Assessing Butterflies in Europe -

Tools to support butterfly monitoring and analysis

Butterfly
CONSERVATION EUROPE

Assessing Butterflies in Europe - Tools to support butterfly monitoring and analysis

Technical report

Photo: Chris van Swaay

Assessing Butterflies in Europe -
Tools to support butterfly monitoring and analysis
Technical report

Schmucki, R.[1], Sevilleja, C.[2], Dennis, E.B.[3], Harrower, C.A.[1], Kazlauskis, K.[4], Rathod, B.[1], van Breda, G.[5], van Breda, J.[6], Van Swaay, C.A.M.[2,7] & Roy, D.B.[1] (2020). *Tools to support butterfly monitoring and analysis*. Butterfly Conservation Europe.

[1] UK Centre for Ecology & Hydrology, Wallingford, United Kingdom
[2] De Vlinderstichting/Dutch Butterfly Conservation, Wageningen, Netherlands
[3] Butterfly Conservation, East Lulworth, Dorset, United Kingdom
[4] Flumens Ltd
[5] VB IT
[6] Biodiverse IT
[7] Butterfly Conservation Europe

Citation

Schmucki, R., Sevilleja, C., Dennis, E.B., Harrower, C.A., Kazlauskis, K., Rathod, B., van Breda, G., van Breda, J., Van Swaay, C.A.M. & Roy, D.B. (2020). *Tools to support butterfly monitoring and analysis*. Butterfly Conservation Europe.

Keywords

Butterfly, Monitoring, rbms, ButterflyCount, Europe, European Union, Biodiversity

November 2020

Acknowledgements:

We thank the European Union for funding the ABLE EU Parliamentary Pilot project via a service contract with DG Environment (Contract no. 07.027742/2018/790285/SER/ENV.D.W).

The European Butterfly Monitoring Scheme (eBMS) is indebted to the constituent National Butterfly Monitoring Schemes, their funders and all volunteers who contribute data. At the time of producing this report, the eBMS is a partnership of the following organisations: the Natural Environment Research Council (acting through the Centre for Ecology & Hydrology), Butterfly Conservation UK, Helmholtz-Zentrum für Umweltforschung GmbH – UFZ, De Vlinderstichting, Catalonia BMS, Finnish Environment Institute (SYKE), Butterfly Conservation Europe (BCE), Research Institute Nature and Forest (INBO), Muséum National d'Histoire Naturelle (MNHN) CNRS-UPMC, Lund University, National Biodiversity Data Centre (NBDC) Ireland, Luxembourg Institute of Science and Technology (LIST), Zerynthia Association, Facultad de Ciencias de la Universidad Autónoma de Madrid, University of Innsbruck (Viel-Falter BMS) and Croatian Natural History Museum (CNHM).

The Austrian BMS is coordinated by Austrian Butterfly Conservation (ABC) in cooperation with the Municipal Department of Environmental Protection Vienna, the University of Vienna, the University of Salzburg, Haus der Natur (Salzburg), Nationalpark Neusiedler See - Seewinkel and Biologische Station Neusiedler See (Illmitz, Burgenland), Bio Forschung Austria and Arbeitsgemeinschaft Österreichischer Entomologen. ABC was founded thanks to the initiative and essential support of Andrea Grill and Daniela Lehner, together with the ABC co-founders Harald Krenn and Manfred Pendl.

Viel-Falter BMS, Austria is coordinated by the Department of Ecology at the University of Innsbruck and implemented together with the Natural History Collections of the Tiroler Landesmuseen Betriebsgesellschaft m.b.H, inatura and EURAC research. It is supported by the states of Tyrol and Vorarlberg and the Foundation Blühendes Österreich.

The Dutch BMS is a co-operation between Dutch Butterfly Conservation and Statistics Netherlands (CBS), part of the Network Ecological Monitoring (NEM) and financed by the Ministry of Agriculture, Nature and Food Quality (LNV).

The German BMS is a cooperation between the Helmholtz Centre for Environmental Research - UFZ, German Butterfly Conservation (GfS) and science4you.

The Irish Butterfly Monitoring Scheme is funded by the Heritage Council and the Department of Culture, Heritage and the Gaeltacht.

The Italian Butterfly Monitoring Scheme is a cooperation among the University of Turin, the University of Florence and the Council for Agricultural Research and Economics of Cosenza.

The Nature Conservation Agency of Latvia has kindly supplied the Latvian data.

The Spanish Butterfly Monitorin Scheme (BMS España) is partially supported by EBD (Estación Biológica de Doñana) and OAPN (Organismo Autónomo de Parques Nacionales) and it is a consortium with contributions by several research institutes, universities, National Parks and nature NGOs.

The ZERYNTHIA Butterfly Monitoring Scheme is supported by the Basque Country Government, Cantabria Government and Valle de Aranguren Council (Navarre).

Jacqueline Loos, Mikael Molander and Lászlo Rákosy were important contributors of the Romanian part of the AMIGA project.

The UK Butterfly Monitoring Scheme is organized and funded by Butterfly Conservation, the Centre for Ecology and Hydrology, British Trust for Ornithology, and the Joint Nature Conservation Committee. The UKBMS is indebted to all volunteers who contribute data to the scheme.

Contents

Table of Figures

Chapter 1 / Introduction

This report summarises the work within ABLE to contribute to the development of a unified system of validated and standardised butterfly counts from across European monitoring schemes. The primary aim is to provide the technical infrastructure to support the development of butterfly monitoring schemes within EU member states and regular reports on butterfly indicators and to enrich research on biodiversity.

To achieve this objective, this report summarises work on the following activities:

1. Produce an effective database for butterfly monitoring records to ensure secure long-term data storage and provide with the capability to produce a suite of butterfly indicators;
2. Establish legal data-sharing agreements with existing monitoring schemes to clarify intellectual property rights and rights of access to data;
3. Gather data annually from existing monitoring schemes, standardise and validate counts, populate database;
4. Extend online data entry systems to allow new countries to submit data efficiently and in a standard format;
5. Develop and test a mobile application for capturing timed counts of butterflies across Europe, supporting a new protocol developed in this project;
6. Provide training for monitoring scheme volunteers and scheme co-ordinators in the use of the online data entry system and data analysis tools;
7. Provide tools, training and online resources for calculating national butterfly population trends;
8. Make BMS records and metadata available in a standard and validated format for research as appropriate.

Chapter 2 / Database of existing butterfly monitoring scheme data for Europe

Building a secure, standardised and reliable database of Butterfly Monitoring Scheme (BMS) data is a key requirement to enable efficient production of butterfly indicators (see the report on Indicators – ABLE task-1), support conservation actions and facilitate scientific advances. In ABLE, we build the European Butterfly Monitoring Scheme (eBMS) database on the following three core principles:

1) Data integrity;
2) Portability;
3) Long-term accessibility.

We worked with national BMS co-ordinators and data experts to develop a formal and rich ontology to link and structure the different concepts used to inform butterfly count data recorded by national BMS (Annexe 1). The data model and protocols used in the eBMS database was designed to enable the integration and standardisation of different data formats, ranging from structure query extract to Excel workbooks and unrelated spreadsheets. In the eBMS database, we use Unicode/UTF-8 encoding to ensure portability and data integrity across operating systems and regions. Inter-operability across systems and users is also facility by the adoption of ISO standards for informing countries, regions, dates and time data in the database.

The eBMS database is built on PostgreSQL[1], a free and open-source relational database management system that can handle spatial component with the rich PostGIS[2] extension. All spatial data is stored in a projected format, using the EPSG:3035 (ETRS89-extended / LAEA Europe) projection system, as recommended by the European Commission Joint Research Centre. This projection system allows an accurate representation of the area and distance across Europe.

Traceability, data integrity and open development

The eBMS data model includes unique identifiers to ensure traceability of the data integrated into the database. This information enables us to effectively relate the data contained in the eBMS database to their sources. Traceability is essential to conduce formal validation of data integrity with the eBMS database, but also to enquire and report about specific issues to national schemes. Beyond facilitating data validation and quality check, this approach has also proven to contribute to reinforce and maintain trust within the network.

The development of the eBMS database follows the principle of open and transparency, on both the intellectual and technological aspects. The eBMS data model and the SQL codes are freely available via the Butterfly Monitoring GitHub repository (Annexe 1). This platform allows efficient version control with Git and sharing of the eBMS data model for re-use and tailoring by national schemes (e.g., BMS Sweden) and is flexible for future development, building capacity with fully portable and open-source technology. The GitHub repository allows users to raise issues and contribute to the development of the tools.

The source data files are sent by national Butterfly Monitoring Schemes (EU and other European countries) and are processed programmatically, using the R environment, where scripts are used to ensuring adequate standardisation and formatting of the data before being integrated into the eBMS database. The development of this protocol is fully documented through Git Version Control and safely stored on a GitHub repository. In our standardisation, we also ensure taxonomic congruence, using the latest European checklist of butterfly [3]. This entire checklist, including synonymy and its GBIF correspondence, is also available on the Butterfly Monitoring GitHub repository.

The eBMS database is updated on an annual cycle, each update being released as a major version (e.g., v3.0), with subsequent corrections and bug fix being identified and released as minor version updates (e.g., v3.2). For data security, we produce a backup of each version and distribute them for safe storage in three geographic locations, with one copy on a secure tape system at UKCEH (Wallingford, UK), one copy at Butterfly Conservation Europe (Wageningen, Netherlands) and the third copy at UFZ (Halles, Germany). Each of these copies can be used to restore and fully recover the data contained in each version of the eBMS database.

[1] The PostgreSQL Global Development Group, & Regents of the University of California. PostgreSQL Database Management System (Release 10). Retrieved from https://postgresql.org/

[2] The PostGIS development team. PostGIS (Version 2.5). Retrieved from https://postgis.net/

[3] Wiemers, M., Balletto, E., Dincă, V., Fric, Z. F., Lamas, G., Lukhtanov, V., … Verovnik, R. (2018). An updated checklist of the European Butterflies (Lepidoptera, Papilionoidea). *ZooKeys*, *811*, 9–45. doi: 10.3897/zookeys.811.28712

The eBMS database in numbers

As of November 2020, the latest version available is eBMS v3.2. This version includes data from 20 schemes that are officially part of the eBMS partnership and five additional datasets that have contributed to the European indicators produced by ABLE (Task 1 report). Butterfly counts are documented at the species per section level, the smallest unit available in BMS protocol. Each section being part of a unique monitoring transect that can vary in length and number of sections. When considered at the section level, the eBMS database v3.2 contains nearly 14 million counts (13,928,931), spanning over a 52-year period (1976 to 2018).

Before 1990, BMS data are available for the UK. Other regions have starting monitoring butterflies with national BMS after 1990 (*e.g.*, The Netherlands - 1990, Belgium Flanders – 1991 and Spain Catalonia – 1994). For this reason, we decided to present metrics computed for the period spanning between 1990 and 2018. In this section, we also decide to present the eBMS data aggregated at the transect level, collating all sections that form a BMS transect. This approach is consistent with the method used to compute indices and trends within the ABLE project and by most National Butterfly Monitoring Schemes.

Since 1990, volunteers have recorded more than 5 million butterfly count events documented at the species level in the eBMS database [4]. These counts have been recorded over more than 900,000 monitoring events (*e.i.*, BMS transect visits). In terms of sampling effort, these visits took place in over 10,816 locations (transects), distributed across 22 countries[5] and 25 schemes (Figure 1).

