

Movement Ecology

MoveApps - a serverless no-code analysis platform for animal tracking data

--Manuscript Draft--

Manuscript Number:	MOVE-D-21-00097R2	
Full Title:	MoveApps - a serverless no-code analysis platform for animal tracking data	
Article Type:	Software article	
Funding Information:	Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg	Dr Martin Wikelski
	Knobloch Family Foundation	Dr Martin Wikelski
	National Aeronautics and Space Administration (80NSSC21K1182)	Ms Sarah Davidson
Abstract:	Background Bio-logging and animal tracking datasets continuously grow in volume and complexity, documenting animal behaviour and ecology in unprecedented extent and detail, but greatly increasing the challenge of extracting knowledge from the data obtained. A large variety of analysis methods are being developed, many of which in effect are inaccessible to potential users, because they remain unpublished, depend on proprietary software or require significant coding skills. Results We developed MoveApps, an open analysis platform for animal tracking data, to make sophisticated analytical tools accessible to a global community of movement ecologists and wildlife managers. As part of the Movebank ecosystem, MoveApps allows users to design and share workflows composed of analysis modules (Apps) that access and analyse tracking data. Users browse Apps, build workflows, customise parameters, execute analyses and access results through an intuitive web-based interface. Apps, coded in R or other programming languages, have been developed by the MoveApps team and can be contributed by anyone developing analysis code. They become available to all user of the platform. To allow long-term and cross-system reproducibility, Apps have public source code and are compiled and run in Docker containers that form the basis of a serverless cloud computing system. To support reproducible science and help contributors document and benefit from their efforts, workflows of Apps can be shared, published and archived with DOIs in the Movebank Data Repository. The platform was beta launched in spring 2021 and currently contains 49 Apps that are used by 316 registered users. We illustrate its use through two workflows that (1) provide a daily report on active tag deployments and (2) segment and map migratory movements. Conclusions The MoveApps platform is meant to empower the community to supply, exchange and use analysis code in an intuitive environment that allows fast and traceable results and feedback. By bringing together analytical experts developing movement analysis methods and code with those in need of tools to explore, answer questions and inform decisions based on data they collect, we intend to increase the pace of knowledge generation and integration to match the huge growth rate in bio-logging data acquisition.	
Corresponding Author:	Andrea Kölzsch Max Planck Institute of Animal Behavior: Max-Planck-Institut für Verhaltensbiologie GERMANY	
Corresponding Author E-Mail:	akoelzsch@ab.mpg.de	
Corresponding Author Secondary Information:		
Corresponding Author's Institution:	Max Planck Institute of Animal Behavior: Max-Planck-Institut für Verhaltensbiologie	
Corresponding Author's Secondary		

Institution:	
First Author:	Andrea Kölzsch
First Author Secondary Information:	
Order of Authors:	Andrea Kölzsch Sarah Davidson Dominik Gauggel Clemens Hahn Julian Hirt Roland Kays Ilona Lang Ashley Lohr Benedict Russell Anne Scharf Gabriel Schneider Candace Vinciguerra Martin Wikelski Kamran Safi
Order of Authors Secondary Information:	
Response to Reviewers:	Dear Editor, We would like to submit our revised manuscript 'MoveApps - a serverless no-code analysis platform for animal tracking data' (ID MOVE-D-21-00097) that you kindly invited for minor revision as a Software paper in Movement Ecology. We are grateful to the associated editor and the third reviewer for their feedback and helpful, constructive comments. Detailed responses to all points including extracts of the new text are listed below. Please note that the example workflows are already uploaded in the Movebank Data Repository with related DOIs and included as references in the manuscript. The DOIs will become active at acceptance of the paper. We hope that we have sufficiently met all the concerns and that you are willing to take our revised manuscript into consideration for publication. We are looking forward to your reaction, Yours sincerely, On behalf of the authors, Dr. Andrea Kölzsch ----- Associate Editor: Dear authors, I read carefully the revised manuscript, your response the the reviews, and the review of Reviewer #3, below. I thank you for addressing the concerns of Reviewers #1 and #2. The revised manuscript is much improved and addresses their concerns in reasonable ways, with good justifications. To some extent, the review of reviewer #3 echoes some of the concerns of Reviewers #1 and #2. In particular, the reviewer writes that some of the claims that the paper makes are better evaluated a few years after the system has been in use. However, the reviewer does not recommend rejecting the paper but a revision; this is based, I

believe, on seeing the value of early publication of such a paper, both in order to promote use of the system, and in order to make the technical discussion of merits and defects of a particular design public. This discourse is valuable.

Therefore, I recommend that the paper be accepted following a minor revision. The review below makes some good suggestions and I think it would be good for you to address as many of them as possible, but I do not think that you must address each and every one of them.

Answer: Thanks for your re-evaluation of our manuscript and your understanding of its value. We have carefully considered each comment of reviewer #3 below and adapted the manuscript accordingly.

Reviewer #3: This is a major revision of a previous manuscript, and I had not seen the previous version, although I have seen the preprint. The authors present some interesting software, MoveApps, which is an online platform that hosts modular code in the form of 'apps', and these apps can be chained together to form analysis workflows. Apps can be released to the platform by user-developers, and incorporated into pipelines by users with either low coding experience or insufficient computational power. The authors have prepared a number of apps, some of which they use in two workflows presented here. Overall, I think this is an interesting concept, there is no doubt the platform works, and the revisions have addressed many of the previous reviewers' concerns.

While the overall platform is easy to use, the manuscript falls short on advice and requirements for app development - it allows nearly anybody to copy one of the existing apps, make some changes, and publish a new app. This permissive approach is not conducive to developing high quality apps. There is no way apart from putting apps into a workflow to know that they work. As a contrasting example the 'move' package is well tested, giving users confidence that it works as intended. One other reason to include tests for each app is that it lets developers know when apps are failing on a specific version of R or one of its dependencies, for example, due to updates. I would strongly suggest the authors have much stricter requirements on apps in terms of quality (especially with regard to logging dependencies, and testing). I think the platform is a nice achievement, but the issue with app development and reproducibility I and previous reviewers have noted will need to be resolved sooner rather than later.

Answer: Thanks for your very helpful evaluation and re-evaluation of previous comments.

Answer: (A) Indeed, we have nowhere explicitly forbidden to submit slightly changed Apps of other users. It has earlier been discussed in our team to include this requirement into our Software Policy (<https://www.moveapps.org/software-policy>), but finally decided to add it rather in terms of a Submission Guideline, as it is more of a grey zone. Unfortunately, we have not yet been able to finish this guideline. However, as you see also in my replies below, each App submission must be evaluated by us as system administrator before Dockerisation, and we will not accept such Apps (but get into contact with the App developers).

Lines 240-242: Each submitted App version is checked by the MoveApps administrators for functionality, and performant custom specifications and possible duplication.

Answer: (B) Making automatic and unit tests mandatory is a task that we have recently realised to be necessary and this is now strengthened by your comments. In this manuscript and the user manual, we add the (still voluntary) suggestion for smoke tests and unit tests (with R package testthat as used in the move package) and will improve the system accordingly.

Lines 209-212: Before submission to MoveApps, the programme code of all Apps must be thoroughly tested. We provide a set of test data that all Apps must be able to process (smoke testing (Chauhan 2014)) and strongly suggest automatic unit tests

(Wickham 2011) that will become mandatory in MoveApps.

Addition in MoveApps Manual: Testing the App. Before submission to MoveApps, all App must be thoroughly tested locally using the file `copilot-\*-sdk.R`, which behaves (almost) like the online MoveApps system. So far, manual testing has worked fine for us, however we require that all Apps must be tested to run for the [four provided data sets] (https://docs.moveapps.org/files/inputs_MoveApps_SmokeTesting.zip) (smoke testing). The datasets are (1) local movements of pigeons, (2) migration tracks of geese, (3) a multiyear track of a white stork and (4) a high-resolution track of local movement of a goat on Mount Etna. Furthermore, we strongly suggest automatic unit tests (e.g. using the R package `testthat`) that will soon become mandatory and integrated into the App submission process.

Answer: (C) Regarding versions and dependencies, I had clarifications by our system developers, and indeed each App runs locally on a specified R version and by the user defined R packages and versions, and can run so in the future on MoveApps. Thus, it is truly reproducible on the platform. Due to resource problems, the R version cannot be user defined, yet, but will be at the latest when other programming languages become available. If the App is updated by the App developer, it can still run into problems if used R packages have changed and versions are not properly selected, but this is in the hands of the App developer. We give advice for avoiding this problem by using less packages and additional tests and joint effort of the community via Git.

Lines 188-190: Thus, each App runs as an independent module in its isolated Docker container with defined software programming language, version, supporting software and packages (incl. versions).

Lines 253-261: When updating single Apps to a new R environment and/or package version(s), they might cease to work properly. The limitation to a minimum number of necessary packages in an App will lower the probability of this to happen. However, due to the modular structure of MoveApps, a workflow can still run, if dysfunctional Apps are removed or replaced by similar but functional Apps, even if the output might differ. Thanks to the open source architecture and the metadata descriptions, the developers of malfunctioning Apps can be contacted by MoveApps users or administrators and the App can be updated, possibly in a joint effort via e.g. Git fork and pull requests.

Lines 453-455: Finally, we require each published workflow to be publicly shared on the MoveApps platform for easy discovery and reuse, allowing any MoveApps user to reproduce the analysis.

DETAILS:

The 'System requirements and design decisions' section has been added in response to previous reviews, and I think it addresses both earlier reviewers' concerns about the choice of platform programming languages, choosing containers over VM-s, and Kubernetes to deploy containers.

