Guidance framework to apply good practices in ecological data analysis: Lessons learned from building Galaxy-Ecology

Royaux Coline[1,2*], Mihoub Jean-Baptiste[3], Jossé Marie[4], Pelletier Dominique[5], Norvez Olivier[6], Reecht Yves[7], Fouilloux Anne[8], Rasche Helena[9], Hiltemann Saskia[10], Batut Bérénice[11,12], Eléaume Marc[13,14], Seguineau Pauline[13,14], Massé Guillaume[15], Amossé Alan[16], Bissery Claire[17,18], Lorrilliere Romain[3], Martin Alexis[19], Bas Yves[3,20], Virgoulay Thimothée[21,22], Chambon Valentin[16], Arnaud Elie[2], Michon Elisa[23], Urfer Clara[2,24], Trigodet Eloïse[21,24], Delannoy Marie[25], Loïs Gregoire[3], Julliard Romain[3], Grüning Björn[26], The Galaxy-E community, Le Bras Yvan[2]

[1] UMR8067 Biologie des Organismes et Ecosystèmes Aquatiques (BOREA, MNHN-CNRS-SU-IRD-UCN-UA), Sorbonne Université, Station Marine de Concarneau - Concarneau, France

[2] Pôle national de données de biodiversité, UAR2006 PatriNat (OFB-MNHN-CNRS-IRD), Muséum National d'Histoire Naturelle, Station Marine de Concarneau - Concarneau, France

[3] Centre d'Écologie et des Sciences de la Conservation (UMR7204 CESCO, MNHN-CNRS-SU), Muséum National d'Histoire Naturelle, Sorbonne Université, Centre National de la Recherche Scientifique - Paris, France

[4] Data Terra, Centre National de la Recherche Scientifique - Brest, France

[5] UMR DECOD (Ifremer-Agrocampus Ouest-INRAE) - Lorient, France

[6] Pôle National de Données de Biodiversité, UAR2006 PatriNat (OFB-MNHN-CNRS-IRD), Fondation pour la Recherche sur la Biodiversité, Muséum national d'Histoire naturelle - Paris, France

[7] Institute of Marine Research - Bergen, Norway

[8] Simula Research Laboratory - Oslo, Norway

[9] Clinical Bioinformatics Group, Department of Pathology, Erasmus Medical Center - Rotterdam, The Netherlands

[10] Institute of Pharmaceutical Sciences, Faculty of Chemistry and Pharmacy, University of Freiburg - Freiburg, Germany

[11] Institut Français de Bioinformatique, CNRS UAR3601 - Évry, France

[12] Mésocentre, Clermont-Auvergne, Université Clermont Auvergne - Clermont-Ferrand, France

[13] Institut de Systématique Evolution, Biodiversité (UMR7205 ISYEB, MNHN-CNRS-SU-EPHE), Département Origines et Évolution, Muséum national d'Histoire naturelle - Paris, France

[14] Institut de Systématique Evolution, Biodiversité (UMR7205 ISYEB, MNHN-CNRS-SU-EPHE), Département Origines et Évolution, Station Marine de Concarneau - Concarneau, France

[15] UMR LOCEAN (CNRS-SU-IRD-MNHN), Centre National de la Recherche Scientifique, Station Marine de Concarneau - Concarneau, France

[16] Muséum National d'Histoire Naturelle, Station Marine de Concarneau - Concarneau, France

[17] Institut français de recherche pour l'exploitation de la mer (Ifremer) – Brest, France

[18] Université Claude Bernard Lyon 1 - Lyon, France

[19] UMR8067 Biologie des Organismes et Ecosystèmes Aquatiques (BOREA, MNHN-CNRS-SU-IRD-UCN-UA), Muséum national d'Histoire naturelle - Paris, France

[20] UAR2006 PatriNat (OFB-MNHN-CNRS-IRD), Muséum national d'Histoire naturelle - Paris, France

[21] Centre d'Écologie et des Sciences de la Conservation (UMR7204 CESCO, MNHN-CNRS-SU), Muséum National d'Histoire Naturelle - Concarneau, France

[22] Université de Montpellier - Montpellier, France

[23] Institut des Sciences de la Mer de Rimouski, Université du Québec à Rimouski - Rimouski, Québec, Canada

[24] Université de Bretagne Occidentale - Brest, France

[25] Fondation pour la Nature et l'Homme - Boulogne-Billancourt, France

[26] Bioinformatics Group, Department of Computer Science, Albert-Ludwigs-University Freiburg - Freiburg, Germany

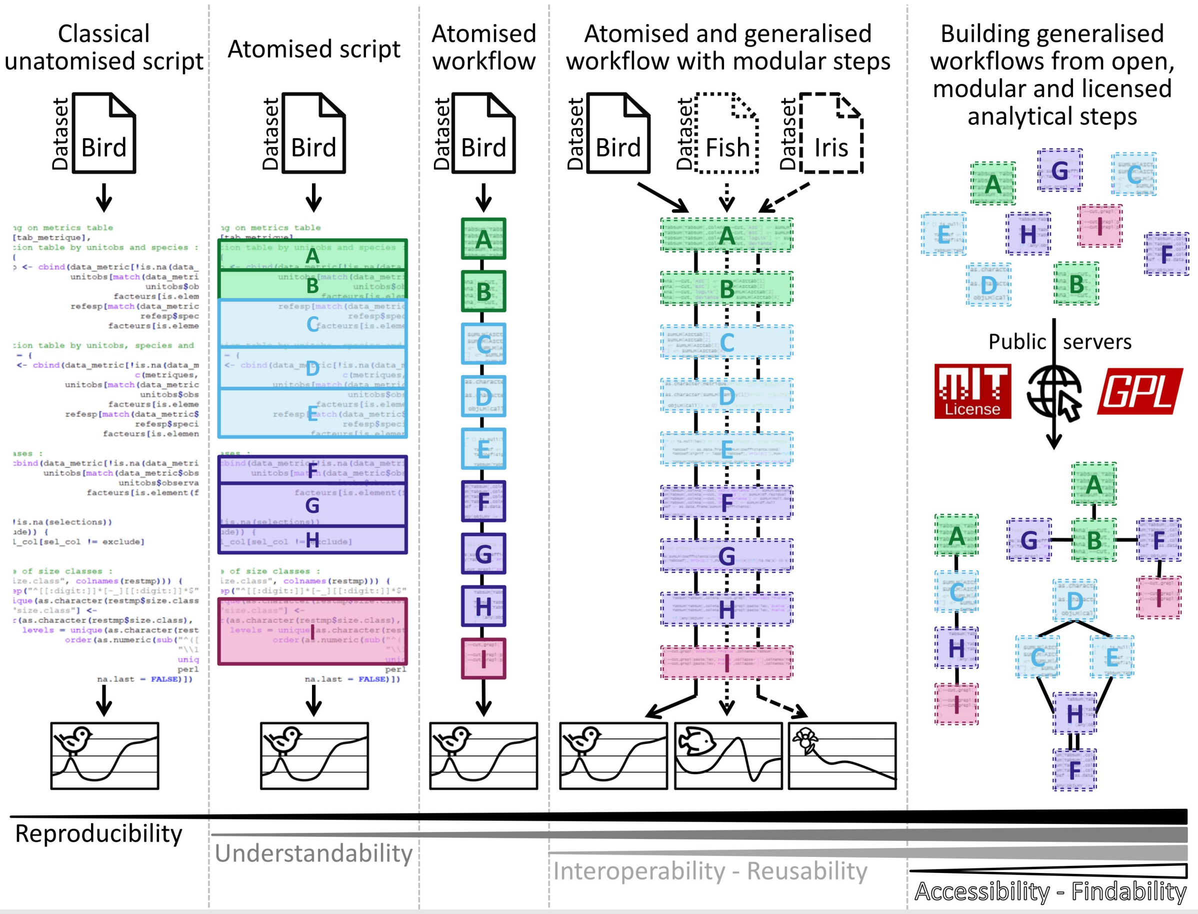
*Corresponding author

Correspondence: coline.royaux@mnhn.fr

ABSTRACT

Numerous conceptual frameworks exist for good practices in research data and analysis (*e.g.* Open Science and FAIR principles). In practice, there is a need for further progress to improve transparency, reproducibility, and confidence in ecology. Here, we propose a practical and operational framework to achieve good practices for building analytical procedures based on atomisation and generalisation. We introduce the concept of atomisation to identify analytical steps which support generalisation by allowing us to go beyond single analyses. These guidelines were established during the development of the Galaxy-Ecology initiative, a web platform dedicated to data analysis in ecology. Galaxy-Ecology allows us to demonstrate a way to reach higher levels of reproducibility in ecological sciences by increasing the accessibility and reusability of analytical workflows once atomised and generalised.

Graphical abstract – Levels of attainable good practices through the atomisation – generalisation framework


Classical unatomised script
Atomised script
Atomised workflow
Atomised and generalised workflow with modular steps
Building generalised workflows from open, modular and licensed analytical steps
Dataset
Bird
Fish
Iris
A
B
C
D
E
F
G
H
I
Public servers
MIT License
GPL
Reproducibility
Understandability
Interoperability - Reusability
Accessibility - Findability

Keywords: Biodiversity; Reproducible analyses; Galaxy; Good practices; Atomisation; Generalisation; Workflows; Ecoinformatics; Conda; Container; Common Workflow Language; RO-CRATE

Introduction

Ecology's Reproducibility Crisis

Research in ecology is increasingly shaped by the availability of novel analytical solutions and statistical tools. Given the ever-growing amount of data available, much attention is often given to the thought process behind statistical analyses to handle different data distributions, pseudo-replication, and sampling biases for instance (*NERC* 2010, 2012; Hampton *et al.*, 2017; Emery *et al.*, 2021). Despite the high-quality standards required by the scientific community from data access to analysis, the level of complexity of ecological systems makes results difficult to reproduce. The ongoing "reproducibility crisis" has also led researchers to pay closer attention to the quality of analyses to increase confidence in their studies and conclusions (Ioannidis, 2022; Fanelli, 2018).

Reproducibility (*i.e.* different teams and experimental setups obtaining similar results; Plesser, 2018) is one of the main criteria for evaluating robust science and reliable conclusions. In ecological sciences, most in-situ observations are not strictly reproducible due to stochasticity. Accordingly, the focus has been directed towards the reproducibility of analyses ("computational reproducibility") over the reproducibility of data collection (Powers & Hampton, 2019; Samota & Davey, 2021). Reproducibility can be achieved at different levels of the analytical workflow, from primary data access to results. Archmiller *et al.*, 2020 and Minocher *et al.*, 2021 tried to evaluate computational reproducibility in 74 studies in wildlife science and 560 studies in biological and behavioural sciences. Although these authors found high rates of computational reproducibility when data and analytical procedures could be fully retrieved, they encountered significant difficulty in retrieving the data files and analytical procedures in most studies.

Given the high complexity and the massive amount of information required to retrieve results using a broad range of data and methods, achieving sufficient reproducibility must be facilitated. In addition, researchers are increasingly challenged to stay up-to-date with the ever-growing number of advanced methods and technologies for data acquisition, storage, and analysis (Hampton *et al.*, 2017). Providing technical and practical support to reduce the perceived complexity of analytical workflows could increase and accelerate the diffusion of good practices in the research community, fostering understanding for a wider audience thereby facilitating transparency and improving reproducibility. Here, we explore how computational reproducibility can be easily implemented in ecological sciences using simple and practical guidelines.

In the current context of the global biodiversity crisis, the scientific community needs to use all available data and provide as robust as possible evidence regarding the state and dynamic of ecological systems, from genetic to ecosystem. At the same time, using analytical tools to provide robust evidence can be complex and may require advanced skills that are not widely available across the scientific community. Therefore, operational solutions and methodological guidelines can allow the analytical workflow to be more accessible without degrading the scientific quality of the analysis, and thus, promote efficient and broad deployment of good practices.