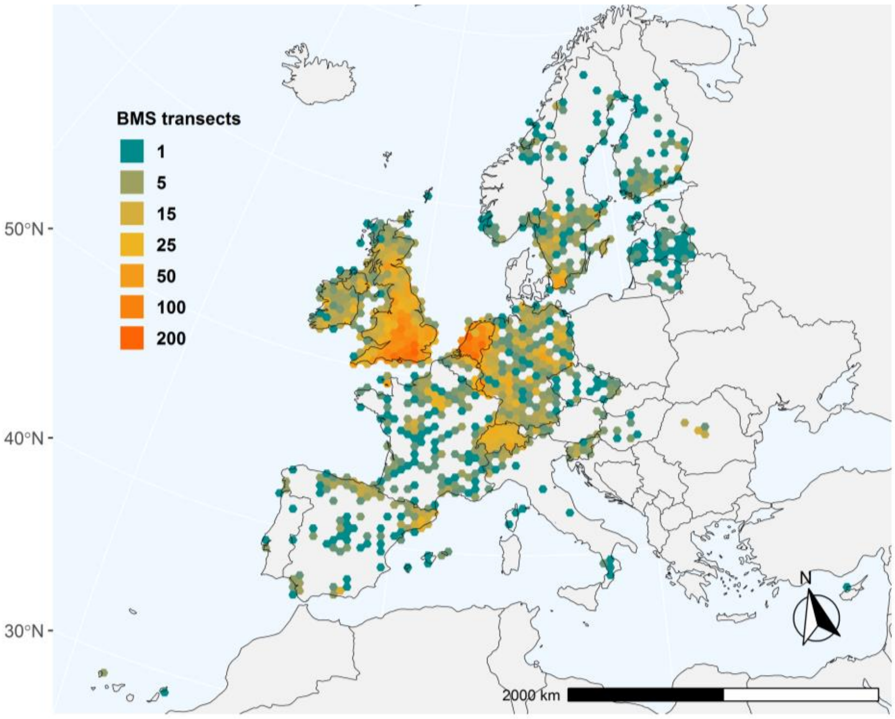


Figure 1. The density of Butterfly Monitoring transects visited per 50km grid across all schemes that have contributed to the eBMS database. Densities are calculated from sites that have been visited at least once since 2000.

[4] Note that 169, 449 counts are available for the UKBMS for the period between 1976-1990.

[5] Note that 19 of the 22 countries are EU27 member states.

Together, these monitoring sites represent a network of 17,333 km where volunteers record (or have recorded) abundance of butterflies on a weekly basis. As of today, this army of skilled volunteers has identified and recorded 312 of the 496 species observed on the European continent [6,7]. Although the protocol and the extent of the BMS contributing to the eBMS database do not capture all species, the species monitored are proportionally representative of their distribution among families and the phylogenetic tree (Figure 2).

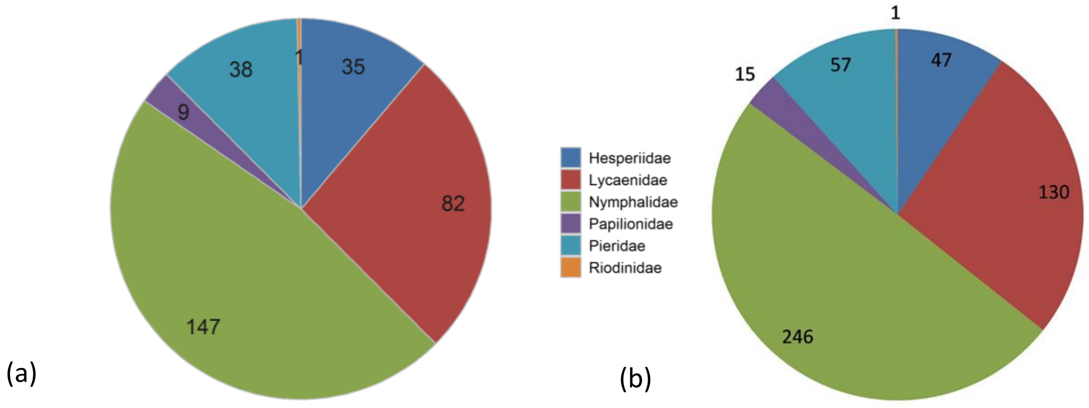


Figure 2. The number of butterfly species by family with data included (a) within the European Butterfly Monitoring Scheme (eBMS) database and, (b) for the whole of Europe [7].

As of 2020, we count on 25 Butterfly Monitoring Schemes that actively contribute with their data to the eBMS database and monitoring data for Europe continues to increase year-on-year (Figure 3).

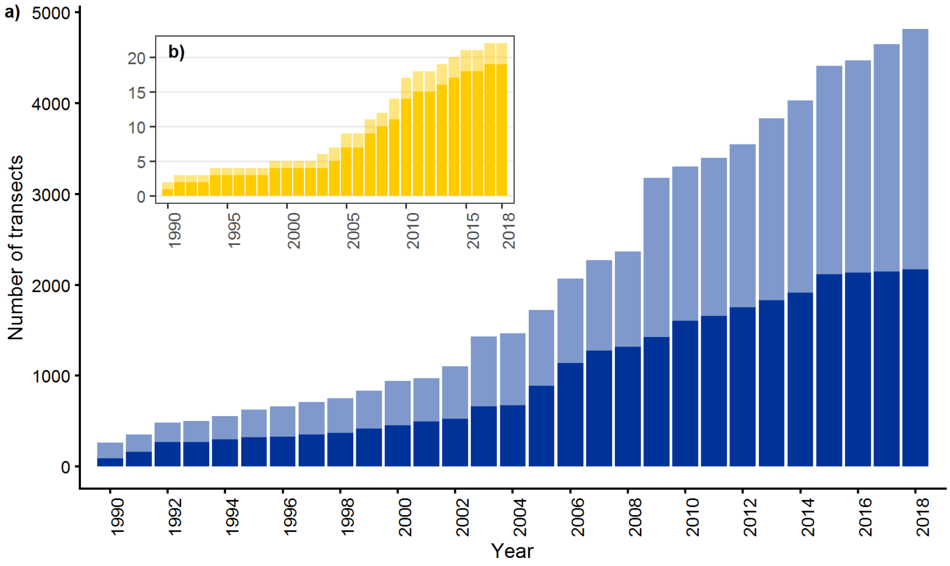


Figure 3. a) Number of Butterfly Monitoring transects that contributed to the European Butterfly Indicator (1990-2018), where schemes within EU27 are represented in dark blue and non-EU schemes in light blue. b) Number of Schemes contributing, with 19 BMS from EU27 member state.

[6] Maes D, Wiemers M, Verovnik R, Warren M, Brosens D, Desmet P (2020). National checklists and red lists for European butterflies. Version 1.1. Research Institute for Nature and Forest (INBO). Checklist dataset https://doi.org/10.15468/ye7whj

[7] Wiemers, M., Balletto, E., Dincă, V., Fric, Z. F., Lamas, G., Lukhtanov, V., … Verovnik, R. (2018). An updated checklist of the European Butterflies (Lepidoptera, Papilionoidea). *ZooKeys*, *811*, 9–45. doi: 10.3897/zookeys.811.28712

With more than 10,000 transects, BMS transects cover a large portion of the European landscape. Together, BMS transects contained in the eBMS database provide a fair representation of the different land cover that characterise EU27 member states (Figure 4) and the continent (Table 1). While most land cover types are well represented in the eBMS data, arable land and forest are underrepresented. Not surprisingly, butterfly monitoring taking place in the agricultural landscape is generally conducted near pastures and heterogeneous agricultural area.

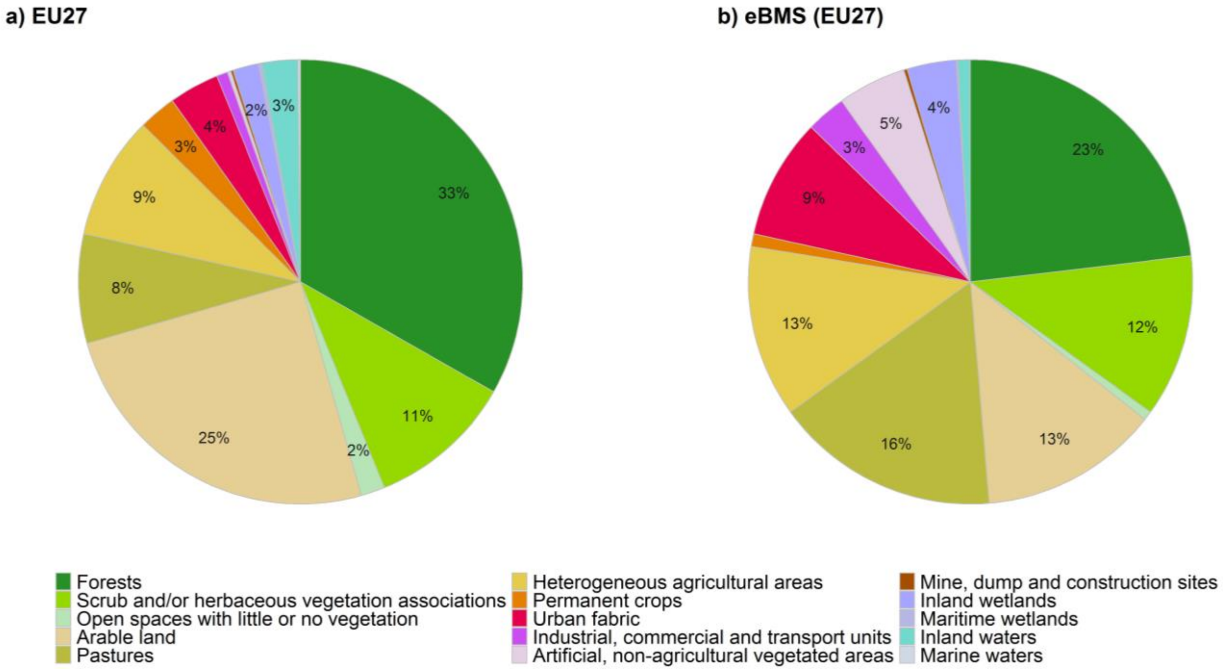


Figure 4. The coverage of CORINE land cover classes (Level-2) across EU27 countries and b) the portion covered by Butterfly Monitoring transects across EU27 countries. Data were extracted from CORINE Land Cover 2018 (v.100 m resolution), using a regular grid (10 km) distributed across EU27 countries and the centroid or each Butterfly Monitoring transect.

Because volunteer citizen scientists are the main contributors to the eBMS database, it is not surprising that urban areas and non-agricultural vegetate areas are both overrepresented in the data (Figure 4 & Table 1). Several schemes have developed alternative protocols to limit these biases, adopting reduced effort protocols or compensating with professionals monitoring to cover underrepresented areas. To a certain level, these biases can also be controlled statistically by weighing the contribution of each transect to produce representative indices and indicators.

Table 1. Land cover coverage expressed in percentage observed across Europe, EU27 and the coverage sampled along Butterfly Monitoring transects located within EU27 country and all countries contributing to the eBMS database. CORINE Land Cover 2018 (100 m resolution), see Figure 4.

CLC LEVEL-1	CLC LEVEL-2	Europe	EU27	eBMS27	eBMS
Forest and semi- natural areas	Forests	**31.96**	**33.24**	**23.13**	**22.44**
	Scrub and/or herbaceous veg. association	11.25	10.60	11.83	11.30
	Open spaces with little or no vegetation	3.90	1.75	0.66	1.16
Agricultural areas	Arable land	**23.42**	**24.94**	13.03	17.22
	Pastures	8.42	7.94	**16.37**	**20.20**
	Heterogeneous agricultural areas	8.15	8.98	12.52	7.24
	Permanent crops	2.40	2.74	0.92	0.48
Artificial surfaces	Urban fabric	**3.55**	**3.66**	**8.79**	**7.72**
	Industrial, commercial and transport units	0.78	0.80	2.88	2.21
	Artificial, non-agricultural vegetated areas	0.32	0.25	5.00	5.49
	Mine, dump and construction sites	0.18	0.19	0.24	0.35
Wetlands	Inland wetlands	**2.56**	**1.87**	**3.58**	**2.66**
	Maritime wetlands	0.29	0.24	0.11	0.51
Water bodies	Inland waters	**2.63**	**2.59**	**0.90**	**0.98**
	Marine waters	0.21	0.21	0.02	0.04

Next steps – eBMS development

One of the main challenges for the eBMS database is to improve our capability to document and validate habitat type across monitoring schemes. This is particularly challenging due to the complexity of reaching a consensual classification of standardised micro-habitat across schemes and European regions. By working together with national schemes and improving the integration of a growing amount of detailed Earth Observation products will help us better inform the context of butterfly occurrence across their distribution range and inform their response to environmental change at both the local and the continental scale.

