



SOFTWARE TOOL ARTICLE

BEpipeR: a user-friendly, flexible, and scalable data synthesis pipeline for the Biodiversity Exploratories and other research consortia

[version 1; peer review: 1 approved with reservations, 1 not approved]

Marcel Glück¹, Oliver Bossdorf², Henri A. Thomassen¹

¹Institute of Evolution and Ecology, Comparative Zoology, Tübingen University, Tübingen, Germany

²Institute of Evolution and Ecology, Plant Evolutionary Ecology, Tübingen University, Tübingen, Germany

v1 First published: 24 Oct 2024, 13:1268
<https://doi.org/10.12688/f1000research.157160.1>
Latest published: 24 Oct 2024, 13:1268
<https://doi.org/10.12688/f1000research.157160.1>

Abstract

Background

Large research consortia can generate tremendous amounts of biological information, including high-resolution soil, vegetation, and climate data. While this knowledge stock holds invaluable potential for answering evolutionary and ecological questions, making these data exploitable for modelling remains a daunting task due to the many processing steps required for synthesis. This might result in many researchers to fall back to a handful of ready-to-use data sets, potentially at the expense of statistical power and scientific rigour. In a push for a more stringent approach, we introduce BEpipeR, an R pipeline that allows for the streamlined synthesis of plot-based Biodiversity Exploratories data.

Methods

BEpipeR was designed with flexibility and ease of use in mind. For instance, users simply choose between aggregating forest or grassland data, or a combination thereof, effectively allowing them to process any experimental plot data of this research consortium. Additionally, instead of coding, they parse most processing information in a user-friendly way through parameter sheets. Processing includes, among others, the creation of a spatially explicit plot-ID template, data wrangling, quality control, plot-wise aggregations, the calculation of derived metrics, data joining to a large composite data set, and metadata compilation.

Open Peer Review

Approval Status ? X

	1	2
version 1	?	X
24 Oct 2024	view	view
1. Elliot Gould , The University of Melbourne, Melbourne, Australia		
2. Matthias Grenié , Universite Grenoble Alpes, Saint-Martin-d'Hères, France		
Laboratoire d'Ecologie Alpine (Ringgold ID: 56837), Grenoble, France		
Any reports and responses or comments on the article can be found at the end of the article.		

Results

With BEpipeR, we provide a feature-rich pipeline that allows users to process Biodiversity Exploratories data in a flexible and reproducible way. This pipeline might serve as a starting point for aggregating the numerous data sets of this and potentially similar research consortia. In this way, it might be a primer for the construction of consortia-wide composite data sets that take full advantage of the consortia's rich information stocks, ultimately boosting the visibility and participation of individual research projects.

Conclusions

The BEpipeR pipeline permits the user-friendly processing and plot-wise aggregation of Biodiversity Exploratories data. With modifications, this framework may be easily adopted by other research consortia.

Keywords

Research consortia, large-scale long-term environmental research, environmental data, data democratization and utilization, reproducibility, R programming language, Biodiversity Exploratories, BExIS



This article is included in the **Bioinformatics** gateway.

Corresponding author: Marcel Glück (marcel.glueck@uni-tuebingen.de)

Author roles: Glück M: Conceptualization, Data Curation, Formal Analysis, Methodology, Project Administration, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; Bossdorf O: Funding Acquisition, Writing – Review & Editing; Thomassen HA: Funding Acquisition, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: German Research Foundation (DFG): 433025806, awarded to HAT and OB.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2024 Glück M *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Glück M, Bossdorf O and Thomassen HA. **BEpipeR: a user-friendly, flexible, and scalable data synthesis pipeline for the Biodiversity Exploratories and other research consortia [version 1; peer review: 1 approved with reservations, 1 not approved]** F1000Research 2024, 13:1268 <https://doi.org/10.12688/f1000research.157160.1>

First published: 24 Oct 2024, 13:1268 <https://doi.org/10.12688/f1000research.157160.1>

Introduction

Large-scale long-term environmental research frameworks such as LTER (Hobbie et al. 2003), TEAM (Rovero and Ahumada 2017), ForestGEO (Anderson-Teixeira et al. 2015, Davies et al. 2021), and the Biodiversity Exploratories (Fischer et al. 2010a, 2010b) are at the forefront of functional biodiversity research. These frameworks are fuelled by well-orchestrated infrastructure projects, unmatched standing scientific expertise, and high-resolution time-series data. This combination of factors allows them to answer some of the most intricate and pressing ecological questions of our time with high statistical power. For instance, they shed light on how land-use shapes biodiversity and ecosystem processes (Allan et al. 2015, Felipe-Lucia et al. 2020, Le Provost et al. 2023), how this gives rise to profound changes in community composition and network interactions (Weiner et al. 2014, Vályi et al. 2015, Blüthgen et al. 2016, Chavarria et al. 2021), and the importance of temporal and spatial heterogeneity in shaping these patterns (Kloss et al. 2011, Allan et al. 2014, Seibold et al. 2019, van Breugel et al. 2019).

Due to their size and the presence of dedicated infrastructure projects, these frameworks continue to benefit from an ever-increasing stock of biological data. For instance, as of 19/03/2024, the Biodiversity Exploratories Information System (BExIS, Chamanara et al. 2021) featured more than 1500 data sets for their experimental forest and grassland plots (EPs). Arguably, while this wealth of information holds great promise for answering even highly intricate research questions, considerable effort is needed to combine these data in a way that allows for their straightforward use. While for a limited number of data sets and at the expense of reproducibility, such processing might be performed in spreadsheet editors such as LibreOffice Calc or Microsoft Excel, this approach becomes increasingly infeasible when more data are incorporated, ultimately asking for a more efficient way of processing. While this often means using programming languages such as R, Python, and Julia, not all ecologists might be used to these languages and learning one can be perceived as daunting (Baker 2017, Custer et al. 2021).

Unsurprisingly, to circumvent these challenges, many research projects within these consortia might rely on a handful of data sets that allow for a straightforward and less time-consuming incorporation into their workflows. By doing so, they might leave out potential data that would have been instrumental in answering their complex scientific questions, ultimately causing a loss in statistical power. Instead, a more compelling approach would be a tool that allows for a user-friendly processing of data sets, rendering the decision between progressing fast or incorporating many data obsolete. To this end, we introduce **BEpipeR**, an R pipeline that allows for the synthesis of EP-level (a)biotic Biodiversity Exploratories data. To maximise its usability and ease of implementation, we purposely limited the amount of coding required. For instance, we allow users to parse most aggregation information through csv files and toggle easily between three aggregation modes (forest, grassland, or combined) that allow for the straightforward processing of data provided by this research framework.

Regardless of the mode selected, BEpipeR performs the following processing: creation of a spatially explicit plot-ID template, data substitution through exact and pattern-based approaches, subsettings, resolving species aggregates issues through fallbacks, data reshaping, variables standardization, mean and median-based outlier detection, data aggregation both within and across data sets, processing and aggregation of climate data generated and extensively pre-processed by TubeDB (Wöllauer et al. 2021); in the following referred to as “BExIS’ climate tool”, normalization by repeated rarefaction, calculating alpha diversity indices, data joining to template, quality control, variables selection by variance inflation factor analyses, and the compilation of metadata from JSON metadata files. Arguably, BEpipeR has the potential to generate large composite data sets in a highly reproducible fashion (Baker 2016). As this might aid the democratization and utilization of available research data, we hope for this pipeline to become a focal point for compiling the vast amount of environmental information generated by the Biodiversity Exploratories and, potentially, similar research consortia.

Methods

Implementation

BEpipeR is written in R v.4.1.1 (R Core Team 2021) and harnesses *renv* v.1.0.3 (Ushey and Wickham 2023) to establish an R project-based reproducible environment. This means that in setting-up the pipeline, all packages that were used to create the pipeline in the first place are automatically installed to a per-project library. These packages include *here* v.1.0.1 (Müller 2020) for a streamlined file and directory referencing, *terra* v.1.7-18 (Hijmans et al. 2022) for spatial processing, *data.table* v.1.14.8 (Barrett et al. 2023), *plyr* v.1.8.8 (Wickham 2011), *Hmisc* v.5.1-1 (Harrell 2023), *tidyverse* v.2.0.0 (Wickham et al. 2019), and *doSNOW* v.1.0.20 (Microsoft Corporation and Weston 2022) for general processing, respectively, *rtk* v.0.2.6.1 (Saary et al. 2017) for rarefaction, *vegan* v.2.6-4 (Dixon 2003) for calculating diversity indices, *usdm* v.2.1-6 (Naimi et al. 2014) for variables selection, and *jsonlite* v.1.8.4 (Ooms 2014) for metadata extraction.

For set-up, we assume the use of **RStudio** integrated development environment (IDE) ([Racine 2012](#)) and a connection to the internet. First, upon downloading the desired release from **GitHub**, the user unzips the compressed pipeline file. Second, the user obtains information on the R version required for running the pipeline by inspecting the top lines of the `renv.lock` file, placed at the root of BEpipeR's directory structure. If the required version is not available on their system, they obtain it from the **Comprehensive R Archive Network** and install it. Additionally, on Windows, they ensure that a compatible version of **RTools** is installed. Third, the user sets the required R version as the default version in RStudio and exits the IDE. Last, BEpipeR's reproducible environment can be unfolded by opening the `BEpipeR.Rproj` file using RStudio, upon which the `renv` package is bootstrapped and all required packages can be installed to the per-project library by typing `'renv::restore()'` and confirming the prompted dialog with 'y'. Subsequently, users may want to increase the number of lines retained in RStudio's console to ensure that all messages generated in running the pipeline are available for post-run inspection. Noteworthy, for visualizing plot locations, the border of Germany must be obtained manually from **GADM** and stored for its use by the pipeline as 'Germany_borders.gpkg' in the pipeline's 'Helpers' directory. For up-to-date set-up instructions, users are referred to the pipeline's GitHub presence.