Is the ecology community failing to meet good practices?

The first step towards reproducibility is knowing current good practices and recommendations. Among them, the FAIR principles (Wilkinson *et al.*, 2016), for which the availability of the data and the code used for each published result is an essential criterion, may be key for appropriate management through the data life cycle (Michener, 2015). The FAIR principles (see also CARE principles by Carroll *et al.*, 2020) are considered as a founding framework to share data along four important elements: "Findable" for humans and machines; "Accessible" with a detailed access procedure; "Interoperable" for interaction with other data or applications; "Reusable" in an identical or different context.

In 2022, Gomes and collaborators identified 12 barriers to data and code sharing, ranging from unclarity of processes to fear of inappropriate use and insecurities around data and code quality (Gomes *et al.*, 2022). Although data accessibility has been substantially improved in ecology during the past decade, sharing analytical scripts and codes remain largely marginal (Ivimey-Cook *et al.*, 2023). According to Culina *et al.*, 2020, in a “random sample of 346 nonmolecular articles published between 2015 and 2019”, 79% had data availability but only 27% had code availability despite a tendency for journals to encourage code-sharing (75% of assessed ecological journals).

Low code availability compared to data availability may suggest a lack of technical solutions for sharing computing codes. Nevertheless, many repositories dedicated to sharing code exist, such as GitHub (https://github.org), which software developers widely use to collaborate and share codes publicly and privately. Besides, the Software Heritage initiative automatically archives all openly available code from GitHub, ensuring long-term preservation (https://archiveprogram.github.com; Di Cosmo & Zacchiroli, 2017). Alternatively, other solutions for data archiving may be used, even if not explicitly focused on code sharing (*e.g.*, Zenodo, national public repositories; see also TRUST principles for data repositories, Lin *et al.*, 2020).

However, even if long-term public archiving of code is necessary to achieve good computational reproducibility, it is insufficient. Therefore, many guidelines and principles have been developed in the recent years. Among others, the utilisation of computational workflows has been suggested as a solution for improving computational reproducibility (Cohen-Boulakia *et al.*, 2017; Grüning *et al.*, 2018) through software such as Snakemake (Köster & Rahmann, 2012), Nextflow (Di Tommaso *et al.*, 2017), or Galaxy (The Galaxy Community, 2022). A workflow is generally defined as a sequence of distinct computational tasks for a particular objective (Goble *et al.*, 2020). As such, a workflow represents the backbone of a single specific analysis. Throughout the analytical procedure, a typical workflow starts with raw data, which can be extracted from several databases or data files and processed through a series of analytical steps. The products resulting from these analytical steps (*i.e.* the outputs of the computational workflow) can be data files, graphic representations and any associated metrics. In this respect, computer code can also be considered as research data (Borgman, 2020).

When properly designed, a certain level of reproducibility can be easily achieved since workflow languages naturally capture the following four key elements (Cohen-Boulakia *et al.*, 2017):

- the specificities of the workflow, the analysis steps and associated tools;
- the workflow entries, datasets and parameters;
- the environment and context of the use of the workflow;
- the results obtained and the outputs of the workflow.

In the original publication of Wilkinson *et al.* (2016), the focus of FAIR principles was mainly on data. However, the principles can be applied to software and computational workflows (Lamprecht *et al.*, 2019; Goble *et al.*, 2020). For instance, a code shared as supplementary material of a non-open access publication could be considered as "Interoperable" but is not easily "Findable", "Accessible", or "Reusable". In contrast, a large block of code consisting of several hundred lines, from data pre-processing to final results and graphics, may require efforts to understand and adapt to other kinds of data ("non-reusable"), mainly if annotations or comments are limited. Similarly, an analytical procedure shared without indicating the versions of hardware, software, and packages has a low chance of producing identical outputs, making it non-reproducible. These issues may harm the scientific community by preventing fully transparent communication among users about knowledge production and practice comparison. They can also be detrimental to individual authors, when they need to update or run new analyses.

Impact on Ecology Research

The efficiency of the expertise and research is greatly affected by the lack of computational reproducibility and FAIRness of analytical procedures. FAIR research data was estimated to save 10.2 billion € per year in Europe (Munafò *et al.*, 2017; European commission, 2019; Gomes *et al.*, 2022). Indeed, analyses and underlying conclusions cannot have a tangible impact if the raw data, the analytical procedures, and the outputs resulting from these procedures are not easily findable, accessible, interoperable and reusable. Moreover, consistent application of reproducibility and FAIR principles will improve trust in research studies and scientific reports (Powers & Hampton, 2019; Lortie, 2021; Jenkins *et al.*, 2023).

The widespread use of computational languages to process large-scale data and analyse complex systems has been a major advance in studying the ecosphere at any spatio-temporal scale (Michener & Jones, 2012; Farley *et al.*, 2018). Even if computational capacity may represent a significant limitation for analysing large data files or using resource-intensive algorithms (Green & Figuerola, 2005), computation clusters nowadays exist to overcome such challenges (Hampton *et al.*, 2017; Larcombe *et al.*, 2017). However, the ever-growing technical and programming skills required to take advantage of such computational solutions by the scientific community raise new challenges.

The use of increasingly complex analytical solutions, paired with different approaches or programming languages, mechanically reduces the number of potential users, limiting collaboration and fragilising fundamental pillars of scientific knowledge such as the peer-review process and critical evaluation. As a response to this situation, adequate training was identified by life science researchers (*Community Survey Report*, 2013; Williams & Teal, 2017; Larcombe *et al.*, 2017), as it would help involve more people in the understanding of current analytical solutions and benefit to scientific cooperation (Touchon & McCoy, 2016; Gownaris *et al.*, 2022). Research is typically structured through a highly competitive organisation, with a potentially detrimental effect on scientific knowledge (Fang & Casadevall, 2015). Instead, fostering collaboration and collective intelligence by promoting transparent sharing of analytical procedures, would offer more persitent and robust ways to achieve actionable science (Ellemers, 2021). Such efforts would be of paramount importance in environmental sciences and the conservation of biodiversity by providing governance and guiding actions with increasingly robust evidence (Keenan *et al.*, 2012).

Are there simple and ready-to-use solutions?

In this note, we aim to promote the reuse of existing concepts and solutions as pillars toward better practices for ecological analyses by providing a streamlined framework. We believe the framework presented in the second part of this note represents an operational and actionable path for researchers and experts to attain levels of good practices (*e.g.* reproducibility, FAIR, open science, R compendium; Casajus N., 2023) with no more investment than they are able or willing to provide (Field *et al.*, 2014).

This framework has been formalised while building the Galaxy-Ecology (Galaxy-E) initiative (see section III). Galaxy (The Galaxy Community, 2022) is a workflow-oriented web platform for sharing and processing research data. It allows sharing, developing, and using various datasets and data processing tools (*e.g.* data formatting, statistical tests, graphic representations). Many scientific workflow management systems, such as Snakemake and Nextflow, operate from the command line. In ecology, numerous initiatives have tried to introduce such systems, starting with more user-friendly solutions. For example, the KNIME and Kepler systems with the CoESRA initiative (Collaborative Environment for Scholarly Research and Analysis) in Australia, or Taverna with the BioVeL initiative (Biodiversity Virtual e-Laboratory) in Europe. These systems are more accessible to new users by offering a graphical interface while achieving high specificity (Berthold *et al.*, 2007; Hardisty *et al.*, 2016). However, good computer programming or scientific workflow management knowledge is still necessary to use these applications correctly.

Galaxy is ready to use and has proved its efficiency and suitability in other research fields, including genomics and climate science (Knijn *et al.* 2020; Serrano-Solano *et al.*, 2022). From a user's point of view, it offers extensive computing power and a graphical interface to use analysis workflows, even without experience in software development. Web-based access allows easy sharing of analytical workflows between collaborators and with a broader audience. Galaxy supports tools in almost any computational language, including R and Python, two of the most used languages in ecology, with many packages dedicated to ecological and biodiversity-oriented analyses incorporated (Lai *et al.*, 2019).

Galaxy enables good reproducibility for data exploration and analyses, helps compute intricate analyses on big data files, enables collaboration, and can support the teaching process. Galaxy-E is a Galaxy server dedicated to ecological analyses maintained by the European Galaxy team (supported by the German Federal Ministry of Education and Research and the German Network for Bioinformatics Infrastructure), and is available at https://ecology.usegalaxy.eu.

Galaxy-E is a demonstration platform for applying good practices such as the FAIR principles and computational reproducibility for analytical procedures in ecology. Hence, this technical note is partly Galaxy-oriented, not to present the platform as a prescriptive solution but to give an operational example of the good practices it helps to achieve. Recommendations described in this note regarding the construction of an analytical procedure on Galaxy are meant to be transposable to local code development or another consistent workflow engine.

Framework towards good practices

Atomisation: what is it and why?

Atomisation is dividing an analytical procedure into several specific steps ("atoms") generating a suite of elementary analytical steps. Breaking down the analytical process into atoms functioning as building blocks allows for better understanding, modularity, and visibility of the analytical flow. It permits making it more accessible to a broader audience or facilitating the peer-review process. Indeed, an extended one-block code that imports raw data, makes pre-processing steps (*e.g.* filter, formatting), conducts analyses (*e.g.* distribution study, modelling), and performs final representations of results (*e.g.* maps, plots) can be challenging to understand and reuse by others or even the same person after some time.

McIntire *et al.* (2022) described the PERFICT approach (Prediction, Evaluation, Reusability, Free access, Interoperability, Continuous workflows, and routine Tests) to set a new foundation for models in predictive ecology. This can be applied more generally to the analytical procedure in ecology and biodiversity. In their article, McIntire and collaborators make an analogy between code development and Lego® construction, similar to our definition of atomisation. Functions are a workflow's most fundamental analytical steps and can be seen as modular pieces, alike single pieces of Lego®. Modules can be created from a single or series of successive functions comparably as in Lego® structures made of several pieces (*e.g.* meant to build cars, houses, or road). These modules (or atoms, tools) can be used as standalone or combined to make simple to complex analytical workflows such as data formatting or curation, running statistical models, or generating graphical elements for visualisation. Doing so, the atomisation approach may facilitate sharing or teaching analytical practices since beginners can easily understand the general organisation of the analytical procedure by simply reading the list of steps in the analysis with a limited degree of complexity. Decoupling programming skills from analytical skills can make data processing more accessible to a wider audience. Indeed, once each elementary step is clearly identified and delimited along the atomisation process, it is easier to grasp the whole analytical procedure and focus on the review of each step at a time or (re)use it. New workflows can further be generated by recombining existing, validated or peer-reviewed elementary steps in innovative ways. This process can save time, increase confidence, and avoid potential programming mistakes, allowing greater focus on understanding the analytical workflow.

Generalisation: what is it and why?

Generalisation is the modification of an analytical procedure to make it applicable to many settings, by removing specificities related to a particular data file or data format. Generalisation aims to optimise the reusability at different times (*e.g.* regular result update), enlarge the application of a given analysis to different input data files while keeping the initial analytical procedure fully reproducible. Generalising an analytical step requires identifying key steps and invariant parameters from those that must be adaptable to allow for the analysis to be applied to specific characteristics of various datasets. These parameters must be implemented to be easily modified if needed. Generalisation can be tricky because the higher the flexibility of an analytical step, the greater the risk of errors in its use. This is why generalisation should be complemented by clear statement and an implementation of red flags and warnings to prevent such events. As with atomisation, generalisation is primarily a conceptual way to build analytical procedures. It requires minor change of practices to reach certain degree of generalisation, avoiding additional effort later on for reusability, reproducibility, and share.