Chapter 3 / Tools for analysis of butterfly monitoring scheme data (rbms)

In ABLE, we developed an R library, also known as R packages, to organise and distribute a collection of functions (codes) especially tailored to analysis Butterfly Monitoring count data. With the **rbms** package, our aim is to facilitate the implementation of statistical and mathematical methods, allowing scientists and national co-ordinators to compute standardised and robust abundance indices from butterfly counts collected by skilled volunteers.

Because butterfly counts are characterised by strong temporal patterns (phenology) related to their specific life cycle[8]. Accounting for these patterns is particularly important when dealing with time-series that contain missing data and that have been sampled over large and heterogeneous areas[9]. To derive robust abundance indices from such data, members of the ABLE project have developed statistical methods and protocols to derive seasonal patterns from repeated counts and inform (inpute) values for missing data[10]. By borrowing strength from observed regional patterns, the method is particularly efficient for analysing butterfly count data at the national or international levels.

With the **rbms** package, we implemented the methods in a set of flexible and computationally performant R functions that offer state-of-the-art computation of annual butterfly indices, confidence intervals and trend estimates. The open-source R environment has often been associated with a steep learning curve, but the development of increasingly intuitive integrated development environments (IDE) and the rapidly growing documentation have removed many of its initial hurdles. These developments have contributed to make R the preferred environment for developing statistical software for analysing ecological data. This allows the **rbms** package to be fully integrated into a rich ecosystem of existing tools and workflows, giving its users the freedom to efficiently manipulate, visualise and analyse their data for producing informative metrics used to assess and report about status and trends of butterfly. The flexibility of the **rbms** package allows its user to work with other tools developed by the community (*e.g.*, **rtrim**, **BRCindicators**), and develop analyses to gain new insights on butterflies' ecology and their response to environmental change.

Although we initially developed the **rbms** package with European monitoring schemes in mind, version 1.0.2 offers enough flexibility to enable its use in other regions where seasonal patterns and monitoring season might differ (*e.g.*, April to Sept. or Nov. to June). The package allows its user to adapt and apply the adequate method for modelling count data (*e.g.*, Poisson, overdispersed Poisson

[8] Roy, D. B., & Sparks, T. H. (2000). Phenology of British butterflies and climate change. Global Change Biology, 6(4), 407–416. doi: 10.1046/j.1365-2486.2000.00322.x

[9] Schmucki, R., Pe'er, G., Roy, D. B., Stefanescu, C., Van Swaay, C. A. M., Oliver, T. H., … Julliard, R. (2016). A regionally informed abundance index for supporting integrative analyses across butterfly monitoring schemes. *Journal of Applied Ecology*, *53*(2), 501–510. doi: 10.1111/1365-2664.12561

[10] Dennis, E. B., Morgan, B. J. T., Freeman, S. N., Brereton, T. M., & Roy, D. B. (2016). A generalized abundance index for seasonal invertebrates. *Biometrics*, *72*(4), 1305–1314. doi: 10.1111/biom.12506

or negative binomial) and set minimum requirements for data quality (*e.g.*, the minimum number of sites, visits, observations).

Tutorials and training

Together with the development of the R package, we have developed a set of tutorials for using **rbms** and provide the necessary know-how to analyse BMS data. At this stage, we developed two vignettes, one where we demonstrate how to derive the flight-curve using the **rbms** package and a second showing how to calculate abundance indices, trends and confidence interval from butterfly data. These vignettes (Annexe II) are actively maintained and freely available from the **rbms** website.

On April 1st 2020, ABLE organised a two-day workshop where the developers of the **rbms** packages presented its functionality and answered questions of users. This training event, which was initially intended to take place at the Future of Butterflies Conference, in Wageningen (The Netherlands), was organised online due to travel restrictions. The event was free and open to all BMS co-ordinators and people interested in BMS data analysis. We counted more than 40 participants attending the two-day event and participants appreciated the hand-on examples and the thorough explanation, including both the technical and theoretical aspects of the methods implemented in the **rbms** package. Although the online format of the workshop limited our capacity to have more informal discussions with users, it had the benefit of opening the event to more participants and facilitated the documentation of the event that is now freely available via the workshop website https://butterfly-monitoring.github.io/bms_workshop/. All recorded sessions are also available for viewing from the resource section in the ABLE website.

Next steps - rbms development

While the current version (v.1.0.2) of the **rbms** package enables its user to analyse BMS data and compute essential indices to produce national statistics, single species trends and multi-species indicators, several functionalities could be added to complete the toolbox. Among them, the user would greatly benefit from extended functionality for handling spatial data effectively within the package. This added functionality would allow computing indices and indicators for a spatial subset (regions) and help including weighting to correct for sampling bias (*e.g.*, land cover or geographic).

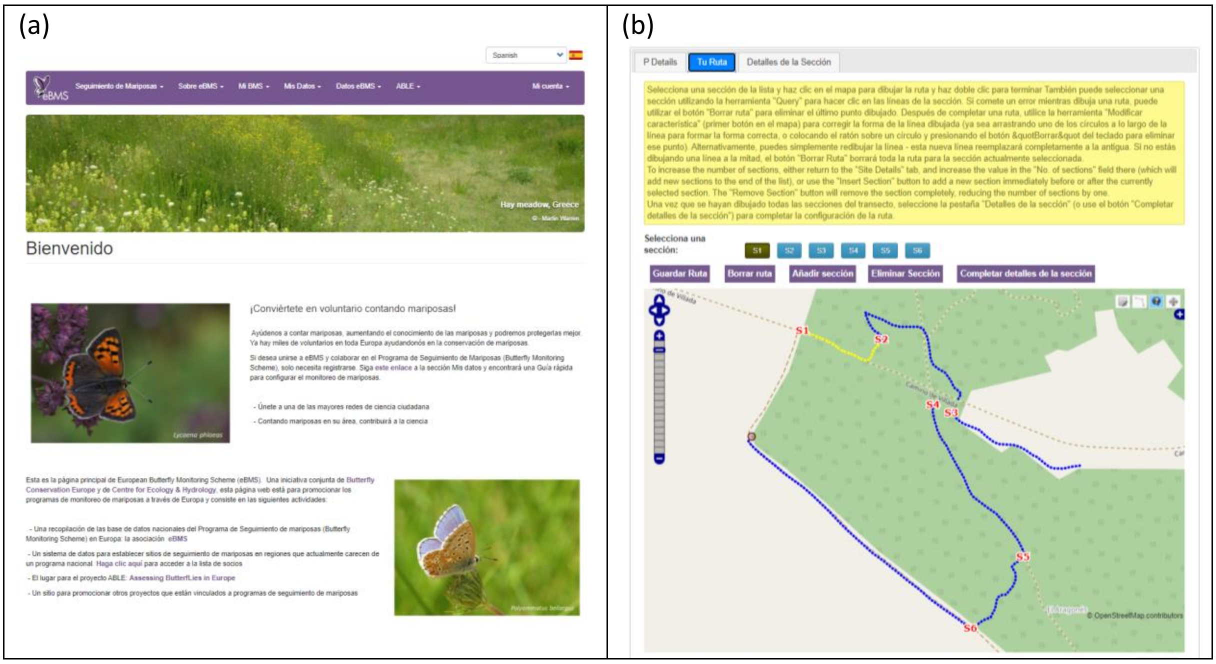
Data and results visualisation functionalities could be improved substantially. By adding such functionalities to the **rbms** package, we would help the production of standardised figures and maps, two components that would help comparison of data and trends between schemes.

A third element that was raised when consulting with users is the value of having a companion package with spatially explicit data product such as the weekly butterfly phenology for the area and the time period covered in the eBMS database. Other essential data product could be made available through the companion package or via R functions giving access to data existing remotely.

Chapter 4 / Online data capture for butterfly monitoring scheme data

A website based upon the Drupal Content Management was developed to support the eBMS, and the ABLE project – www.butterfly-monitoring.net. A key element is an online recording system to facilitate the recording of butterfly transects, and integration with a new timed count protocol developed during this project and implemented within the ButterflyCount mobile application (further details below). The online transect input system is based upon the open-source software Indicia which is managed through GIT, via the public repository https://github.com/indicia-team and extensive documentation is viewable at https://indicia-docs.readthedocs.io/en/latest/. Indicia has been used for the capture of Butterfly Monitoring Scheme data in the UK, Ireland and Luxembourg for several years and was adapted for wider use. It is also widely used in the UK for capture of biological recording and monitoring data for a range of taxa – e.g. iRecord as a tool for capture of opportunistic wildlife sightings, the UK National Plant Monitoring Scheme, the UK BeeWalk monitoring scheme for bumblebees. It has been adapted for wider use throughout Europe via the eBMS website https://butterfly-monitoring.net/mydata. Issue tracking and software updates for the eBMS transect system is managed through GIT via the repository: https://github.com/BiologicalRecordsCentre/able/issues.

Screenshots to illustrate some of the key features of the online data input system is given below (Figure 5). The website is multilingual and configured for 16 languages (English, Bulgarian, Croatian, Czech, Dutch, French, German, Greek, Hungarian, Italian, Polish, Portuguese, Slovenian, Spanish, Swedish and Turkish).



¡Conviértete en voluntario contando mariposas!

Ayúdenos a contar mariposas, aumentando el conocimiento de las mariposas y podremos protegerlas mejor. Ya hay miles de voluntarios en toda Europa ayudandonós en la conservación de mariposas.

Si desea unirse a eBMS y colaborar en el Programa de Seguimiento de Mariposas (Butterfly Monitoring Scheme), solo necesita registrarse. Siga este enlace a la sección Mis datos y encontrará una Guía rápida para configurar el monitoreo de mariposas.

- Únete a una de las mayores redes de ciencia ciudadana
- Contando mariposas en su área, contribuirá a la ciencia

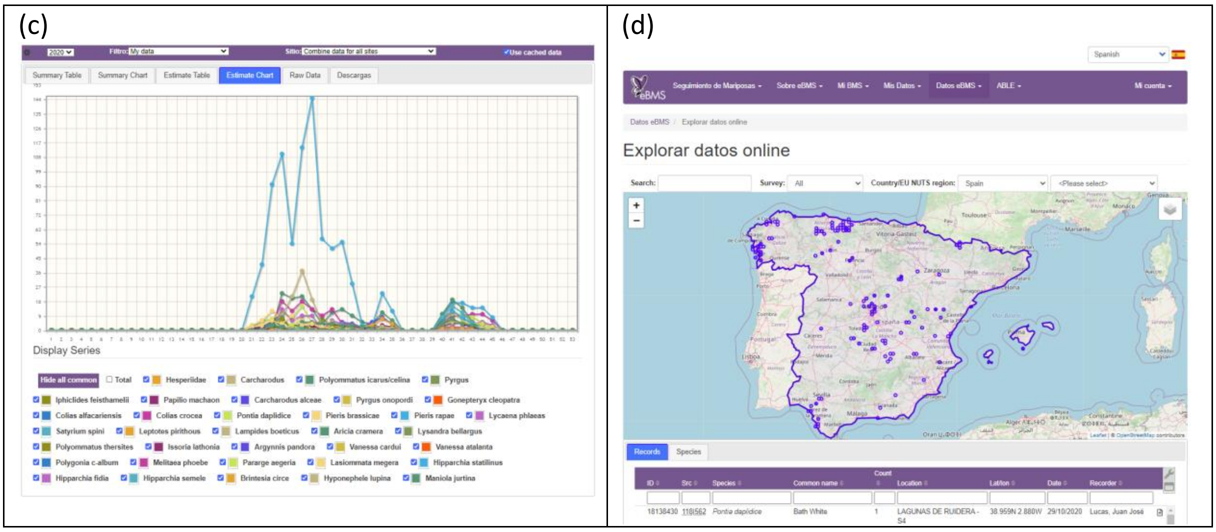
Esta es la página principal de European Butterfly Monitoring Scheme (eBMS). Una iniciativa conjunta de Butterfly Conservation Europe y de Centre for Ecology & Hydrology, esta página web está para promocionar los programas de monitoreo de mariposas a través de Europa y consiste en las siguientes actividades:

- Una recopilación de las base de datos nacionales del Programa de Seguimiento de mariposas (Butterfly Monitoring Scheme) en Europa: la asociación eBMS
- Un sistema de datos para establecer sitios de seguimiento de mariposas en regiones que actualmente carecen de un programa nacional. Haga clic aquí para acceder a la lista de socios
- El lugar para el proyecto ABLE: Assessing ButterfLies in Europe
- Un sitio para promocionar otros proyectos que están vinculados a programas de seguimiento de mariposas

Selecciona una sección de la lista y haz clic en el mapa para dibujar la ruta y haz doble clic para terminar También puede seleccionar una sección utilizando la herramienta "Query" para hacer clic en las líneas de la sección. Si comete un error mientras dibuja una ruta, puede utilizar el botón "Borrar ruta" para eliminar el último punto dibujado. Después de completar una ruta, utilice la herramienta "Modificar característica" (primer botón en el mapa) para corregir la forma de la línea dibujada (ya sea arrastrando uno de los círculos a lo largo de la línea para formar la forma correcta, o colocando el ratón sobre un círculo y presionando el botón &quotBorrar&quot del teclado para eliminar ese punto). Alternativamente, puedes simplemente redibujar la línea - esta nueva línea reemplazará completamente a la antigua. Si no estás dibujando una línea a la mitad, el botón "Borrar Ruta" borrará toda la ruta para la sección actualmente seleccionada.