The section makes four technical claims (L. 170 - 175), and claim (1) is self-evident. However, re: claim (2), it's not clear how the workflows are meant to be made reproducible in the long run. Yes, the apps are Docker containers, but the specifics of how users should build their own apps (I did look at the online documentation, and the copilot-sdk Rproject provided), and how the MoveApps team containerises the apps, are unclear (see questions on app development). For example, it is evident that the apps' Github repositories are not sufficient to build containers - e.g. they do not contain a Dockerfile - so where is this important step happening? If the authors have a workflow to build Docker images from apps' R code, then they should ideally release that as well.

Answer: Regarding the reproducibility, it is suggested in the User Manual and now added to the manuscript, that published workflows be also made into public workflows on MoveApps so that all users can copy them to their dashboard. These public workflows will consist of the Apps in exactly the versions and with the R version(s) that were used to build (and publish) the workflow and can be used also in the future, thus fully reproducible. Note my comment below that earlier concerns that this functionality

will cease, have proven incorrect. I am sorry for the confusion. See the relevant text passages:

Lines 453-455: Finally, we require each published workflow to be publicly shared on the MoveApps platform for easy discovery and reuse, allowing any MoveApps user to reproduce the analysis.

Answer: Furthermore, I have now added a description of the dockerisation process, that is half-automated and based on information provided by App developers in the App specification file:

Lines 243-247: The MoveApps administrator specifies the Docker file in a semi-automated manner that relies on the dependency details (packages and versions) given in the App's custom specification file (see above). After automatic deployment of the App version by the system's build infrastructure, the App becomes available to all users on MoveApps.

Answer: Indeed, the Github repos don't contain dockerfiles, because we chose to keep the process of contributing App as simple as possible, to lower the threshold for users to contribute. However, with the copilot-sdk.R file, information about package versions and the App function code, the calculations of the App can be run on a local system.

With regard to claim (3), the examples shown in the workflows use data presumably from 2015 (based on the reference to Koelzsch et al. 2015), so this does not really show an example of application to "near-real-time data feeds". The authors mention a project in which MoveApps are currently used (<https://ceg.osu.edu/animal-tracking-y2y>) - that website links to data collected 1999 - 2009; perhaps an ongoing study would be a better example.

Answer: We are aware of the age of the used data set, however, up-to-date will always be old once a manuscript is actually read by somebody. We have decided not to change the example use cases, but have added two up-to-date examples of collaborations. For example, see this website: https://ceg.osu.edu/Y2Y_Room2Roam

Lines 285-289: MoveApps is integrated into multiple ongoing conservation-focused projects (e.g. Room To Roam: Y2Y Wildlife Movement by Ohio State University, Cluster-based Detection of Vulture Poisoning by North Carolina Zoo), and additional workshops, user training sessions and hackathons are planned for 2022/2023.

Finally, with regard to claim (4), there is some mention of tackling high usage using Kubernetes for scalability (L. 190 - 194) - but some practical things are unclear: (a) How many apps and/or workflows can be run at the same time? (b) Where are the major system bottlenecks? (c) Does the runtime of apps and workflows increase if there are many dozens of apps running at once? (d) If there are limits on this, what sort of expansion is planned?

Answer: We have now added some of the suggested details. The planned expansion to Amazon web services or similar has been mentioned before:

Lines 379-381: Workflows are managed to concurrently always activate two Apps, thus reserving system memory, which is the main bottleneck in App execution. In the present system, up to 20 workflows can run at once, additional requests are cued.

Lines 544-548: As demand might increase in the future or the request for faster processing of workflows becomes critical, the use of Kubernetes orchestration in MoveApps allows distributed computing with the possibility to involve commercial partners like Amazon, Google, IBM, Microsoft, Baidu or institutional cloud computing resources for improved performance and scaling.

The section 'Comparison with other movement analysis tools' has been added in response to previous reviews. While I appreciate the current text, the authors have brushed off the suggestion to compare with existing frameworks for organising reproducible computational pipelines, instead comparing with existing, single-purpose tools such as 'ctmmweb', or R itself which is an entire language. The reviewer linked

paper (doi:10.1111/1365-2656.13610) presents conceptual guidance for organising computational pipelines, and MoveApps clearly allows such pipelines to be practically implemented in a neat, modular way that can extend beyond pre-processing. I would agree with the earlier reviewer that this comparison would be useful and is required.

Answer: We see the linked paper as an example of a useful workflow, but not as a framework comparable with MoveApps. We have now mentioned it in relation to the usefulness of workflows and pipelines:

Lines 534-536: In addition, the combination of Apps into workflows allows for an unprecedented ability to run more complex analyses and computational pipelines (Gupte et al. 2022).

In relation to reviewer 1's main concerns, (2, 5) are reasonably addressed.

(1) I agree with the authors' reasoning regarding the use of containers. Yet, details about the actual containerisation of the apps' R code are missing - especially with regard to how the correct R version and package versions are restored to the container environment. It would be good to mention what the containerisation process actually involves.

Answer: As mentioned above, the correct R package versions are specified in the App specifications file (appspec.json) that is used in the half-automated dockerisation process. Right now, the administrator selects the most recent R version, but soon also this selection will be made by the App developer in a new section of the appspec.json. This has been integrated into the text:

Lines 243-246: The MoveApps administrator specifies the Docker file in a semi-automated manner that relies on the dependency details (packages and versions) given in the App's custom specification file (see above). After automatic deployment of the App version by the system's build infrastructure, the App becomes available to all users on MoveApps.

(3) I think specifically addressing the issue of security vulnerabilities is beyond the scope of this software paper, but I would recommend the authors think about and mention some basic issues - for example, are there sensible limits on how many apps users can launch at once, or on how much data they can request from Movebank? Can a poorly coded app crash the system?

Answer: There is presently no limit, how many workflows (apps) a user can launch manually on MoveApps, however the scheduled/automated launching is limited by quotas that have to be renewed. Data request by Movebank is not limited, but safe, because users have to login with their Movebank accounts. Our system can only handle up to 2 million locations in a workflow due to memory limits, which might work as a download limit. A poorly coded App cannot crash the system, only lead to errors in workflows, because the attributed memory for each App is fixed. Users will be able to report such problems with certain Apps to us via a feedback-page of user messages, which is planned, or by our support Email address that is in place.

Lines 417-420: To avoid system overload by scheduled workflows that are not used any more, we have set a quota of 12 or 30 repeats (depending on run intervals) that needs to be reset by the user. A note on the current state of the quota is included in each notification E-mail.

Lines 371-372: Presently, analyses on data sets of up to 2 million locations are possible in a MoveApps workflow.

(4) The use of 'moveStack' as the main data format is also reasonably addressed. Yet, one major process in animal movement analysis is linking positions with environmental data in the form of raster layers. I would recommend addressing whether and how the platform intends to allow users to incorporate such data (for example, from local upload, or Google Earth Engine, potentially requiring Python or JavaScript as well as in-app Google authentication).

Answer: Such annotation tasks have repeatedly been requested to us. We are

planning to adapt the platform to allow making Apps that require additional files from users that they can provide via their Dropbox or Google Docs account that can be linked to MoveApps. This will work similarly to the Cloud Storage App, where such files can be selected to upload directly. This is shortly mentioned in the conclusion/outlook section of the manuscript:

Lines 568-571: The inclusion of additional data that are commonly used in analyses of animal tracking data, such as remote sensing information, will be further defined in the coming year with the addition of planned Apps that incorporate such sources.

Regarding reviewer 1's lesser concerns, (1, 4, 6) have been reasonably addressed. (2) I share reviewer 1's concern on app development (see more below). The current addition L. 236 - 237 is insufficient to explain how apps are prepared for the platform. The authors should be much more strict about what's expected of app developers, in terms of app organisation and testing (for example, structuring apps as packages with scripts; see below). This also applies to the apps currently available - none of them appear to include any unit tests on their central R function. This is not really criticism of the platform, but since a major draw of the whole enterprise is the apps already available, they should ideally follow software best practices.

Answer: See also above. For the available Apps we have used manual testing, but have recently become aware of the necessity of automatic unit testing. This will become mandatory soon, meanwhile we provide 4 data sets that can be used for smoke testing.

Lines 209-212: Before submission to MoveApps, the programme code of all Apps must be thoroughly tested. We provide a set of test data that all Apps must be able to process (smoke testing (Chauhan 2014)) and strongly suggest automatic unit tests (Wickham 2011) that will become mandatory in MoveApps.

(3) The authors should really consider some form of continuous integration/deployment, because the system of app building and updating they describe will become unsustainable once the platform has many users (analysts and developers alike). If they already do, this should be mentioned.

Answer: Thanks for the suggestion. Indeed, we already use automatic deployment in a half-automated review and docker-building tool that is part of MoveApps (administration view). We have added a description of it to the manuscript.

Lines 243-246: The MoveApps administrator specifies the Docker file in a semi-automated manner that relies on the dependency details (packages and versions) given in the App's custom specification file (see above). After automatic deployment of the App version by the system's build infrastructure, the App becomes available to all users on MoveApps.

(5) I think the reviewer's concern has been addressed with reference to the version archiving. Yet the response makes an intriguing statement: "... the Apps may not be able to run properly in an updated MoveApps system with updated R environment in several years ...". Is it not possible for apps to specify which R version and R package versions are to be used, and these to be used in their container? This leads me to imagine that all apps' R code is wrapped into a container that uses 'rocker/r-base' or 'rocker/geospatial', which provides the current R version. A possible consequence of this implementation is that apps will eventually become incompatible with the current R version and need updates - but why should this be the case, if a simple app using an old R version is performing fine?