Atomisation and generalisation are related and complementary concepts. Atomisation into adequate elementary steps is necessary to properly generalise an analytical procedure as it permits to enhance the modularity of the procedure and its capacity to be tailored to different data types. Atomisation and generalisation must be applied from the earliest stages of the programming development of any analytical procedure in order to achieve:

- Greater transparency, even for beginners, since the relevance and coherence of each step and their successive arrangement along the analytical procedure should be appraised independently of the programming skills;
- Time savings;
- Greater reusability;
- Modularity of the elementary steps, to rearrange them differently if needed.

How to do atomisation and generalisation: Finding balance

Breaking down codes into elementary steps to achieve atomisation is not an intuitive task et first as it may target a single function or a more intricate set of several functions. There could be different degrees of atomisation, depending on the grain required to decompose the analytical process (fig. 1; tab. 1). The application of general guidelines and good practices implies finding a balance between the most appropriate degree of atomisation and generalisation. This depends on the type of analytical procedure or the targeted audience (*e.g.* with different interests and programming skills). Attention to this balance is critical to ensure that the analytical procedures could be reused. For instance, a workflow in which each function would be considered as a unique elementary step would optimise the flexibility but may likely add unnecessary complexity. At the other extreme, considering a whole analytical workflow as an elementary step may make it ready-to-use and simplify its application, but would be too coarse and therefore limit flexibility by violating the principle of atomisation.

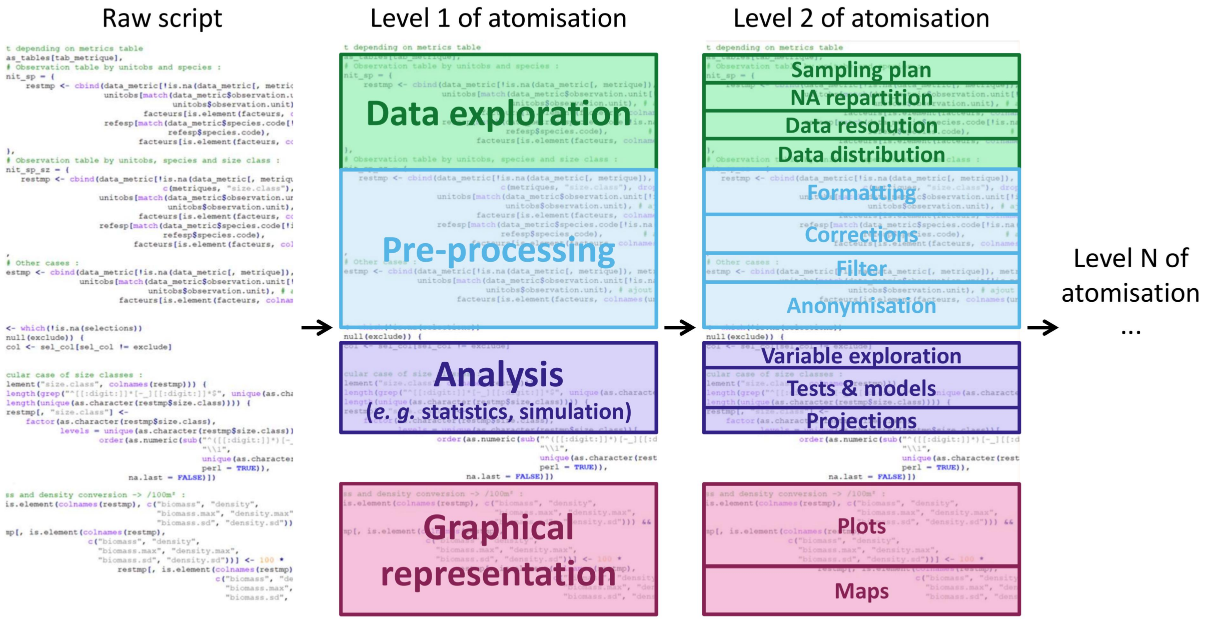


Figure 1 - Illustration of the atomisation of an existing code

Table 1 - Example of atomisation levels

Level 1 - big shape	Level 2	Level 3
Data exploration	Sampling plan	Complete Balanced
	Missing values	Proportion Distribution
	Data granularity	Geographic resolution Temporal resolution Measure resolution
	Data distribution	Geographic coverage Temporal coverage Measures ranges Summaries
	…	…
Pre-processing	Formatting	Change file format Change general format
	Corrections	Remove special characters Remove low trust observations Correct measures
	Filtering	Remove unwanted observations
	Anonymisation	Anonymise names Anonymise localities Anonymise species
	…	…
Analysis	Variable exploration	PCA Collinearity Correlation
	Unimodal tests	Linear Models χ^2 Student
	Statistical models	Generalised Linear Models Generalised Additive Models Random Forest
	Models Evaluation	Evaluation metrics (*e.g.* AIC, Jaccard) Validation methods
	Projections	Geographical projections Temporal projections
	…	…
Representation	Plot	Raw variables Modelled results
	Map	Observations Projections
	…	…

Few changes in code-writing habits can enhance the reusability of the analytical procedure by generating easy-to-understand analytical procedure without investing much time. It is best to develop each elementary step directly in separate code files and to give details of the order in which elementary steps are used for each analytical workflow. To ensure reproducibility and traceability of the results, each computation of the analytical workflow should be associated with the details of the parameters settings and datasets used. From a practical point of view, a couple of recommendations could be made for coding elementary steps in order to facilitate generalisation and ease the reuse. Once each elementary step is defined, we recommend all dependencies (*e.g.* software version, packages, libraries and their versions) to be set at the same place, at the start of the code, followed by modular parameters (*e.g.* input file location and name, column selection, modelling parameters, data specificities, output saving location). When the script of the elementary step is completed, modular parameters should be the only part of the code that may be modified in future reuse. Dependencies and subsequent computational tasks should be left untouched to ensure the integrity of the analysis and then, reproducibility. In the end, it is best to add an open-source license to any analytical procedure shared publicly (*e.g.* MIT, GPL). It permits to clearly state the terms and conditions of diffusion, share and reuse.

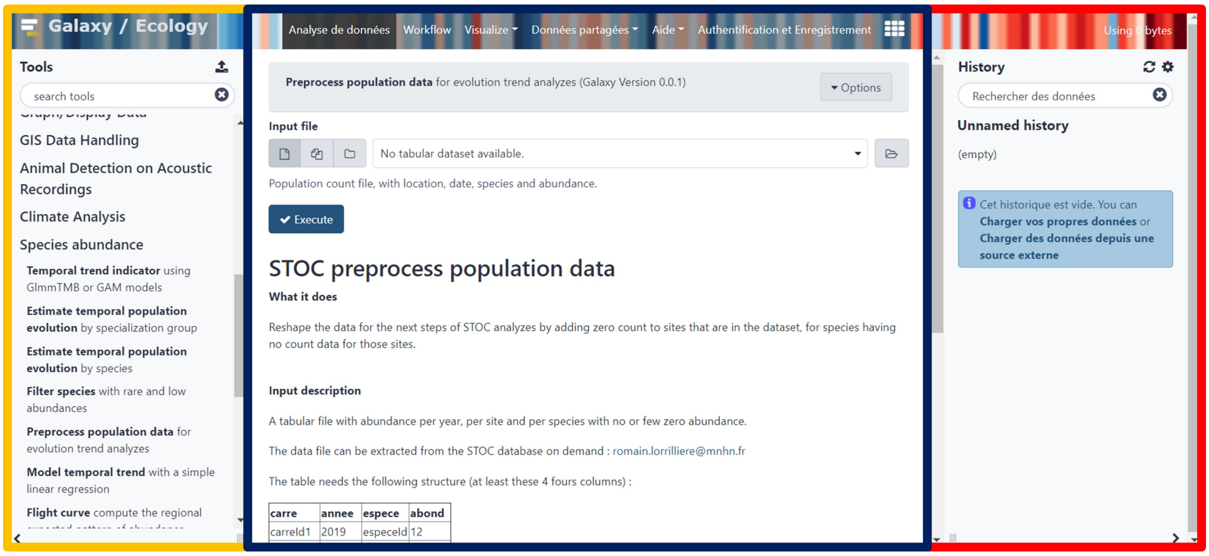
As such, atomisation and generalisation may overcome social or psychological barriers related to transparent sharing, either related to securing ownership (*e.g.* DOI) and to embarrassment or fear during a peer-review process (Gomes *et al.*, 2022).

Entering a new dimension: the Galaxy-E initiative example

Developing open and properly atomised and generalised analytical procedures can already represent a significant step forward in terms of good practice. Galaxy as a demonstration platform to package analyses in an accessible and user-friendly manner can help achieve a further level of FAIRness. Any analytical procedure can be adapted on the platform and Galaxy can be used through the whole data life cycle (https://rdmkit.elixir-europe.org/galaxy_assembly). Throughout this note, many ways to contribute to Galaxy are discussed in their conceptual and methodological aspects. One can use off-the-shelf tools, workflows, and tutorials to design an analytical procedure, or suggest, develop, and share new workflows and tutorials, two aspects that do not require coding skills. Eventually, one can modify or develop entirely new tools with any computational language to make them accessible to all users on any Galaxy server. The Galaxy platform emphasises (i) accessibility of tools and data even without programming experience, (ii) reproducibility through the easy creation and reuse of analysis workflows, (iii) transparency through the open-source distribution of underlying codes; and (iv) community support.

There are different Galaxy servers, at global, continental, and national levels (European and French levels for example), but also according to the fields (*e.g.*, biomedical, ecology, climate). The Galaxy-E initiative is hosted by European (https://ecology.usegalaxy.eu) and French (https://ecology.usegalaxy.fr) servers.

Datasets can be uploaded on a Galaxy server from a local device, an online server, or a database. Users can then access every available tools (fig. 2, left panel) to modify, explore, and analyse their data. All tools used, parameters, and data (inputs and outputs) of the analysis are saved in a private “Galaxy history” (fig. 2, right panel), documenting every step of the analytical procedure and recording the provenance of each output. From any history, the user can extract a workflow (fig. 3) or directly share or publish the history itself.



Galaxy / Ecology

Tools

search tools

GIS Data Handling

Animal Detection on Acoustic Recordings

Climate Analysis

Species abundance

Temporal trend indicator using GlmmTMB or GAM models

Estimate temporal population evolution by specialization group

Estimate temporal population evolution by species

Filter species with rare and low abundances

Preprocess population data for evolution trend analyzes

Model temporal trend with a simple linear regression

Flight curve compute the regional

Analyse de données | Workflow | Visualize | Données partagées | Aide | Authentification et Enregistrement

Preprocess population data for evolution trend analyzes (Galaxy Version 0.0.1)

Options

Input file

No tabular dataset available.

Population count file, with location, date, species and abundance.

✓ Execute

STOC preprocess population data

What it does

Reshape the data for the next steps of STOC analyzes by adding zero count to sites that are in the dataset, for species having no count data for those sites.

Input description

A tabular file with abundance per year, per site and per species with no or few zero abundance.