To increase the number of sections, either return to the "Site Details" tab, and increase the value in the "No. of sections" field there (which will add new sections to the end of the list), or use the "Insert Section" button to add a new section immediately before or after the currently selected section. The "Remove Section" button will remove the section completely, reducing the number of sections by one.

Una vez que se hayan dibujado todas las secciones del transecto, seleccione la pestaña "Detalles de la sección" (o use el botón "Completar detalles de la sección") para completar la configuración de la ruta.



2020 Filtro: My data Sitio: Combine data for all sites ☑ Use cached data

Explorar datos online

Search: Survey: All Country/EU NUTS region: Spain <Please select>

ID	Src	Species	Common name	Count	Location	Lat/lon	Date	Recorder
18138430	118I562	*Pontia daplidice*	Bath White	1	LAGUNAS DE RUIDERA - S4	38.959N 2.880W	29/10/2020	Lucas, Juan José

Figure 5 Screenshots illustrating functionalities of the eBMS online recording system for butterfly transects

The website has been particularly useful in helping new schemes develop without requiring major investment in IT facilities, e.g. used to consolidate the Luxembourg and Spain BMS and to help support new schemes in Italy, Portugal and Austria (Figure 6).

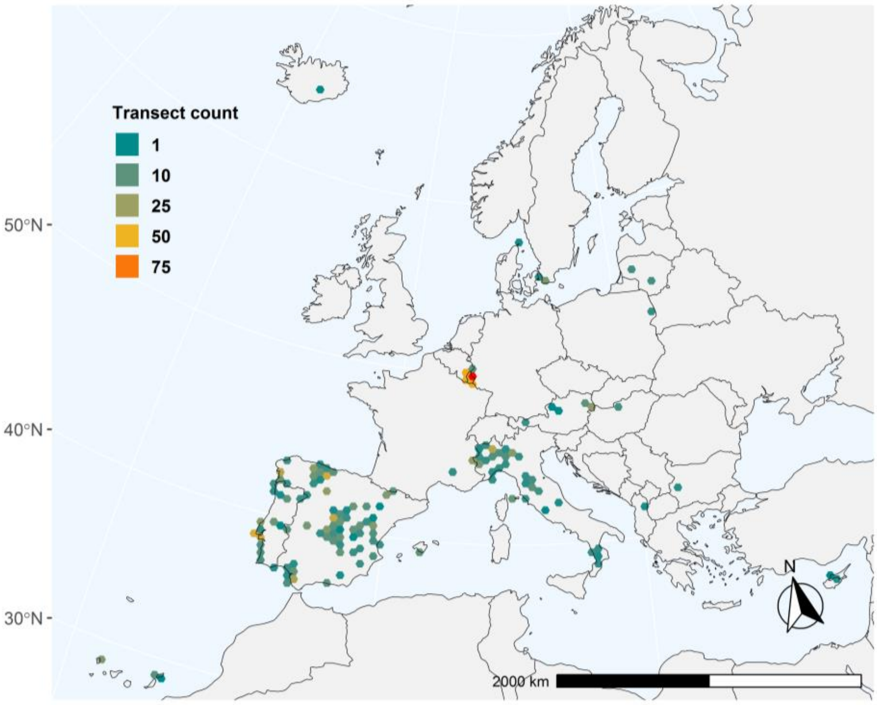


Figure 6. The number of butterfly transect samples recorded per 50 km via the eBMS online data entry system.

Historical data has been uploaded for the Spain BMS to consolidate all the scheme data into a single accessible location. Data from more than 200 transect sites is included within the website for 2018, 2019 and 2020. Data submitted has grown steadily in recent years (Figure 7a) although the number of sampling visits per transect site was noticeably reduced in 2020, particularly in Spain due to the impact of restrictions on accessing the countryside due to the Coronovirus pandemic (Figure 7b). The number of species recorded on transects continued to increase as the spatial coverage of monitoring expanded in all schemes (Figure 7c).

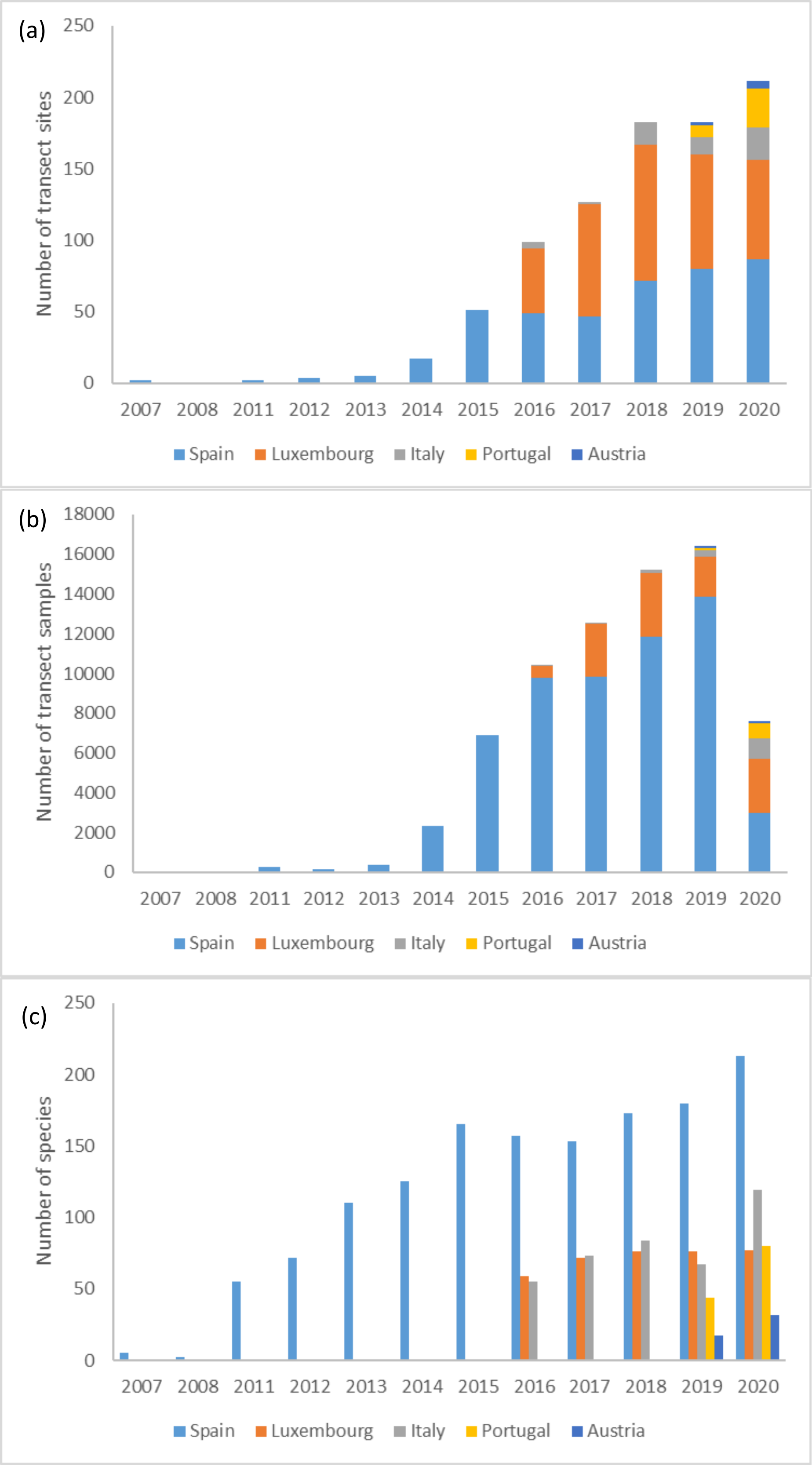


Figure 7. Data within the eBMS online data capture system for (a) number of transect sites, (b) number of sampling visits and (c) number of species recorded each year.

Chapter 4 / Mobile applications for butterfly monitoring in Europe

Introduction

A major priority of the ABLE project has been to build capacity to support the development of national butterfly monitoring schemes, as the most effective approach to monitoring. We recognise the challenges in establishing such schemes based on 'traditional' sampling approaches of butterfly transects that are fixed sites visited at high frequency each year (ideally each week of the year) and over time (consistent high level of sampling year-on-year). The scientific value of such standardised schemes such as butterfly monitoring transect schemes has been widely recognised, but such programmes typically require a high level of central support. They also have much greater barriers to participation and therefore rely on fewer dedicated, skilled volunteers; the abilities of these participants to undertake biodiversity monitoring may be comparable with those of professional scientists.

Expanding butterfly monitoring more widely across the EU, therefore, requires additional, complementary approaches that provide standardised sampling of butterfly abundance data but require less effort and expertise from volunteers, thereby offering greater opportunity to expand the spatial coverage of butterfly monitoring across EU Member States. Through the ABLE project, we have therefore developed a complementary ***timed count protocol*** for butterfly monitoring that shares some of the benefits of both butterfly transects and relatively unstructured opportunistic (presence-only) sightings.

The design and promotion of the timed count protocol have been purposefully kept as simple as possible to minimise barriers to participation. The protocol involves counting all butterflies seen within a fixed amount of time (15 minutes) with associated information on the area searched – through a GPS track or an area. A complete list of the butterflies seen during timed counts will provide absence information (non-detection) that is highly valuable for statistical analysis. The timed count has been integrated within the mobile application (ButterflyCount) to enable the method to be used anywhere in Europe at any time, and will be run exclusively using online systems (including the eBMS website to support the mobile application) for efficiencies and to attract younger audiences. We see particular merit in targeting the method at remote locations (where regular transects are logistically difficult), as well as in urban areas where there is a large target audience. There is growing research evidence that simpler timed count schemes such as developed through ABLE for butterflies, given sufficient data, can produce comparable estimates of trends as structured schemes. This has been demonstrated for birds in North America (comparing eBird transect checklists with North American Breeding Bird Survey data, (Munson et al., 2010) and for butterflies in the UK (comparing a short-duration citizen science project – Big Butterfly Count with a long-running, standardised UK Butterfly Monitoring Scheme, (Dennis, Morgan, Brereton, Roy, & Fox, 2017).

