

APPLICATION

BarKeeper—a versatile web framework to assemble, analyse and manage DNA barcoding data and metadata

Sarah Wiechers¹  | Lara Marie Kösters^{2,3}  | Dietmar Quandt⁴  | Thomas Borsch⁵  |
Susann Wicke^{2,3}  | Kai Frank Müller¹ 

¹Institute for Evolution and Biodiversity, Evolution and Biodiversity of Plants, University of Münster, Münster, Germany

²Institute for Evolution and Biodiversity, Plant Evolutionary Biology, University of Münster, Münster, Germany

³Institute for Biology, Plant Evolution and Biodiversity, Humboldt-Universität zu Berlin, Berlin, Germany

⁴Nees-Institute for Biodiversity of Plants, University of Bonn, Bonn, Germany

⁵Botanischer Garten Berlin, Freie Universität Berlin, Berlin, Germany

Correspondence

Kai Frank Müller

Email: kaimueller@uni-muenster.de

Funding information

Bundesministerium für Bildung und Forschung, Grant/Award Number: 01LI1101 and 01LI1501; Deutscher Akademischer Austauschdienst; University of Muenster

Handling Editor: Samantha Price

Abstract

1. DNA barcoding aids in rapidly identifying specimens in various contexts by comparing short stretches of DNA to a reference database. Barcoding initiatives generate large reference databases of carefully curated high-quality sequences to maximize identification success.
2. *BarKeeper* provides a flexible tool for barcoding initiatives and in the context of phylogenetic studies to foster shared work on large datasets of raw sequence and associated metadata. It is free, open-source and available as a set of Docker containers for easy setup. After setting it up once, all project members can use it independently of their operating system or location.
3. *BarKeeper* offers features to collect and manage data and metadata about specimens, taxa and DNA sequences from Sanger sequencing and high-throughput sequencing (HTS) technologies. It provides excellent flexibility by not being restricted to specific markers or taxon groups. Users can view and edit records and associated metadata while the app assists them by trimming and assembling reads. It automatically checks the quality of generated barcodes with a taxonomy-based tool and offers a wide range of options for data analysis. Extensive search features allow querying the database for specific groups of records and saving the search results in the user's profile or downloading them in various file formats.
4. *BarKeeper* combines multiple tools to aid barcoding projects in every step, from a reference taxa list to finished barcode sequences, thereby minimizing the number of laborious, potentially error-prone manual steps and enabling efficient collaborative workflows.

KEYWORDS

DNA assembly, DNA barcoding, metadata, phylogenetics, reference database, taxonomy, web framework

1 | INTRODUCTION

DNA barcoding uses short stretches of DNA to identify one taxon or multiple taxa of a sample by comparing them to a reference database.

In a zoological context, researchers commonly employ a part of the mitochondrial *Cytochrome Oxidase 1* (*CO1*) gene as a marker region for this purpose (Hebert, Cywinska, et al., 2003). In many animal taxa, very high identification success rates can be achieved using *CO1*, often

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial](https://creativecommons.org/licenses/by-nc/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2023 The Authors. *Methods in Ecology and Evolution* published by John Wiley & Sons Ltd on behalf of British Ecological Society.

down to the species level (Ahmed et al., 2022; Hebert, Ratnasingham, et al., 2003; Meyer & Paulay, 2005). In plants, mitochondrial genes evolve slower and barely show substitutions between closely related species. Markers from the plastid and/or nuclear genome are, therefore, better suited for plant barcoding due to their higher levels of sequence variation (Hollingsworth et al., 2011). To achieve identification success rates comparable to *CO1* in animals, plant barcoding initiatives usually use a combination of two or more marker regions (Geiger et al., 2016; Hollingsworth et al., 2016; Korotkova et al., 2011), while a universally agreed-upon set of marker regions still does not exist for plants (Hollingsworth et al., 2016; Kolter & Gemeinholzer, 2021; Shneyer & Rodionov, 2019). In fungi, the ITS region of the nuclear ribosomal DNA serves as a standard. However, a lack of resolution in some lineages and hybridization currently lead to the inclusion of other genomic regions as barcode markers (Lücking et al., 2020).