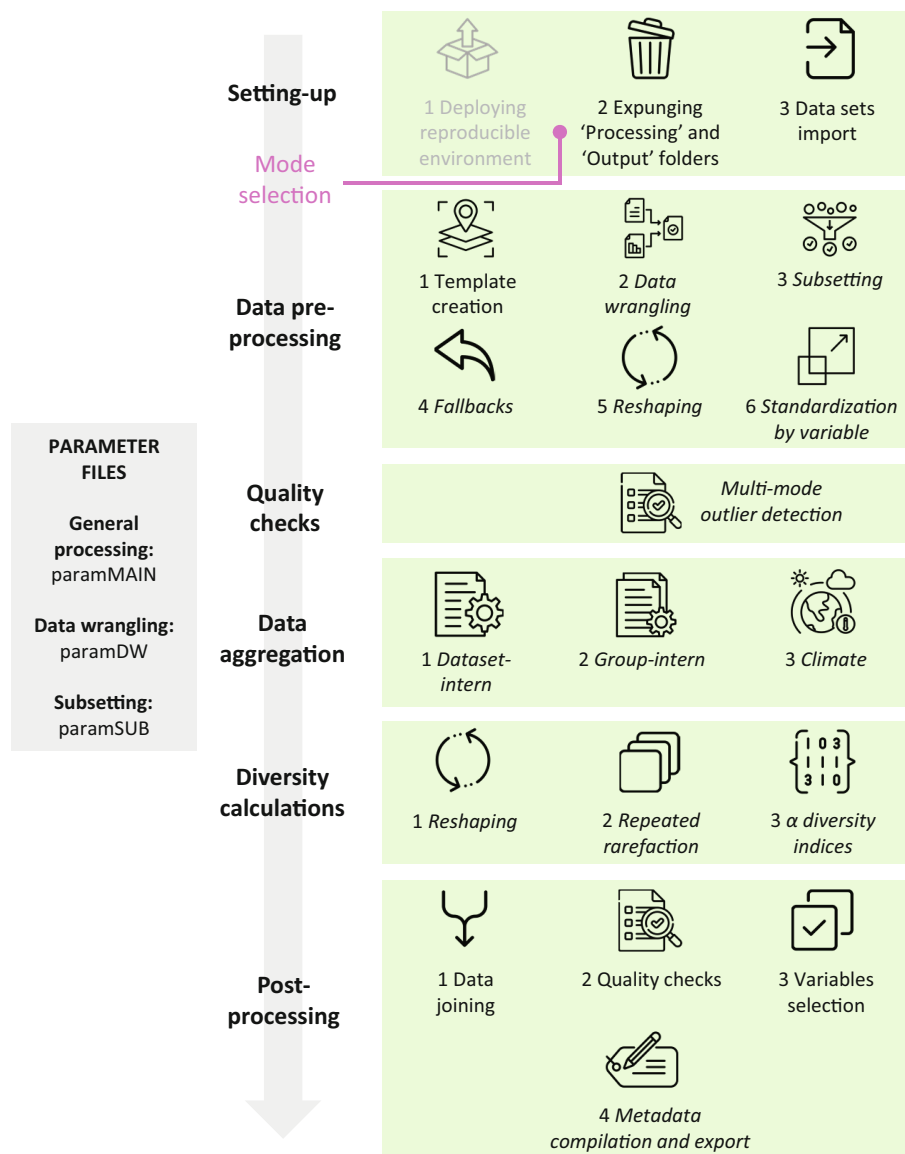


Figure 1. Overview of the BEpipeR pipeline. Included are the parameter files used in its operation (left-hand side), its major processing steps (centre), and their sub-steps (right-hand side; italic: optional). Deploying the reproducible environment in setting-up the pipeline is only performed once, and hence greyed out.

Operation

Parsing information

With few exceptions (see below), BEpipeR's flow of operations ([Figure 1](#)) is controlled through three csv files (paramMAIN, paramDW, and paramSUB) that are used to parse processing information in a user-friendly fashion. Of these three, paramMAIN is the most instrumental and holds the majority of the aggregation information, whereas paramDW and paramSUB are helper files that coordinate the data wrangling (DW) and subsetting (SUB) steps, respectively. As the Excel versions of these files support users in providing processing information through conditional formatting and functions, we suggest that users provide processing information to these versions first, followed by exporting them to csv file format. While we purposely minimized user interventions to the pipeline's code, they could not be avoided completely. Currently, user actions might be required at five points ([Table 1](#)) marked with the comment string "ACTION POTENTIALLY REQUIRED" in the R script. Users are advised to familiarize themselves with these interventions before executing the pipeline productively.

To demonstrate BEpipeR's flow of operations and guide users in interpreting the pipeline's output, we distribute BEpipeR in a 'just-ran' state. This means that the pipeline comes with both exemplary input data and the results produced by processing these files (see Use Cases). As indicated ([Table 2](#)), exemplary input files must be replaced with real-world Biodiversity Exploratories data when using the pipeline productively to ensure the correctness of results.

We support users in providing the processing information required with dictionaries to all three param files that can be found as additional sheets in their respective Excel files. We recommend that users consult this information before starting to work with the pipeline. For exhaustive information on data processing, users are referred to the R script itself, which features comments on the reasoning for each step performed as well as further background information throughout. Additionally, in the following, we provide an in-depth description of the workflow, including details on how to encode the information required, the processing performed, and the output generated, to guide users in familiarizing themselves

Table 1. Information on running parameters to be parsed directly to BEpipeR. The following parameters cannot be parsed through BEpipeR's parameter files. Instead, they must be provided directly to the pipeline's source code at locations marked with the string "ACTION POTENTIALLY REQUIRED".

Variable name	Expected input	Input class	Default value	Function
BEpipeR_mode	Either "forest", "grassland", or "combined" for aggregating forest or grassland data, or a combination thereof, respectively.	string	"combined"	Specifies the mode for processing input data.
CLIM_min_years	An integer without leading zeros.	integer	4	The minimal number of years a variable must have data for to be retained in the climate data set (ID: 19007).
DI_reshape_whitelist	Quoted data set IDs without version information (i.e., base IDs); separated by commas.	string	empty	Prevents the reshaping from long to wide format for the data sets specified.
FQC_plots_to_remove	Quoted EP plot designations with leading zeros; separated by commas.	string	empty	Allows for the exclusion of plots whose inclusion would result in the discarding of many or all variables in excluding variables with any NA (Not Available) values.
VS_protected_variables	Quoted full variable names as provided in the FQC (i.e., quality-controlled) composite data set; separated by commas.	string	empty	Allows for the protection of variables from being excluded through stepwise variance inflation factor analysis.

Table 2. BEpipeR's core directories, their expected/generated main content, and processing-related information. 'Provisioning' describes whether the content is generated automatically or must be provided by the user. Placeholders are surrounded by square brackets. BEXIS: Biodiversity Exploratories Information System.

Directory	(Expected) content and processing information	Provisioning
Helpers	The main parameter files (paramMAIN), data wrangling (paramDW) and subsetting (paramSUB) helpers; the data set used for constructing the spatially explicit plot IDs template (currently data set 20826_7, dummy data provided must be replaced with real Biodiversity Exploratories data); a GeoPackage file with the border of Germany for visualizing plot locations (see Implementation section for more information).	Manual
Metadata	Metadata files to all data sets flagged as included in paramMAIN. For data not obtained through BEXIS' climate tool, this is the corresponding '[baseID]_[version]_datastructure.txt' file. For climate data obtained through BEXIS' climate tool, this equates to its sensor description csv file renamed to match the scheme '[baseID]_[version]_sensor_description.csv'.	Manual
Output	Files generated by BEpipeR throughout its execution (see Table 3). This directory is expunged at the start of each pipeline run.	Automatic
Processing	Files copied here from 'Source' for processing through BEpipeR. This directory is expunged at the start of each pipeline run.	Automatic
R_scripts	The R programming language script of the BEpipeR pipeline.	Automatic
renv	Files required for setting-up and maintaining the pipeline's reproducible environment.	Automatic
Source	All data sets in csv file format to be processed by BEpipeR following their automatic transfer to the 'Processing' directory. Naming scheme: '[baseID]_[version]_data.csv'.	Manual
Temp	Temporary files written by the <i>rtk</i> R package in performing repeated rarefaction. This directory is created by BEpipeR.	Automatic

with the pipeline. Potential abbreviations provided in parentheses after the names of processing steps refer to the prefixes of variable names used in the BEpipeR R script.

Data retrieval, exploration, and curation

Data retrieval from BEXIS database, as well as their exploration and curation are not performed by BEpipeR. Instead, they are performed best by the user on a dataset-to-dataset basis. This approach acknowledges that the decisions on incorporation and processing depend on the user's aims as well as the unique combination of data and metadata. It further allows users to harness existing workflows for inspecting tabular data. Upon examining both data and metadata, the user decides whether the data set at hand should be processed by BEpipeR, and if yes, provides all the information required for its processing to paramMAIN and, if applicable, paramDW and paramSUB (see below). Subsequently, they copy the respective data set in csv file format (named '[baseID]_[version]_data.csv'; square brackets denote placeholders) to BEpipeR's 'Source', as well as its '[baseID]_[version]_datastructure.txt' file to the 'Metadata' directory.

Setting-up

Following the successful deployment of the reproducible environment through *renv* and before executing the pipeline, the user decides on one of three possible processing modes: i) forest, for aggregating the Biodiversity Exploratories' forest data, ii) grassland, for grassland data, or iii) combined, for aggregating both forest and grassland data at the same time. They then provide the corresponding string to the BEpipeR_mode variable in the pipeline's source code (Table 1). Noteworthy, both 'Processing' and 'Output' directories are expunged immediately after the start of each pipeline run to avoid potentially outdated files from being mistaken as up-to-date ones. Subsequently, the 'Processing' directory is populated by copying all csv data sets from the 'Source' directory to this folder. Please note, as data sets are retrieved by their base IDs, BEpipeR does not allow multiple data sets with the same base ID to be present in the 'Processing' directory. If this issue is detected, the user is informed and asked to solve the issue.