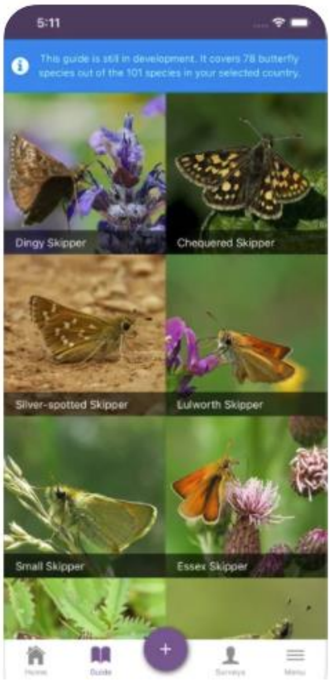
ButterflyCount mobile application

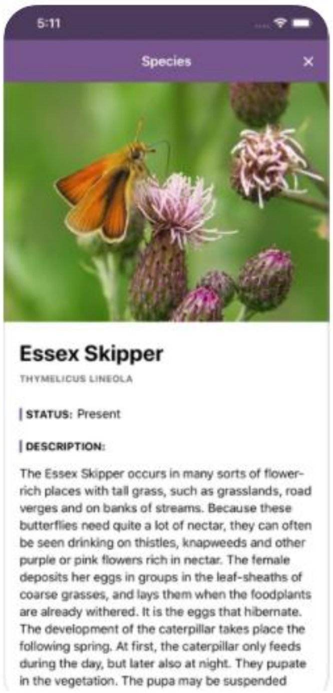
The ButterflyCount mobile application has been developed and tested across Europe (https://butterfly-monitoring.net/ebms-app) during the ABLE project. The app has been translated into 11 languages (English, Lithuanian, Hungarian, Swedish, Spanish, Finnish, French, Russian, German, Italian, Dutch). Translations have been co-ordinated through a dedicated project within the Transifex system that enables multiple translators to contribute – BCE partners have voluntarily provided all translations using this system.

The app enables data on butterfly abundance to be captured via the timed area count protocol (15-minute counts) designed for this project. The timed count protocol involves the app tracking any route walked or entered as an area on a map, with a list of butterflies seen over a 15 minutes period. The app includes the latest checklist of European butterflies (Wiemers et al. 2018), plus, additional aggregate taxa for species that cannot be reliable identified to species in the field – a total of 504 taxa. A guide to butterflies is included within the app with photos and descriptions for most species. Common names are included for a number of languages, including English, Dutch, Swedish, German. To expand the potential of the app to collect data for other insect groups, full checklists are also included for moths, dragonflies and bumblebees.

The ButterflyCount mobile application also includes a survey mode to enter butterfly counts for traditional transect routes. This includes options to record the weather conditions during sampling visits; the same weather information is also collected for both 15-minute timed counts. To simplify the data collection for recorders, the weather conditions are filled in automatically using the nearest available meteorological station, using a service from https://openweathermap.org/current.

The timed area count complements BMS transects to enable data to be collected from a wider audience (e.g. butterfly recorders who do not walk BMS transects) and from additional locations (e.g. urban areas, farmland, remote areas) in order to complement BMS transect that predominantly sample protected areas. App users can map and download their data via the eBMS website through dedicated website reports (https://butterfly-monitoring.net/elastic/my-records). National co-ordinators can download all data for their region via a dedicated web page. Detailed instructions for using the ButterflyCount app are available via the eBMS website; example screenshots are given below (Figure 8).


This guide is still in development. It covers 78 butterfly species out of the 101 species in your selected country.



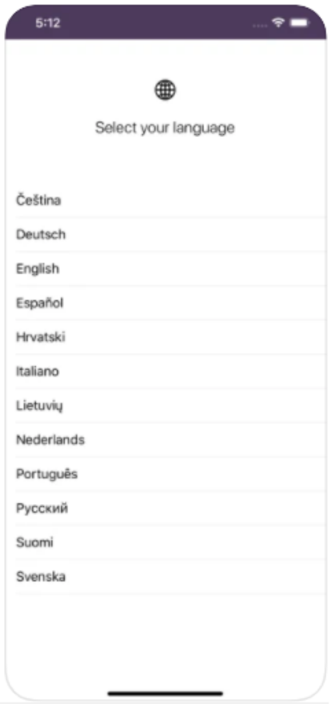
Essex Skipper

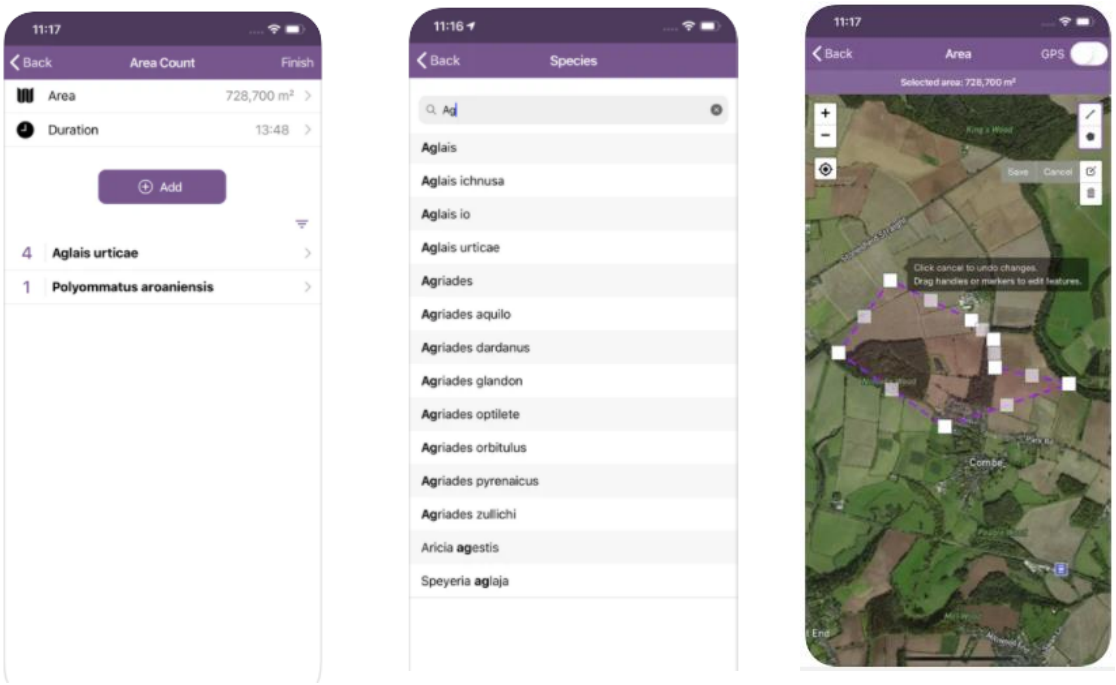
THYMELICUS LINEOLA

STATUS: Present

DESCRIPTION:

The Essex Skipper occurs in many sorts of flower-rich places with tall grass, such as grasslands, road verges and on banks of streams. Because these butterflies need quite a lot of nectar, they can often be seen drinking on thistles, knapweeds and other purple or pink flowers rich in nectar. The female deposits her eggs in groups in the leaf-sheaths of coarse grasses, and lays them when the foodplants are already withered. It is the eggs that hibernate. The development of the caterpillar takes place the following spring. At first, the caterpillar only feeds during the day, but later also at night. They pupate in the vegetation. The pupa may be suspended





11:17

Back | Area Count | Finish

Area 728,700 m²

Duration 13:48

Add

4 Aglais urticae

1 Polyommatus aroaniensis

11:16

Back | Species

Ag

Aglais

Aglais ichnusa

Aglais io

Aglais urticae

Agriades

Agriades aquilo

Agriades dardanus

Agriades glandon

Agriades optilete

Agriades orbitulus

Agriades pyrenaicus

Agriades zullichi

Aricia agestis

Speyeria aglaja

11:17

Back | Area | GPS

Selected area: 728,700 m²

Save | Cancel

Click cancel to undo changes.
Drag handles or markers to edit features.

Figure 8. Example screenshots from the ButterflyCount mobile application

The app was formally launched in early 2020 after field testing by the ABLE project team and BCE contacts. Further field testing was disrupted due to restricted access to the countryside during the Coronavirus pandemic. Despite this, the app was downloaded over 700 times (~400 via apple devices and ~350 via android) and, as of the end of October 2020, a total of 9912 butterfly counts were submitted via the app from across Europe (Figure 9).

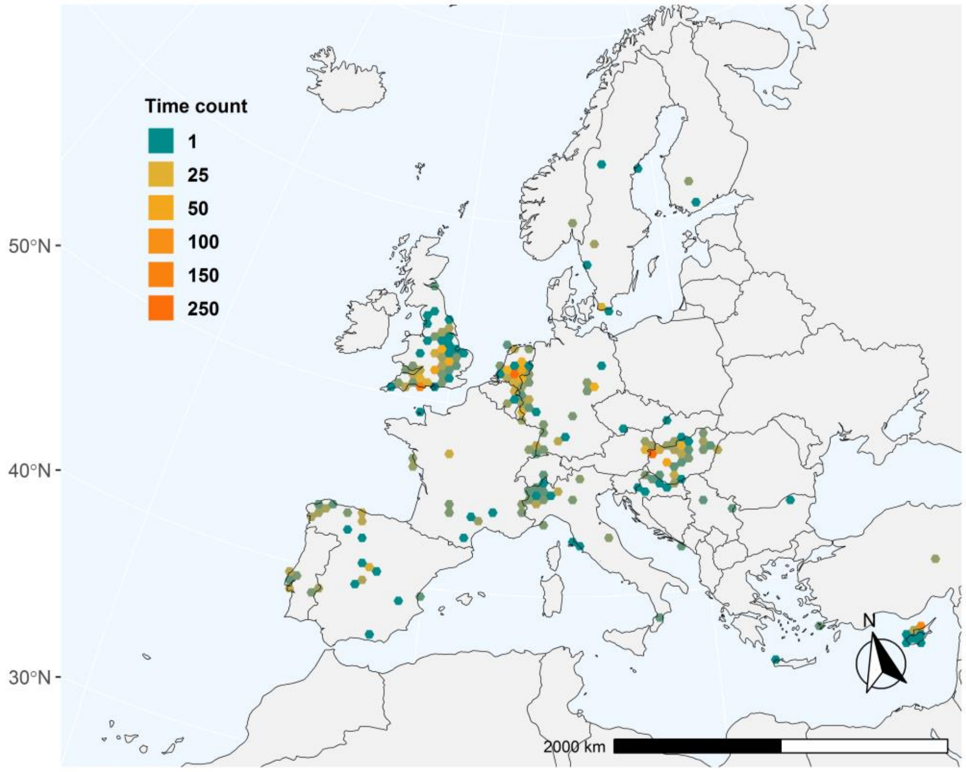


Figure 9. The number of timed counts (15 minutes) recorded per 50 km via the ButterflyCount mobile application

Annexe I. eBMS data model

The eBMS database is built in PostgreSQL, a powerful open-source object-relational database system. The eBMS data model provides a solution for including essential details about butterfly counts from the most local (section along transect) to the wider continental scales. In this data model, ensure that the data are fully traceable to enable efficient reporting and verification with the source (i.e. National BMS). Although fully traceable, the database contains no personal information on recorders as their ID is anonymised.