As some markers reach aligned lengths of up to 2500 nucleotides (Korotkova et al., 2011), compiling reference data from all possible markers per taxon is more challenging for these taxa than for animals. Even Sanger-sequenced barcoding markers require automation for an efficient assembly of multiple reads instead of merely manual editing and curation; for high-throughput sequencing (HTS) approaches, an entirely algorithmic assembly is without an alternative. Sequence data diversity paired with the enormous breadth of organismal complexity of projects engaging in plant or fungal barcoding demands exceptionally high flexibility regarding data storage, processing and assembly tools.

A suitable barcoding platform should be versatile enough to be tailored to the specific needs of the barcoding project using it and ideally provide a collaborative platform to work on the same datasets from multiple locations. To make use of existing resources and archives, the tool should be able to retrieve information from other databases and deliver resulting barcodes to a central reference database. Communication between the various databases requires the support of data standards for both updating data from and publishing data to external services.

To assure the highest possible quality of the provided sequences, built-in quality control is essential for a barcoding tool. First, users should be able to look at, edit and verify the results of all automatic sequence assemblies. Second, all sequences should undergo a regular taxonomy-based check to systematically find problematic barcode sequences, such as contamination during the laboratory workflow or misidentifications of the specimens under study.

Previous shortcomings of HTS technologies, such as short read lengths or high error rates, are now compensated by the circular consensus sequencing (CCS) technique, during which base calls are reanalyzed from multiple readthroughs (Quail et al., 2012; Rhoads & Au, 2015). With up to 200,000 CCSs generated per run, costs for this technique are substantially lower than for Sanger sequencing. Furthermore, since every molecule in a DNA extract is represented by one CCS readout (Hebert et al., 2018), analysing mixed samples in a single run, as done in metabarcoding studies, becomes possible and straightforward. A versatile barcoding application, therefore, should enable users to efficiently exploit the advantages of all sequencing

technologies in that it reliably handles the resulting data from both Sanger and HTS platforms without losing quality and robustness.

To meet all these requirements, we developed the *BarKeeper* application. Its features are helpful for larger DNA barcoding initiatives, especially those focusing on non-animal groups and data compilation in the context of phylogenetic studies.

2 | MATERIALS AND METHODS

2.1 | Basic implementation

BarKeeper is based on Ruby on Rails (versions 2.7 of Ruby and 5.2 of Rails) as its primary development framework, with PostgreSQL (version 14.1) as a database management system. PostgreSQL provides numerous valuable features like support for array and hash data types as attribute values, performant full-text search and materialized views. Pages in *BarKeeper* are rendered server-side, with some components refreshed via Ajax without needing page reloads. More graphic-heavy and/or highly interactive view components are rendered entirely via JavaScript, using frameworks such as *jQuery* (version 1.12.4) and *d3.js* (version 5.4). More computationally intensive processes like assembling more than four primer reads or uploading pherogram data are run in the background using *Sidekiq* and *Redis* to allow users to continue working during their execution.

The app and its dependencies can be installed on a self-administrated server to guarantee maximum flexibility and control over the stored data. After this setup, all project members can use it via a web browser without the need to install the software themselves. *BarKeeper* can be obtained as a set of connected Docker containers, so project managers will need little skills in server administration to install the app and customize it to the project's specific needs.