Answer: This issue has led to a more detailed discussion with our system development team, and indeed the above statement has been wrong. Only the base system (Linux) is fixed and the same for all containers (as opposed to VMs), not the R environment. Thus, indeed each App is specified with an R version and R package versions and can run like that in any future. Thus, Apps don't need updates and reproducibility of public workflows is given.

Lines 188-190: Thus, each App runs as an independent module in its isolated Docker

	container with defined software programming language, version, supporting software and packages (incl. versions). The statement L. 241 - 242 is good to know, but it seems mostly designed with the actual function of an app changing over time (e.g. output a PNG rather than KML). I also appreciate the added reference to the archiving service (L. 438 - 444), but what the authors describe sounds as though users who want to replicate a specific pipeline that is no longer compatible with the (future) current R version, will have to download the archived code, and restore the correct R and package versions from CRAN themselves. Users who are using MoveApps due to low coding skills would find this challenging, so I think this aspect does not really serve the community MoveApps is targeting. So overall, I don't think that solving the reproducibility problem theoretically is sufficient when releasing new software for community uptake. Answer: See my previous reply. It is possible to run public workflows on MoveApps (that are related to published ones) in their original App versions and containers, which makes them fully reproducible on MoveApps. The addition of the ability to rebuild the system locally is given for backup or if people prefer local reproduction of the workflow. Lines 453-455: Finally, we require each published workflow to be publicly shared on the MoveApps platform for easy discovery and reuse, allowing any MoveApps user to reproduce the analysis.
Additional Information:	
Question	Response
Is this study a clinical trial? A clinical trial is defined by the World Health Organisation as 'any research study that prospectively assigns human participants or groups of humans to one or more health-related interventions to evaluate the effects on health outcomes'.	No

MoveApps - a serverless no-code analysis platform for animal tracking data

Andrea Kölzsch* ^{1,2}, Sarah C. Davidson^{1,2,3,4}, Dominik Gauggel⁵, Clemens Hahn⁵, Julian Hirt⁵,
Roland Kays^{6,7}, Ilona Lang⁸, Ashley Lohr⁶, Benedict Russell⁵, Anne K. Scharf^{1,2}, Gabriel
Schneider⁸, Candace M. Vinciguerra^{1,6}, Martin Wikelski^{1,2,4}, Kamran Safi^{1,2}

¹ Department of Migration, Max Planck Institute of Animal Behavior, Radolfzell, Germany

² Department of Biology, University of Konstanz, Konstanz, Germany

³ Department of Civil, Environmental and Geodetic Engineering, The Ohio State University,
Columbus, OH, USA

⁴ Centre for the Advanced Study of Collective Behaviour, University of Konstanz, Konstanz,
Germany

⁵ couchbits GmbH, Konstanz, Germany

⁶ North Carolina Museum of Natural Sciences, Raleigh, NC, USA

⁷ Department of Forestry and Environmental Resources, North Carolina State University,
Raleigh, NC, USA

⁸ Communication, Information, Media Centre, University of Konstanz, Konstanz, Germany

20 * Corresponding author: Andrea Kölzsch, Department of Migration, Max Planck Institute of
21 Animal Behavior, Am Obstberg 1, 78315 Radolfzell, Germany, akoelzsch@ab.mpg.de

22

Abstract

24

Background

Bio-logging and animal tracking datasets continuously grow in volume and complexity, documenting animal behaviour and ecology in unprecedented extent and detail, but greatly increasing the challenge of extracting knowledge from the data obtained. A large variety of analysis methods are being developed, many of which in effect are inaccessible to potential users, because they remain unpublished, depend on proprietary software or require significant coding skills.

Results

We developed MoveApps, an open analysis platform for animal tracking data, to make sophisticated analytical tools accessible to a global community of movement ecologists and wildlife managers. As part of the Movebank ecosystem, MoveApps allows users to design and share workflows composed of analysis modules (Apps) that access and analyse tracking data. Users browse Apps, build workflows, customise parameters, execute analyses and access results through an intuitive web-based interface.

Apps, coded in R or other programming languages, have been developed by the MoveApps team and can be contributed by anyone developing analysis code. They become available to all user of the platform. To allow long-term and cross-system reproducibility, Apps have public source code and are compiled and run in Docker containers that form the basis of a serverless cloud computing system. To support reproducible science and help contributors

document and benefit from their efforts, workflows of Apps can be shared, published and archived with DOIs in the Movebank Data Repository.

The platform was beta launched in spring 2021 and currently contains 49 Apps that are used by 316 registered users. We illustrate its use through two workflows that (1) provide a daily report on active tag deployments and (2) segment and map migratory movements.

Conclusions

The MoveApps platform is meant to empower the community to supply, exchange and use analysis code in an intuitive environment that allows fast and traceable results and feedback. By bringing together analytical experts developing movement analysis methods and code with those in need of tools to explore, answer questions and inform decisions based on data they collect, we intend to increase the pace of knowledge generation and integration to match the huge growth rate in bio-logging data acquisition.

Keywords

animal movement, movement ecology, bio-logging, method sharing, community empowerment, analysis code publication, reproducibility, cloud infrastructure, serverless computing

Background

The growing field of bio-logging and animal tracking allows us to follow and document the movement behaviour and ecology of animals and species to an unprecedented extent and level of detail (Kays et al. 2015; Wilmers et al. 2015). However, as data volume and complexity have expanded, the extraction of knowledge has become increasingly challenging. The field of movement ecology has joined the big-data sciences: Tracking and bio-logging datasets comply with the "Four Vs Framework" (Volume, Variety, Veracity, Velocity) and their analysis "exceeds the capacity or capability of current or conventional methods and systems" (Farley et al. 2018).

For many users of bio-logging devices, the ability to fully exploit the information contained in tracking data increasingly lags behind the technological capacities (Holyoak et al. 2008). Some devices provide so much and such complex information that basic exploration of the data becomes a first major obstacle (Slingsby and van Loon 2016). As a result, experienced field biologists and wildlife managers must join forces with computational movement ecologists to process data appropriately in the quest to answer underlying ecological, management and conservation questions (Williams et al. 2020; Joo et al. 2020b). After collection, organisation and quality control, data are typically visually and analytically explored and processed in an iterative approach (Gupte et al. 2022). Following initial analysis, results often provide important insight leading to data re-analysis, data fusion (i.e. association with other ancillary information such as remote sensing data) or integration of additional data collected that were ignored in initial processing. This process results in new and often bespoke methodological workflows and analysis code (Reichman et al. 2011), but

is tedious and not particularly sustainable or transparent (Lowndes et al. 2017) and requires accessory effort and investment to bring together the right combination of skills and interests in the research teams.

Ideally, these methods, workflows and analysis code compilations should be shared, compared, assessed and re-used or adapted across research groups and management agencies (Peng 2011). Indeed, many standard as well as novel analytic methods are being made available as open access code or functions in R-packages (Joo et al. 2020a; R-Core-Team 2021). R has become by far the most preferred software package for (movement) ecology, because it is open source and a large community contributes and maintains packages, continuously extending its scope and user community (Lai et al. 2019; Joo et al. 2020b). R-code and R-functions can allow efficient processing, exploration and robust analysis of datasets that cannot easily be accessed using software that has traditionally been used by field biologists (e.g. Excel or Google Earth). However, for some biologists, applied wildlife managers and those new to the discipline, the discovery, evaluation and use of this growing amount of code (Mislán et al. 2016) presents a major hurdle to being able to optimally benefit from state-of-the-art methods. Particularly for applied monitoring, conservation and management applications, it is of utmost importance that the information and insight gained from animal movement data can correctly and reliably inform decision makers, as well as support the possibility for near-real-time response when data are transmitted remotely from deployed tags.

108 The challenge of maximising the creation of knowledge from, and the beneficial use of,
109 heterogeneous bio-logging data has been raised before, regarding the storage,
110 standardisation and sharing of complex tracking datasets (Campbell et al. 2016; Sequeira et
111 al. 2021). Online tracking databases have been established that allow researchers to stream,
112 harmonise and store data from different types of tags, such as Movebank (movebank.org)
113 (Kranstauber et al. 2011), Ocean Tracking Network (oceantrackingnetwork.org) and the
114 EuroMammals family of databases (Urbano et al. 2010). These platforms perform vital steps
115 to enable efficient analysis of tracking data, for example by standardising the coordinate
116 reference system of location estimates and time zone and format of dates and times that
117 define animal occurrences, and by providing shared data access protocols. In combination
118 with appropriate metadata provided by data owners, long-term storage and exchange
119 between researchers is made possible (Davidson et al. 2020). As sharing, publishing and
120 combining data across groups and studies has become easier, so have collaborative projects
121 and an interest in novel and accessible methods that can be applied to research, teaching,
122 applied management and public engagement. The circle of ecologists participating in these
123 databases has grown, increasing the taxonomic, geographic and temporal scope of
124 harmonised data. At the same time, so has the community of developers contributing to the
125 creation of new and innovative movement analysis methods, with potential to reach the
126 crowd and citizen science community (Franzoni and Sauermann 2014).

128 One intrinsic complication of standardised and open software lies in the differences in
129 development, maintenance, update and adaptation to novel computing infrastructure.
130 Depending on requirements and preference, users may deploy new methods specific to
131 particular working environments and operating systems that often are hard to combine. In

132 addition, all operating systems need regular update and maintenance, and code and
133 hardware degrade and become obsolete over time. Software packages require continuous
134 code updates, and must often dynamically communicate with programmes and packages
135 that also change regularly. The optimal utilisation of hardware resources at the currently
136 highest performance levels requires additional maintenance and significant development
137 effort. As a consequence, maintaining reproducibility of analysis code over long periods is
138 very hard to achieve (Powers and Hampton 2019). Although code can be archived, including
139 information about the used software, versions and settings, changing computing
140 environments might make it near to impossible to execute code in future systems.