The data file can be extracted from the STOC database on demand : romain.lorrilliere@mnhn.fr

The table needs the following structure (at least these 4 fours columns) :

carre	annee	espece	abond
carreId1	2019	especeId	12

Using bytes

History

Rechercher des données

Unnamed history

(empty)

Cet historique est vide. You can **Charger vos propres données** or **Charger des données depuis une source externe**

Figure 2 - Galaxy-Ecology users' interface https://ecology.usegalaxy.eu. Yellow panel on the left: analysis tool list; blue panel in the middle: current tool interface; red panel on the right: Galaxy analysis history

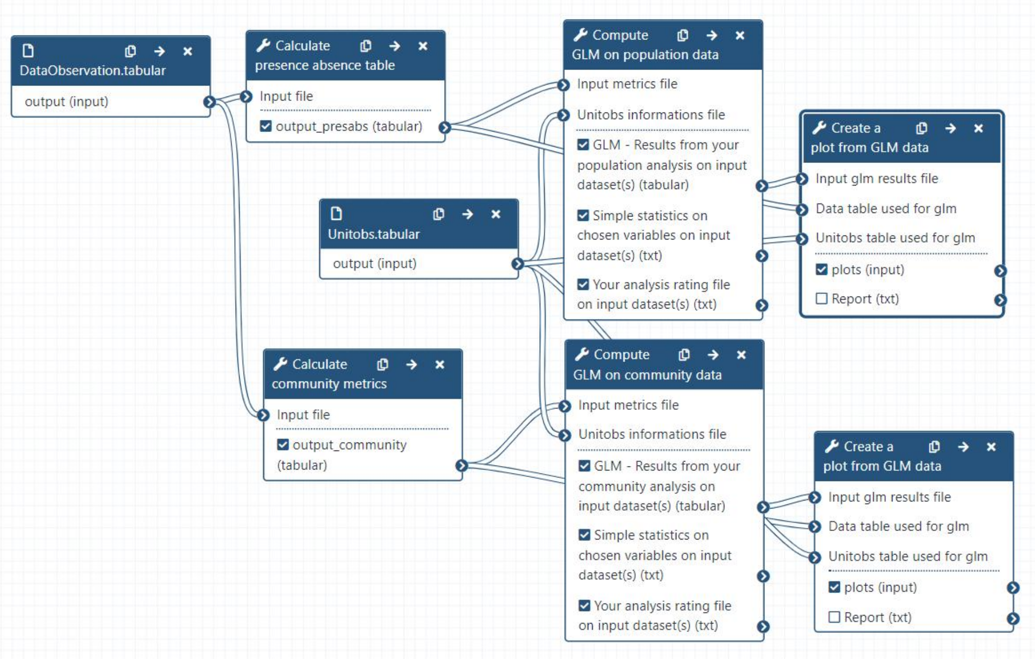

DataObservation.tabular
output (input)
Calculate presence absence table
Input file
☑ output_presabs (tabular)
Unitobs.tabular
output (input)
Calculate community metrics
Input file
☑ output_community (tabular)
Compute GLM on population data
Input metrics file
Unitobs informations file
☑ GLM - Results from your population analysis on input dataset(s) (tabular)
☑ Simple statistics on chosen variables on input dataset(s) (txt)
☑ Your analysis rating file on input dataset(s) (txt)
Compute GLM on community data
Input metrics file
Unitobs informations file
☑ GLM - Results from your community analysis on input dataset(s) (tabular)
☑ Simple statistics on chosen variables on input dataset(s) (txt)
☑ Your analysis rating file on input dataset(s) (txt)
Create a plot from GLM data
Input glm results file
Data table used for glm
Unitobs table used for glm
☑ plots (input)
☐ Report (txt)
Create a plot from GLM data
Input glm results file
Data table used for glm
Unitobs table used for glm
☑ plots (input)
☐ Report (txt)

Figure 3 - Representation of a Galaxy workflow in the editing interface of a Galaxy server. Each box represents an analysis tool, and the lines represent the flow of data through the tools

Anyone can use the tools on Galaxy and/or develop new tools and workflows to make them available to all by publishing them in the shared Galaxy ToolShed (https://toolshed.g2.bx.psu.edu/) which ensures that the tools and dependencies can be installed on any Galaxy servers. By definition, a Galaxy workflow already has a degree of atomisation (each tool represents an elementary step) and generalisation and benefits from the same advantages as the framework presented in the previous section in good practices (tab. 2).

Table 2 - Comparison between the atomisation-generalisation framework and Galaxy for the achievement of good practices. Limitations are occasionally raised with short advice to mitigate them when relevant

		Atomised-generalised code	Galaxy
Reproducibility and transparency	Environment, software and package versions	Can be indicated but possibly hard to manage Can also be set as an output of the analysis (*e.g.* session info) Packages written in each coded elementary step or using a versioning system such as Conda	Entirely packaged with Conda package manager and BioContainers Possibility to store analytical procedures as containers for persistent execution
	Inputs and parameters	One must keep track of different parametrisation and input settings at each computation	Automatically tracked and shareable with the “Galaxy history”
	Peer-review	Organisation of the analytical procedure reviewable by non-code developers Code developers might be able to detect errors as it is easier in shorter scripts Transparency over the development process achievable through Git	Reviewable “Galaxy history” and re-executable workflow Peer-reviewed tools with open-source code Transparency over the development process through Git The workflows can be reviewed by the Intergalactic Workflow Commission (IWC) for best practices
	Output provenance	Can be tracked and reproduced in some cases	Tracked with the “Galaxy history” and reproducible with workflow
FAIR principles	Findable	If properly shared	Web-based solution Unified system for data and software citation and attribution Tools can be made available on several servers Tools can be linked to tools registries and annotated with different ontologies Annotated workflows findable on WorkflowHub (https://workflowhub.eu) and Dockstore (https://dockstore.org)
	Accessible	If properly shared	Free distribution of tools via the Galaxy ToolShed and workflows via WorkflowHub and Dockstore under an open-source licence
	Interoperable	When properly generalised, different elementary steps should be useable in interaction with each other	Use different software, computational language and library versions on a single platform with the Conda package management system Workflows exportable in JSON and shareable through several standards (*e.g.* Common Workflow Language; Crusoe *et al.*, 2022 and Research Object Crate; Soiland-Reyes *et al.*, 2022)
	Reusable	Generalised elementary steps are reusable and adaptable with different analytical procedure, parametrisation and/or inputs	Tools, histories and workflows are re-executable, reusable and adaptable with different analytical procedure, parametrisation and/or inputs. Open-source code can be used outside of a Galaxy server
Technical and knowledge gaps	Understandability	The analytical procedure is clearer when properly atomised	Tools interface, workflow annotations, help sections and tutorials are a valuable help
	Teaching opportunities	Learning the analytical procedure design separately from computing languages, giving structure to trainees Reusability of elementary steps for trainees	Experimenting with intricate analyses without computer code first Tutorials and videos from Galaxy Training Network Galaxy community
	Computing capacity	Need for a computation cluster if large data or demanding algorithm	HPC (High Performance Computing) through an interface Bulk (meta)data manipulation
Collaboration and attribution	Analysis design and development	Achievable through collaborative code-editing applications	With anyone through a Galaxy server
	Citation	Easy reuse of openly shared elementary steps could lead to higher citation rates	Each tool, workflow, and tutorial are provided with a unique identifier for proper attribution and citation

The 12 barriers to data and code-sharing raised by Gomes *et al.*, (2022) can be at least partially addressed by Galaxy (see fig. S1).

Galaxy is a powerful platform enabling researchers to readily move towards good practices. The Galaxy interface mitigates the difficulties associated with library management and code development, which permits simpler access to complex analytical methods. One can focus on the analysis itself and its concepts, rather than on syntax difficulties or cluster programming, disconnecting the study of data analysis concepts from the study of computing languages.

The Galaxy Training Network (GTN) is a valuable asset to the accessibility and reusability of tools and workflows (Batut *et al.*, 2018; Hiltemann *et al.*, 2023). The Galaxy Training platform (https://training.galaxyproject.org) is an open, FAIR, collaborative platform compiling a variety of tutorials written by researchers, administrators, developers, and other contributors. These tutorials not only aim to teach how to use Galaxy, and take advantage of advanced features such as Interactive Tools (*i.e.* interactive applications within Galaxy, *e.g.* Windows desktop, Rstudio, R Shiny apps), but also how to run and interpret scientific analyses through detailed step-by-step guides.

Levels of good practice

As highlighted in previous sections, there are many good practices and recommendations existing for analytical procedures, data management, and computational code development. The levels of application of these good practices fall within a continuum offering many possibilities. From the lowest to the highest good practice levels for a published work there can be for example:

- Raw data and analytical procedure are not shared, only processed and interpreted results along with a brief description of methods.
- Pre-processed data is shared, and methods are described in the word-limit given by the publisher (example: tables of metrics and how it was calculated).
- Raw data and source code are shared on a repository. Software and package versions are not specified and there is no guaranty to be able to reproduce the analytical procedure.
- Raw data and atomised – generalised source codes are shared on a repository with specified hardware, software and dependencies versions. Input parameters are recorded in an attached file.
- Raw data is shared with proper metadata and an actionable version of the whole analytical procedure is traceable, ready to use and eventually reuse on other data types. Such level can be attained notably using Galaxy.
- All results and conclusions are published as an executable paper with analyses and workflows implemented and executable directly in the shared article (Strijkers *et al.*, 2011).

Executable Papers (Strijkers *et al.*, 2011) can require significant time and resource investment as well as good knowledge of programming languages, making it an admirable but hard-to-attain goal. On Galaxy, any available tool can be easy to use. Sharing a complete, detailed and (re-)executable analytical procedure is facilitated as provenance is tracked and metadata is automatically enriched. Finally, a Galaxy history or workflow can be made accessible to anyone (See methods section for details on the use of Galaxy). More effort may be required on Galaxy when an additional analytical step needs to be developed, but the Galaxy community can be an efficient crutch on which hard-pressed scientists can rely. Indeed, one can ask for help on the implementation of tools whether one knows computing languages and can share their code or not.

A deeply collaborative initiative

Galaxy is an utterly participative platform. Any analysis history or workflow can be shared and enriched in parallel by several users, facilitating teamwork. As discussed earlier, several ways to participate to Galaxy exist depending on one’s skills, available time, and needs. In the methods section, three ways to participate to Galaxy are distinguished: “as a user”, “as a developer” and “as a trainer”. One is not confined to only one of these roles; this distinction is more of a handy way to give structure to the methodology depending on one’s skills, available time and needs. Anyone can participate to the Galaxy-Ecology initiative by notably:

- Sharing datasets, histories and workflows;
- Giving feedback on servers, tools, and workflows;
- Sharing tools and workflows ideas (eventually with code) through Git issues;
- Asking for tool modifications through issues;
- Modifying existing tools or proposing new tools through GitHub or GitLab;
- Writing or contributing to a GTN tutorial on a specific functionality or a workflow on the Galaxy Training Network platform;
- Create learning pathways, a set of tutorials curated by community experts to form a coherent set of lessons around a topic, building up knowledge (https://training.galaxyproject.org/training-material/learning-pathways);
- Propose training events and help users in the utilisation of a workflow and tutorial.

Galaxy is community-driven which permits continuous peer review of the platform and of the tools, workflows and tutorials provided. If enough researchers and experts start using and contributing to the platform, the number and content of available analytical procedures could expand at the same pace as latest analytical methodologies are integrated to research processes. If a different platform fits best and is more widely used by ecological and biodiversity scientific communities in the end, the work done on Galaxy will not be lost as tools are easily transposable to other interfaces (*e.g.* scripts directly usable with R, Python, etc., translation of workflows to other workflow engines), histories shareable as files and workflows reusable through WorkflowHub (https://workflowhub.eu) or Dockstore (https://dockstore.org) and exportable in CWL and RO-CRATE standards.

Galaxy-Ecology has implemented workflows for biodiversity data exploration, eDNA processing, general population and community metrics and models, ecoregionalisation, NDVI (Normalised difference vegetation index) computation with Sentinel-2 data among others (see some examples: https://workflowhub.eu/workflows/657) and tutorials for several of them are available on the GTN platform (see https://training.galaxyproject.org/training-material/topics/ecology).

Conclusion

This article showcases a simple proposition to achieve good practices in analytical procedures with two plain guidelines: atomisation and generalisation. This straightforward framework represents a different manner to think and build analytical procedures; it doesn't require using a new technology or learning to use a new software. Relying on existing solutions as much as possible is, in our perspective, an efficient way to achieve a better understanding of good practices and their implications. Given the current environmental crisis, science has the major political and social responsibility to maintain good levels of transparency, reproducibility and efficiency.