2.2 | Main features

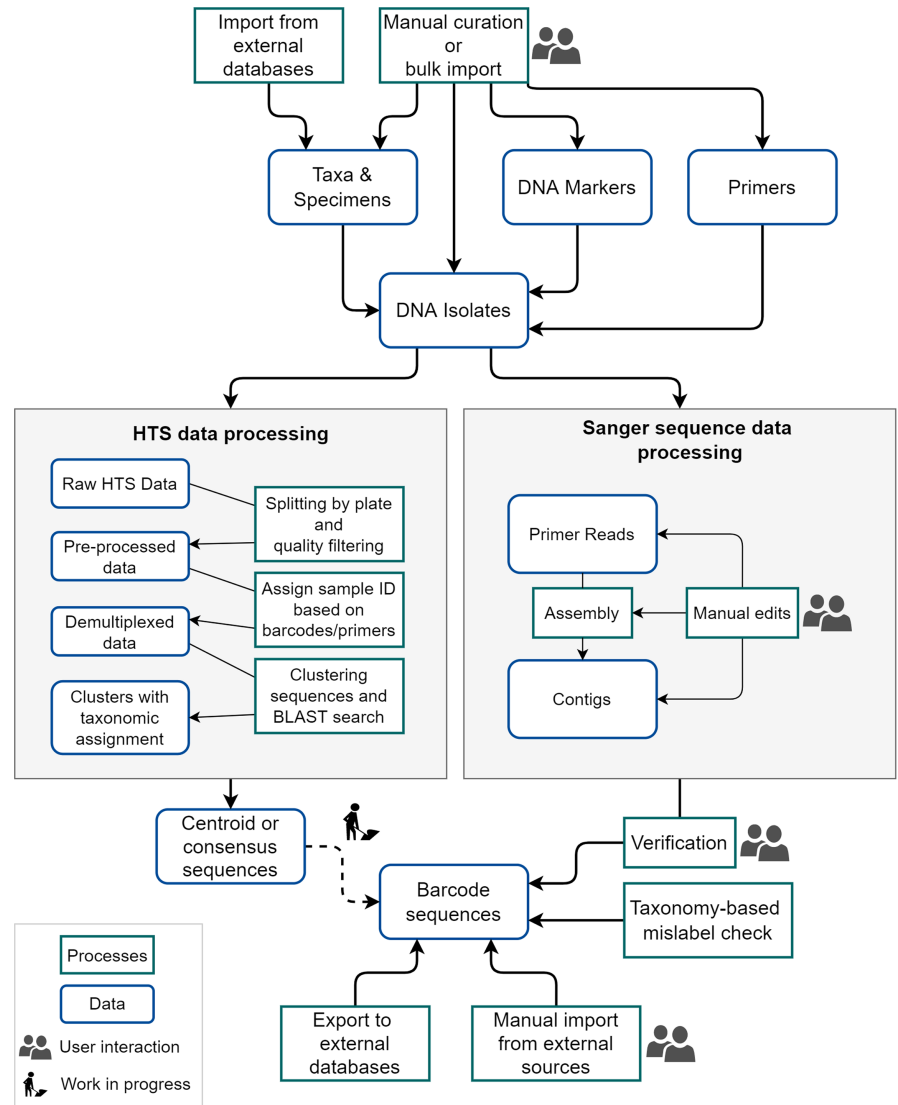
Users can import or manually enter any number of taxa and create a hierarchy among them. An interactive tree view allows navigating to or searching for specific taxa and quickly accessing related database entries.

BarKeeper allows for any number of specimens (herbarium specimens or tissue samples) to be linked to each taxon. DNA sequences generated for a marker region from an isolate (barcode sequences) are associated with the isolate of this specimen, always reflecting potential changes in the taxon concept or nomenclature (Figure 1).

BarKeeper offers components of a classic laboratory information management system (Figure 1). Users can add information about markers and primers used in the project, tissue types of sampled specimens, labs partaking in the project and equipment like freezers, racks and plates.