141
142 As a consequence of the above challenges and unexploited opportunities, the next step in
143 improving the efficiency and benefits of analysis in movement ecology is, in our view, to
144 foster more coordinated and inclusive cooperation between field ecologists, movement
145 analysts and programmers. Such an effort could expand access to state-of-the-art methods
146 and computing power, extend the community of experts that participate in analysis, support
147 communication and exchange between those collecting data and those developing analysis
148 methods, and secure reproducibility and scalability. Here, we introduce MoveApps
149 (moveapps.org, Fig. 1), a web-based analysis-platform for animal movement data, developed
150 with the aim to connect people who develop and drive the field of code analysis methods
151 with people that use these tools for their newly collected datasets to answer research
152 questions and inform decisions. The platform will make movement analysis methods more
153 readily available and provide fast and tractable feedback, fostering communication across
154 the range of skills and experience present in the research community. The platform enables
155 data owners and analysts to work independently with opportunity for close exchange with

each other. MoveApps is based on a serverless cloud computing system that is independent of changing infrastructure (Kearse et al. 2012; Perez et al. 2018), thus supporting long-term and flexible functionality of analysis code. The beta version of MoveApps was released in February 2021.

Implementation

System requirements and design decisions

We designed MoveApps as a modular, open-source online platform that allows the secure use and exchange of interactive, user-developed analysis modules (Apps). Similar to other modular systems (e.g. Scratch (<https://scratch.mit.edu>), Node-RED (<https://nodered.org>)), the Apps can be linked and combined into data analysis workflows (Fig. 1). This modularity maximises flexibility and minimises each App's complexity, likelihood for errors and development effort. Each individual App is a simple analysis building block that is defined by its input and output type. The analysis executed by an App is meant to be independent of a specific programming language, version, or system structure. We specified MoveApps as a serverless platform (Perez et al. 2018) that runs on a cloud computing system, thus (1) operating independent of the users' hardware, (2) providing reproducibility of workflows over a long time, (3) supporting automated routines that can be applied to near-real-time data feeds and (4) allowing scalability to future high usage by distributed and scalable computing.

For compatibility with other systems and clouds, MoveApps was designed using widely used open-source tools and languages. Its platform background is programmed in Kotlin and Java (Ardito et al. 2020). For realising it as a serverless cloud computing system, we decided to implement Apps as containers instead of Virtual machines. Both provide virtual environments in which processes can run in isolation, but instead of emulating their own host operating system, containers share an underlying host (Cito et al. 2017). That makes them faster and requires less overhead, which is sensible for our platform of many different small Apps (coded by many different developers) working together. As underlying host, we use the open-source operating system Linux GNU. The two most widely accepted container systems for Linux are LXC and Docker (Bernstein 2014). In the light of distributed computing, we selected Docker for its better portability across machines (Boettiger 2015). Thus, each App runs as an independent module in its isolated Docker container with defined programming language, version, supporting software and packages (incl. versions). This minimises cascading errors in overly variable, interconnected or interdependent sequences of Apps. The library of separately developed Apps in the form of Docker containers is automatically deployed, scaled and managed by Kubernetes (kubernetes.io), a widely used open-source container-orchestration system (Bernstein 2014). This system ensures that the Apps can interface and exchange their inputs and outputs in a safe and standardised way and supports scalability as the platform grows.

App development

The base of our modular analysis platform are the Apps: each App is meant to be developed to independently perform one or a few main function(s) on the input dataset and then output its results for further handling by a subsequent App. Apps in development (Fig. 1.1)

for the platform are managed in public Git repositories. Each repository contains the programme code for executing the App, a custom specification of the App and a documentation file adhering to our template. All functional development and testing of the App's programme code is done in the user's typical compiler/editor. In the presently running, first beta version of MoveApps, only R and R-shiny Apps are supported. We currently provide Software Development Kits with an initial R-Studio project that allows Apps under development to be locally run and perform as if they were launched on the MoveApps platform. Before submission to MoveApps, the programme code of all Apps must be thoroughly tested. We provide a set of test data that all Apps must be able to process (smoke testing (Chauhan 2014)) and strongly suggest automatic unit tests (Wickham 2011) that will become mandatory in MoveApps.

The required custom specification file (named appspec.json) can be compiled with the help of a settings editor that is provided on the MoveApps platform (moveapps.org/apps/settingseditor). This meta-information file must contain all parameter definitions, system dependencies, a selected license, language, keywords, author names and a link to the App documentation. Additional information that would be used during workflow publication can be specified, including references and funding sources. To improve metadata quality and interoperability with other services (Schneider et al. 2021), we have designed the structure and options to incorporate the DataCite metadata scheme (DataCite-Metadata-Working-Group 2021) and well-known identifiers, such as ORCID (<https://orcid.org>).

Each App requires a defined input and output type. The only input types currently supported are movement data in the “moveStack” format of the R-move-package (Kranstauber et al. 2020) or a specified .csv data frame that is internally transformed to a “moveStack”. Similarly, supported output types are "moveStack" data and, if the App can serve as a workflow endpoint, an interactive user interface (R-Shiny). The present limitation to “moveStack” ensures the proper use for movement analyses, but easy transformation to data frames and other formats in R allows future portability and the option to extend the range of interchangeable input data types. Apps can produce additional output "artefacts" (in some cases also called “products”), which are files that can be downloaded from the MoveApps platform in various formats, such as .pdf or .csv. The dataset created as the output of each App can be downloaded in R format (.rds).

After initialisation of a new App in MoveApps, which includes the definition of the runtime environment, input and output data formats and provisioning of a link to the Git repository, a first App version must be created and submitted. Each submitted App version is checked by the MoveApps administrators for functionality, performant custom specifications and possible duplication. Upon passing this short review, the submitted App is wrapped in a Docker container. The MoveApps administrator specifies the Docker file in a semi-automated manner that relies on the dependency details (packages and versions) given in the App’s custom specification file (see above). After automatic deployment of the App version by the system’s build infrastructure, the App becomes available to all users on MoveApps. Improved App versions can be submitted at any time and become available to respective App users by notification of the possibility for update of Apps they used in existing workflows. All versions of an App are stored and can be reintegrated upon demand.

248

249 The use of an open-source language such as R, to which huge numbers of developers
250 contribute, brings the challenge of interdependencies and possible inconsistencies as
251 packages and the R environment are updated. When updating single Apps to a new R
252 environment and/or package version(s), they might cease to work properly. The limitation to
253 a minimum number of necessary packages in an App will lower the probability of this to
254 happen. However, due to the modular structure of MoveApps, a workflow can still run, if
255 dysfunctional Apps are removed or replaced by similar but functional Apps, even if the
256 output might differ. Thanks to the open source architecture and the metadata descriptions,
257 the developers of malfunctioning Apps can be contacted by MoveApps users or
258 administrators and the App can be updated, possibly in a joint effort via e.g. Git fork and pull
259 requests.

260

261 **Empower the community to share and contribute**

262 One major aim of MoveApps is to empower all members of the bio-logging and movement
263 ecology community to easily contribute, use and benefit from the platform. Therefore, its
264 dashboard is arranged in a user-friendly interface to intuitively browse and select data, Apps,
265 App settings and options, and workflows by point-click-track. The users as well as App
266 developers need not be familiar with or accommodate their work to the background
267 infrastructure and can instead focus on their scientific or management questions and
268 contents of the relevant data and Apps.

269

The MoveApps platform has been developed in the spirit of Open Science, sharing and joint improvement (Franzoni and Sauermann 2014; Nosek et al. 2015; Gewin 2016; Powers and Hampton 2019). While we are providing an initial offering of Apps and sample workflows, the bulk of development of Apps to the platform is meant to be taken over by a growing movement ecology community. A thorough user manual and tutorials (docs.moveapps.org) enable (i) App users to combine Apps and create workflows for analysis of their movement data and (ii) App developers to create and submit innovative Apps to the MoveApps platform for the community to discover and adopt. Over the past year, we have introduced the platform to potential users through workshops, conference presentations and personal meetings with dozens of government agencies, non-profit organizations and academic institutions, which have also served as an opportunity to identify pressing needs and prioritize functions to implement in the first phases of App development. MoveApps is integrated into multiple ongoing conservation-focused projects (e.g. Room To Roam: Y2Y Wildlife Movement by Ohio State University, Cluster-based Detection of Vulture Poisoning by North Carolina Zoo), and additional workshops, user training sessions and hackathons are planned for 2022/2023.

All submitted Apps must be provided under a selected open license for further use. We currently allow the choice between five widely used open (software) licenses: GNU General Public License, MIT License, GNU Affero General public License, 3-Clause BSD License and Creative Commons Attribution Share Alike (for more details see <https://choosealicense.com/licenses/>). Each of these options allows free use of the App by any App user in MoveApps as well as the copying of code for further use or archiving. This

builds the basis for true reproducibility and iterative improvement of the data analysis process (Fidler et al. 2017; Powers and Hampton 2019).

The MoveApps Terms (moveapps.org/terms-of-use) clearly state that the user is responsible for evaluating the functionality and suitability of each App and workflow. MoveApps and App developers cannot be held responsible for errors or unexpected output in such a community supported open source project. However, App developers must not knowingly include malware and need to provide a current contact E-mail address. We foster an environment of active personal collaboration and productive exchange between App developers and with MoveApps to jointly improve the system and App usability. However, the containerised architecture of MoveApps allows for safe execution of code (because inputs and outputs are defined by the system) and provides the opportunity to withdraw Apps, for example if flaws are identified that cannot be feasibly resolved, without breaking workflows permanently.