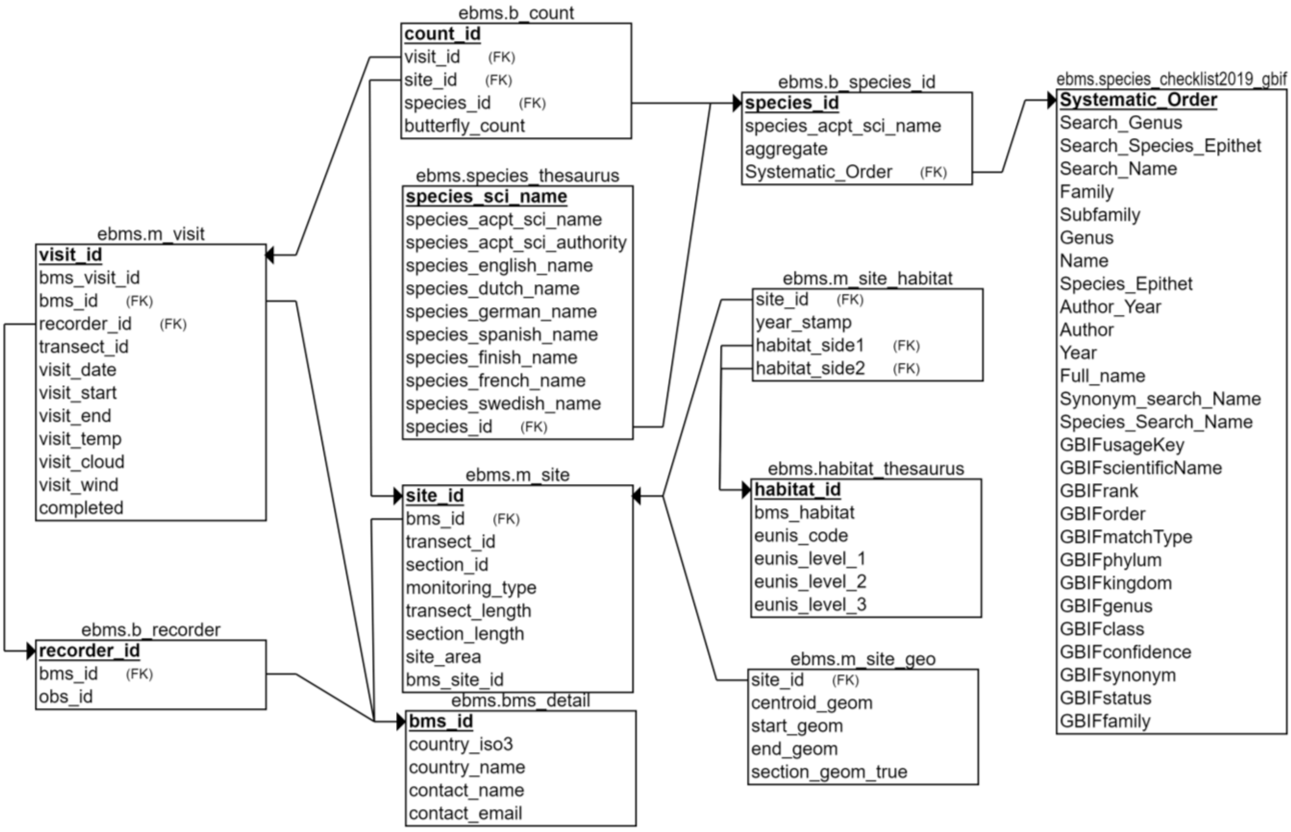
Database structure

The eBMS database is a set of ten tables related to each other to inform about the abundance, the taxonomy, the geography and the habitat of each collected data. The tables are divided into three broad categories; (1) **B**utterfly count, (2) **M**onitoring context, and (3) **T**hesaurus.

In eBMS v3.0, we follow that most recent checklist available for European butterfly (Wiemer *et al.* 2018). For habitat classification, we used the EUNIS habitat classification defined to level-3. The use of comprehensive thesaurus enables constant update without having to alter the original data sets. With this approach, we can also include taxonomic aggregates in our data model.

All scripts and necessary data for constructing the eBMS data model are available at https://github.com/butterfly-monitoring/ebms_data_model

Entity Relationship Diagram



Annexe 1. Entity Relationship Diagram (ERD) depicting the data model of the eBMS database, Version 3.0.

Ontology and table definition

m_visit

Column	Variable type	Description
visit_id (PK)	integer (serial)	Visit ID for a specific monitoring event in eBMS
bms_visit_id	Integer	Visit ID for a specific monitoring event used in the original BMS (source data)
bms_id (FK)	varchar (255)	Code of the National BMS (e.g., UKBMS)
recorder_id (FK)	integer	eBMS anonymous code for recorder
transect_id	varchar (255)	Unique ID for monitoring transect (e.g., UKBMS.368)
visit_date	date	Date with no time of day (YYYY-MM-DD)
visit_start	time	Time of day when specific monitoring event started
visit_end	time	Time of day when specific monitoring event ended
visit_temp	integer	The temperature in degree Celsius
visit_cloud	integer	Percentage cloud cover (0-100)
visit_wind	integer	Wind speed measured with the Beaufort wind scale (0-12)
completed	boolean	If the monitoring walk was completed

b_count

Column	Variable type	Description
count_id (PK)	integer (serial)	Unique ID of the butterfly count
visit_id (FK)	integer	Unique ID of the visit
site_id (FK)	integer	Unique ID of the site
species_id (FK)	integer	Unique ID of the species
butterfly_count	integer	Number of butterflies recorded

b_species_id

Column	Variable type	Description
species_id (PK)	integer (serial)	ID code for the species accepted name
species_acpt_sci_name	varchar (255)	Scientific name accepted in Fauna Europea
Aggregate	Boolean	Is an aggregate of multiple species (ID > 1000)
Systematic_Order (FK)	Integer	ID used in the checklist 2019

species_thesaurus

Column	Variable type	Description
species_sci_name (PK)	varchar (255)	Scientific name used by the National BMS
species_acpt_sci_name	varchar (255)	Accepted scientific name in checklist 2019
species_acpt_sci_authority	varchar (255)	Accepted authority in checklist 2019
species_english_name	varchar (255)	Common name in English
species_dutch_name	varchar (255)	Common name in Dutch
species_german_name	varchar (255)	Common name in German
species_spanish_name	varchar (255)	Common name in Spanish
species_finish_name	varchar (255)	Common name in Finish
species_french_name	varchar (255)	Common name in French
species_swedish_name	varchar (255)	Common name in Swedish
species_id (FK)	integer	ID code for the accepted scientific name.

m_site_habitat

Column name	Variable type	Description
site_id (FK)	integer	Unique ID for monitoring site in the eBMS database (e.g., section)
year_stamp	date	Year of the site habitat classification
section_length	numeric	Length of the section in meter
site_area	numeric	Area of the site in square meter
habitat_side1	integer	Habitat classification on one side of the site
habitat_side2	integer	Habitat classification on the other side of the site

m_site

Column	Variable type	Description
site_id (PK)	integer (serial)	Unique ID for monitoring site in the eBMS database (e.g., section)
bms_id (FK)	varchar (255)	Name of the National BMS (e.g., UKBMS
transect_id	varchar (255)	Unique ID for monitoring transect (e.g., UKBMS.368)
section_id	integer	ID for a specific section along a transect
monitoring_type	varchar (255)	Monitoring protocol used where 1=point, 2=area, 31=normal transect, 32=single species transect, 33=egg count plott

m_site_geo

Column name	Variable type	Description
site_id (FK)	integer	Unique ID for monitoring site (e.g., section)
centroid_geom	geometry (point)	Geometry at the centroid of a monitoring unit (e.g., points, section, area) *
start_geom	geometry (point)	The geometry of the start point *
end_geom	geometry (point)	The geometry of the end point *
section_geom_true	boolean	Is geometry accurate at the section level

*** Projection system used: EPSG:3035 (ETRS89 / ETRS-LAEA)**

habitat_thesaurus

Column	Variable type	Description
habitat_id (PK)	integer (serial)	Numeric code for the habitat defined in national BMS
bms_habitat	text	Name of the habitat used in the National BMS
eunis_code	varchar (255)	EUNIS 3 levels habitat code
eunis_level_1	varchar (255)	EUNIS level 1 habitat name
eunis_level_2	varchar (255)	EUNIS level 2 habitat name
eunis_level_3	varchar (255)	EUNIS level 3 habitat name

b_recorder

Column	Variable type	Description
recorder_id (PK)	integer (serial)	Unique anonymised identifier used in eBMS
bms_id (FK)	varchar (255)	Code for the National BMS (e.g. UKBMS)
obs_id	varchar (255)	Recorder ID code used in the National BMS

bms_detail

Column	Variable type	Description
bms_id (PK)	varchar (255)	Code used to identify the National BMS (e.g., UKBMS)
country_iso3	varchar (255)	Country ISO 3-letter code
country_name	varchar (255)	Country full name
contact_name	varchar (255)	Name of the main contact person that signed the agreement on behalf of the National BMS
contact_email	varchar (255)	Email of the main contact person for the National BMS

species_checklist2019_GBIF *

Column	Variable type	Description
Systematic_Order (PK)	integer	ID used in the checklist 2019
Search_Genus	varchar (255)	The scientific name of the genus of the taxa to search
Search_Species_epithet	varchar (255)	The scientific name of the species epithet of the taxa to search
Search_Name	varchar (255)	The scientific name of the taxa to search
Family	varchar (255)	The scientific name of the family of the taxa
Subfamily	varchar (255)	The scientific name of the subfamily of the taxa
Genus	varchar (255)	The scientific name of the genus of the taxa
Name	varchar (255)	The scientific name of the taxa
Species_Epithet	varchar (255)	The scientific name of the species epithet of the taxa
Author_Year	varchar (255)	Taxonomic author and year
Author	varchar (255)	Taxonomic author
Year	integer	Taxonomic year
Full_name	varchar (255)	Full scientific names (Genus, species, Author, year)
Aggregate	boolean	Is an aggregate of species
Synonym_search_Name	boolean	Is a synonym
Species_search_Name	varchar (255)	Search as species concept
GBIFusageKey	integer	GBIF key for the matching taxa
GBIFscientificName	varchar (255)	Scientific name from the GBIF backbone
GBIFrank	varchar (255)	Taxonomic level (SPECIES)
GBIForder	varchar (255)	The scientific name of the order in which the taxon is classified
GBIFmatchType	Varchar (255)	NONE, EXACT, FUZZY or HIGHERRANK matching used
GBIFkingdom	varchar (255)	The full scientific name of the kingdom in which the taxon is classified
GBIFgenus	varchar (255)	The scientific name of the genus in which the taxon is classified
GBIFclass	varchar (255)	Taxonomic group (INSECT)
GBIFconfidence	integer	Level of confidence (0-100)
GBIFsynonym	boolean	Is a synonym
GBIFstatus	varchar (255)	Accepted or Synonym
GBIFfamily	Varchar (255)	The scientific name of the family in which the taxon is classified

* All GBIF fields are extracted via the GBIF species API, using the `rgbif` R package.

Annexe II. Getting started with the `rbms` package

Phenology – from counts to flight curve

In this tutorial, we show how to fit the flight curve function on butterfly counts recorded on a weekly base. We use the R functions implemented in the **`rbms`** package and the data bundle within the same package. The data are genuine Butterfly Monitoring Scheme counts with transect visit dates. The flight curve computation is based on spline fitted on the counts collected across multiple sites and standardised to sum to 1 (area under the curve is one).

Load package and data included in the package

```
library(rbms)


##  Welcome to rbms, version 1.0.2
##  This package has been tested by users, but is
##  in active development and feedbacks are welcome
##  https://github.com/RetoSchmucki/rbms/issues


data(m_visit)
data(m_count)
```

Here, the visit and count data are both packaged in data.table format but can also be provided as data.frame. The function converts them into data.table because this format is more efficient for handling large data sets. While the input format can vary, the header names need to be consistent, and some columns are essential for the functions to work.

Visit data represent the visit date at which each site was visited, and butterflies were monitored. If no butterfly was observed during a visit, the abundance for that specific visit would be set to zero [0]. This allows to subset positive non-zero counts from the count data set, which result in smaller objects to handle. The visit data may contain many columns, but only two are essential for the function.