Data pre-processing

- 1 **Template creation:** Combining Biodiversity Exploratories data is complicated by two factors. First, most data sets are not spatially explicit per se, meaning they do not feature location information that allows for a straightforward calculation of inter-plot distances. Second, plot information is not provided in a harmonized fashion. This means that the column holding plot designations might be arbitrarily named (e.g., EP, EpPlotID, EP_plot_ID, Useful_EP_PlotID, Plot, PlotID, Plot ID, Plot_ID, or plotid_withzero) and located within the data set. This is further complicated by the presence or absence of leading zeros in plot numbers (e.g., AEW1 vs. AEW01), and two alternative plot encoding schemes (e.g., AEW01 in EP is A18422 in grid plot (GP) encoding). To allow for joining the data regardless of encoding and to maximise its downstream usability, the plot IDs template constructed by BEpipeR holds EP as well as GP designations with and without leading zeros, respectively.

To allow users to seamlessly use the data generated by BEpipeR in downstream spatially explicit statistical frameworks, the pipeline enriches the template with plot location information harmonized to the World Geodetic System 1984 (WGS84, EPSG: 4326) and informs the user about the spatial imprecision introduced by reprojecting location data from DHDN (Deutsches Hauptdreiecksnetz) to this unified coordinate reference system. Subsequently, the csv version of paramMAIN is imported to the R session as 'datasets_table' and filtered for instrumental columns and data sets flagged for inclusion. This table outlines the processing to be performed on each data set and is updated after each major processing step to reflect the progress of the pipeline. Template creation is concluded by various quality checks, including warnings if 'datasets_table' features data sets not found in the 'Processing' directory, or the system-wide memory available might not be sufficient for executing the pipeline (see Minimal system requirements).

- 2 **Data wrangling (DW):** Removing or replacing factually incorrect values is essential in pre-processing data. BEpipeR supports users therein by allowing them to replace or remove these values through exact and pattern-based approaches. To enable this option, the user sets rDW in paramMAIN for the respective data set to 'yes'. Subsequently, they provide additional information to paramDW, the helper file for this operation. This information includes the data set's base ID (Dataset_ID), whether the replacement is value- (Class = value) or pattern- (Class = pattern) based, the value to replace (Value_old), and the value to replace with (Value_new). Noteworthy, the pattern-based approach even allows for the deletion of matching rows by specifying Value_new as NULL. In contrast, value-based row deletions are best performed by subsetting (see next step). Generally, modifications are applied in the order of listing (i.e., from top to bottom). This means that multiple modifications can be applied to the same data set by listing the same base ID in multiple rows of paramDW, each time with a different modification.
- 3 **Subsetting (SUB):** BEpipeR's data wrangling capabilities are enriched by its subsetting function, which allows for the filtering of rows using exact matches. To achieve this, upon setting rSUB for the respective data set in paramMAIN to 'yes', users provide the following information to this operation's helper file, paramSUB: the data set's base ID (Dataset_ID), the name of the column to perform the subsetting on (Subset_variable), the comparison operator to use (Operator), and the entry to retain or remove (Subset_level). As for DW, multiple modifications that are applied consecutively from top to bottom can be requested for the same data set.
- 4 **Fallbacks (FB):** Data sets that establish relationships between taxonomic entities and their abundances often feature taxonomic levels not resolved to completion. For instance, a tree species data set might feature aggregated species, such as *Quercus spec.*, alongside species that were fully resolved. To remedy this issue, BEpipeR allows users to perform fallbacks to more basal (taxonomic) levels. To invoke this operation, users set rFB in paramMAIN to 'yes' in addition to providing the following information: the name of the column to perform the fallback on (FBcol), the separator used to delineate the different levels of information in FBcol (FBsep), and FBsub, the index of the substring of interest. BEpipeR uses this information to string-split the information in FBcol at the separator specified, upon which the substring of interest is retained by its index. Subsequently, BEpipeR sums abundances per plot at the newly generated factor level, effectively collapsing abundances at a more basal (taxonomic) level. Abundance scores harmonized in this way may seamlessly be used in downstream processing steps.
- 5 **Reshaping (RES):** Usually, plot data are most easily processed in wide format, with rows denoting plots and columns representing the respective variables. In this step, BEpipeR allows users to cast data to this format while coding absent combinations as Not Available (NA). To flag a data set for reshaping, the user sets rRES in

paramMAIN to 'yes', upon which they supply the factors column that will be used in constructing new column names to RESvar. Please note: i) Factors not used in reshaping are collapsed by calculating plot-wise means. Hence, the resulting data set features unique plots in the first column, followed by column-wise environmental data. ii) Due to NA as missing combination value, RES is mutually exclusive with calculating alpha diversity indices (DI, see below). Hence, if you would like to calculate these indices, keep the data in long format. Do not reshape climate data obtained through BExIS' climate tool to be processed in CLIM (see below) either.

- 6 **Standardization by variable (STD):** Abundance scores often rely on sampling effort, with differential effort potentially giving rise to differential abundance, preventing meaningful plot-based comparisons. While various data-dependent normalization/standardization approaches exist (e.g., [Weiss et al. 2017](#), [Lin and Peddada 2020](#), [Xia 2023](#)), data sets that feature information on sampling effort are best normalized using this information. To achieve this, BEpipeR allows all numeric variables of a data set to be normalized by information provided in a user-specified column of that data set. To do so, users set rSTD for the respective data set to 'yes' and provide the variable's name to be used for data set-wide standardization (STDvar) to paramMAIN. The result is standardized variables that permit a meaningful and straightforward integration in downstream processing steps.

Quality checks

Multi-mode outlier detection (QC): To support users in their data exploration and to spot potential invalid values, such as un-disclosed numeric NA values or species aggregates that result in artificially high or low abundance scores, BEpipeR performs column-wise outlier detection. This is done upon setting QC in paramMAIN to 'yes'. To avoid false alarms in non-combined mode (i.e., forest or grassland), plots not conforming to the desired ecosystem are excluded in the detection procedure. In combined mode, outlier detection is performed separately for each ecosystem. Currently, BEpipeR features two outlier detection approaches that are based on constructing confidence intervals as multiples of standard deviation (sd, default: 18) around column means and medians. Noteworthy, lower confidence interval bounds are adjusted according to their column means/medians. More specifically, for a given column, if the lower confidence interval bound is negative but the corresponding column mean/median is positive, the respective lower bound is adjusted to zero. This approach acknowledges that many environmental data might be positive and increases the detection sensitivity of BEpipeR at the lower end of the data distribution. Flagged data sets, filtered for columns with potential outliers, are exported to the global R environment for inspection with the naming scheme 'QC_[baseID]_flagged_[MEAN|MEDIAN]_[forest|grassland]'. If justified, outliers may be removed by the user through DW, SUB, or a combination thereof. Note that, by default, BEpipeR will not perform QC on climate data obtained through BExIS' climate tool (base ID: 19007), regardless of what users set QC for this data set to. This is because these data have already been extensively quality-checked; hence, any outliers found are most likely false positives. Future releases of BEpipeR will enhance the outlined detection approaches by utilizing data structure information exclusively obtained through BExIS' API.

Data aggregation

- 1 **Dataset-intern aggregation (DIA):** The aim of this step is to construct data-set and plot-level-wise aggregation metrics, regardless of whether data is provided in long or wide format. To accommodate both data structures, BEpipeR provides two aggregation approaches of which one must be provided to the DIAappr column in paramMAIN upon setting DIA for the respective data set to 'yes'. For either aggregation approach and the grouping variables provided, BEpipeR computes mean, median, sd, and median absolute deviation (mad) values.

DIAappr = 2 allows users to aggregate data in long format. Currently, up to three grouping variables (plot IDs + two non-plot variables) are supported and might be provided to paramMAIN's DIAcol1, DIAcol2, and DIAcol3 columns.

DIAappr = 3 permits the plot-wise aggregation of data in wide format, meaning only plot IDs as grouping variable are currently supported (i.e., both DIAcol2 and DIAcol3 must be kept empty).

- 2 **Group-intern aggregation (GIA):** This step allows users to combine multi-measurement (e.g., multi-year) data split across multiple data sets with the subsequent calculation of summary statistics (mean, median, sd, and mad), while maintaining up to three grouping variables (plot IDs + two non-plot variables). This processing is invoked by setting GIA in paramMAIN to 'yes', followed by providing grouping variables to the GIAcol1, GIAcol2, and GIAcol3 columns. Noteworthy, to reduce file sizes, amplicon sequencing data sets might have been shrunk by i) omitting plot \times taxonomic unit combinations with zero abundance, and/or ii) deleting all-zero

abundance taxonomic units, resulting in deliberately not covering all taxonomic units across all years. The first issue can be corrected for by enabling abundance correction (GIAabcorr = 'yes'), which effectively rebuilds the plot \times taxonomic unit matrix with missing combinations coded as zeros. The second issue is resolved by BEpipeR upon setting taxonomic units correction (GIAtaxcorr) to 'yes', which ensures that all taxonomic units are present across all years (absent units are introduced with all-zero abundance). Critical, in paramMAIN's Group column, users must assign a unique number to the data sets to be combined. In addition, for all focal variables, it is the user's responsibility to ensure that they are shared between the group's data sets and that their order of listing in paramMAIN is identical. Future releases of this pipeline will automatize these steps by falling back to the group's shared variables, followed by their re-organization and processing.

- 3 **Climate (CLIM):** BEpipeR's ability to process environmental data is enriched by its ability to process yearly climate aggregates obtained through BExIS' climate tool (Wöllauer et al. 2021). To obtain data processable by BEpipeR, users choose the following parameters in the web tool's graphical user interface for generating their data: Spatial aggregation: separate plots; Aggregation of time: year; options: 'write all plots in one CSV-File', 'one plot timeseries after another', 'write header in CSV-Files', 'include column "plotID"', Calendar columns: year. Additionally, they request the parameter description file to be included in the zip archive to be generated. We recommend users to set 'quality check of measured values' to '3: physical range + step + empirical check' to obtain climate data that fulfils the highest quality standards. Users are free to choose whether they enable the interpolation of missing values. If they opt to do so, we advise that they request the inclusion of the 'qualitycounter' column in their aggregated climate data, which provides information on the total and interpolated number of data points underlying each of the yearly climate aggregates. The presence of this column in the climate data is used as indicator for BEpipeR to remove weakly supported data points (percentage interpolated > 60%), a step that is skipped if this column is not found in the data. If interpolated information is provided by BExIS' climate tool, users are advised to not re-arrange the data column-wise, as this will break the association between the 'qualitycounter' and data columns. However, row-wise operations, such as the exclusion of undesired years through SUB, are permitted.