Comparison with other movement analysis tools

Apart from the large list of R-packages that allow the analysis of movement data (Joo et al. 2020a), there are several standalone, specific software tools for movement ecology analyses ((Resheff et al. 2014; Calabrese et al. 2016; Dodge et al. 2021); see also www.movebank.org/cms/movebank-content/software). Compared to MoveApps, these tools require a local installation or data upload from the local computer, limiting repeatability across users/devices and usefulness for users without access to sufficient computing power. Furthermore, some existing applications are partly commercial, imposing

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

316 licensing costs and subscription plans, and by that additionally increase the hurdles of
317 interacting and analysing movement data. Monolithic standalone applications further suffer
318 from potential obscurity of the actual functionality and the underlying algorithms of the
319 implemented functions provided, and were often developed to meet a specific need, with
320 limited support and intent to offer future growth in functionality or customisation to support
321 user requests.

322
323 The one system in ecology that is somewhat comparable with MoveApps, even if not
324 serverless, is R with R-Studio itself and shinyapps.io. R is open access and most people use it
325 in a local install instance (server-based installations are possible). It allows the addition of
326 packaged functions by the community, as well as exchange and collaboration via Git.
327 However, R-Studio as frontend can only be used by coding, which is the hurdle that
328 MoveApps attempts to overcome. Shinyapps.io is a commercial online platform that allows
329 the deployment, sharing and use of R-Shiny Apps. One example is “ctmmweb” which allows
330 easy calculation of various home range measures (Calabrese et al. 2021). Similar to above
331 discussed standalone software tools, R-Shiny Apps tend to become bespoke and often
332 monolithic tools, that are difficult to adapt and alter. With its modular container structure,
333 multi-language design and open source availability of Apps, MoveApps overcomes those
334 limitations. It allows flexible and parallel improvements and variations of Apps and
335 workflows as a community service. We chose to prioritise integration of R and R-Shiny into
336 MoveApps in part to encourage integration of functions from these existing popular analysis
337 packages (Joo et al. 2020a) into the platform early on.

Results

Workflow compilation, use and scheduling

Within MoveApps, Apps can be combined into workflows (Fig. 1.2), which define an ordered set of steps to access, process and analyse data. The process of building workflows is simple and intuitive in the platform's graphical user interface, where users can browse Apps, view details of an App's developers, purpose and documentation and select chosen Apps to add to a workflow. The list of Apps is alphabetically ordered, includes a short description of each App and is searchable by keywords. Each workflow is visually represented by connected containerised Apps, including access points to e.g. App details, options with descriptions for available settings and result overviews, as well as buttons to initiate or stop workflow runs. Workflows can be saved, edited and run for specific use cases.

Every workflow starts with a core App that loads data into the system (Fig. 1.4). As MoveApps has been set up as a partner platform to the Movebank data base within the Movebank Ecosystem (Kays et al. 2022), it is most convenient to directly import animal movement data stored in Movebank using the "Movebank" App. This core App allows users to log into Movebank to browse and securely transfer data based on their user access permissions within the Movebank data base, which accommodates both public and controlled-access data , provides support to harmonize data to a shared format and vocabulary, and supports live data feeds (Kays et al. 2022). Relying on Movebank for input of data to MoveApps thus provides a secure method to share data between collaborators, allows users without access to data storage or a fast internet connection to input large data volumes, reduces problems in analysis caused by inconsistent or unknown data formats, and

supports automated reporting procedures during data collection (see example workflows below). Alternatively, uploading data files (.rds or .csv) from a personal cloud folder (Dropbox, Google Drive) is supported. This option offers flexibility to prepare multi-study datasets prior to importing to MoveApps, as well as to support Apps that incorporate other local data sources as part of tracking data analysis. The data are then passed on to the next App in the appropriate format and processed accordingly. Presently, analyses on data sets of up to 2 million locations are possible in a MoveApps workflow.

After data import, subsequent Apps can be added by selection from a list of all available Apps that accept the appropriate input and provide output in the required format. Input and output formats are filtered and matched automatically by the system. Once a workflow is compiled, it can be executed (Fig. 1.3). The user can follow the progress of each App in a workflow by the colour-indication of its state (idle, starting, working, post-processing or in error). Workflows are managed to concurrently always activate two Apps, thus reserving system memory, which is the main bottleneck in App execution. In the present system, up to 20 workflows can run at once, additional requests are cued.

Because MoveApps is cloud based, workflows run independently of the local machine and results from complex and time-intensive workflows can be checked after login at a later time. While the container structure of the workflow leads to somewhat longer runtimes in MoveApps than if the code was executed locally (see example workflows below), we consider this downside to be more than offset by the increased flexibility by users and other advantages of containers (see above). The workflow run can be stopped or re-started at any

time. R-shiny Apps that invoke user interfaces can be opened after the App has finished and its results can be examined and users can interact with it according to the App's programming features (Fig. 1.5).

App details can be viewed at any time by opening the App menu. From this menu, user can change settings or access logs (process run, warning or error messages). Users can also "pin" a workflow at a certain App to retain the results of an App and all preceding Apps in the workflow. As a result, only subsequent Apps to the "pinned" App are re-executed when a workflow is re-started. The purpose is to avoid re-running e.g. initial data access and preparation steps that can be time-consuming with large datasets, thus providing ease of use when iteratively composing workflows and testing App settings. Each App that returns data also generates a short summary of the output data (e.g. time interval, number of animals and positions), which can be viewed easily at any time after the App has finished running. This allows the user to swiftly review App results, identify possible errors or unexpected results of the App, and better understand how each App relates to the workflow output. Finally, each workflow can be cloned into several workflow instances that analyse different datasets or are run using different user-specified parameter settings in one or more of its Apps. Managed by Kubernetes, this allows parallel execution for easy exploration of the influence of the workflow's parameter space on the results. All workflows and their instances are saved in the user account for future reference.

Workflow instances can be started manually or scheduled to run automatically and without further interaction at fixed time intervals. This is especially useful when up-to-date

information about tagged animals are required on a regular basis. Results of the scheduled runs can be accessed in the MoveApps platform or via a secure API (Fig. 1.6). Users have the option to request an E-mail notification after each scheduled run is completed, containing either a link to the MoveApps site for output access and download or including selected output files as attachments. The integration of alert notifications in the E-mail is e.g. possible with the “Email Alert” App. To avoid system overload by scheduled workflows that are not used any more, we have set a quota of 12 or 30 repeats (depending on run intervals) that needs to be reset by the user. A note on the current state of the quota is included in each notification E-mail.

Share, cite and publish

For replication, collaboration or other joint work, it is possible to share workflows with other MoveApps users (Fig. 1.7). Workflows can be either shared publicly or with specific users. Recipients can load a shared workflow into their account's dashboard and edit it there independently of the original workflow. It is possible to add two kinds of messages with shared workflows: (1) an open text field that allows the user to provide a brief description of the workflow and (2) a data source message which is by default filled with details of the dataset used by the original workflow creator. Thus, sensitive data are not transferred. Recipients of workflows must access the input data from their own accounts, which maintains the integrity of data sharing rights as managed by users in Movebank.

The importance of transparency and reproducibility based on open data and open code/methods has been repeatedly highlighted (Nosek et al. 2015; Fidler et al. 2017),

especially if ecological applications are involved that can have important or controversial implications for science or management and are hard to impossible to replicate (Powers and Hampton 2019). Further, there is a need to ensure that researchers receive professional benefit and recognition for sharing code (Reichman et al. 2011). Therefore, MoveApps provides a citation for all Apps (Fig. 1.8) and offers the option to publish and acquire a digital object identifier (DOI) for workflows that are related to a published paper and dataset (Fig. 1.9).

To support reproducibility and comprehensive documentation of published analyses, the published workflows, their related Apps (including settings and source code) and metadata describing the operating system, libraries, packages and run-time versions used are archived in the Movebank Data Repository (Fig. 1.9). This is a free and well-established repository in the movement ecology community (Schneider et al. 2021; Kays et al. 2022) that provides persistent identifiers for future access and is accepted by scientific journals. The repository is developed in accordance with the FAIR (Wilkinson et al. 2016) and TRUST (Lin et al. 2020) data principles. For publication and archiving of workflows, users are required to provide a description of the workflow and each contained instance, the names of all contributors, funding sources and license type. Similar information for each App used in the workflow is extracted from their custom specification files. Finally, we require each published workflow to be publicly shared on the MoveApps platform for easy discovery and reuse, allowing any MoveApps user to reproduce the analysis. Thus, in combination with MoveApps' serverless and modular structure, this archiving service helps to ensure the future reusability of code and replicability of published results, as well as the possibility to assess, modify and improve code and related analytical methods. For replication outside of MoveApps, archived

workflows can be downloaded for local use, and old R-environments and R-package versions can be accessed from the CRAN website.

Example workflows

We illustrate the use of MoveApps with two example workflows that address common analysis needs: using the “Morning Report” and the “Migration Mapper”, we analyse a published set of migration tracks of greater white-fronted geese (*Anser a. albifrons*; Movebank study: “Migration timing in white-fronted geese (data from Kölzsch et al. 2016)”, (Kölzsch et al. 2016a). These workflows were developed to showcase the use of the platform and discuss possible extensions to the beta version. The workflows have been made public on MoveApps to be used by all registered users and have been published in the Movebank Data Repository (Kölzsch and Wikelski 2021; Kölzsch et al. 2021).