Similar to a typical field information management system, users can manually add specimen metadata or obtain it from external

FIGURE 1 Simplified app structure of *BarKeeper*. Colour-coding distinguishes data types (blue boxes) from processes (yellow boxes) of the app. Arrows indicate the direction of interactions. Users can upload information about markers, primers, DNA isolates, taxa and specimens manually or via spreadsheets. Taxonomic information can also be automatically retrieved from certain databases. The app will automatically attempt to assemble uploaded Sanger sequencing reads to create contigs. These can then be manually edited and verified. Barcode sequences are generated from verified contigs and are regularly checked for potential taxonomic mislabels automatically. Finished sequences can be viewed or downloaded. Raw HTS data will be processed by the app by first splitting it by plate and performing extensive quality filtering, then assigning sample IDs based on barcodes (tags) and primers and finally clustering the sequences and performing a BLAST search to confirm the assigned taxonomic label.



databases if an appropriate ID is provided. In the current version of the app, this is possible with information stored in the GGBN data portal of the Berlin Botanical Garden and Museum (BGBM; Droege et al., 2014) and the BioCASE collection data access services (Holetschek et al., 2012).

For taxa, primers and isolates, users can import more extensive lists of records from table format files in CSV or XLS format or add data manually. Template files on how to structure tables for batch import can be downloaded within the app. For each isolate, unlimited Sanger sequencing reads can be imported as SCF or AB1 files. If read data uploads comply with the app's naming conventions, *BarKeeper* will automatically extract information about the used primer and the isolate. *BarKeeper* will then assign the sequences either to an existing contig or create a new one. Reads are automatically trimmed according to a configurable set of parameters describing the quality of the base calls. If necessary or preferred by the user, it is also possible to trim reads manually and edit single base calls using a detailed view of the pherogram and read sequence (Figure 2a). This view shows the trimmed regions, single base calls and the quality (Phred score) per base.

BarKeeper will automatically assemble all trimmed reads into a contig unless the user indicates otherwise (Figure 1). More time-consuming or multiple assemblies will run in the background in a separate process while users can continue working with the app. Smaller, single-locus assemblies from Sanger data run directly. The app uses a semi-global variant of the Needleman-Wunsch algorithm for Sanger read assembly. If necessary, the default parameters used for an assembly can be adjusted for each contig.

Contigs can be viewed alongside all associated reads in a single browser window. If necessary, reads can then be edited further, resulting in an automatic re-assembly of the contig. Within the contig view, users can change the order of the displayed reads for better visualization of overlap regions, include or exclude reads from an assembly, or open a more detailed view for any of the included reads (Figure 2b).

Users can approve (verify) a contig if they are satisfied with its quality. Consequently, a marker sequence will be created or updated that reflects the consensus sequence of the contig. This sequence represents the final barcode, which can be downloaded, used in further analyses, or uploaded to an external reference database. Within



FIGURE 2 (a) Sanger read view. Users can examine a read, including a detailed view of the pherogram and base call qualities. Single base calls can be edited, and the read can be trimmed manually, if desired. Parameters for automatic trimming and assembly can be changed here. (b) Contig view. Users can view the assembled reads and get information about the read's status. Users can modify the display order of the reads in a context menu. A more detailed view of each read can be opened either in the same browser window above the contig view or in a new tab. Clicking on a specific base will open the detailed view directly at its position. Potential issues with taxonomic labeling implicated during the regular taxonomy-based analysis will be highlighted in this view for both contigs and original reads.

the app, the barcode will be included in an interactive diagram showing the project's progress. Should a user discover that a contig needs refinement, verification can be undone? The app tracks the users' verification activities.

All newly created marker sequences will be periodically checked for possible issues in a taxonomy-based mislabel check. This taxonomy cross-check relies on the SATIVA algorithm (Kozlov et al., 2016), which builds a phylogenetic tree from all approved marker sequences

and conducts a systematic leave-one-out analysis to evaluate the taxonomically-conformant placement of a sequence in the tree. Misplaced sequences will be tagged and highlighted accordingly. Incongruences can result from contamination of DNA samples or other problems in the experimental workflow. If multiple sequences per isolate show placement issues, *BarKeeper* will tag the associated isolate and specimen, prompting users to check for potential specimen misidentifications underlying the problematic taxonomic label. Users can then review the problematic sequences and associated records and either make changes to the data or mark the issues solved without making changes, for example, in case of a false-positive SATIVA result.