Methods - How to Galaxy-fy your analytical procedure?

Analyses are rarely computed only once. Any analysis with a generalisation potential is a suitable candidate to be Galaxy-fied. This methodological framework is presented at three levels depending on potential interests, computing language skills, and willingness to invest more or less time in the process: (i) 'user' relying on existing Galaxy tools and workflows to analyse data (lower time investment), (ii) 'developer' relying on existing and validated analytical procedure to develop Galaxy tools and workflows (highest time investment), and (iii) 'trainer' relying on existing Galaxy tools to share workflows and create training material (variable time investment). Of course, learning to use a new platform and trying to look differently at analyses is time consuming in the short term, but saves time in the long run. Even if in the end the analysis is not made available on Galaxy, the work is not lost as each step helps the analysis to reach a higher level of good practice.

Guidelines “as a user”

Whether one wants to design a new analysis directly on Galaxy or has already an established analytical procedure and wants to adapt it on Galaxy to make it easier to review and reuse, the following steps are approximately the same. As Galaxy already is a workflow-oriented platform with atomisation of steps, “atoms” of the analysis are apparent while building the analysis on Galaxy.

The Galaxy platform offers many options that can be explored using the guided tours of the interface (on the welcome page or tab “Help – Interactive Tours”). Several tutorials are also available on the Galaxy Training Network (https://training.galaxyproject.org) to learn how to use Galaxy (*e.g.* topics

"Introduction to Galaxy Analyses", "Using Galaxy and Managing your Data"). Main steps of the implementation of an analytical procedure on Galaxy as a user are represented on figure 4.

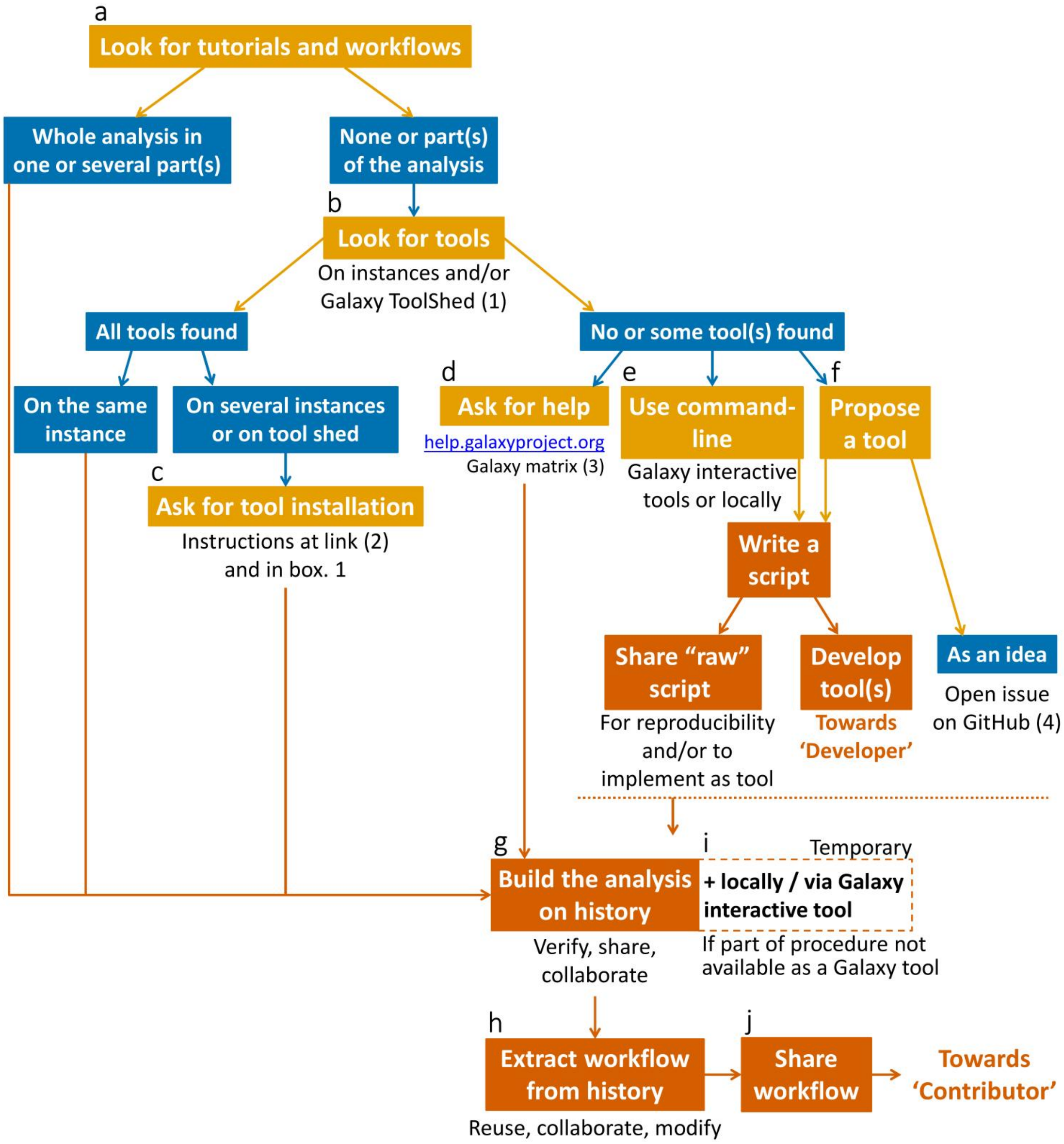

a
Look for tutorials and workflows
Whole analysis in one or several part(s)
None or part(s) of the analysis
b
Look for tools
On instances and/or Galaxy ToolShed (1)
All tools found
No or some tool(s) found
On the same instance
On several instances or on tool shed
c
Ask for tool installation
Instructions at link (2) and in box. 1
d
Ask for help
help.galaxyproject.org
Galaxy matrix (3)
e
Use command-line
Galaxy interactive tools or locally
f
Propose a tool
Write a script
Share "raw" script
For reproducibility and/or to implement as tool
Develop tool(s)
Towards 'Developer'
As an idea
Open issue on GitHub (4)
g
Build the analysis on history
Verify, share, collaborate
i
Temporary
+ locally / via Galaxy interactive tool
If part of procedure not available as a Galaxy tool
h
Extract workflow from history
Reuse, collaborate, modify
j
Share workflow
Towards 'Contributor'

Figure 4 - Decision tree and framework for Galaxy users relying on existing tools and workflows. The orange boxes represent actions. The blue boxes represent possible situations one may encounter during the procedure. The red boxes represent steps where one could stop, share the work, and then attain better reproducibility and FAIRness. Letters at the top left of boxes indicate which paragraph it refers to in the text.
Links: (1) https://toolshed.g2.bx.psu.edu (2) https://usegalaxy-eu.github.io/posts/2020/08/22/three-steps-to-galaxify-your-tool
(3) https://matrix.to (4) https://github.com/galaxyecology

(a) The first thing to do when starting an analysis on Galaxy is to look for tutorials on the Galaxy Training platform to benefit from others' experience. One tutorial may be enough to set the tracks for the whole analytical procedure, but it is also possible to use sub-parts of tutorials and/or associate several tutorials to complete steps of the procedure. Numerous ready-to-use workflows are also available on the Galaxy servers (tab “Shared Data – Workflows”) or could be imported from WorkflowHub or Dockstore, one may find one or several workflows to complete its analysis. High-quality peer-reviewed Galaxy workflows are reported by the Intergalactic Workflow Commission (IWC, https://github.com/galaxyproject/iwc). Additionally, it is possible to seek for help by asking on the Matrix channel (https://gitter.im/Galaxy-Training-Network/Lobby) or by opening a topic on the Galaxy Help (https://help.galaxyproject.org).

(b) If the whole analytical procedure has not been fully covered with available tutorials and workflows, almost 10,000 tools are available on the Galaxy Tool Shed (https://toolshed.g2.bx.psu.edu) to connect the dots.

(c) One or several helpful tools might not be installed on the used Galaxy server and one may need to ask for an installation (See box. 1 Ask for tool installation).

Box 1 - Ask for tool installation. See https://usegalaxy-eu.github.io/posts/2020/08/22/three-steps-to-galaxify-your-tool/ for more details

Fork: Act of creating a copy of a repository in one’s personal space
Commit: Act of submitting a modification to a file
Pull Request (PR): Act of proposing one or several Commit(s) to be integrated
Merge: Act of accepting the PR and integrate the modification proposed on the repository

Galaxy tools installation process is accessible to anyone, it is often explained directly in the “Read me” file on the server tools repository (usually on GitHub or GitLab). To ask for the installation of a tool one must:

- Look for the tool repository on the Galaxy Tool Shed;
- Look for the domain tools repository (*e.g.* https://github.com/usegalaxy-eu/usegalaxy-eu-tools for all Galaxy Europe servers; https://gitlab.com/ifb-elixirfr/usegalaxy-fr/tools for Galaxy France);
- Fork this repository and look for the .yaml file corresponding to the used server (*e.g.* ecology.yaml for the https://ecology.usegalaxy.eu and https://ecology.usegalaxy.fr servers);
- In the .yaml file, make a Commit to add the following lines with the name and owner of the tool (written on the tool repository on the Galaxy Tool Shed) along with a suggested tool panel section in which the tool can be sorted:

~~~
```
name: pampa_presabs
owner: ecology
tool_panel_section_label: 'Species abundance'
```;
~~~

- PR the modification on the domain tools repository and wait for server maintainers’ approval (merge) and/or suggestions. The installation of tools might be rejected if the peer-review process or relevance of the proposed tool is not adequate in the server maintainers’ opinion.

If there are still gaps in the analytical procedure that none of the existing tools can fill, several options are available:

(d) Ask for help (see end of bullet a).

(e) Temporarily fill the gap with a command-line code locally or through a Galaxy Interactive Tool (*e.g.* Rstudio, Jupyter notebook and Ubuntu desktop interactive tools). The code can be shared or not.

(f) Propose a new tool by sharing the idea through a GitHub issue (https://github.com/galaxyecology; preferably along with a code if existing). Details on the task aimed and awaited input and output (*i.e.* full specifications) of the tool along with references are of great help for potential developers who may take over tool development. If one wants to try tool development, see section ‘As a developer’.

(g) Through these steps of looking for tutorials, workflows, and tools, the analytical procedure is progressively designed on the Galaxy history. As each

Galaxy tool, parametrisation and provenance of each file produced is tracked in the Galaxy history, one can try several tools with different parameters to compare and find out which configuration seems the best. The Galaxy history can be shared to anyone through a link to collaborate on the analysis or in a peer-review process.

(h) When parametrisation stage is done and the analytical procedure is complete, one can extract a workflow to reuse the analytical procedure on new datasets.

(i) In the case of a missing tool and part of the analytical procedure is temporarily performed outside Galaxy, one can build separate workflows, between which data is downloaded to make required steps locally. A better temporary solution is to program the launch of Galaxy Interactive Tools (*e.g.* Posit (R), Jupyter notebooks, and Ubuntu desktop interactive tools) in the workflow to keep most of the procedure on Galaxy. In this case, provenance tracking can be secured partially by saving created objects, command history (*e.g.* Rhistory), and running environment for example.

(j) Extracted workflow(s) can be shared with others for feedback or collaboration, but it can also be shared publicly on Galaxy server(s) and/or integrated to an article. When starting to share openly workflow(s), one is a Galaxy contributor as well as a user (see section “As a trainer”).

Guidelines “as a developer”

Developing Galaxy tools requires time investment, especially at the beginning to understand how Galaxy works and the architecture of the tools. The development procedure can vary depending on the origin of the analytical workflow idea which can be (i) existing code, a package, or a workflow implemented elsewhere, (ii) an idea from a user proposal, (iii) a published article or a personal need, and even (iv) an analytical procedure using originally several interfaced tools. When an analytical procedure was originally designed with atomisation and generalisation of elementary steps in mind, the process of developing Galaxy tools should take a lot less time. Main steps of the implementation of an analytical procedure on Galaxy as a developer are represented on figure 5.