To calculate reliable multi-year summary statistics (mean, median, sd, mad, min, and max), BEpipeR allows users to exclude variables not based on a minimal number of data points (i.e., years). By default, this filter is set to four but may be adjusted by users interested in retaining variables that satisfy a more stringent filtering approach (CLIM_min_years, Table 1). These users should keep in mind that, depending on the years they want to obtain temporal coverage over, the replacement of plot HEW02 with HEW51 in 2016 might complicate or even negate the acquisition of long-term time-series climate data. To assist users in this filtering, BEpipeR issues a warning if their strategy is too stringent and results in retaining only few or no climate variables at all.

Diversity calculations

To go beyond a simple description of abundances, we allow users to calculate alpha diversity indices. To do so, they set DI for the respective data set in paramMAIN to 'yes'. First, this triggers the reshaping of the respective data set to wide format with zero as value for combinations not present. This step may be skipped for data sets already in this format by providing their base ID to the DI_reshape_whitelist variable (Table 1). Second, users might opt to normalize the data provided through rarefaction by setting RF in paramMAIN to 'yes' and providing the number of repetitions to perform (as multiple of ten) to RFnrep. We are well aware of the ongoing debate on proper normalization and the alleged shortcomings of rarefaction (McMurdie and Holmes 2014, Schloss 2024). We acknowledged this by purposely deciding on repeated rarefaction, as an extension of normal rarefaction, for the following reasons: i) rarefaction is a highly tractable and easy-to-grasp concept, ii) data normalized through rarefaction might seamlessly be used in calculating alpha diversity indices (e.g., Walters and Martiny 2020, Schloss 2024), iii) rarefaction might still be the most frequently used normalization technique for amplicon data, and implemented in many processing pipelines such as QIIME (Caporaso et al. 2010) and mothur (Schloss et al. 2009), and iv) rarefaction noticeably decreases the discrepancy between OTU and ASV data (Walters and Martiny 2020, Chiarello et al. 2022), allowing for a higher degree of comparability regardless of the type of clustering applied. Most importantly, repeated rarefaction addresses the often-criticised data loss by random subsampling through performing these subsamplings repeatedly, effectively reducing the impact of single stochastic processes in normalizing data (Cameron et al. 2021). Noteworthy, before repeated rarefaction, potential decimal values in the abundance table that resulted from upstream multi-year aggregations are rounded up to the next integer, an approach that prevents positive values smaller than 0.5 from falsely being set to zero (i.e., absence). Users might gauge the success of the normalization by inspecting rarefaction curves and/or slopes exported to the 'Output' directory (Table 3). Following repeated rarefaction and the rounding of potentially produced decimal abundance scores to their nearest integer, BEpipeR computes alpha diversity indices, including species richness, Menhinick (Menhinick 1964), Margalef

(Margalef 1973), Shannon-Wiener (Shannon and Weaver 1949), Simpson (Simpson 1949), and the inverse Simpson index. Because most alpha diversity indices are meaningless for empty sampling units, only plots with non-zero richness are retained for later joining.

Importantly, some data sets may be incorporated into BEpipeR's workflow as they are (i.e., without the need for DIA, GIA, CLIM, RF, or DI). This must be signalled to BEpipeR by setting `AsIt` in `paramMAIN` to 'yes', upon which no aggregation is performed on these data sets. This functionality allows BEpipeR to incorporate highly sophisticated ready-to-use data sets in a straightforward fashion.

Post-processing

- 1 **Data joining (MRG):** Upon ensuring that all data have been processed fully by inspecting the relevant `datasets_table` columns, BEpipeR left-joins all available data to the plot IDs template constructed upstream. For data sets with leftover grouping variables (apparent by the data set having more rows than the plot IDs template), BEpipeR attempts to accommodate these by repeated reshaping to wide format until the data set's number of rows conforms to the expectation, or no more potential grouping variables are found. In the latter case, the user is warned. In joining data, BEpipeR appends complete data set IDs to column headers to allow for a straightforward back-tracing of information to their origin.
- 2 **Quality checks (FQC):** The aim of this processing step is two-fold. First, BEpipeR performs several quality checks to ensure data consistency and the successful execution of upstream processing steps. For instance, it warns if additional rows were introduced in left-joining, plot designations are found in the values matrix, or if duplicated column headers or headers without data set ID are found, and it maximises the data's downstream usability by replacing potential spaces in column names with underscores. Second, it removes undesired information from the composite data set constructed in MRG to prepare the data for variables selection (see below) or direct use. This is achieved by replacing NaN (Not a Number) and Inf (infinite) values with NA and excluding non-numeric columns. The resulting intermediate composite data set (`FQC_env_var_composite_intermediate.csv`, Table 3) might still contain NA cells and mono-value columns. However, it may already be of interest to users who apply statistical frameworks capable of tolerating such input data. This composite data set is processed further by excluding mono-value columns and plots not conforming to the BEpipeR mode specified. Additionally, as some plots might render obtaining a large complete-cases data set difficult by breaking up otherwise continuous long-term time-series data (e.g., HEW51, established in 2016), we allow users to exclude these plots (`FQC_plots_to_remove`, Table 1) before removing any columns with NA values. The resulting complete-cases composite data set is subsequently exported (`FQC_env_var_composite_complete.csv`, Table 3). For users applying statistical frameworks capable of processing multi-colinear data, this file may already be used as input for their analyses.
- 3 **Variables selection (VS):** Understanding the correlation structure underlying explanatory data is pivotal for the thought- and meaningful interpretation of statistical models. In this processing step, we support users in two ways: i) through BEpipeR, we provide information on correlations underlying the complete composite data set produced in FQC, and ii) condense the data to a set of less correlated variables. Insights into the correlation structure are gained by calculating Pearson correlation coefficients (r) and associated false discovery rate-corrected (Benjamini and Hochberg 1995) P values between all variable pairs in `FQC_env_var_composite_complete.csv` (Table 3). This information is further used to warn users if significant ($P < 0.05$) pairwise comparisons with unusually high goodness of fit ($r \sim 1$) are observed, upon which the user decides whether these comparisons are justifiable or instead indicative of issues in upstream data processing.

Reducing the data set to a suite of less correlated variables is achieved by variables selection through variance inflation factor (VIF) analyses. Noteworthy, users might often have justified *a priori* assumptions about focal variables, and hence would like to retain these in their data set for easier model interpretation. We acknowledge this and provide users with the ability to protect their focal variables from removal by supplying their names to the `VS_protected_variables` variable in the script (Table 1). To maximise the downstream usability of the data generated, BEpipeR performs variables selection for a range of VIF thresholds from two to ten, with smaller values denoting a more stringent exclusion approach. For each VIF threshold, multiple files are exported to the 'Output' directory (Table 3).

- 4 **Metadata compilation and export (COMD):** To allow for a straightforward data re-usage, the provisioning of metadata is a cumbersome yet necessary duty to all data scientists. The Biodiversity Exploratories provides

Table 3. Information on the files generated by the BEpipeR pipeline and exported to the 'Output' directory. Placeholders are in square brackets.

Processing step	File name	Description
Rarefaction (RF)	RF_[baseID]_rarefaction_curves_subsample_[subsampleSize].png	Rarefaction curves depicting the relationship between subsample size on the x and richness on the y axis. The vertical line marks the subsample size used for rarefaction that is also provided in the file's name. Horizontal lines visualize plot-based richnesses following a single rarefaction. Generated with <code>vegan's rarecurve()</code> function.
Rarefaction (RF)	RF_[baseID]_rarefaction_slopes_subsample_[subsampleSize].csv	The slopes of rarefaction curves constructed with <code>vegan's rarecurve()</code> function at the subsample size specified in the file's name. Generated with <code>vegan's rareslope()</code> function.
Final quality control (FQC)	FQC_env_var_composite_intermediate.csv	The composite data set constructed by left-joining all data to the spatially explicit plot IDs template with the subsequent replacement of NaN (Not a Number) and Inf (infinite) values with NA (Not Available), as well as the exclusion of non-numeric columns. Metadata columns provide experimental (EP) and grid plot (GP) designations with (PlotID) and without (PlotID) leading zeros, as well as location information in World Geodetic System 1984 (WGS84). This information is followed by the variables produced, with their headers carrying processing information and the full data set IDs they originate from.
Final quality control (FQC)	FQC_env_var_composite_complete.csv	FQC_env_var_composite_intermediate.csv after excluding plots not in concordance with the BEpipeR mode specified, in addition to user-defined ones (Table 1). Mono-value columns and those with NAs have been excluded as well, making this data set a complete-cases one.
Variables selection (VS)	VS_pearson_corrMat.csv	The Pearson's r matrix as produced by <code>Hmisc's rcorr()</code> function for all non-metadata variables in FQC_env_var_composite_complete.csv.
Variables selection (VS)	VS_pearson_numObs.csv	The number-of-observations matrix underlying the values in VS_pearson_corrMat.csv.
Variables selection (VS)	VS_pearson_pVals.csv	The P values matrix to the r values stored in VS_pearson_corrMat.csv.
Variables selection (VS)	VS_corr_flat_complete.csv	A flattened representation of VS_pearson_corrMat.csv and VS_pearson_pVals.csv. Diagonal values as well as false discovery rate (FDR)-corrected P values are provided.
Variables selection (VS)	VS_VIFVIFthreshold_VS_analysed_vars.csv	The names of the variables that underwent variables selection by variance inflation factor (VIF) analysis with the VIF threshold specified. Typically, this is all non-metadata variables from FQC_env_var_composite_complete.csv.
Variables selection (VS)	VS_VIFVIFthreshold_VS_excluded_vars.csv	The variables excluded by <code>usdm's vifstep()</code> function at the VIF threshold specified.