The “Morning Report” workflow (Fig. 2a, doi:10.5441/001/1.h4c0p8bv, (Kölzsch and Wikelski 2021)) is made up of two Apps, the “Movebank” App and the “Morning Report” App, where the latter extracts an overview of a dataset with times of tag activity, plots of tag properties and a small interactive map. This is meant to be used for projects with active tags to explore tag performance, identify changes in behaviour and possibly find the animals in the field. Four Apps (called “Morning Report pdf Overview”, “Morning Report pdf Attribute Plots”, “Morning Report pdf Property Plots” and “Morning Report pdf Maps”) were recently developed, which can be combined into a workflow that provides “.pdf” artefact files containing a time overview for all animals/tags, various data properties and track maps for download. These files can be taken into the field, sent by E-mail or accessed via API.

The user interface output of the workflow (Fig. 2b) reveals that there were (at least) six different animals with available data during the past 5 months in the dataset. The time range, number of locations and distances moved are indicated. For the selected animal, we can see that from mid-June to the end of August, no data were available. After this period, autumn migration commenced and the large displacements and route are visible in the plots and map. To assess performance, we ran the workflow on both MoveApps and on a local installation of R-Studio. The workflow took 3:15 min to run on MoveApps, of which the longest part was taken up by loading the data (2:55 min). In comparison, on a local system R-Studio (IntelCore i7, 16GB RAM, Windows 10 64-bit), running the same code required 2:55 min in total, with 2:46 min for loading the data. Relative performance will vary based on the available processing power available to users outside of MoveApps.

The “Migration Mapper” workflow (Fig. 3a, doi:10.5441/001/1.7tq16jr8, (Kölzsch et al. 2021)) is a more complex workflow made up of six Apps that load data from Movebank, remove outliers, thin the data, filter by season, segment the data by speed and then plot the remaining locations as a density raster. The raster plot is provided as a user interface in which the user can change raster size for more detail vs. better visibility. The division of the workflow’s functionality into the many small Apps has notable advantages: Modular runs of independent Docker instances are more stable and run on less resources than one large, complex App. Furthermore, each App can be used in new workflows or can be replaced in the present workflow by different or more advanced App versions or Apps that have similar functionality.

The user interface outputs of the two different workflow instances show the routes of greater white-fronted geese during spring migration (Fig. 3b) and autumn migration (Fig. 3c). Densely travelled areas become visible by the heat map colours and indicate movement rather than resting, because only flight locations were selected using the “Segment Data by Speed” App. The maps confirm the known differences between the two migrations: During spring the geese fly in a wide front, using many different routes, whereas during autumn most of them use the coastal route which they pass quickly (Kölzsch et al. 2016b). The runtimes of the workflow for spring and autumn migration only differed minimally, each taking about 5:20 min on MoveApps, and 3:00 min on local R-Studio (see above).

Conclusions

In a time of extreme growth of size and complexity of datasets (Wilmers et al. 2015; Joo et al. 2020b), we present the MoveApps platform as a unique tool to improve our ability to analyse movement data with the best methods in a comprehensible and efficient way. Our development showcases how movement ecology as a scientific community can be empowered to make analysis methods more accessible, in particular for to ecologists and wildlife managers. The platform offers opportunities for interactive participation by those less comfortable with command line programming, shared methods and collaboration across projects and agency jurisdictions, and management and research strategies that take advantage of dynamic monitoring and analysis of data as they are being collected.

Beyond its user-friendly interface, the MoveApps platform with searchable and citable Apps will help the community stay up to date with and explore the rapidly growing list of methods

for movement data analysis (Joo et al. 2020a). Methods will become easily accessible as citable, reproducible and community-approved Apps and can be tested, compared and further improved by the community. In addition, the combination of Apps into workflows allows for an unprecedented ability to run more complex analyses and computational pipelines (Gupte et al. 2022). Hence, the MoveApps platform is intended to accelerate scientific work, discovery and collaboration between research groups and communities.

As a serverless cloud computing facility, MoveApps runs independently of soon outdated operation systems and can be scaled to the needs of the community (Talia 2013). It can provide computing power to researchers or communities that might not have such facilities at their home institutions or who work in the field. Presently, MoveApps is hosted on the cloud infrastructure of the Max Planck Society and is free and practically unlimited for all users. As demand might increase in the future or the request for faster processing of workflows becomes critical, the use of Kubernetes orchestration in MoveApps allows distributed computing with the possibility to involve commercial partners like Amazon, Google, IBM, Microsoft, Baidu or institutional cloud computing resources for improved performance and scaling. This would come with the caveat that the running costs charged by the commercial service providers would need to be covered by the users justified by their need to analyse their data. We hope that this concept of flexible and integrative cloud-based analytics, deliberately designed to accommodate Open Science procedures can serve as a model for other research infrastructure applications in the future.

Finally, MoveApps provides a new way of making scientific research reproducible in all steps. Currently, scientific papers and datasets can be published with DOI, adding analysis methods complements this list and closes an often-encountered gap (Powers and Hampton 2019). Owing to its serverless structure, analysis methods and code in MoveApps can be permanently stored and are reproducible and openly accessible for use and improvement (Nosek et al. 2015). We believe this to be a necessary step to better promote Open Science and expect that our idea will be taken up by other research communities.

MoveApps launched its beta version in February 2021 and presently contains 49 functioning Apps that are used by 319 registered users. We invite the community to test it, provide feedback and contribute their own Apps and/or workflows. In the near future, we plan to provide more interfaces for communication between users and App developers, and include the capability to submit Apps in programming languages other than R. Based on community demands and as part of ongoing projects, Python will be integrated next, but there are no technical restrictions to extending this selection. The inclusion of additional data that are commonly used in analyses of animal tracking data, such as remote sensing information, will be further defined in the coming year with the addition of planned Apps that incorporate such sources. Additional App input and output formats will lead to different types of Apps which can be combined in various ways, leading to rapid growth and scalability of the system. To ensure an open invitation to participate and broad community input, we introduce the platform in its beta release, while the platform is available and offers basic functionalities, and while feedback can still drive the direction and priorities for future development. We encourage the community to contribute, exchange ideas and help define the future of MoveApps.

570

571 **Availability and requirements**

572 **Project name:** MoveApps

573 **Project home page:** <https://www.moveapps.org>

574 **Operating system(s):** platform independent

575 **Programming language:** Kubernetes, Docker, Kotlin/Java, R

576 **Other requirements:** none

577 **License:** General MoveApps Terms (<https://moveapps.org/terms-of-use>); selection of open
578 software licenses for contributed Apps

579 **Any restrictions to use by non-academics:** none

580

581 **List of abbreviations**

582 DOI - digital object identifier

583

584 **Declarations**

585 **Ethics approval and consent to participate**

586 Not applicable.

587

588 **Consent for publication**

Not applicable.

Availability of data and materials

The example tracks of greater white-fronted geese are available from the Movebank Data

Repository: <https://doi.org/10.5441/001/1.31c2v92f> (Kölzsch et al. 2016a) and can be

accessed from the open Movebank study "Migration timing in white-fronted geese (data

from Kölzsch et al. 2016)".

The example workflows including R-code and system specifications are available from the

Movebank Data Repository (<https://doi.org/10.5441/001/1.h4c0p8bv> (Kölzsch and Wikelski

2021) and <https://doi.org/10.5441/001/1.7tq16jr8> (Kölzsch et al. 2021)) and are globally

shared workflows in MoveApps ("Morning Report" and "Migration Mapper").

Conflict of interest

The authors declare that they have no competing interests.

Funding

The initial development of MoveApps was funded by the Ministry for Science, Research and

the Arts of Baden-Württemberg and the Knobloch Family Foundation.

Authors' contributions

KS, MW and AKS conceived and specified the idea for the platform. DG, CH, JH and BR set up, programmed and support the platform. AK and AKS programmed the Apps. AK coordinated the system development, wrote the user manual and supports users. SCD provided expertise of Movebank. AL, CMV and RK tested the platform and brought up improvements. GS, IL and SCD developed the publication and citation process. AK led the writing of the manuscript. All authors contributed critically to the drafts of the manuscript and gave final approval for publication.

Acknowledgements

We are grateful to Michael Quetting for project coordination and to Babak Naimi for contribution to the very early conceptions of the MoveApps idea and its start. SCD acknowledges support from the NASA Ecological Forecasting Program Grant 80NSSC21K1182. Thanks to three anonymous reviewers for comments on an earlier version of this manuscript.