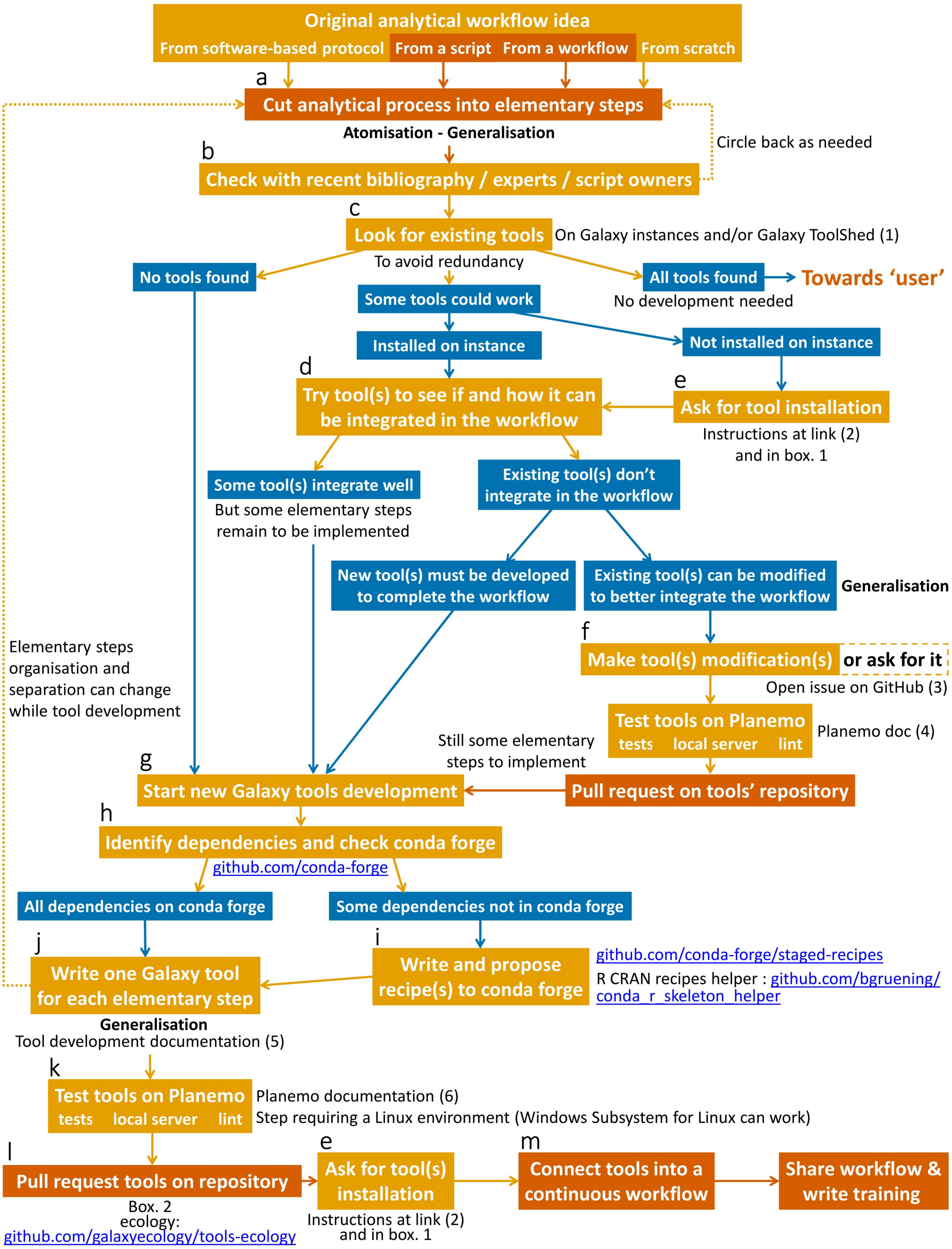

Original analytical workflow idea
From software-based protocol
From a script
From a workflow
From scratch
a
Cut analytical process into elementary steps
Atomisation - Generalisation
Circle back as needed
b
Check with recent bibliography / experts / script owners
c
Look for existing tools
On Galaxy instances and/or Galaxy ToolShed (1)
To avoid redundancy
No tools found
All tools found
Towards 'user'
No development needed
Some tools could work
Installed on instance
Not installed on instance
d
Try tool(s) to see if and how it can be integrated in the workflow
e
Ask for tool installation
Instructions at link (2) and in box. 1
Some tool(s) integrate well
But some elementary steps remain to be implemented
Existing tool(s) don't integrate in the workflow
New tool(s) must be developed to complete the workflow
Existing tool(s) can be modified to better integrate the workflow
Generalisation
f
Make tool(s) modification(s)
or ask for it
Open issue on GitHub (3)
Test tools on Planemo
tests local server lint
Planemo doc (4)
Elementary steps organisation and separation can change while tool development
Still some elementary steps to implement
g
Start new Galaxy tools development
Pull request on tools' repository
h
Identify dependencies and check conda forge
github.com/conda-forge
All dependencies on conda forge
Some dependencies not in conda forge
j
Write one Galaxy tool for each elementary step
i
Write and propose recipe(s) to conda forge
github.com/conda-forge/staged-recipes
R CRAN recipes helper : github.com/bgruening/conda_r_skeleton_helper
Generalisation
Tool development documentation (5)
k
Test tools on Planemo
tests local server lint
Planemo documentation (6)
Step requiring a Linux environment (Windows Subsystem for Linux can work)
l
Pull request tools on repository
Box. 2
ecology:
github.com/galaxyecology/tools-ecology
e
Ask for tool(s) installation
Instructions at link (2) and in box. 1
m
Connect tools into a continuous workflow
Share workflow & write training

Figure 5 - Decision tree and framework for Galaxy developers. Orange boxes represent actions, blue boxes represent possible situations one may encounter during the process and red boxes represent shareable steps where one could stop and still attain better reproducibility and FAIRness. Letters at the top left of boxes indicate which paragraph it refers to in the text.

Links: (1) https://toolshed.g2.bx.psu.edu (2) https://usegalaxy-eu.github.io/posts/2020/08/22/three-steps-to-galaxify-your-tool
(3) https://github.com/galaxyecology
(4) https://planemo.readthedocs.io/en/latest/index.html
(5) https://docs.galaxyproject.org/en/latest/dev/schema.html
(6) https://planemo.readthedocs.io/en/latest/index.html

(a) The atomisation process starts at early stage of the design of an analytical workflow before writing any computer code. Atomisation into elementary steps provides clarity to the development phases. Ultimately, one elementary step equals one Galaxy tool and the modular parameters identified in the code for generalisation would be those that appear on the tool interface.

(b) One can start by splitting essential steps of the analysis (*e.g.* pre-processing, analyses, representations) and detailing each elementary step afterward to get different atomisation resolutions (tab. 1; fig. 1). The first atomisation is not a permanent choice and will certainly be refined over the course of the development process. It is mainly useful as a medium for researchers and other scientists to give feedback on the projected architecture of the workflow and to have an overview of the analytical procedure. As for any analysis, one must check if potential issues or red flags were raised by the community on the methods used and take it into account in the architecture of the workflow. At this point, any products generated from the atomisation process can be shared and be useful to the scientific community. For example, sharing a written description or a schematic representation of the steps and organisation of an analytical procedure (coded or not) is a valuable help for anyone trying to make a similar analysis.

(c) As a user would do and before starting tool development, one must look for existing tools on Galaxy servers and Galaxy ToolShed (https://toolshed.g2.bx.psu.edu) to avoid redundancy. If all needed tools are available, one can directly build their workflow on Galaxy, see 'As a user' section. Many tools are available on Galaxy for data manipulation. If one needs a particular format or type of data there is high probability that it can already be handled on Galaxy.

(d) If some tools could work in the workflow, one must test it to see if and how it can be integrated.

(e) In the case some tools are not installed on the Galaxy server, ask for tool installation (see box.1)

(f) Selected tools might not integrate precisely as aimed, if the input or the output is not formatted as projected in the primary workflow design, other tools added before and/or after might solve the problem. If such tools are not available or the problem is more about a missing parameter or methodology, it might be more coherent to modify existing tool(s) than developing entirely new ones. One can open a new GitHub issue to ask for modifications on the tool repository (found on the Galaxy ToolShed) or directly suggest modifications on the tool. When modifying a tool, the process is approximately the same as for developing an entirely new tool (explained in the next paragraph) only the Pull Request for modifications should be opened on the tool repository.

(g) The Galaxy community has made available a lot of documentation resources for tool development on the GTN Training platform (category "Development in Galaxy"; https://training.galaxyproject.org/training-material/topics/dev) and on the General Galaxy documentation (https://docs.galaxyproject.org; https://docs.galaxyproject.org/en/latest/dev/schema.html).

Galaxy tools have a common architecture (fig. 6). Each tool consists of an XML (Extensible Markup Language) wrapper which defines input file(s) and parameters that are presented to the end-user in the Galaxy web interface ("ToolName.xml" in fig. 6). Inputs provided through the interface can be processed with code in any computing language (“ExeToolName.r” in fig. 6). Outputs of the code are also specified in the XML file and are made available to the user in the Galaxy history at the end of the computation.

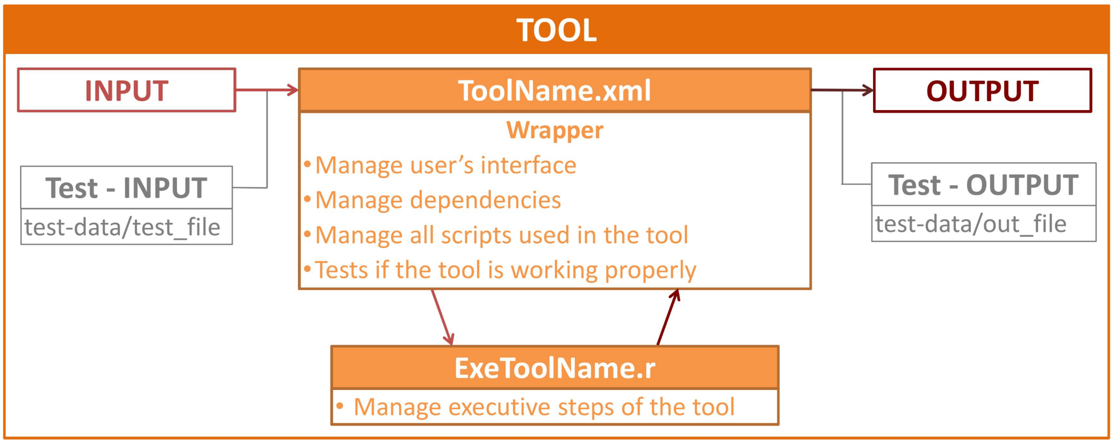


Figure 6 - Schematic representation of the simplified architecture of an example Galaxy tool using R language. From the input files and parameters provided by the user, the tool will launch an analytical procedure through the XML and R files to produce the outputs.

At least one unit test is mandatory to make sure a tool works and produces the expected outputs. This also facilitates maintenance, as tests will indicate if the functionality is preserved after tool updates. To do so, the test is written in the XML file with all parameter settings, input and expected output files (stored in a sub-directory “test-data”) or characteristics of the expected output.

This organisation can be more elaborate, especially when developing several tools at the same time. For example, parts of XML files may repeat themselves in the different tools and one can create a supplementary XML file to write this repeating part once as a macro and call (‘expand’) it as needed, which saves time and space. The same type of repeating patterns can occur in the computing code and one should create a functions file to avoid copy-pasting of many lines in several separate code files.