Table 3. *Continued*

Processing step	File name	Description
Variables selection (VS)	VS_VIF[VIFthreshold]_VS_corr_matrix.csv	A Pearson's r matrix for the variables retained by vifstep() at the VIF threshold specified.
Variables selection (VS)	VS_VIF[VIFthreshold]_VS_retained_vars_scores.csv	The VIF scores of variables retained at the VIF threshold specified.
Variables selection (VS)	VS_VIF[VIFthreshold]_VS_composite.csv	FQC_env_var_composite_complete.csv after excluding variables listed in VS_VIF[VIFthreshold]_VS_excluded_vars.csv.
Compiling metadata (CMD)	COMD_metadata_compiled.csv	<p>The compiled metadata of non-metadata variables in FQC_env_var_composite_complete.csv. For each variable, the following information is provided:</p> <ul style="list-style-type: none"> - The data set the variable originates from (with (FullID) and without (BaseID) version suffix). - Its name as extracted from FQC_env_var_composite_complete.csv (Composite_var). - Its name after removing processing suffixes (Composite_var_trimmed), as well as the processing information extracted (Aggr_string_1, Aggr_string_2). - Its metadata as extracted from Biodiversity Exploratories JSON 'datastructure' files (Variables.Id, Variables.Label, Variables.Description, Variables.unit.Name, Variables.unit.Description, and Variables.dataType.Name). - The processing performed by BEpipeR (in separate columns: rSUB, rDW, rSTD, rRES, rFB, DIA, GIA, RF, and DI; in concatenated fashion: Proc_info). See paramMAIN's dictionary for more information.

these metadata in JSON ‘datastructure’ files for normal data sets and in a csv file for data generated through BExIS’ climate tool. BEpipeR utilizes this information to generate metadata for variables featured in the complete composite data set. To do so, BEpipeR strips away data set IDs and aggregation suffixes from headers. Subsequently, for variable names isolated this way, their metadata (such as variable description and unit information) are extracted, enriched with information on the processing performed through BEpipeR, and exported as tabular data to the ‘Output’ directory (Table 3). Note that, as for data sets, metadata files are retrieved by their base ID, and hence, multiple metadata files with the same base ID are not supported in the ‘Metadata’ directory.

Minimal system requirements

To facilitate the adoption of the pipeline, we designed BEpipeR to be executable even on entry-level consumer hardware. CPU-wise, BEpipeR should execute fine on machines with ≥ 2 physical cores. RAM-wise, its minimal requirements are primarily dictated by the size of the input data sets users opt to process, as well as the type of processing requested. For instance, input files < 100 KB might consume negligible amounts of working memory, while large amplicon sequencing data sets (> 200 MB) might require significantly more, in particular if they are rarefacted with an excessively high number of repetitions. Still, to prevent working memory from becoming a limiting factor on typical consumer-level hardware, repeated rarefaction is performed in chunks of ten, temporary files might be written to disk (‘Temp’ directory, Table 2), and large elements are cleared from the pipeline’s workspace immediately after they have become obsolete. With respect to processing times, BEpipeR might spend most of its time on the repeated rarefaction of large amplicon data sets, as well as performing variables selection on large composite data sets (> 150 rows, > 1000 columns). However, since these steps harness parallel processing, they can be sped up considerably by switching to more capable hardware.

Use cases

To demonstrate BEpipeR’s rich functionalities with minimal effort to the user, we ship the pipeline with exemplary data, including ten input data sets, corresponding metadata, and filled-out parameter files (Table 2). Both input and metadata files mimic real Biodiversity Exploratories information, which cannot be included for various reasons. In addition, the pipeline includes all files produced by processing the provided input data with default settings (Table 1) to allow users to familiarize themselves with the output produced (Table 3). In the following, we provide a concise summary of the BEpipeR workflow using the provided input data; for brevity, data sets are referred to by their base ID, and only steps required for understanding the provided example are listed. i) Plot designations and location information in data set 20826 are used to construct the spatially explicit plot IDs template. ii) Species abundance data in 19848 contain a non-valid numerical NA value (-88888888), which is replaced with NA in DW. iii) With the entry ‘None’, the abundance data set 19849 contains a non-valid factor level in its plot ID column. This information is excluded through SUB. iv) Species in data set 18269 could not be completely resolved (Genus3_spec), complicating meaningful comparisons between the taxonomic entities in this data set. This is solved by collapsing abundance information at the genus level through FB, followed by reshaping this data to wide format in RES. v) Abundance data in 18526 are not standardized/normalized for sampling effort. Instead, this information is provided in the ‘nobs’ column of this data set, which is subsequently used to normalize abundance scores and restore the inter-plot comparability of the data. vi) After all data sets have passed QC, the two pH measurements per plot provided in 14447 are summarized plot-wise through DIA. vii) Multi-year abundance data split over the data sets 19848, 19849, and 19850 are summarized at plot and species level through GIA. viii) Multi-year climate data (19007) obtained through BExIS’ climate tool is processed by CLIM. Note that, because no ‘qualitycounter’ column is provided, the removal of weakly supported data points is skipped. ix) Amplicon sequencing data in 25067 are first reshaped to wide format, followed by their repeated rarefaction with 150 repetitions, and the calculation of alpha diversity indices. Subsequently, all data are left-joined to the plot IDs template (MRG). Noteworthy, because data set 14567 did not require any processing, it is incorporated as is. The resulting composite data set is quality-controlled and filtered (FQC), and variables selection (VS) is performed. Processing is concluded by the compilation and export of metadata (COMD) to the variables in the complete composite data set constructed in FQC.

Discussion

With BEpipeR, we provide a feature-rich pipeline for processing and synthesizing Biodiversity Exploratories data. To our knowledge, this is the first framework of this consortium to do so in a user-friendly and highly reproducible fashion. We acknowledge that embedding it in the Biodiversity Exploratories with its many projects comes with both challenges and benefits. We recognise that providing a comprehensive framework for the processing of the consortium’s many data sets is a daunting task, as many, potentially conflicting, interests need to be satisfied. Hence, for the near-time development of this pipeline, we see the following three focal points for improvement: i) Streamlining the user experience by the improved handling of errors, increasing the pipeline’s verbosity, and providing the ability to parse even more aggregation information through the existing parameter files. ii) The extension of existing features, such as data

normalization through transformation (e.g., [McKnight et al. 2019](#), [Boshuizen and Te Beest 2023](#)). iii) Increasing rigour in ensuring data integrity by implementing access to BExIS' API and, thereby, information obtainable solely through this channel.

Noteworthy, many re-usability issues BEpipeR corrects for would have been prevented in the first place by adopting more stringent standard operational procedures that ensure data re-usage with minimal user effort. Briefly, we restrict ourselves to the issues encountered most often while working on a subset (~ 150 data sets from 2009 onwards) of the Biodiversity Exploratories' information stock: i) Non-harmonized plot information: Data re-usage might be drastically improved by encoding plot information in a unified way. This includes, among others, making the plot ID column the first column of data sets, using unified column names for this type of information, and enforcing the experimental plot scheme with leading zeros throughout. ii) Non-harmonized NA and NODATA values: Consortium-wide non-numerical NA and NODATA values would prevent the (mis)use of numerical values for encoding this information. This issue is exacerbated by the fact that information on these values can only reliably be obtained through BExIS' API, a resource most scientists might not be aware of or familiar with. iii) Non-harmonized encoding of factors: Factors should be encoded as character strings to facilitate their detection and prevent aggregation over these values. These and other potential improvements should be accompanied by more stringent quality control and data curation through the Biodiversity Exploratories' data management team to prevent mal-formatted, incomplete, or erroneous data sets from being listed as ready-to-use in BExIS database. We also highlight the need to revise any data sets that may not adhere to these standards. While most of these suggestions mean minimal effort for data owners who upload new data sets, and a reasonable yet essential one for this consortium's data management team, they might drastically reduce hands-on time for scientists that re-use these data, and ultimately pave the way to making these data exploitable through large inter-framework databases ([Finkel et al. 2020](#)).

In constructing BEpipeR, we aimed to balance its specificity to the Biodiversity Exploratories with general applicability. This means that while this pipeline was written to solve numerous Biodiversity Exploratories-specific data issues, it might nevertheless be adapted to the needs of other large research consortia. This might be achieved most easily by, among others, implementing a step that recodes other consortia's plot designations to the Biodiversity Exploratories' experimental plot designation scheme, preventing them from having to adjust most regular expression-based pattern matching in BEpipeR. These consortia may also benefit from the modularity of BEpipeR, where each major loop is a well-defined processing step, allowing for straightforward modifications to the workflow. Additionally, parsing-wise, users may tailor paramMAIN to their needs by deleting or replacing all columns not strictly required for BEpipeR's operation (as indicated in the file's dictionary). Finally, changes to BEpipeR's source code are aided by a consistent and traceable naming scheme for variables, as well as detailed comments on the code and the underlying reasoning.

To conclude, even though this project might be facing substantial challenges, it is the Exploratories' large base of researchers and scientific staff that has the potential to render this endeavour a success. People interested can contribute both conceptually, by providing suggestions for future implementations, and preferably, by coding. In the best case, their participation is fuelled by having understood the nature of this framework, that is, its capability to boost each project's data visibility and impact by providing it in a composite data set for the most straightforward re-use possible. As we will demonstrate elsewhere, BEpipeR can be used to generate expansive composite data sets with the potential to further insights into complex evolutionary and ecological matters.

Ethics and consent

Ethical approval and consent were not required.

Data availability

Example data used in this publication are available as part of BEpipeR on [GitHub](#) and [Zenodo](#) ([Glück et al. 2024](#)).

Data are available under the terms of the [Creative Commons Attribution 4.0 International license](#) (CC-BY 4.0).

Software availability

Software and source code available from: <https://github.com/marcelglueck/BEpipeR>

Archived source code at time of publication: <https://doi.org/10.5281/zenodo.13838117> ([Glück et al. 2024](#))

License: LGPL-3.0

Acknowledgements

We thank the Copyright Office of Tübingen University for their assistance in finding a suitable license for this pipeline and Andreas Ziegler for helping with its shipping. We also acknowledge support from the Open Access Publishing Fund of Tübingen University for covering publication fees. Icons in [Figure 1](#) were obtained from [flaticon.com](#); in their order of first appearance: xnimrodx, lakonicon, Bharat Icons, gravisio, Ida Desi Mariana, phatplus, Freepik, Kharisma, POD Gladiator, Uniconlabs, Iconjam, KP Arts, Mayor Icons, karthiks_18, and IwitoStudio. This work is based on data obtained within the DFG Priority Program 1374 'Infrastructure-Biodiversity-Exploratories'. We thank the staff of the three exploratories, the BE office and the BEXIS team for their work in maintaining the plot and project infrastructure, and Markus Fischer, the late Elisabeth Kalko, Eduard Linsenmair, Dominik Hessenmöller, Jens Nieschulze, Daniel Prati, Ingo Schöning, François Buscot, Ernst-Detlef Schulze, and Wolfgang W. Weisser for their role in setting-up the Biodiversity Exploratories project.