1) **SITE_ID** (can be numbers of characters and treated as a non-numeric factor)
2) **DATE** (by default, the format is "%Y-%m-%d" (e.g., 2019-11-28]). If a different the format needs to be specified, use the argument DateFormat)

```
##        SITE_ID       DATE
##     1:       1 2000-04-07
##     2:       1 2000-04-19
##    ---
## 12139:     193 2004-09-15
## 12140:     193 2004-09-28
```

Count data must be provided in columns with specific headers; more column can be provided, but rbms only use the following four:

1) **SITE_ID**
2) **DATE**
3) **SPECIES**
4) **COUNT**

```
##       SITE_ID       DATE SPECIES DAY MONTH YEAR COUNT
##    1:       1 2000-04-07       2   7     4 2000     5
##    2:       1 2000-04-19       2  19     4 2000     3
##   ---
## 5306:     193 2004-06-19       2  19     6 2004     2
## 5307:     193 2004-07-10       2  10     7 2004     1
```

Organise the data to cover the time-period and monitoring season of the BMS

With the visit and count data, we need to merge them into a new data.table object that covers the entire time-series of interest. This time-series is structured with the start and end date of the monitoring season and define on a weekly or a daily base (resolution of the flight curve). In a first step, we initialise the time-series with day-week-month-year information.

```
ts_date <- rbms::ts_dwmy_table(InitYear = 2000, LastYear = 2003, WeekDay1 =
'monday')
```

We add the monitoring season to the new time-series, providing the StartMonth and EndMonth arguments. The definition of the monitoring season can be refined with more arguments (StartDay, EndDay). We also define the resolution of the time-series (weekly or daily), where TimeUnit = 'w' will compute the flight curve based on weekly counts. In the alternative, 'd' is used for building a daily based flight curve. The ANCHOR argument adds zeros (0) before and after the monitoring season. This ensures that the flight curve starts and end at zero.

```
ts_season <- rbms::ts_monit_season(ts_date, StartMonth = 4, EndMonth = 9,
StartDay = 1, EndDay = NULL, CompltSeason = TRUE, Anchor = TRUE,
AnchorLength = 2, AnchorLag = 2, TimeUnit = 'w')
```

NOTE: for species with overwintering adult and early counts, having an Anchor set to zero might sound wrong, and we are currently working on finding an alternative for these cases to represent the flight curve of those species better.

After the monitoring season is defined for a specific time-period and monitoring season, we use the ts_monit_site() function to expand and inform the time-series with the site visits. For this, we use the visit data and link it with the time series contained in ts_season object.

```
ts_season_visit <- rbms::ts_monit_site(m_visit, ts_season)
```

The observed counts are then added to the data.table object, using the count and selecting the species of interest with the argument 'sp'. Here we use the count recorded for species "2" (species names can also be a string of characters).

```
ts_season_count <- rbms::ts_monit_count_site(ts_season_visit,
m_count, sp = 2)
```

The resulting data.table object contains zeros and positive counts recorded along each BMS transects for species "2", over the entire time-series, but only within the focal monitoring season. This time-series contains the week or days of the counts. When the counts are missing because the site was not visited, the count is informed as NA.

Compute the yearly flight curve for the data

With the object constructed above, we can compute the flight curve for each year that has sufficient data. The flight_curve function assumes that the annual phenology shares the same shape across sites.

The objective of this function is to extract this shape that we call the flight curve. Here, we want to impose some minimal threshold to control the quality of the data used to inform this model. This is done by setting the Minimum number of visits MinNbrVisit, the minimum number of occurrences MinOccur and the number of sites MinNbrSite to use to fit the model.

These values can influence the model and its sensitivity will depend on the species and thedata set, but as a minimum requierement, MinOccur should be set >= 2, the MinNbrVisit > MinOccur and MinNbrSite >= 5. These thresholds affect the data that inform your model and the resulting flight curve. When higher values are chosen, fewer sites will be available and if insufficient, the thresholds need to be revised.

```
 ts_flight_curve <- rbms::flight_curve(ts_season_count, NbrSample = 300,
MinVisit = 5, MinOccur = 3, MinNbrSite = 5, MaxTrial = 4, GamFamily = 'nb',
SpeedGam = FALSE, CompltSeason = TRUE, SelectYear = NULL, TimeUnit = 'w')

## [1] "Fitting the flight curve spline for species 2 and year 2000 with 76
sites, using gam() : 2020-11-01 19:24:05 -> trial 1"
```

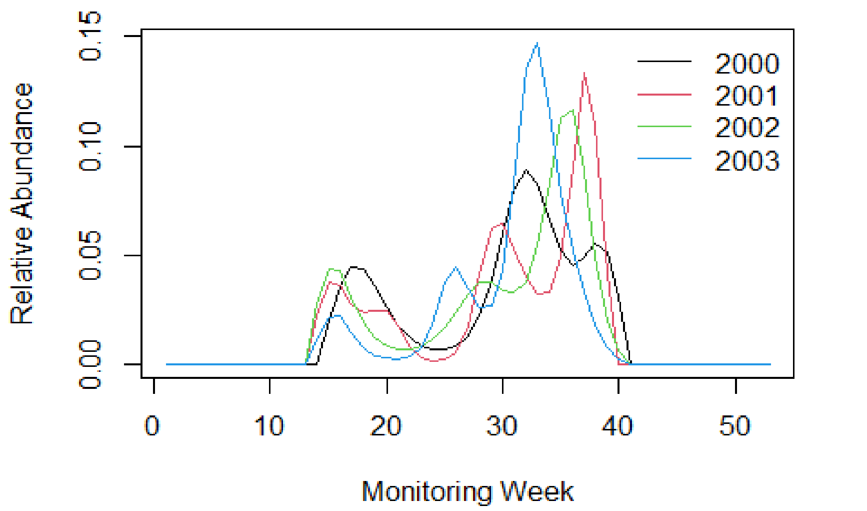
NOTE: for the **flight_curve** function, you will also have to define some parameters for the distribution for the GAM model as well as the maximum number of times to try to fit the model and the number of samples to use. The later will take a random sample from the data set if it contains more site than the number specified.

From the flight_curve() function, we can retrieve a list of 3 objects:

- pheno that contain the standardised phenology curve derived by fitting a GAM model, with a cubic spline to the count data;
- model that contains the result of the fitted GAM model;
- data the data used to fit the GAM model.

We can now extract the pheno object, a data.frame that contains the shape of the annual flight curves, standardised to sum to 1. The flight-curves contained in the pheno object can be visualised with the following line of codes.

```
## Extract phenology part
pheno <- ts_flight_curve$pheno

## add the line of the first year
yr <- unique(pheno[order(M_YEAR), as.numeric(as.character(M_YEAR))])

if("trimWEEKNO" %in% names(pheno)){
  plot(pheno[M_YEAR == yr[1], trimWEEKNO], pheno[M_YEAR == yr[1], NM], type
= 'l', ylim = c(0, max(pheno[, NM])), xlab = 'Monitoring Week', ylab =
'Relative Abundance')
} else {
  plot(pheno[M_YEAR == yr[1], trimDAYNO], pheno[M_YEAR == yr[1], NM], type
= 'l', ylim = c(0, max(pheno[, NM])), xlab = 'Monitoring Day', ylab =
'Relative Abundance')

}
## add individual curves for additional years
if(length(yr) > 1) {
i <- 2
  for(y in yr[-1]){
    if("trimWEEKNO" %in% names(pheno)){
      points(pheno[M_YEAR == y , trimWEEKNO], pheno[M_YEAR == y, NM], type
= 'l', col = i)
    } else {
      points(pheno[M_YEAR == y, trimDAYNO], pheno[M_YEAR == y, NM], type =
'l', col = i)
    }
    i <- i + 1
  }
}

## add the legend
legend('topright', legend = c(yr), col = c(seq_along(c(yr))), lty = 1, bty
= 'n')
```



Collated index - Inputting missing counts

From the flight curve and the observed counts, we can derive expected values for weeks or days where a site has not been monitored. Together, observed and imputed counts are used to compute abundance indices for across sites. Site indices are then used to calculate annual collated indices.

See Get started with rbms to compute the flight curve object used below.

```
library(data.table)
library(rbms)
data(m_visit)
data(m_count)


ts_date <- rbms::ts_dwmy_table(InitYear = 2000, LastYear = 2003, WeekDay1 = 'monday')
ts_season <- rbms::ts_monit_season(ts_date, StartMonth = 4, EndMonth = 9, StartDay = 1, EndDay =
NULL, ComplSeason = TRUE, Anchor = TRUE, AnchorLength = 2, AnchorLag = 2, TimeUnit = 'w')
ts_season_visit <- rbms::ts_monit_site(m_visit, ts_season)
ts_season_count <- rbms::ts_monit_count_site(ts_season_visit, m_count, sp = 2)
ts_flight_curve <- rbms::flight_curve(ts_season_count, NbrSample = 300, MinVisit = 5, MinOccur = 3,
MinNbrSite = 5, MaxTrial = 4, GamFamily = 'nb', SpeedGam = FALSE, CompltSeason = TRUE,
SelectYear = NULL, TimeUnit = 'w')
```

```
## [1] "Fitting the flight curve spline for species 2 and year 2000 with 76 sites, using gam() : 2020-11-
02 18:43:00 -> trial 1"
## [1] "Fitting the flight curve spline for species 2 and year 2001 with 76 sites, using gam() : 2020-11-
02 18:43:01 -> trial 1"
## [1] "Fitting the flight curve spline for species 2 and year 2002 with 88 sites, using gam() : 2020-11-
02 18:43:03 -> trial 1"
## [1] "Fitting the flight curve spline for species 2 and year 2003 with 103 sites, using gam() : 2020-
11-02 18:43:05 -> trial 1"
```

Impute predicted counts for missing monitoring dates

The impute_count() function uses the count data generated from the ts_season_count() function and the outcome of the flight_curve() function to calculate the flight curves (ts_flight_curve$pheno). the function looks for the phenology available (using the nearest year) to estimate and inpute missing values; the extent of the search can be limited by setting the YearLimit parameter, which by default is not restricted. Like in other **rbms**s functions, the imputation can be made on a weekly or daily basis ('w' or 'd').

```
## extract phenology data from the ts_fligh_curve list
pheno <- ts_flight_curve$pheno

impt_counts <- rbms::impute_count(ts_season_count=ts_season_count, ts_flight_curve=pheno,
YearLimit= NULL, TimeUnit='w')
```

The impute_count() function produces a data.table that contains the original COUNT values, a series of IMPUTED_COUNT over monitoring season, TOTAL_COUNT per site and year, TOTAL_NM which is the proportion of the flight curve covered by the visits, and the SINDEX which is the site index and correspond to the sum of both observed and imputed counts over the sampling season.

If the flight curve is missing of a specific year, the `impute_count()` function uses the nearest phenology found. If none is available within the limit of years set by the `YearLimit` parameter, the function will return no `SINDEX` for that specific year.

```
impt_counts_1year <- rbms::impute_count(ts_season_count=ts_season_count,
ts_flight_curve=pheno[M_YEAR != 2001, ], YearLimit= 1, TimeUnit='w')

## Warning in FUN(X[[i]], ...): We used the flight curve of 2000 to compute
## abundance indices for year 2001

impt_counts_0year <- rbms::impute_count(ts_season_count=ts_season_count,
ts_flight_curve=pheno[M_YEAR != 2001, ], YearLimit= 0, TimeUnit='w')

## [1] "No reliable flight curve available within a  0  year horizon of 2001"
```

The site and collated indices

From the imputed count, site index can be calculated for each site or with a filter that will only keep the sites that have been monitored at least a certain proportion of the flight curve. In this example, we set the threshold to 10%, using the `MinFC` parameter.

```
sindex <- rbms::site_index(butterfly_count = impt_counts, MinFC = 0.10)
```

With the site indices, annual collated indices can be estimated by fitting a Generalised Linear Model (GLM), where sites and years are modelled as factors. Here we also use the proportion of the flight curve sampled by the observation as a weight for the GLM. Finally, we also remove all sites where the species was not observed, setting the parameter rm_zero = TRUE, this facilitates the model fit.

```
co_index <- collated_index(data = sindex, s_sp = 2, sindex_value = "SINDEX", glm_weights = TRUE,
rm_zero = TRUE)
```

The collated index computed by the `collated_index()` function can be interpreted as the mean total butterfly count expected on a BMS transect in a given year.