Detailed documentation of the XML wrapper files is available in Galaxy, see https://docs.galaxyproject.org/en/master/dev/schema.html, as well as tutorials (https://gxy.io/GTN:T00117). An empty Galaxy tool template in R language is available in the following repository: https://github.com/ColineRoyaux/Galaxy_Templates/tree/main/R_Tool_template.

(h) To begin development, it is best to have knowledge of the required informatics dependencies of the tool(s) such as software versions, packages and their versions to directly check their availability on Conda Forge (https://conda-forge.org/feedstock-outputs).

(i) Some dependencies might not be available, and, in this case, one must write and propose a recipe to the Conda Forge on GitHub (https://github.com/conda-forge), for guidelines see https://conda-forge.org/#add_recipe. For Python and R packages available on Pypi or CRAN respectively, helper codes are available to automatically generate recipes, see https://github.com/conda/grayskull and https://github.com/bgruening/conda_r_skeleton_helper (by B. Grüning), respectively. Dependencies of the Galaxy tools are called in the XML file.

(j) Generalisation of computational code is especially important while developing the Galaxy tool to make sure the tool is useful to the largest audience. It is difficult to think about all possible purposes of a tool, one will likely miss some aspects but as Galaxy is a participative platform, anyone can ask for modifications or make it themselves. The format of the input file is a critical aspect of developing a Galaxy tool, while other aspects of the format can be left to the users' choice or imposed. For example, on Galaxy, the preferred format for table input is tab-separated values (TSV or "tabular"). Many tools on Galaxy are available to convert file formats (*e.g.* from CSV to tabular).

For example, a typical choice to make as a developer when developing a tool dealing with tables is to ask the user to specify through the interface which column contains a specific variable, or to require a column name to be present in the input file for the tool to find the variable. The first option is more generalised as it is easier for the user to select a column directly on the interface rather than change column names in the data files. The second option can however be chosen when the tool uses a lot of columns in different input tables or has a lot of intricate parameters to avoid unnecessary complexity of the tool interface. This option can also be consistent for tools using input data file written in a standardised way, as Darwin-core data standard for example.

Depending on the type of manipulations and analyses made in the tool, many parameters might be useful for users to customise such as the type of model, the distribution law of the data, the corrections to make on the data, the level of resolution or the type and format of output(s). Prior discussions on the workflow with experts and researchers on the analytical procedure can permit to raise important parameters for the users to set. Another good way to get a view on what kind of parameters can be useful for users is to check directly for parameters in the functions used in the computational code and identify which ones are important for the computation and might be critical for users to set. These parameters can be provided with default values if the user does not provide a custom value. An "advanced parameters" collapsible section can also be implemented to keep the interface simple while still permitting flexibility for experimented users. Finally, to check if a workflow is properly generalised, one can seek input files of different origins from open data repositories or ask scientists to test their tools.

It is impossible to prevent all possible misuses of software and such events occur also when using command-line functions. Implementation of error and warning messages in the computing code is the best way to avoid misuse (*e.g.* wrong input format or parameter selection). One can also use the interface, the help section of tools, and training to help users to set parameters properly and raise red flags on the use of tools and workflows (*e.g.* the tool cannot be used on some types of data, types of modelling interact badly with some parameters settings or data distributions). If possible, implementing verification steps in the tools to give feedback to the user on how the computation went is also a good way for the user to get hindsight on the results (*e.g.* quantity of data that couldn’t be used in the tool, models’ evaluation variables, summary plots).

(k) To verify tools syntax (lint), run unitary tests (test), and deploy a local Galaxy server to test tools interface (serve), one must use Planemo, the Galaxy Software Development Kit (Bray *et al.*, 2023). Planemo is a command–line tool used on a Linux environment (see documentation https://planemo.readthedocs.io/en/latest. For Windows users, Planemo can work on a WSL (Windows Subsystem for Linux) or using cloud development environment like GitPod. Galaxy Tool development can take many forms; the computational code can be developed beforehand on the local environment or, together with the XML file and be tested directly through a local interface deployed for testing. Each strategy has different pros and cons depending on the type of analytical procedure, the origin of the workflow, and the developer personal preference and knowledge.

(l) When ready, tool(s) can be proposed to a collaborative Galaxy tool repository (for ecology: https://github.com/galaxyecology/tools-ecology; see box. 2 for procedure on GitHub) for peer-review by the community.

Box 2 - Definitions of Git terminology and procedure for proposing a tool to a Galaxy repository

Fork: Act of creating a copy of a repository in one’s personal space
Commit: Act of submitting a modification to a file
Pull Request (PR): Act of proposing one or several Commit(s) to be integrated
Merge: Act of accepting the PR and integrate the modification proposed on the repository

One has to fork the repository to add their new tool with a Commit and propose a PR against the original Galaxy repository with a brief description of the aims of developed tool(s) (PR example: https://github.com/galaxyecology/tools-ecology/pull/50). When a PR is opened on the repository, verification (“Check jobs”) of the tool(s) compatibility, syntax, development good practices and proper running are made automatically. If there are problems, one can check output logs of what went badly and try to correct it while scientists invested in the Galaxy community give feedback on the tool(s). When checks are finally passed and code is peer-reviewed by the community, the PR is merged and the tool(s) made available on the Galaxy ToolShed within a few days. One may then ask for tool installation on any server (see box. 1 Ask for tool installation).

(m) Once all developed tools are available on the Galaxy server, one can build a workflow as a user would do, share it and eventually write a training on the use of the workflow, see section “as a trainer”.

Guidelines “as a trainer”

Main steps of the implementation of an analytical procedure on Galaxy as a trainer are represented on figure 7.

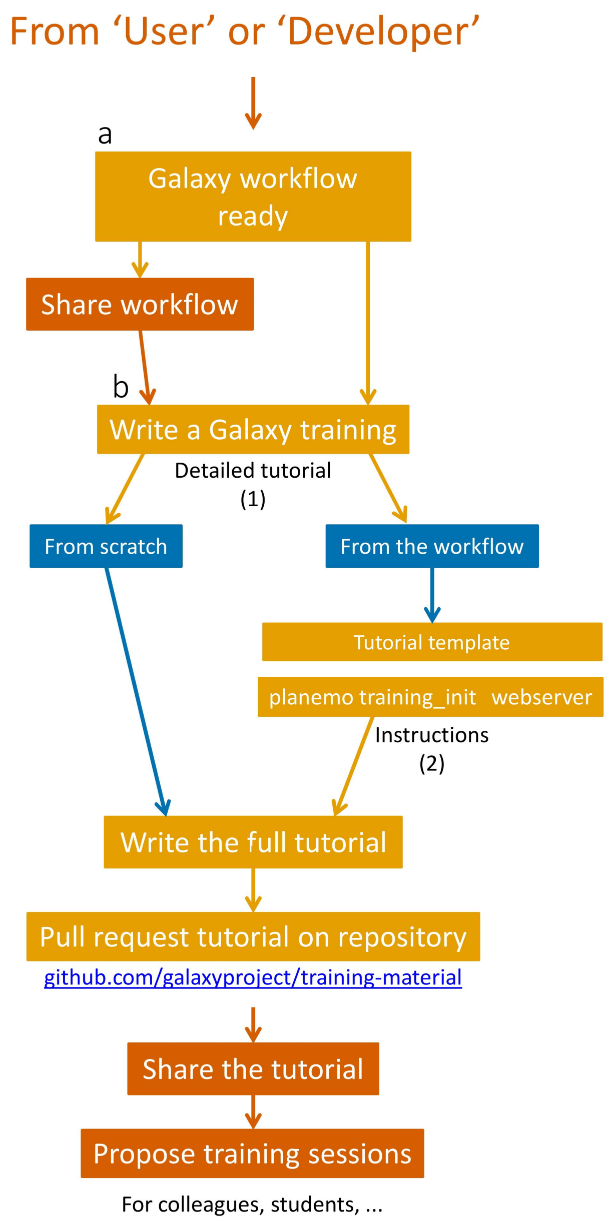

From 'User' or 'Developer'
a
Galaxy workflow ready
Share workflow
b
Write a Galaxy training
Detailed tutorial (1)
From scratch
From the workflow
Tutorial template
planemo training_init webserver
Instructions (2)
Write the full tutorial
Pull request tutorial on repository
github.com/galaxyproject/training-material
Share the tutorial
Propose training sessions
For colleagues, students, ...

Figure 7 - Decision tree and framework for Galaxy trainers. Orange boxes represent actions, blue boxes represent possible situations one may encounter during the process and red boxes represent shareable steps where one could stop and still attain better reproducibility and FAIRness. Letters at the top left of boxes indicate which paragraph it refers to in the text.
Links: (1) https://training.galaxyproject.org/training-material/topics/contributing/tutorials/create-new-tutorial/tutorial.html (2) https://training.galaxyproject.org/training-material/topics/contributing/tutorials/create-new-tutorial/tutorial.html#create-the-skeleton-of-the-tutorial

(a) When an analytical procedure is built on Galaxy, one can extract a workflow from the history created. This workflow can be modified afterward to add annotations, comments, and flags. To make their workflow more generalised, one can leave parameters empty and users will have to set these parameters each time the workflow is launched. This workflow can be shared to contribute to Galaxy. Ultimately, it could be submitted to IWC and be made available on WorkflowHub and/or Dockstore.

(b) Eventually, one can write a tutorial on the GTN or a blog post on the Galaxy Community Hub to get better visibility and broadcast valuable elements on the use of the workflow. GTN tutorials are written in markdown. One can start from scratch, but it is easier to start from a template generated from an existing Galaxy workflow using the dedicated webserver (https://ptdk.apps.galaxyproject.eu) or the command-line software Planemo (documentation: https://planemo.readthedocs.io/en/latest). Indeed, this approach only requires adding any needed explanations between the auto-generate "hands-on" boxes containing tools and parameters instructions. Many tutorials explain the different ways to contribute to the GTN (*e.g.* tutorials, slides, videos, training sessions, quizzes) in the contributing topic on the GTN: https://training.galaxyproject.org/training-material/topics/contributing Introduction on the creation of a new hands-on tutorial is detailed in this tutorial: https://training.galaxyproject.org/training-material/topics/contributing/tutorials/create-new-tutorial/tutorial.html. Like tools, contributions to Galaxy Training Material are proposed through GitHub (https://github.com/galaxyproject/training-material). Available tutorials are publicly and freely available and can be openly shared to colleagues and students and be used during courses and training sessions.

Appendices

Table S1 - Barriers and solutions to data and code-sharing raised by Gomes *et al.* (2022), along with corresponding solutions on the Galaxy platform.