References

- Allan E, Bossdorf O, Dormann CF, *et al.*: **Interannual variation in land-use intensity enhances grassland multidiversity.** *Proc. Natl. Acad. Sci.* 2014; **111**(1): 308–313.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Allan E, Manning P, Alt F, *et al.*: **Land use intensification alters ecosystem multifunctionality via loss of biodiversity and changes to functional composition.** *Ecol. Lett.* 2015; **18**(8): 834–843.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Anderson-Teixeira KJ, Davies SJ, Bennett AC, *et al.*: **CTFS-Forest GEO: a worldwide network monitoring forests in an era of global change.** *Glob. Chang. Biol.* 2015; **21**(2): 528–549.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Baker M: **Reproducibility crisis.** *Nature.* 2016; **533**(26): 353–366.
- Baker M: **Scientific computing: Code alert.** *Nature.* 2017; **541**(7638): 563–565.
[Publisher Full Text](#)
- Barrett T, Dowle M, Srinivasan A: **data.table: Extension of 'data.frame'.** 2023.
[Reference Source](#)
- Benjamini Y, Hochberg Y: **Controlling the false discovery rate: a practical and powerful approach to multiple testing.** *J. R. Stat. Soc. Series B.* 1995; **57**(1): 289–300.
[Publisher Full Text](#)
- Blüthgen N, Simons NK, Jung K, *et al.*: **Land use imperils plant and animal community stability through changes in asynchrony rather than diversity.** *Nat. Commun.* 2016; **7**(1): 10697.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Boshuizen HC, Te Beest DE: **Pitfalls in the statistical analysis of microbiome amplicon sequencing data.** *Mol. Ecol. Resour.* 2023; **23**(3): 539–548.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Cameron ES, Schmidt PJ, Tremblay BJ-M, *et al.*: **Enhancing diversity analysis by repeatedly rarefying next generation sequencing data describing microbial communities.** *Sci. Rep.* 2021; **11**(1): 22302.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Caporaso JG, Kuczynski J, Stombaugh J, *et al.*: **QIIME allows analysis of high-throughput community sequencing data.** *Nat. Methods.* 2010; **7**(5): 335–336.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Chamanara J, Gaikwad J, Gerlach R, *et al.*: **BEXIS2: A FAIR-aligned data management system for biodiversity, ecology and environmental data.** *Biodivers. Data J.* 2021; **9**.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Chavarria KA, Saltonstall K, Vinda J, *et al.*: **Land use influences stream bacterial communities in lowland tropical watersheds.** *Sci. Rep.* 2021; **11**(1): 21752.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Chiarello M, McCauley M, Villéger S, *et al.*: **Ranking the biases: The choice of OTUs vs. ASVs in 16S rRNA amplicon data analysis has stronger effects on diversity measures than rarefaction and OTU identity threshold.** *PLoS One.* 2022; **17**(2): e0264443.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Custer GF, van Diepen LT, Seeley J: **Student perceptions towards introductory lessons in R.** *Nat. Sci. Educ.* 2021; **50**(2): e20073.
[Publisher Full Text](#)
- Davies SJ, Abiem I, Salim KA, *et al.*: **ForestGEO: Understanding forest diversity and dynamics through a global observatory network.** *Biol. Conserv.* 2021; **253**: 108907.
[Publisher Full Text](#)
- Dixon P: **VEGAN, a package of R functions for community ecology.** *J. Veg. Sci.* 2003; **14**(6): 927–930.
[Publisher Full Text](#)
- Felipe-Lucia MR, Soliveres S, Penone C, *et al.*: **Land-use intensity alters networks between biodiversity, ecosystem functions, and services.** *Proc. Natl. Acad. Sci.* 2020; **117**(45): 28140–28149.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Finkel M, Baur A, Weber TK, *et al.*: **Managing collaborative research data for integrated, interdisciplinary environmental research.** *Earth Sci. Inf.* 2020; **13**: 641–654.
[Publisher Full Text](#)
- Fischer M, Bossdorf O, Gockel S, *et al.*: **Implementing large-scale and long-term functional biodiversity research: The Biodiversity Exploratories.** *Basic Appl. Ecol.* 2010a; **11**(6): 473–485.
[Publisher Full Text](#)
- Fischer M, Kalko EK, Linsenmair KE, *et al.*: **Exploratories for large-scale and long-term functional biodiversity research. Long-Term Ecological Research: Between Theory and Application.** 2010b; 429–443.
[Publisher Full Text](#)
- Glück M, Bossdorf O, Thomassen HA: **BEpipeR: a user-friendly, flexible, and scalable data synthesis pipeline for the Biodiversity Exploratories and other research consortia.** *Zenodo.* 2024.
[Publisher Full Text](#)
- Harrell F: **Hmisc: Harrell Miscellaneous.** 2023.
[Reference Source](#)
- Hijmans RJ, Bivand R, Forner K, *et al.*: **Package 'terra'.** 2022.
[Reference Source](#)
- Hobbie JE, Carpenter SR, Grimm NB, *et al.*: **The US long term ecological research program.** *Bioscience.* 2003; **53**(1): 21–32.
[Publisher Full Text](#)
- Kloss L, Fischer M, Durka W: **Land-use effects on genetic structure of a common grassland herb: a matter of scale.** *Basic Appl. Ecol.* 2011; **12**(5): 440–448.
[Publisher Full Text](#)
- Le Provost G, Schenk NV, Penone C, *et al.*: **The supply of multiple ecosystem services requires biodiversity across spatial scales.** *Nat. Ecol. Evol.* 2023; **7**(2): 236–249.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Lin H, Peddada SD: **Analysis of microbial compositions: a review of normalization and differential abundance analysis.** *NPJ Biofilms Microbiomes.* 2020; **6**(1): 60.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Margalef R: **Information theory in ecology.** 1973.
- McKnight DT, Huerlimann R, Bower DS, *et al.*: **Methods for normalizing microbiome data: an ecological perspective.** *Methods Ecol. Evol.* 2019; **10**(3): 389–400.
[Publisher Full Text](#)
- McMurdie PJ, Holmes S: **Waste not, want not: why rarefying microbiome data is inadmissible.** *PLoS Comput. Biol.* 2014; **10**(4): e1003531.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Menhinick EF: **A comparison of some species-individuals diversity indices applied to samples of field insects.** *Ecology*. 1964; **45**(4): 859–861.

[Publisher Full Text](#)

Microsoft Corporation: **Weston S: doSNOW: Foreach Parallel Adaptor for the 'snow' Package.** 2022.

[Reference Source](#)

Müller K: **here: A Simpler Way to Find Your Files.** 2020.

[Reference Source](#)

Naimi B, Hamm NA, Groen TA, *et al.*: **Where is positional uncertainty a problem for species distribution modelling?** *Ecography*. 2014; **37**(2): 191–203.

Ooms J: **The jsonlite package: A practical and consistent mapping between json data and r objects.** 2014. arXiv preprint arXiv:1403.2805.

R Core Team: **R: A Language and Environment for Statistical Computing.** 2021.

[Reference Source](#)

Racine JS: **RStudio: a platform-independent IDE for R and Sweave,** *JSTOR*. 2012.

Rovero F, Ahumada J: **The Tropical Ecology, Assessment and Monitoring (TEAM) Network: An early warning system for tropical rain forests.** *Sci. Total Environ.* 2017; **574**: 914–923.

[PubMed Abstract](#) | [Publisher Full Text](#)

Saary P, Forslund K, Bork P, *et al.*: **RTK: efficient rarefaction analysis of large datasets.** *Bioinformatics*. 2017; **33**(16): 2594–2595.

[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Schloss PD: **Rarefaction is currently the best approach to control for uneven sequencing effort in amplicon sequence analyses.** *mSphere*. 2024; e00354-00323.

Schloss PD, Westcott SL, Ryabin T, *et al.*: **Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities.** *Appl. Environ. Microbiol.* 2009; **75**(23): 7537–7541.

[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Seibold S, Gossner MM, Simons NK, *et al.*: **Arthropod decline in grasslands and forests is associated with landscape-level drivers.** *Nature*. 2019; **574**(7780): 671–674.

[PubMed Abstract](#) | [Publisher Full Text](#)

Shannon CE, Weaver W: *The mathematical theory of communication.* University of Illinois Press; 1949.

Simpson EH: **Measurement of diversity.** *Nature*. 1949; **163**(4148): 688–688.

[Publisher Full Text](#)

Ushey K, Wickham H: **renv: Project Environments.** 2023.

[Reference Source](#)

Vályi K, Rillig MC, Hempel S: **Land-use intensity and host plant identity interactively shape communities of arbuscular mycorrhizal fungi in roots of grassland plants.** *New Phytol.* 2015; **205**(4): 1577–1586.

[PubMed Abstract](#) | [Publisher Full Text](#)

van Breugel M, Craven D, Lai HR, *et al.*: **Soil nutrients and dispersal limitation shape compositional variation in secondary tropical forests across multiple scales.** *J. Ecol.* 2019; **107**(2): 566–581.

[Publisher Full Text](#)

Walters KE, Martiny JB: **Alpha-, beta-, and gamma-diversity of bacteria varies across habitats.** *PLoS One*. 2020; **15**(9): e0233872.

[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Weiner CN, Werner M, Linsenmair KE, *et al.*: **Land-use impacts on plant-pollinator networks: interaction strength and specialization predict pollinator declines.** *Ecology*. 2014; **95**(2): 466–474.

[PubMed Abstract](#) | [Publisher Full Text](#)

Weiss S, Xu ZZ, Peddada S, *et al.*: **Normalization and microbial differential abundance strategies depend upon data characteristics.** *Microbiome*. 2017; **5**: 1–18.