All table views in the app can be thoroughly searched and sorted by all columns. In addition to this quick local search, a global search across all record types is available. Contigs, marker sequences and specimens can be simultaneously filtered by several criteria, such as sequence length, associated taxa, verification status or presence of any issues. Users can save those searches, resulting in “smart lists” whose contents will be updated continuously.

Search results for marker sequences and contigs can be downloaded in FASTA or PDE (the native format of the alignment editor *PhyDE*, Müller et al., 2010) format. Contigs can be downloaded as an archive file containing associated Sanger read data in SCF or AB1 format. Furthermore, lists of all taxa and specimen data, including associated marker sequences, can be downloaded in CSV format.

BarKeeper supports communication with the GGBN data portal of the BGBM (Droege et al., 2014) via their BioCAsE Provider Software. For isolate IDs found in this database, the app will automatically fetch information about these isolates and the associated taxa, herbarium specimens and tissue samples.

2.3 | High-throughput sequencing data handling

The built-in HTS barcoding pipeline accepts as input HTS data in FASTA, FASTQ, or BAM format and tar archives containing files in those formats. Uploading these data is possible by providing a local path or WebDAV address. Users can upload additional files specifying sequence IDs, amplicon primers, tags and indices, if multiple tagged plates were run. HTS data are processed in the background, either locally or on a connected, more powerful server.

HTS data processing consists of three significant steps: (i) data pre-processing, including extensive quality filtering, (ii) demultiplexing and data sorting and (iii) taxonomic assignment (Figure 1). Before starting an analysis, users can supply target taxa. Each centroid sequence is used in a BLAST search against Genbank to add a taxonomic assignment to the cluster. Should this taxonomic assignment not match the target taxa provided by the user, this cluster will be flagged for manual review. Additional run statistics, such as the number of high-quality sequences or the number of taxon clusters, can also be reviewed inside the app.

2.4 | Miscellaneous features

Admin users can create unlimited projects within the app and assign records such as lab and taxonomic data and user accounts to those. All users can filter the entire app content by any project to which they have been assigned; the currently selected project can be set within each user's profile. This filtering affects all tables and diagrams. The project structure itself can be used to organize records belonging to different subprojects. Furthermore, users can be assigned specific responsibilities, like taxonomic or experimental work. Such an assignment influences the types of records users can access, newly create, edit or delete. Guest users only have the right to view records. Only admins can manage users, projects and responsibilities.

BarKeeper can run more computationally intensive jobs on a more powerful external server, like taxonomy-based quality control or HTS data handling. This configuration allows users to set up the app itself on a machine with limited computational capabilities, which is particularly useful if resource-intensive jobs are run relatively infrequently.

3 | RESULTS AND DISCUSSION

BarKeeper was created to meet the needs of plant barcoding in a single, web-based application and thereby poses a flexible and powerful solution for barcoding initiatives and research groups engaging in phylogenetic research. It does not restrict the number or type of used markers. This feature distinguishes our app from other tools, like the Barcode of Life Data Systems (Ratnasingham & Hebert, 2007), which are limited to specific barcoding markers.

BarKeeper allows users to collect, store and manage biodiversity data concerning their collected specimens and target taxa. Although other software products like the *Diversity Workbench* (Triebel et al., 1999; Weibulat et al., 2013) also provide this functionality, they do not model associated molecular data. *BarKeeper* offers extensive options for assembling and working with Sanger sequencing reads and HTS data. This feature is similar to what can be found in software specifically focusing on this, like *Geneious* (Kearse et al., 2012), while also allowing users to manage laboratory or taxon information connected to the stored molecular data.

By combining these features, *BarKeeper* covers all steps from generating reference taxa lists and collecting specimens over generating molecular sequences and DNA barcodes, in addition to regular checks of sequence quality and correctness of taxonomic labels. Admin users can structure the data and users by creating subprojects and assigning user roles to grant specific access and editing rights based on the project's demands.