References

- Ardito L, Coppola R, Malnati G, Torchiano M (2020) Effectiveness of Kotlin vs. Java in android app development tasks. Information and Software Technology 127:106374.
<https://doi.org/10.1016/j.infsof.2020.106374>
- Bernstein D (2014) Containers and Cloud: From LXC to Docker to Kubernetes. IEEE Cloud Computing 1:81–84. <https://doi.org/doi:10.1109/MCC.2014.51>

630 Boettiger C (2015) An introduction to Docker for reproducible research, with examples from
631 the R environment. SIGOPS Oper Syst Rev 49:71–79.
632 <https://doi.org/10.1145/2723872.2723882>

633 Calabrese JM, Fleming CH, Gurarie E (2016) ctm: an r package for analyzing animal
634 relocation data as a continuous-time stochastic process. Methods in Ecology and Evolution
635 7:1124–1132. <https://doi.org/10.1111/2041-210X.12559>

636 Calabrese JM, Fleming CH, Noonan MJ, Dong X (2021) ctmweb: A Graphical User Interface
637 for Autocorrelation-Informed Home Range Estimation. Wildlife Society Bulletin 45:162–169.
638 <https://doi.org/10.1002/wsb.1154>

639 Campbell HA, Urbano F, Davidson S, et al (2016) A plea for standards in reporting data
640 collected by animal-borne electronic devices. Animal Biotelemetry 4:1.
641 <https://doi.org/10.1186/s40317-015-0096-x>

642 Chauhan VK (2014) Smoke testing. Int J Sci Res Publ 4.1:2250–3153

643 Cito J, Schermann G, Wittern JE, et al (2017) An Empirical Analysis of the Docker Container
644 Ecosystem on GitHub. In: 2017 IEEE/ACM 14th International Conference on Mining Software
645 Repositories (MSR). pp 323–333

646 DataCite-Metadata-Working-Group (2021) DataCite Metadata Schema Documentation for
647 the Publication and Citation of Research Data and Other Research Outputs v4.4. 82 pages.
648 <https://doi.org/10.14454/3W3Z-SA82>

649 Davidson SC, Bohrer G, Gurarie E, et al (2020) Ecological insights from three decades of
650 animal movement tracking across a changing Arctic. Science 370:712–715.
651 <https://doi.org/10.1126/science.abb7080>

652 Dodge S, Toka M, Bae CJ (2021) DynamoVis 1.0: an exploratory data visualization software
653 for mapping movement in relation to internal and external factors. *Movement Ecology* 9:55.
654 <https://doi.org/10.1186/s40462-021-00291-5>

655 Farley SS, Dawson A, Goring SJ, Williams JW (2018) Situating Ecology as a Big-Data Science:
656 Current Advances, Challenges, and Solutions. *BioScience* 68:563–576.
657 <https://doi.org/10.1093/biosci/biy068>

658 Fidler F, Chee YE, Wintle BC, et al (2017) Metaresearch for Evaluating Reproducibility in
659 Ecology and Evolution. *BioScience* 67:282–289. <https://doi.org/10.1093/biosci/biw159>

660 Franzoni C, Sauermann H (2014) Crowd science: The organization of scientific research in
661 open collaborative projects. *Research Policy* 43:1–20.
662 <https://doi.org/10.1016/j.respol.2013.07.005>

663 Gewin V (2016) Data sharing: An open mind on open data. *Nature* 529:117–119.
664 <https://doi.org/10.1038/nj7584-117a>

665 Gupte PR, Beardsworth CE, Spiegel O, et al (2022) A guide to pre-processing high-throughput
666 animal tracking data. *Journal of Animal Ecology*. <https://doi.org/10.1111/1365-2656.13610>

667 Holyoak M, Casagrandi R, Nathan R, et al (2008) Trends and missing parts in the study of
668 movement ecology. *PNAS* 105:19060–19065. <https://doi.org/10.1073/pnas.0800483105>

669 Joo R, Boone ME, Clay TA, et al (2020a) Navigating through the r packages for movement.
670 *Journal of Animal Ecology* 89:248–267. <https://doi.org/10.1111/1365-2656.13116>

671 Joo R, Picardi S, Boone ME, et al (2020b) A decade of movement ecology. *arXiv:200600110*
672 [q-bio]

673 Kays R, Crofoot MC, Jetz W, Wikelski M (2015) Terrestrial animal tracking as an eye on life
674 and planet. *Science* 348:aaa2478. <https://doi.org/10.1126/science.aaa2478>

675 Kays R, Davidson SC, Berger M, et al (2022) The Movebank system for studying global animal
676 movement and demography. *Methods in Ecology and Evolution* 13:419–431.
677 <https://doi.org/10.1111/2041-210X.13767>

678 Kearse M, Moir R, Wilson A, et al (2012) Geneious Basic: An integrated and extendable
679 desktop software platform for the organization and analysis of sequence data.
680 *Bioinformatics* 28:1647–1649. <https://doi.org/10.1093/bioinformatics/bts199>

681 Kölzsch A, Hirt J, Safi K (2021) Migration Mapper. Movebank Data Repository MoveApps
682 Workflow. <https://doi.org/10.5441/001/1.7tq16jr8>

683 Kölzsch A, Kruckenberg H, Glazov P, et al (2016a) Data from: Towards a new understanding
684 of migration timing: slower spring than autumn migration in geese reflects different decision
685 rules for stopover use and departure. Movebank Data Repository.
686 <https://doi.org/10.5441/001/1.31c2v92f>

687 Kölzsch A, Müskens GJDM, Kruckenberg H, et al (2016b) Towards a new understanding of
688 migration timing: slower spring than autumn migration in geese reflects different decision
689 rules for stopover use and departure. *Oikos* 125:1496–1507.
690 <https://doi.org/10.1111/oik.03121>

691 Kölzsch A, Wikelski M (2021) Morning Report. Movebank Data Repository MoveApps
692 Workflow. <https://doi.org/10.5441/001/1.h4c0p8bv>

Kranstauber B, Cameron A, Weinzerl R, et al (2011) The Movebank data model for animal tracking. *Environmental Modelling & Software* 26:834–835.
<https://doi.org/10.1016/j.envsoft.2010.12.005>

Kranstauber B, Smolla M, Scharf AK (2020) move: Visualizing and Analyzing Animal Tracking Data. Version 4.0.4

Lai J, Lortie CJ, Muenchen RA, et al (2019) Evaluating the popularity of R in ecology. *Ecosphere* 10:e02567. <https://doi.org/10.1002/ecs2.2567>

Lin D, Crabtree J, Dillo I, et al (2020) The TRUST Principles for digital repositories. *Sci Data* 7:144. <https://doi.org/10.1038/s41597-020-0486-7>

Lowndes JSS, Best BD, Scarborough C, et al (2017) Our path to better science in less time using open data science tools. *Nat Ecol Evol* 1:1–7. <https://doi.org/10.1038/s41559-017-0160>

Mislan KAS, Heer JM, White EP (2016) Elevating The Status of Code in Ecology. *Trends in Ecology & Evolution* 31:4–7. <https://doi.org/10.1016/j.tree.2015.11.006>

Nosek BA, Alter G, Banks GC, et al (2015) Promoting an open research culture. *Science* 348:1422–1425. <https://doi.org/10.1126/science.aab2374>

Peng RD (2011) Reproducible Research in Computational Science. *Science* 334:1226–1227. <https://doi.org/10.1126/science.1213847>

Perez A, Moltó G, Caballer M, Calatrava A (2018) Serverless computing for container-based architectures. *Future Generation Computer Systems* 83:50–59.
<https://doi.org/10.1016/j.future.2018.01.022>

714 Powers SM, Hampton SE (2019) Open science, reproducibility, and transparency in ecology.
 715 Ecological Applications 29:e01822. <https://doi.org/10.1002/eap.1822>
 716 R-Core-Team (2021) R: A language and environment for statistical computing. R Foundation
 717 for Statistical Computing, Vienna, Austria
 718 Reichman OJ, Jones MB, Schildhauer MP (2011) Challenges and Opportunities of Open Data
 719 in Ecology. Science 331:703–705. <https://doi.org/10.1126/science.1197962>
 720 Resheff YS, Rotics S, Harel R, et al (2014) AcceleRater: a web application for supervised
 721 learning of behavioral modes from acceleration measurements. Movement Ecology 2:27.
 722 <https://doi.org/10.1186/s40462-014-0027-0>
 723 Schneider G, Kölzsch A, Safi K (2021) MoveApps - Etablierung eines Dienstes zur Entwicklung,
 724 Veröffentlichung und langfristigen Nachnutzung fachspezifischer Forschungssoftware
 725 Sequeira AMM, O'Toole M, Keates TR, et al (2021) A standardisation framework for bio-
 726 logging data to advance ecological research and conservation. Methods in Ecology and
 727 Evolution n/a: <https://doi.org/10.1111/2041-210X.13593>
 728 Slingsby A, van Loon E (2016) Exploratory Visual Analysis for Animal Movement Ecology.
 729 Computer Graphics Forum 35:471–480. <https://doi.org/10.1111/cgf.12923>
 730 Talia D (2013) Clouds for Scalable Big Data Analytics. Computer 46:98–101.
 731 <https://doi.org/10.1109/MC.2013.162>
 732 Urbano F, Cagnacci F, Calenge C, et al (2010) Wildlife tracking data management: a new
 733 vision. Philosophical Transactions of the Royal Society B: Biological Sciences 365:2177–2185.
 734 <https://doi.org/10.1098/rstb.2010.0081>

Wickham H (2011) testthat: Get Started with Testing. The R Journal 3:5.
<https://doi.org/10.32614/RJ-2011-002>

Wilkinson MD, Dumontier M, Aalbersberg IJ, et al (2016) The FAIR Guiding Principles for scientific data management and stewardship. Sci Data 3:160018.
<https://doi.org/10.1038/sdata.2016.18>

Williams HJ, Taylor LA, Benhamou S, et al (2020) Optimizing the use of biologgers for movement ecology research. Journal of Animal Ecology 89:186–206.
<https://doi.org/10.1111/1365-2656.13094>

Wilmers CC, Nickel B, Bryce CM, et al (2015) The golden age of bio-logging: how animal-borne sensors are advancing the frontiers of ecology. Ecology 96:1741–1753.
<https://doi.org/10.1890/14-1401.1>