Wickham H: **The split-apply-combine strategy for data analysis.** *J. Stat. Softw.* 2011; **40**: 1–29.

Wickham H, Averick M, Bryan J, *et al.*: **Welcome to the Tidyverse.** *J. Open Source Softw.* 2019; **4**(43): 1686.

[Publisher Full Text](#)

Wöllauer S, Zeuss D, Hänsel F, *et al.*: **TubeDB: An on-demand processing database system for climate station data.** *Comput. Geosci.* 2021; **146**: 104641.

[Publisher Full Text](#)

Xia Y: **Statistical normalization methods in microbiome data with application to microbiome cancer research.** *Gut Microbes*. 2023; **15**(2): 2244139.

[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Open Peer Review

Current Peer Review Status: ? ✖

Version 1

Reviewer Report 23 January 2025

<https://doi.org/10.5256/f1000research.172574.r351632>

© 2025 Grenié M. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Matthias Grenié 

¹ Université Grenoble Alpes, Saint-Martin-d'Hères, Auvergne-Rhône-Alpes, France

² Laboratoire d'Ecologie Alpine (Ringgold ID: 56837), Grenoble, Auvergne-Rhône-Alpes, France

I was asked to review the BEPipeR manuscript, as a great tool for the Biodiversity Exploratories. While clearly a huge amount of work was put into designing and creating the pipeline, I do have strong reserves regarding its ease of use and clarity as a tool.

One first question I had while reading the manuscript was to know what BEPipeR were. From reading the manuscript, this wasn't clear from the beginning. I would have expected BEPipeR to be an R package or a shiny application, as it aimed to simplify data access and analysis for the user. I understood, after finishing the introduction that it was indeed an R pipeline, but I was wondering what the added benefit was compared to an R package or a shiny app? It seems to me that a pipeline is more brittle to change than a well defined R package. It's also harder to distribute as it doesn't use the main mechanism to spread analyses through R which are R packages. It also doesn't leverage the "Research Compendium" suggested by Marwick et al. (2018) and others at the same time (see <https://github.com/benmarwick/rrtools>). That infrastructure tries to provide the best of both world between R packages and pipelines. Research compendia can be spread through <https://docs.r-universe.dev/> for easy install and leveraging the default installation mechanism of R. I think the author should also be clear why choosing not to create an R package was better to help user and instead creating a full pipeline with many specificities.

One main question I think the authors should take some time to answer is to identify their target audience. Who are they addressing to? Are the users going to be people comfortable editing an R script? Would they rather edit a clear configuration file? How about people who want to tweak their own pipeline? I think this work should be preliminary to building such a complex tool, as BEPipeR is targeting several user types without clearly facilitating their work: you both have to edit the R script for specific values and get your own configuration files. One good way of identifying the target audience, is to run tests runs of the software with potential users to gather feedback about what was clear and what wasn't. This is invaluable when building a tool, to make sure what you're creating is (1) going to serve the users' needs; (2) be used; (3) easy to use.

Whether it was within the manuscript or the GitHub repository, I found difficult to get a quick overview of what was achievable with BEPipeR. Before following the entire script it isn't clear what the main functions of BEPipeR are. What are the core functions of BEPipeR? You could show quick snippets of plots generated through the BEPipeR pipeline or data analyzed with it on the GitHub page or within the manuscript. The fact that I had to manually go through the script to understand what was achieved concretely by BEPipeR prevents me from using it as I didn't know how it could help before hand. In addition to "Setting up" instructions, you could provide a "Where to start" section to clear define the goals and possibilities offered by BEPipeR. Also, these instructions should be available within the GitHub repository, as not all users are going to refer to the manuscript to understand the basic features of BEPipeR. One thing that would be nice would be to see where BEPipeR stands in terms of the analysis pipeline of the Biodiversity Exploratories datasets.

Regarding installation of BEPipeR, I had a hard time setting up with the exact version of R and thus installing the Pipeline, I had to install `[rig]()` to be able to switch R versions as R 4.1.1 is already considered quite old. I had to use admin privileges to install ``rig``, then use a command line instruction to install the specific R version, then load RStudio with the good version of R, then restore the ``renv`` file. I'm unsure if the average user of the pipeline would be confident to perform such a long and complicated process. Also, it doesn't seem to be a good idea to have several R versions laying around your computer. This can definitely confuse your users. While I understand the need to fix package version to ensure maximum reproducibility, I urge the authors to point the users to a "simple" way of doing so. After about an hour of struggle, I managed to install all versions of needed software (previous R version, previous version of RTools, and the ``renv`` environment). I really wonder to what extent this process is easily extendable to other users, and this, to me, emphasizes the fact that building an R package with clearly defined package versions would make the full process easier. Also, if BEPipeR requires all that, while needing an R connection, I don't understand why it requires to manually download the GADM German gpkg file instead of providing a function to download it at the good place. Also, it puts the burden on the users' shoulders where they have not only to download but also rename it properly and move it in the appropriate folder. A user can stumble over any of each of these steps.

Regarding the pipeline, as advertised in the manuscript as "user-friendly processing of data sets", I was expecting a clear separation between the features provided by BEPipeR and the script that needs to be tweaked by the users. What I found was quite hard to understand: a 3000-lines R script, with both user-input values mixed up with actual functional code to process the dataset. I found this structure impossibly complex for the feature. I would have expected some functionality to be "packaged" somehow, if not in an R package, at least in some R/ folders with local functions. Understanding what parts of this script were to be modified and which ones shouldn't was quite difficult. Even though everything was explained in much details in the manuscript, I had trouble connecting the manuscript to the actual script. I would have expected the manuscript to more explicitly show parts of the script to explain how it works. Instead the manuscript goes in great length explaining what the script does, without explaining how to use it! For example, the `param*.csv` files are loaded in the script at three different locations, why is that? Do you expect the user to go through these three locations to understand what's happening? The three files could be loaded at the top of the script to easily separate between user input and data loaded automatically. Also, even though there are dictionaries provided for the design of each of these files, the structure for variable selection is quite hard to understand what these files are and how

to create them. I think extensive documentation of these files, with verbal description and schemes showing how they connect to the pipeline would help. The three param files are mentioned in Figure 1, but it's really unclear how they connect to the pipeline exactly. What do they provide and how are they reused throughout the analysis?

Regarding script structure, one thing possible would be to divide the script into smaller scripts to abstract away some details and better structure the code. With each script calling common functions, or at least calling one another. The main script doesn't follow RStudio's syntax to create clear parts that the user can jump through using the "Show Document Outline" feature, the convention is that any comment in R followed by text and four times the same character is considered "a part" of the script. This makes navigating the script harder, especially if you're expecting the user to use RStudio.

Another thing that puzzled me, whether in the manuscript or the software documentation is that nowhere was explained how I should run the script. Should I run `source()` directly on the main script? With which working directory? Which parameters should I make sure to have changed? Should I run the script line-by-line or section by section? This should be clearly stated as a naive user would want to know this information rapidly when downloading the pipeline.

While the paper is a great companion to explain how to use BEPipeR, I think the pipeline should be self-sufficient and contain enough documentation to be able to operate the pipeline without referring to external documentation. The documentation should be extensive about the steps, both about what analyses are performed but also, and it's quite important for a pipeline, how to perform them, with which possible options.

Have you tested it against users? What was their feedback? I would be curious to know as I expect the use of such a script to be quite complex for naive users. I would definitely recommend the authors to simplify the features they want to work on, to compartmentalize their pipeline through well-defined functions made available to the users, and simplify the final script exposed to the user, with easy to run routine. Like "run this main script that is going to call internal functions". The authors could also look into building a [targets](<https://docs.ropensci.org/targets/>) pipeline for ease of reproducibility.

I had two additional small remarks: the first one is I congratulate the authors for having thought about archiving their software on Zenodo in addition to GitHub as it allows near-permanent archival. The second is that I enjoyed seeing the hexagonal logo of BEPipeR but it doesn't seem specific enough to be related to Biodiversity Exploratories. Maybe you could think about a logo that reuses parts of the graphical identity of Biodiversity Exploratories.

References

1.) Marwick et.al., 2018 (Ref 1)

References

1. Marwick B, Boettiger C, Mullen L: Packaging data analytical work reproducibly using R (and friends). 2018. [Publisher Full Text](#)

Is the rationale for developing the new software tool clearly explained?

Partly

Is the description of the software tool technically sound?

Partly

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: biodiversity metrics, biodiversity tools

I confirm that I have read this submission and believe that I have an appropriate level of expertise to state that I do not consider it to be of an acceptable scientific standard, for reasons outlined above.

Reviewer Report 10 December 2024

<https://doi.org/10.5256/f1000research.172574.r341491>

© 2024 Gould E. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Elliot Gould 

The University of Melbourne, Melbourne, Victoria, Australia

Summary:

Research Consortia involved in large-scale long-term environmental research frameworks, such as Biodiversity Exploratories, continuously accrue vast amounts of data. However, a great deal of effort, technical expertise, and data processing infrastructure is required to leverage these data to answer even trivial research questions, let alone more intricate questions.

Glück et al. have developed the BEpipeR tool to facilitate the streamlined synthesis of plot-based Biodiversity Exploratories data. BEpipeR is written in R, one of the most (if not *the most*) popular programming languages in ecology. The approach taken by BEpipeR is a template-based

approach whereby the specifications for the data-processing and synthesis pipeline are provided as parameters within .csv files, which are then read in as R objects provided to the main pipeline script.

While there might be more sophisticated ways of approaching the task (e.g. the Portal Project) than using parameter inputs or dealing with dynamic analysis contexts via static templates (i.e. data could be in long or wide formats and the user has to specify whether the data is long or wide instead of this being handled automatically); the strength of a simpler approach is that potential errors should be more easily detectable and resolvable by more novice R users, the pipeline can be easily modified, or expanded to other projects.

Although the full computing environment for reproducing and operating the pipeline is not provided, the software environment is provided utilising the *renv* R package, which is supported by RStudio / Posit, and is a well-known, documented and supported tool. This is a good choice given the target user base of the software are mostly undertaking data synthesis manually with spreadsheets. Although providing the full computing environment through a tool like docker/rocker might enable greater reproducibility and longevity / stability, it would not be approachable to the target user.