All features of *BarKeeper* rely on a single underlying database structure. Therefore, its functions interact and interconnect seamlessly without additional tools or programs. As a result, workflows become more efficient while reducing data loss, for example, due to file format conversions. Since *BarKeeper* is based on open web

technology, users can use it on every device capable of running a web browser, independent of the operating system. Setting *BarKeeper* up is possible on any machine with Docker installed and only needs to be done once to allow all project users access to the web application.

BarKeeper is released under the terms of the GNU Affero General Public Licence (AGPL 3) and can be used by any group of researchers free of charge; costs apply only to maintaining server infrastructure. Having all components on self-managed infrastructure ensures full control over critical and sensitive research data.

3.1 | Usage in GBOL

Initially, the application presented in this paper was developed in the context of the GBOL5 project, a sub-project of the German Barcode of Life initiative aimed at barcoding German land plant species. The website <https://gbol5.de> shows the app's current state as used in this project.

As more features were added, it quickly became apparent that other researchers could benefit from a similar app. Therefore, the code was refactored to be less project-specific and more widely applicable. Finally, as described in this paper, the resulting app code and its dependencies were combined into a set of Docker containers.

A demo application of *BarKeeper* is available at <http://barkeeper.uni-muenster.de/>. The credentials for a test user account are user@example.com, with the password being "barkeeper2022".

3.2 | Future development

BarKeeper is under active development. Next to minor changes and improvements to ensure a stable functionality and satisfying user experience, we will add more tools to work with the generated barcoding data, such as statistical evaluations of intra- versus infraspecific variability and sizes of barcoding gaps.

With the ever-increasing significance of HTS data in barcoding studies, new features to deal with different raw input data will become necessary. An important question is how marker sequences can be generated from read populations and if that is the best strategy. Future reference databases might also contain read populations against which sequence data can be queried.

Users who want to join the development process and add custom features can retrieve the app code from GitHub. The app code includes a test suite covering unit and integration tests to ensure future updates to the code will not break existing functionality.

AUTHOR CONTRIBUTIONS

Kai Frank Müller, Dietmar Quandt and Thomas Borsch conceived the overall app design and database structure. Kai Frank Müller and Sarah Wiechers lead-programmed the application. Kai Frank Müller supervised programming and testing. Susann Wicke designed the HTS pipeline module and supervised its testing. Lara Marie Kösters lead-programmed the HTS pipeline module. Sarah Wiechers wrote the article with contributions and a critical review of all authors.

ACKNOWLEDGEMENTS

We would like to thank the IT services of WWU Münster and HU Berlin for their quick and excellent support. Kai Frank Müller, Dietmar Quandt and Thomas Borsch received funding from the German Federal Ministry of Education and Research via the GBOL initiative (BMBF FKZ 01L1101 and 01L11501). Kai Frank Müller and Dietmar Quandt received support from the DAAD via the CoCiBio project. Financial Support for IT and server infrastructure from the Faculty of Biology at WWU Münster to Susann Wicke is also gratefully acknowledged. We acknowledge support from the Open Access Publication Fund of the University of Muenster. Open Access funding enabled and organized by Projekt DEAL.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

The complete source code for *BarKeeper* is accessible via GitHub (<https://github.com/DNA-BarKeeper/barkeeper>) under the terms of GNU AGPL3. The software as presented in this manuscript is archived with Zenodo <https://doi.org/10.5281/zenodo.7319392> (Wiechers et al., 2022). A demo application can be reviewed at <http://barkeeper.uni-muenster.de/>. The credentials for a test user account are user@example.com, with the password being 'barkeeper2022'.

