relationships (keys), typically managed with SQL.

RESTful API: Standardized web interface that allows software components to communicate over HTTP using stateless requests. It enables external systems and applications to access, retrieve, and manipulate data from the back-end in a structured manner, supporting integration with other tools and workflows.

Reusable: Metadata and data should be accompanied by well-described documentation to enable replication and/or integration in different usages (https://www.go-fair.org/fair-principles).

Scalability: Ability of a system to handle increasing volumes of data or computational workload.

Veg-X standard: It is designed to be used to share and merge vegetation-plot data of different kinds. It supports observations at both the individual plant level and aggregated community levels, enabling consistent representation of vegetation data across different sampling designs and study types (https://iavs-org.github.io/VegX/articles/VegX-Standard.html; Wiser et al., 2011).

Wallacean shortfall: The lack of knowledge about the geographic distributions of species, even for those that have already been described taxonomically.