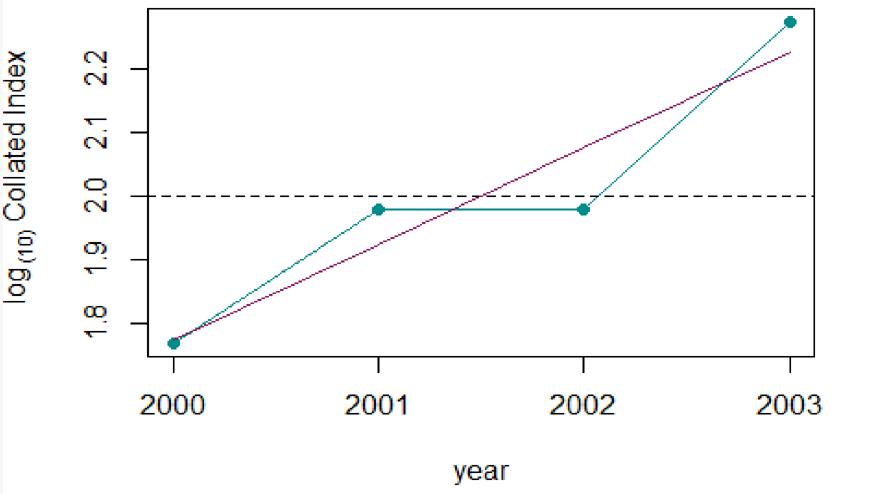
```
## $col_index
##    BOOTi M_YEAR NSITE NSITE_OBS COL_INDEX
## 1:     0   2000   124       108  19.23040
## 2:     0   2001   108       100  31.12038
## 3:     0   2002   114       107  31.13623
## 4:     0   2003   122       113  61.25559
##
## $site_id
##   [1] "1"   "14"  "157" "158" "159" "160" "161" "162" "163" "164" "165" "166"
##  [13] "15"  "167" "168" "169" "170" "171" "172" "173" "174" "175" "176" "16"
##  [25] "177" "178" "179" "185" "193" "51"  "82"  "31"  "41"  "154" "19"  "180"
##  [37] "181" "182" "187" "183" "184" "186" "21"  "23"  "24"  "25"  "26"  "27"
##  [49] "2"   "28"  "29"  "30"  "32"  "34"  "36"  "37"  "39"  "42"  "43"  "3"  ...
```

This index can be scaled on a log(10) scale

```
co_index <- co_index$col_index
co_index_b <- co_index[COL_INDEX > 0.0001 & COL_INDEX < 100000, ]
co_index_logInd <- co_index_b[BOOTi == 0, .(M_YEAR, COL_INDEX)][, log(COL_INDEX)/log(10), by =
M_YEAR][, mean_logInd := mean(V1)]

## merge the mean log index with the full bootstrap dataset
data.table::setnames(co_index_logInd, "V1", "logInd"); setkey(co_index_logInd, M_YEAR);
setkey(co_index_b, M_YEAR)
co_index_b <- merge(co_index_b, co_index_logInd, all.x = TRUE)
```

The log scaled indices can then be plotted with the following code, where the average is centred to two.

```
col_pal <- c("cyan4", "orange", "orangered2")

b1 <- data.table(M_YEAR = co_index_b$M_YEAR, LCI = 2 + co_index_b$logInd -
co_index_b$mean_logInd)
b2 <- data.table(M_YEAR = co_index_b[BOOTi == 0, M_YEAR], LCI= 2 + co_index_b[BOOTi == 0,
logInd] - co_index_b[BOOTi == 0, mean_logInd])


lm_mod <- try(lm(LCI ~ M_YEAR, data = b2), silent=TRUE)


plot(b1, col = adjustcolor( "cyan4", alpha.f = 0.2),
   xlab = "year", ylab = expression('log '['(10)']*' Collated Index'),
   xaxt="n", type = 'n')
axis(1, at = b2$M_YEAR)
points(b2[!is.na(LCI),], type = 'l', lty=2, col="grey70")
points(b2, type = 'l', lwd=1.3, col = col_pal[1])
points(b2[!is.na(LCI),], col= col_pal[1], pch=19)
abline(h=2, lty=2)
if (class(lm_mod)[1] != "try-error"){
  points(b2$M_YEAR, as.numeric(predict(lm_mod, newdata = b2, type = "response")),
  type = 'l', col='maroon4', lwd = 1.5, lty = 1)
 }
```



Bootstrap confidence interval

The confidence interval around the collated indices can be computed from a bootstrap sample, where *n* site indices are randomly resampled *k* time to produce a distribution of the annual collated indices. From this distribution, we can derive the confidence intervals around the collated indices.

We first define *k* bootstrap sample, with replacement, using the function boot_sample(). Here we set *k* to 200 samples, but for a reliable confidence interval, *k* should larger than 1000.

```
bootsample <- rbms::boot_sample(sindex, boot_n = 200)
```

Using the collated_index() function in a loop, with bootstrap samples informing the argument boot_ind, we can now compute *k* collated indies over the entire time-series.

```
co_index <- list()
## for progression bar, uncomment the following
## pb <- txtProgressBar(min = 0, max = dim(bootsample$boot_ind)[1], initial = 0, char = "*",  style = 3)
for(i in c(0,seq_len(dim(bootsample$boot_ind)[1]))){
 co_index[[i+1]] <- rbms::collated_index(data = sindex, s_sp = 2, sindex_value = "SINDEX", bootID=i,
boot_ind= bootsample, glm_weights=TRUE, rm_zero=TRUE)
## for progression bar, uncomment the following
## setTxtProgressBar(pb, i)
}
## collate and append all the result in a data.table format
co_index <- rbindlist(lapply(co_index, FUN = "[[","col_index"))
```

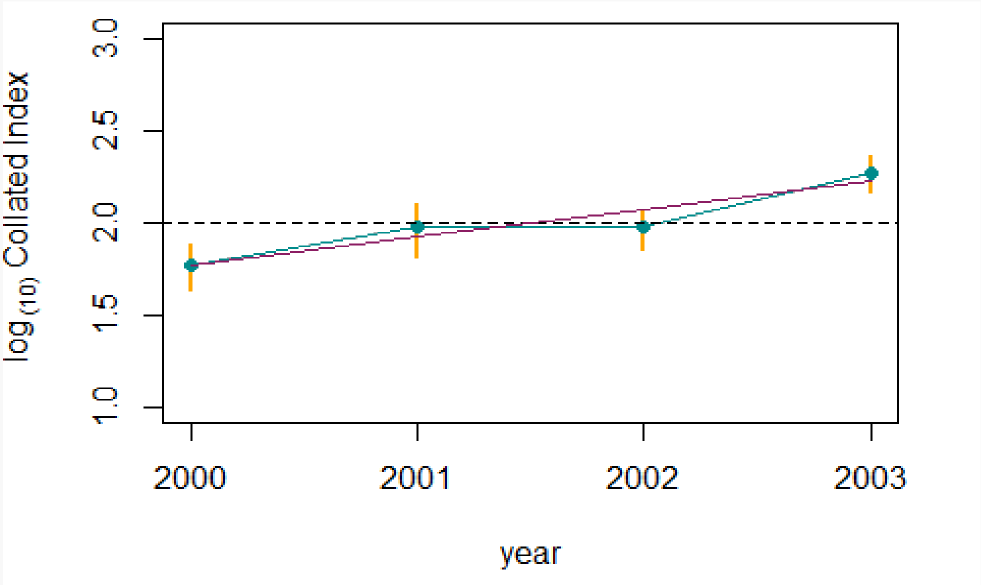
Annual log indices, as well as their average, are then computed from the original sample. Similarly, we compute annual log indices for each bootstrap sample.

```
co_index_b <- co_index[COL_INDEX > 0.0001 & COL_INDEX < 100000, ]
co_index_logInd <- co_index_b[BOOTi == 0, .(M_YEAR, COL_INDEX)][, log(COL_INDEX)/log(10), by =
M_YEAR][, mean_logInd := mean(V1)]

## merge the mean log index with the full bootstrap dataset
data.table::setnames(co_index_logInd, "V1", "logInd"); setkey(co_index_logInd, M_YEAR);
setkey(co_index_b, M_YEAR)
co_index_b <- merge(co_index_b, co_index_logInd, all.x = TRUE)

data.table::setkey(co_index_b, BOOTi, M_YEAR)
co_index_b[ , boot_logInd := log(COL_INDEX)/log(10)]
```

From the bootstrap samples, we can derive a 95% Confidence Interval, using the corresponding percentiles (*i.e.*, 0.025 and 0.975).

```
b1 <- data.table(M_YEAR = co_index_b$M_YEAR, LCI = 2 + co_index_b$boot_logInd -
co_index_b$mean_logInd)


b2 <- data.table(M_YEAR = co_index_b[BOOTi == 0, M_YEAR], LCI= 2 + co_index_b[BOOTi == 0,
logInd] - co_index_b[BOOTi == 0, mean_logInd])
```

```
b5 <- b1[co_index_b$BOOTi != 0, quantile(LCI, 0.025, na.rm = TRUE), by = M_YEAR]
b6 <- b1[co_index_b$BOOTi != 0, quantile(LCI, 0.975, na.rm = TRUE), by = M_YEAR]
lm_mod <- try(lm(LCI ~ M_YEAR, data = b2), silent=TRUE)

## define graph axis limits and color scheme
yl <- c(floor(min(b5$V1, na.rm=TRUE)), ceiling(max(b6$V1, na.rm=TRUE)))

col_pal <- c("cyan4", "orange", "orangered2")

## draw the plot for the selected species
plot(b1, ylim = yl, col = adjustcolor( "cyan4", alpha.f = 0.2),
    xlab = "year", ylab = expression('log '['(10)']*' Collated Index'),
    xaxt="n", type = 'n')

axis(1, at = b2$M_YEAR)

segments(x0 = as.numeric(unlist(b5[,1])), y0 = as.numeric(unlist(b5[,2])),
      x1 = as.numeric(unlist(b6[,1])), y1 = as.numeric(unlist(b6[,2])),
      col = col_pal[2], lwd = 2)
points(b2[!is.na(LCI),], type = 'l', lty=2, col="grey70")
points(b2, type = 'l', lwd=1.3, col = col_pal[1])
points(b2[!is.na(LCI),], col= col_pal[1], pch=19)
abline(h=2, lty=2)

if (class(lm_mod)[1] != "try-error"){
  points(b2$M_YEAR, as.numeric(predict(lm_mod, newdata = b2, type = "response")),
  type = 'l', col='maroon4', lwd = 1.5, lty = 1)
 }
```



Annexe III. rbms documentation

boot_sample()	boot_sample Generate n bootstrap sample of the monitoring sites to be used for each iteration.
check_names()	check_names Verify for the required column names in the data
check_package()	check_package Internal function to verify the required package is installed
check_pheno()	check_pheno Check for the flight curve of a specific year. If the specific year is missing, use the nearest year available within 5 years to impute missing count. A function used in impute_count.
collated_index()	collated_index compute a collated index from the site indices, using a Generalised Linear Model.
collated_index_old()	collated_index_old compute a collated index from the site indices, using a Generalized Linear Model.
df_visit_season()	df_visit_season Link each recorded visit to a corresponding monitoring season; this function is used in ts_monit_site
fit_gam()	fit_gam fit a Generalised Additive Model to butterfly count data along with a temporal variable and accounting for site effect when multiple sites are available.
flight_curve()	flight_curve Compute the annual flight curve from butterfly count data collated across sites.
get_nm()	get_nm Compute the normalised flight curve by fitting a spline in a Generalised Additive Model for one year 'y' to butterfly count data.
get_nny()	get_nny find the nearest year with a computed flight curve
impute_count()	impute_count
initiate_project()	initiate_project Build the initial folder structure for a typical research project
m_count	Toy data set with butterfly count for x species across y sites
m_visit	Toy data set with the date when the sites have been visited for monitoring
set_anchor()	set_anchor Add Anchors of "zeros" at the determined distance on each side of the monitoring season with specific weight (length), this function is used by ts_monit_season()
site_index()	site_index Extract abundance indices per site and year based on flight curve imputation.
ts_date_seq()	ts_date_seq Generate a time-series of dates (per day) from the beginning of a starting year to the end of an ending year.
ts_dwmy_table()	ts_dwmy_table Generate a time-series of dates with day, week, month and year (dwmy) from one initial to an end year.
ts_monit_count_site()	ts_monit_count_site Generate a full-time series of the observed counts, for all sites and each day since a starting and ending years of the defined time-series
ts_monit_season()	ts_monit_season Build a time-series of dates with specific detail about the monitoring season
ts_monit_site()	ts_monit_site Augment the time series in m_season with all sites and visits with "zeros", leaving all non-visited day with and <NA>