ORCID

Sarah Wiechers  <https://orcid.org/0000-0002-3899-3461>

Lara Marie Kösters  <https://orcid.org/0000-0002-7913-2377>

Dietmar Quandt  <https://orcid.org/0000-0003-4304-6028>

Thomas Borsch  <https://orcid.org/0000-0002-5724-2786>

Susann Wicke  <https://orcid.org/0000-0001-5785-9500>

Kai Frank Müller  <https://orcid.org/0000-0002-0962-3824>

REFERENCES

- Ahmed, S., Ibrahim, M., Nantasenamat, C., Nisar, M. F., Malik, A. A., Waheed, R., Ahmed, M. Z., Ojha, S. C., & Alam, M. K. (2022). Pragmatic applications and universality of DNA barcoding for substantial organisms at species level: A review to explore a way forward. *BioMed Research International*, 2022, 1–19. <https://doi.org/10.1155/2022/1846485>
- Droege, G., Barker, K., Astrin, J. J., Bartels, P., Butler, C., Cantrill, D., Coddington, J., Forest, F., Gemeinholzer, B., Hobern, D., MacKenzie-Dodds, J., Ó Tuama, É., Petersen, G., Sanjur, O., Schindel, D., & Seberg, O. (2014). The global genome biodiversity network (GGBN) data portal. *Nucleic Acids Research*, 42(D1), D607–D612. <https://doi.org/10.1093/nar/gkt928>
- Geiger, M. F., Astrin, J. J., Borsch, T., Burkhardt, U., Grobe, P., Hand, R., Hausmann, A., Hohberg, K., Krogmann, L., Lutz, M., Monje, C., Misof, B., Morinière, J., Müller, K., Pietsch, S., Quandt, D., Rulik, B., Scholler, M., Traunspurger, W., ... Wägele, W. (2016). How to tackle the molecular species inventory for an industrialized nation—Lessons from the first phase of the German Barcode of Life initiative GBOL (2012–2015). *Genome*, 59(9), 661–670. <https://doi.org/10.1139/gen-2015-0185>
- Hebert, P. D. N., Braukmann, T. W. A., Prosser, S. W. J., Ratnasingham, S., deWaard, J. R., Ivanova, N. V., Janzen, D. H., Hallwachs, W., Naik, S., Sones, J. E., & Zakharov, E. v. (2018). A Sequel to Sanger: Amplicon