Figure captions

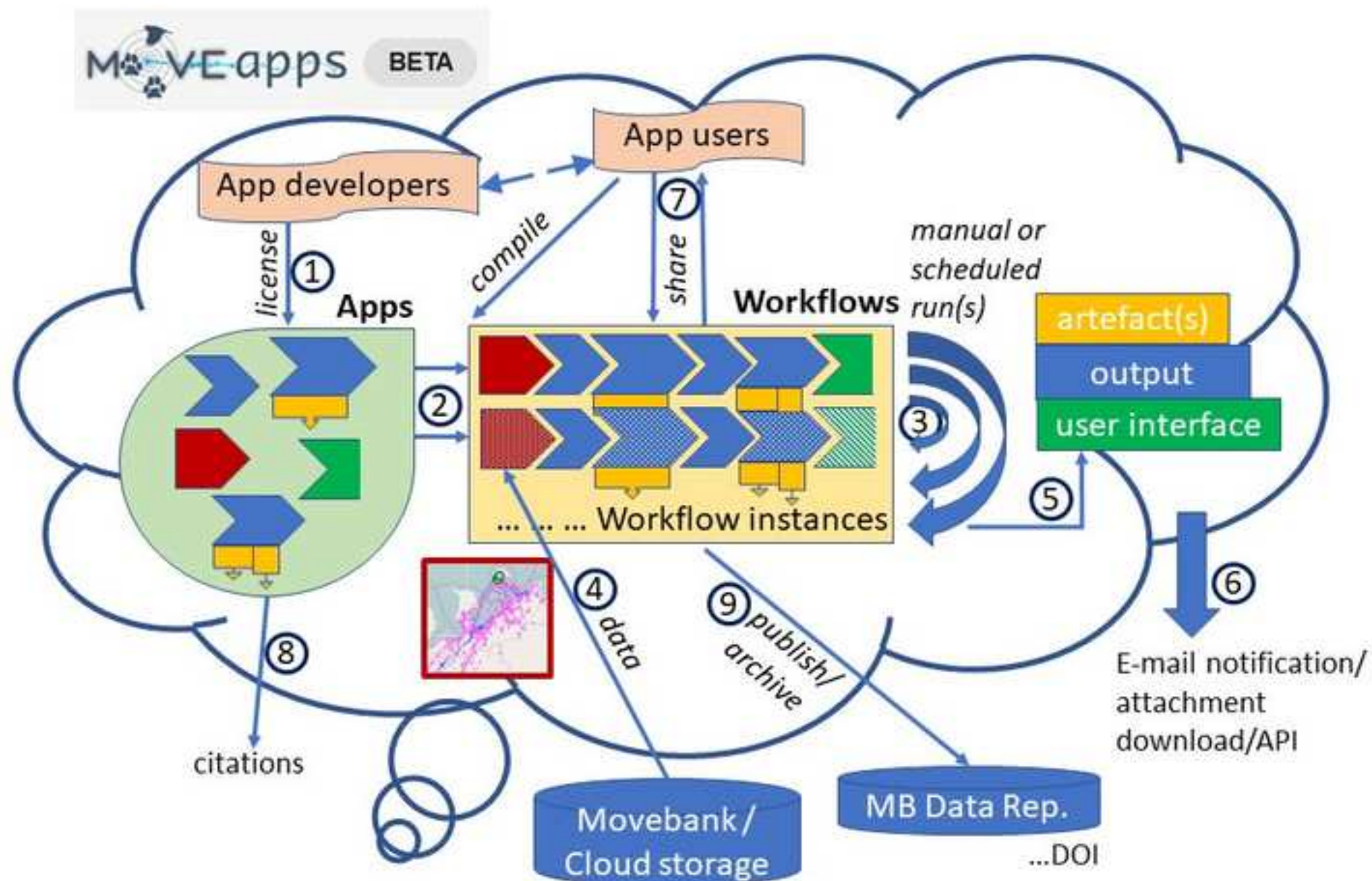
Figure 1. Schematic representation of the "cloud" computing MoveApps platform (beta version). (1) App developers provide Apps with defined input and output format under an open license. Apps can upload data (red), process data (blue), show results in an interactive user interface (green) and create artefacts for download (yellow). (2) App users can combine those Apps to specific workflows to analyse their movement data. Workflows can consist of several workflow instances that can be (3) run manually or scheduled to analyse (4) tracking data. (5) The calculated results can be explored in a user interface or (6) downloaded as output and artefact files directly or via API. Notification E-mails can be sent of finished scheduled runs. (7) Workflows can be shared in the platform. (8) Citations for Apps are

provided and (9) workflows can be published with a digital object identifier (DOI) and archived in the Movebank Data Repository. Registered MoveApps users can be App developers or App users (compiling workflows) or both.

Figure 2. Example workflow "Morning Report". Screenshots of the (a) workflow representation (order and names of combined Apps) and (b) workflow user interface output for an example dataset of greater white-fronted goose (*Anser a. albifrons*) tracks. Note that only tracks with data during 2014 were explored with the selected settings.

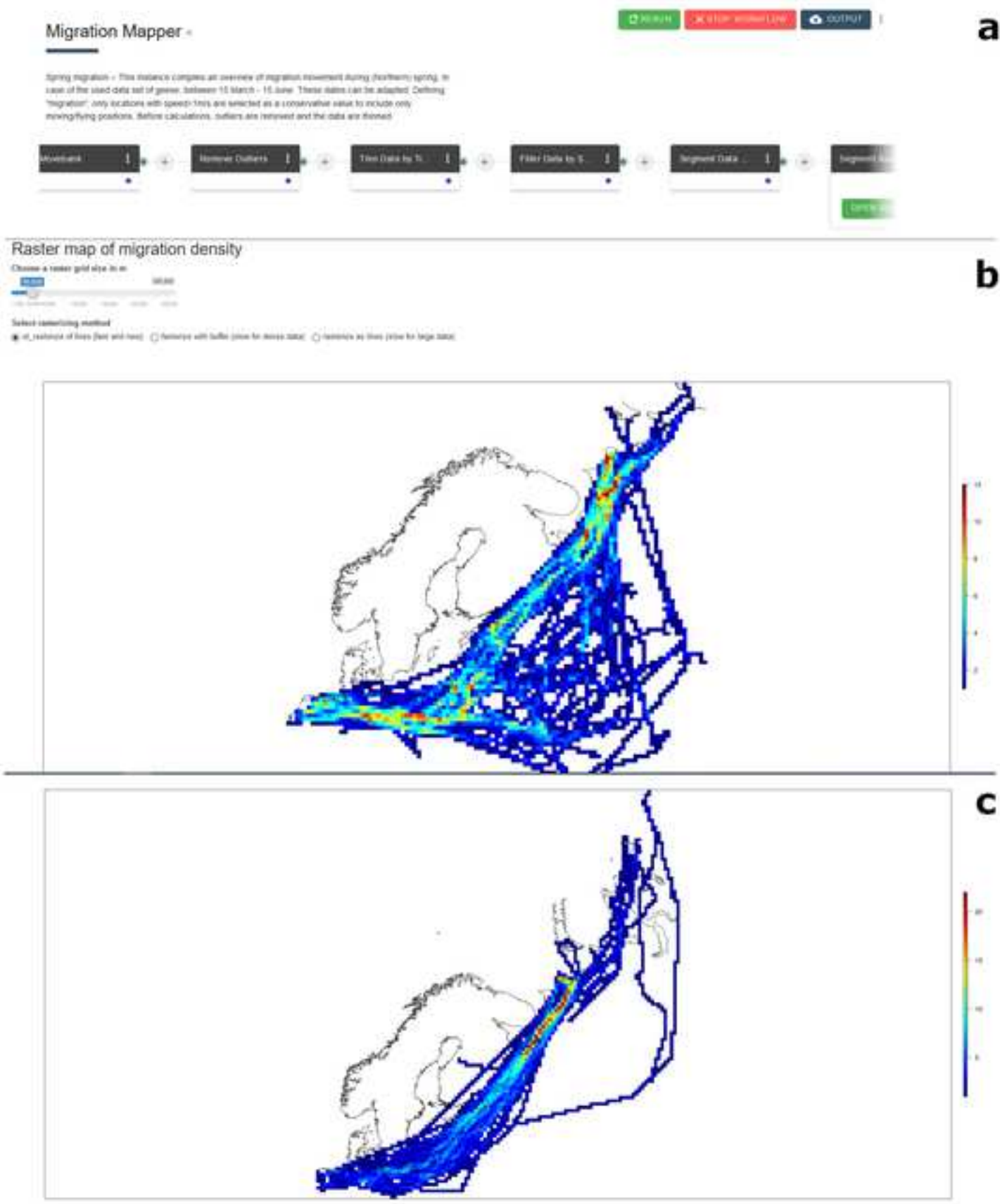
Figure 3. Example workflow "Migration Mapper". Screenshots of the (a) representation (order and names of combined Apps) of the workflow instance "Spring migration", (b) workflow user interface output for Spring migration and (c) Autumn migration of an example dataset of greater white-fronted goose (*Anser a. albifrons*) tracks. Note that tracks of all years are combined.

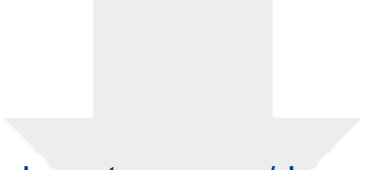
Figure 1



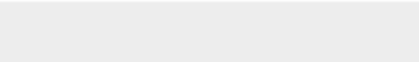
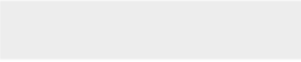
[Click here to access/download;Figure;Fig2_MorningReport.jpg](#) 

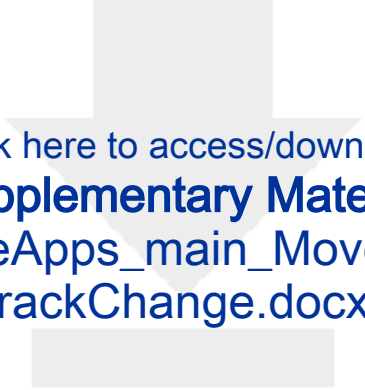






[Click here to access/download](#)
Supplementary Material
Reply letter to review 2.docx





[Click here to access/download](#)

Supplementary Material

KoelzschEtAl_MoveApps_main_MoveEcol_Resub2_F_T
rackChange.docx