The software can be downloaded from GitHub, and the user is expected to execute the software using the RStudio IDE, which is also an excellent choice given its popularity, and approachability for new and experienced R users alike.

Software documentation mainly comprises the manuscript under review, as well as the associated README and some markdown files embedded in subdirectories, which gives a neat overview of the purpose and contents of the repository. It's always nice to see a pretty HEX sticker too.

I have not had sufficient time to properly play with and test the software / code in detail as I would prefer (Ivimey-Cook et al. 2023). But I have briefly explored the main code script and associated files, such as the parameter files, so the majority of my review comprises critiques about the manuscript, and some minimal aspects of implementation.

The manuscript was well-calibrated in the level of detail and explanation for the software's target user, and included a good level of detail, in terms of set up / implementation, as well as some nice design choices that facilitate use by less familiar R users. e.g. the choice to include an excel spreadsheet version of the parameter specification files and associated data dictionary sheet, and the distribution of the software in a 'just-ran' state, so that users can get a handle on how the outputs relate to inputs and specification settings. I also particularly liked Figure 1, which gives a great visual overview of the data pipeline.

Major Issues & Improvements

A major issue with the manuscript were the small but frequent grammatical errors and stylistic expression that hampered clarity of meaning. Secondly, the rationale for the software could be more tightly synthesized with relevant literature, which, if addressed, would help in highlighting the utility and merits of the software. Similarly, the discussion could reference related approaches / literature in to contextualise the intended application and contrast alternative approaches to this problem. Minor issues include the need for further clarifying some aspects of the pipeline in

appropriate logical sequence, and some further explanation or provision of further resources / user guides for the more complex elements of the software. These are addressed in detail below.

Rationale

- In reference to your sentence "By doing so, they might leave out potential data that would have been instrumental in answering their complex scientific questions, ultimately causing a loss in statistical power," a loss of statistical power is certainly a big problem caused by not utilising all available data. I think this problem is actually more serious/nuanced than reduced statistical power. For example, another problem is that including or excluding particular subsets of the data may have a large influence on the results and therefore findings (check out some many-analyst style studies for examples of how analytic decisions may have large bearings on the research findings, e.g. Gould et al. in press -- disclaimer, I am first author on this paper, there is no pressure to cite, but we did find that data subsetting could swing numerical results wildly at the boundaries of expected values).
- Secondly, this ultimately is also a form of 'research waste.' You could also consider citing Buxton et al. (2021) Ref 2, or Purgar et al. (2022) Ref 3 here.
- Considering these additional problems related to a lack of appropriate infrastructure for leveraging available data would further underscore the need for the *BEpipeR* package.
- Taking a positive perspective, improving data usage within- and among- consortiums, through utilising tools similar to *BEpipeR* would contribute to the task of data / evidence synthesis, and potentially tackling broader empirical research questions, around generality, for example.
- Tools, such as *BEpipeR*, support good data stewardship and data management with a long-term vision which ultimately support scientific discovery and progress (Wilkinson et al. 2016) Ref 4

Other Related Literature / Applications

- Portal Project and associated package (addresses the same problem as BEpipeR, but for the Portal Project): <https://portal.weecology.org> <https://github.com/weecology/portalr> (Christensen et al. 2019) Ref 1
- "Regularly updated data" (Yenni et al. 2019) Ref 5-- a potential use-case for BEpipeR as new monitoring data is collected over time.
- "Near Term forecasting" (White et al. 2019) Ref 6-- another potential use-case for BEpipeR that could be incorporated into automated pipelines when the software matures into the future.

Use-case demonstration

- I found the textual summary of the use-case was rather onerous to read. This could be better presented as a diagram / schematic, perhaps with snapshots of the data, and a summary of the user-settings at various phases of the pipeline.

Minor Issues

- *abstract:*
 - *Background:* I think a clause qualifying what Biodiversity Exploratories is ("Large-scale long-term environmental research frameworks") in the abstract would be handy, perhaps at the end of the final sentence in the 'Background' component of the abstract.
 - *Implementation:*

- It was nice to see an explanation of the setup that includes an explanation of `renv`, I suggest providing a link to the user-guide (<https://docs.posit.co/ide/user/ide/guide/environments/r/renv.html>), given the target audience are not experienced R users. I've personally experienced a bit of difficulty with `renv` at times, even though I consider myself to be at the pointy end of things.

- "For up-to-date set-up instructions, users are referred to the pipeline's GitHub presence" - I suggest qualifying the exact location. When I went to find where this information might be, I could only find the same content stored as a [markdown file](#).

- "As the Excel versions of these files support users in providing processing information [...]" - the fact that there are .xlsx versions of the same .csv files hasn't yet been introduced in this paragraph. I suggest stating so clearly and rewording. I had to go to the repository to be sure of the intended meaning here.

- *Table 2:*

- Move "Source" row above "Processing" since the content of "Processing" is the product of "Source": "Files copied here from 'Source' for processing through BEpipeR. This directory is expunged at the start of each pipeline run."

- "The R programming language script of the BEpipeR pipeline": replace "of the" with, "that executes", or similar.

- *Data preprocessing:*

- "R session as 'datasets_table'" should be "global environment for the active R session as the object 'datasets_table'".

- In my experience reshaping data between long and wide formats can sometimes be non-trivial. I would suggest giving users a warning here to check the outputs are as expected after reshaping. An example of the data reshaping inputs and outputs might also help exemplify this process a little further. Perhaps as a vignette in the repository or in a text-box. I found the reshaping a little hard to mentally visualize from the text description.

- The 'standardization by variable' paragraph was also a little tricky to follow. It's stated that "BEpipeR allows all numeric variables of a data set to be normalized by information provided in a user-specified column of that data set", what is the specific normalization procedure used? This should be mentioned so that the user can understand the calculation fully. I assume from the previous sentence in the paragraph mentioning sampling effort, and from reading below that the standardization approach is rarefaction?

- *Quality Checks*

- It seems that outlier checks might be performed once (maybe twice?) by the user, and that after running the pipeline once to flag potential outliers, the user can then remove outliers, and then re-run the pipeline to completion. After doing so, should the user then turn off the quality checking? Figure 1 gives the impression of a somewhat linear pipeline, but perhaps the process is more iterative depending on actions taken after QC. A little more explanation of the overall workflow here would be handy.

- *Diversity calculations*

- Which exact procedure is used to "calculate alpha diversity indices"? This isn't mentioned until the very end of the paragraph, which follows very detailed treatment of the contentious nature of rarefaction.

- *Post-processing*

- "BEpipeR left-joins all available data to the plot IDs template constructed upstream." The target user is not going to know what a left-join is, most likely. Perhaps you could provide a link to R4ds, which explains this procedure further: <https://r4ds.hadley.nz/joins> . <https://r4ds.had.co.nz/relational-data.html#outer-join> has a really nice diagram depicting and

explaining this.

- *Conclusion*

- "People interested can contribute both conceptually, by providing suggestions for future implementations, and preferably, by coding." This needs a little further explanation, i.e. people can contribute by making pull-requests to the package's code-base on GitHub.

Referencing

I don't think Baker (2016) is the right reference for the statement "Arguably, BEpipeR has the potential to generate large composite data sets in a highly reproducible fashion." Either you meant to cite Baker (2017), which is also in your reference list, or you should cite something more focussed on the intersection of data pipelines / infrastructure and reproducibility.

Software Citation Style

Instead of having a very long sentence chewing up a whole paragraph containing all of the software used in your package, consider using a table of package citations, e.g. following the citation report approach taken by the **grateful** package, [summarising packages within a table](#).

Expression

- At times the writing was hyperbolic and not so succinct, e.g. "fuelled by" (perhaps replace with, "underpinned by") and "unmatched standing expertise", "To conclude, even though this project might be facing substantial challenges, it is the Exploratories' large base of researchers and scientific staff that has the potential to render this endeavour a success." I think an additional edit for succinctness and clarity of expression is needed.
- Overuse of demonstrative particles, such as 'this', 'that', 'these' combined with nominalization at times hampered clarity and slowed my understanding, e.g. of nominalisation: "that allow for a straightforward and less time-consuming incorporation into their workflows". Suggest switching to more active language. This will also help with succinctness.
- "executing the pipeline productively" it's not clear what 'productively' means here. Do you mean 'in production'?
- "Additionally, in the following, we provide an", replace with "Below we provide an"
- Shift "to guide users in familiarizing themselves with the pipeline" after "an in-depth description of the workflow".

Grammatical, spelling, typographical errors

- "data subsettings": 'subsetting' is plural.
- "data substitution through exact and pattern-based approaches", do you mean 'subsetting'?
- "resolving species aggregates issues", replace "species aggregates" with 'species aggregation' or similar.
- First sentence of last paragraph in introduction is extremely long, and the clause following the semi-colon seems incomplete grammatically.

References

1. Christensen E, Yenni G, Ye H, Simonis J, et al.: portalr: an R package for summarizing and using the Portal Project Data. *Journal of Open Source Software*. 2019; **4** (33). [Publisher Full Text](#)

2. Buxton R, Nyboer E, Pigeon K, Raby G, et al.: Avoiding wasted research resources in conservation science. *Conservation Science and Practice*. 2021; **3** (2). [Publisher Full Text](#)
3. Purgar M, Klanjscek T, Culina A: Quantifying research waste in ecology. *Nat Ecol Evol*. 2022; **6** (9): 1390-1397 [PubMed Abstract](#) | [Publisher Full Text](#)
4. Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, et al.: The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data*. 2016; **3**: 160018 [PubMed Abstract](#) | [Publisher Full Text](#)
5. Yenni GM, Christensen EM, Bledsoe EK, Supp SR, et al.: Developing a modern data workflow for regularly updated data. *PLoS Biol*. 2019; **17** (1): e3000125 [PubMed Abstract](#) | [Publisher Full Text](#)
6. White E, Yenni G, Taylor S, Christensen E, et al.: Developing an automated iterative near-term forecasting system for an ecological study. *Methods in Ecology and Evolution*. 2019; **10** (3): 332-344 [Publisher Full Text](#)

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Applied ecology / ecological modelling, research software development for data analysis pipelines.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research