- sequencing that scales. *BMC Genomics*, 19(1), 1–14. <https://doi.org/10.1186/S12864-018-4611-3/FIGURES/10>
- Hebert, P. D. N., Cywinska, A., Ball, S. L., & DeWaard, J. R. (2003). Biological identifications through DNA barcodes. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(1512), 313–321. <https://doi.org/10.1098/rspb.2002.2218>
- Hebert, P. D. N., Ratnasingham, S., & de Waard, J. R. (2003). Barcoding animal life: Cytochrome *c* oxidase subunit 1 divergences among closely related species. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(suppl. 1), S96–S99. <https://doi.org/10.1098/rsbl.2003.0025>
- Holetschek, J., Dröge, G., Güntsch, A., & Berendsohn, W. G. (2012). The ABCD of primary biodiversity data access. *Plant Biosystems - An International Journal Dealing with All Aspects of Plant Biology*, 146(4), 771–779. <https://doi.org/10.1080/11263504.2012.740085>
- Hollingsworth, P. M., Graham, S. W., & Little, D. P. (2011). Choosing and using a plant DNA barcode. *PLoS ONE*, 6(5), e19254. <https://doi.org/10.1371/journal.pone.0019254>
- Hollingsworth, P. M., Li, D.-Z., van der Bank, M., & Twyford, A. D. (2016). Telling plant species apart with DNA: From barcodes to genomes. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 371, 20150338. <https://doi.org/10.1098/rstb.2015.0338>
- Kearse, M., Moir, R., Wilson, A., Stones-Havas, S., Cheung, M., Sturrock, S., Buxton, S., Cooper, A., Markowitz, S., Duran, C., Thierer, T., Ashton, B., Meintjes, P., & Drummond, A. (2012). Geneious basic: An integrated and extendable desktop software platform for the organization and analysis of sequence data. *Bioinformatics*, 28(12), 1647–1649. <https://doi.org/10.1093/bioinformatics/bts199>
- Kolter, A., & Gemeinholzer, B. (2021). Plant DNA barcoding necessitates marker-specific efforts to establish more comprehensive reference databases. *Genome*, 64(3), 265–298. <https://doi.org/10.1139/gen-2019-0198>
- Korotkova, N., Borsch, T., Quandt, D., Taylor, N. P., Müller, K. F., & Barthlott, W. (2011). What does it take to resolve relationships and to identify species with molecular markers? An example from the epiphytic Rhipsalideae (Cactaceae). *American Journal of Botany*, 98(9), 1549–1572. <https://doi.org/10.3732/ajb.1000502>
- Kozlov, A. M., Zhang, J., Yilmaz, P., Glöckner, F. O., & Stamatakis, A. (2016). Phylogeny-aware identification and correction of taxonomically mislabeled sequences. *Nucleic Acids Research*, 44(11), 5022–5033. <https://doi.org/10.1093/nar/gkw396>
- Lücking, R., Aime, M. C., Robbertse, B., Miller, A. N., Ariyawansa, H. A., Aoki, T., Cardinali, G., Crous, P. W., Druzhinina, I. S., Geiser, D. M., Hawksworth, D. L., Hyde, K. D., Irinyi, L., Jeewon, R., Johnston, P. R., Kirk, P. M., Malosso, E., May, T. W., Meyer, W., ... Schoch, C. L. (2020). Unambiguous identification of fungi: Where do we stand and how accurate and precise is fungal DNA barcoding? *IMA Fungus* 2020 11:1, 11(1), 1–32. <https://doi.org/10.1186/S43008-020-00033-Z>
- Meyer, C. P., & Paulay, G. (2005). DNA barcoding: Error rates based on comprehensive sampling. *PLoS Biology*, 3(12), e422. <https://doi.org/10.1371/journal.pbio.0030422>
- Müller, K., Quandt, D., Müller, J., & Neinhuis, C. (2010). *PhyDE, version 0.9971: Phylogenetic data editor*.
- Quail, M. A., Smith, M., Coupland, P., Otto, T. D., Harris, S. R., Connor, T. R., Bertoni, A., Swerdlow, H. P., & Gu, Y. (2012). A tale of three next generation sequencing platforms: Comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. *BMC Genomics*, 13(1), 341. <https://doi.org/10.1186/1471-2164-13-341>
- Ratnasingham, S., & Hebert, P. D. N. (2007). BOLD: The barcode of life data system: Barcoding. *Molecular Ecology Notes*, 7(3), 355–364. <https://doi.org/10.1111/j.1471-8286.2007.01678.x>
- Rhoads, A., & Au, K. F. (2015). PacBio sequencing and its applications. In *Genomics, proteomics & bioinformatics* (Vol. 13, Issue 5, pp. 278–289). Beijing Genomics Institute. <https://doi.org/10.1016/j.gpb.2015.08.002>
- Shneyer, V. S., & Rodionov, A. V. (2019). Plant DNA Barcodes. *Biology Bulletin Reviews*, 9(4), 295–300. <https://doi.org/10.1134/S2079-08641904008X>
- Triebel, D., Hagedorn, G., Jablonski, S., & Rambold, G. (1999). *Diversity Workbench - A virtual research environment for building and accessing biodiversity and environmental data*.
- Weibulat, T., Rollinger, G., Reichert, W., Weiss, M., Neubacher, D., Volz, B., Rambold, G., & Triebel, D. (2013). DiversityMobile as part of the diversity workbench platform - A data pipeline from smartphone recording up to GBIF access. In *TDWG 2013 annual conference*.
- Wiechers, S., Kösters, L. M., Quandt, D., Borsch, T., Wicke, S., & Müller, K. F. (2022). DNA-BarKeeper/barkeeper: Full release (v1.0.0). *Zenodo*. <https://doi.org/10.5281/zenodo.7319392>

How to cite this article: Wiechers, S., Kösters, L. M., Quandt, D., Borsch, T., Wicke, S., & Müller, K. F. (2023). *BarKeeper*—a versatile web framework to assemble, analyse and manage DNA barcoding data and metadata. *Methods in Ecology and Evolution*, 14, 799–805. <https://doi.org/10.1111/2041-210X.14047>