Barriers	Solutions and arguments from Gomes *et al.* (2022)	How Galaxy addresses the barrier
Unclear sharing process	Use FAIR principles Try, even if it is not perfect Look for online resources Ask editorial support staff and institutional libraries	FAIR and workflow-oriented platform Easy sharing of computational procedures ("Galaxy history" and/or workflow) as a link or a file attached to a publication Available online resources and forums for help
Complex workflows	Process and clean data with reproducible code Detailed description of data processing steps Use non-proprietary files or softwares Avoid manual tasks	Reproducible workflows and visualisation of analytical procedure with the interface (fig. 3) "Galaxy history" tracks provenance of outputs and details of the data processing steps Possibility to add annotations and write a tutorial Open source platform Manual tasks can be recorded in workflows
Large data files	Free cloud storage Bundle smaller datasets	Free cloud storage (storage extension on demand) and High Performance Computing
Insecurity	Share to trusted peers and/or on pre-prints servers before formal peer-review Review before publication ensues in higher-quality results Foster an inclusive environment promoting growth over criticism and shame "Perfect code" doesn't exist	"Galaxy history" and workflow record the whole analytical procedure, it is private by default and can be shared to specific users or through a link making review by trusted peers easier and faster before public sharing Peer-reviewed tools
Unclear value	Uncertainty about potential reuse should not present a barrier to sharing	Sharing an analytical procedure is not only relevant for others' reuse but also for collaboration, peer review, and teaching Sharing tools or workflows with Galaxy enables overcoming this uncertainty Methods of the note aims to facilitate this process and ensure it is properly made, adding a layer of clarity regarding the value of shared codes
Inappropriate use	Metadata information with thorough description of datasets and processes, terms and consideration of	Raise major red flags or potential misuse in the help section and/or in the tool execution by validating input before tool

	reuse and any limitations, assumptions, caveats, and shortcomings Include contact information	execution. Implemented errors and warnings in the code to prevent directly prohibitive use of tools. Write execution suggestions and guidelines in the workflow annotations and/or associated tutorial. Possibility to produce editable report when executing a workflow or from the “Galaxy history”
Rights	Use open repositories instead of attaching code and data directly to the article as supplementary material Use data and code licenses Seek for help with institutional libraries and offices dedicated to copyright, open science and commercialisation	Open-source platform and tools shared through public servers prevents copyright issues Each Galaxy tool related code must have a license. Annotation of workflows with license Use of GitHub (or GitLab) to share code and workflows
Sensitive content	Aggregating, generalising or anonymising data	Sharing data and analytical procedure is up to the user Available tool to anonymise geographical coordinates on Galaxy
Transient storage	Archive data in permanent repositories Avoid proprietary files (e. g. Microsoft suite files) Use tools to promote backwards compatibility and portability of softwares and packages within different operating systems (e. g. containers, Jupyter notebooks)	Use of Software Heritage through GitHub to archive code Promotes non-proprietary files (e. g. TSV, fasta) Version-controlled tools to ensure the consistency and persistence of analyses even over updates Conda package manager and BioContainers to ensure cross-operating system compatibility for any programming language Containerisation to ensure cross-infrastructure compatibility (Grüning *et al.*, 2018) Possibility to execute and share Jupyter notebooks Development repositories available in the Galaxy ToolShed
Scooping	Data and code sharing increases opportunities for collaborations Use pre-print servers to make first claim to a research project “Those who collect data and develop code remain best positioned to undertake future analyses” (pp. 6)	Credit of tools are displayed on the interface Users creating a “Galaxy history” can export a reference list of each tool used, facilitating credit attribution Data can be shared privately through a link while being prepared for publication, or while under embargo.
Lack of time	“Despite the upfront time required, sharing research data and code can ultimately save time for individual researchers and their collaborators, as well as for others who want to reuse it” (pp.7) Begin the research project taking account of future sharing of data and code	More time-consuming in the short term as learning to use a new tool is time-costly but time is saved in the long-run as analyses can be re-executed with different parameters, data, or by different users It can help reduce peer review time with possible reproduction of results and easy access to analysis details through the workflow interface
Lack of incentives	“Sharing data and code can increase visibility and recognition of a researcher within the scientific community […]. It can also help develop open science habits that increase efficiency, and contribute to a better understanding of one’s own data and code” (pp.7)	Facilitates sharing and reuse of analytical methods, broader citations of the article associated with the analysis or collaborations could naturally emerge

Acknowledgements

Authors want to thank Sandrine Pavoine for its highly relevant and helpful advices and reviews on both the content and the form of the article.

Authors contribution statement

C. R. drafted the article text, tables, and figures.

C. R. conceptualised the atomisation – generalisation framework with J.-B. M. and Y. L.B. while working on the development of Galaxy workflows.

J.-B. M. and Y. L.B. reviewed and helped rewrite many parts of the draft.

Y. R. and D. P. helped inspire and were invested in the early design of the article.

M. J. and P. S. tested and approved the appliance of the framework.

O. N., M. J., Y. R., M. E., B. B., A. F., H. R. and S. H. highly enhanced the quality of the redaction in both form and content at several stages of the draft.

H. R, S. H., B. B., A. F., and B. G. are involved in the Galaxy-E initiative and provided many advices on the redaction of the article and/or on the development of the initiative.

M. E. and G. M. are involved in Antarctic-oriented Galaxy tool and workflow development coordination.

C. B., R. L., A. M., Y. B., A. A., T. V. and V. C. developped scripts, tools and/or Galaxy workflows to contribute to the Galaxy-E initiative.

E. A. developped R scripts and apps used to integrate R Shiny apps as Galaxy interactive tools and initiate "Research Data management Galaxy tools".

E. M. and C. U. developed the first training materials for Galaxy-E.

E. T. worked on the use of the first Galaxy-E analysis.

M. D., G. L. and R. J. were coordinating the prefiguration of Galaxy-E through the 65 Millions d'Observateurs project.

Additionnally, all authors reviewed and approved the article draft.

Funding

Funding were provided by the European Union through the Erasmus+ project; the Agence Nationale de la Recherche through the 65 Million d'Observateurs and the IA-Biodiv projects; the French National Fund for Open Science through the OpenMetaPaper project; the European commission through the H2020, the EOSC-Pillar, and the H2020 GAPARS projects; the GO FAIR initiative through the BiodiFAIRse Implementation Network; the Blue Nature Alliance; and the Antarctic and Southern Ocean Coalition. Finally, funding by the French Ministry of Higher Education and Research were provided for the “Pôle national de données de biodiversité” e-infrastructure.

Conflict of interest disclosure

The authors declare that they comply with the PCI rule of having no financial conflicts of interest in relation to the content of the article.

References

Archmiller AA, Johnson AD, Nolan J, Edwards M, Elliott LH, Ferguson JM, Iannarilli F, Vélez J, Vitense K, Johnson DH, Fieberg J (2020) Computational Reproducibility in The Wildlife Society's Flagship Journals. *Journal of Wildlife Management*, **84**, 1012–1017. https://doi.org/10.1002/JWMG.21855

Batut B, Hiltemann S, Bagnacani A, Baker D, Bhardwaj V, Blank C, Bretaudeau A, Brillet-Guéguen L, Čech M, Chilton J, Clements D, Doppelt-Azeroual O, Erxleben A, Freeberg MA, Gladman S, Hoogstrate Y, Hotz HR, Houwaart T, Jagtap P, Larivière D, Le Corguillé G, Manke T, Mareuil F, Ramírez F, Ryan D, Sigloch FC, Soranzo N, Wolff J, Videm P, Wolfien M, Wubuli A, Yusuf D, Taylor J, Backofen R, Nekrutenko A, Grüning B (2018) Community-Driven Data Analysis Training for Biology. *Cell Systems*, **6**, 752-758. https://doi.org/10.1016/j.cels.2018.05.012

Berthold MR, Cebron N, Dill F, Gabriel TR, Kötter T, Meinl T, Ohl P, Sieb C, Thiel K, Wiswedel B (2007) KNIME: The Konstanz Information Miner. *Studies* *in Classification, Data Analysis, and Knowledge Organization*, 319–326. https://doi.org/10.1007/978-3-540-78246-9_38

Borgman CL (2020) Qu'est-ce que le travail scientifique des données ? Big data, little data, no data. https://doi.org/10.4000/BOOKS.OEP.14692

Bray S, Chilton J, Bernt M, Soranzo N, van den Beek M, Batut B, Rasche H, Čech M, Cock PJA, Grüning B, Nekrutenko A (2023) The Planemo toolkit for developing, deploying, and executing scientific data analyses in Galaxy and beyond. *Genome Research*, **33**, 261–268. https://doi.org/10.1101/gr.276963.122

Carroll S, Garba I, Figueroa-Rodríguez O, Holbrook J, Lovett R, Materechera S, Parsons M, Raseroka K, Rodriguez-Lonebear D, Rowe R, Sara R, Walker J, Anderson J, Hudson M (2020) The CARE Principles for Indigenous Data Governance. *Data Science Journal*, **19**, 43. https://doi.org/10.5334/dsj-2020-043

Casajus N. (2023) {rcompendium} {An} {R} package to create a package or research compendium structure.

Cohen-Boulakia S, Belhajjame K, Collin O, Chopard J, Froidevaux C, Gaignard A, Hinsen K, Larmande P, Bras Y Le, Lemoine F, Mareuil F, Ménager H, Pradal C, Blanchet C (2017) Scientific workflows for computational reproducibility in the life sciences: Status, challenges and opportunities. *Future Generation Computer Systems*, **75**, 284–298. https://doi.org/10.1016/j.future.2017.01.012

Crusoe MR, Abeln S, Iosup A, Amstutz P, Chilton J, Tijanić N, Ménager H, Soiland-Reyes S, Goble C (2022) Methods Included: Standardizing Computational Reuse and Portability with the Common Workflow Language. *Communications of the ACM*, **65**, 54–63. https://doi.org/10.1145/3486897

Culina A, van den Berg I, Evans S, Sánchez-Tójar A (2020) Low availability of code in ecology: A call for urgent action. *PLOS Biology*, **18**, e3000763. https://doi.org/10.1371/JOURNAL.PBIO.3000763

Di Cosmo R, Zacchiroli S (2017) Software Heritage: Why and How to Preserve Software Source Code.

Di Tommaso P, Chatzou M, Floden EW, Barja P., Palumbo E, Notredame C (2017) Nextflow enables reproducible computational workflows. *Nature Biotechnology*, **35**, 316–319. https://doi.org/10.1038/nbt.3820

Ellemers N (2021) Science as collaborative knowledge generation. *British Journal of Social Psychology*, **60**, 1–28. https://doi.org/10.1111/BJSO.12430

EMBL Australia Bioinformatics Resource (2013) Community Survey Report https://www.embl-abr.org.au/news/braembl-community-survey-report-2013/

Emery NC, Crispo E, Supp SR, Farrell KJ, Kerkhoff AJ, Bledsoe EK, O'Donnell KL, McCall AC, Aiello-Lammens ME (2021) Data Science in Undergraduate Life Science Education: A Need for Instructor Skills Training. *BioScience*, **71**, 1274–1287. https://doi.org/10.1093/BIOSCI/BIAB107

European Commission, Directorate-General for Research and Innovation (2018) Cost-benefit analysis for FAIR research data : cost of not having FAIR research data. *Publications Office*. https://doi.org/10.2777/02999

Fanelli D (2018) Is science really facing a reproducibility crisis, and do we need it to? *Proceedings of the National Academy of Sciences of the United*

States of America, **115**, 2628–2631. https://doi.org/10.1073/pnas.1708272114

Fang FC, Casadevall A (2015) Competitive Science: Is Competition Ruining Science? *Infection and Immunity*, **83**, 1229–1233. https://doi.org/10.1128/IAI.02939-14

Farley SS, Dawson A, Goring SJ, Williams JW (2018) Situating Ecology as a Big-Data Science: Current Advances, Challenges, and Solutions. *BioScience*, **68**, 563–576. https://doi.org/10.1093/BIOSCI/BIY068

Field B, Booth A, Ilott I, Gerrish K (2014) *Using the Knowledge to Action Framework in practice: a citation analysis and systematic review*. *Implementation Science*, **9**, 172. https://doi.org/10.1186/s13012-014-0172-2

Goble C, Cohen-Boulakia S, Soiland-Reyes S, Garijo D, Gil Y, Crusoe MR, Peters K, Schober D (2020) FAIR Computational Workflows. *Data Intelligence*, **2**, 108–121. https://doi.org/10.1162/dint_a_00033

Gomes DGE, Pottier P, Crystal-Ornelas R, Hudgins EJ, Foroughirad V, Sánchez-Reyes LL, Turba R, Martinez PA, Moreau D, Bertram MG, Smout CA, Gaynor KM (2022) Why don't we share data and code? Perceived barriers and benefits to public archiving practices. *Proceedings of the Royal Society B*, **289**, 20221113 https://doi.org/10.1098/rspb.2022.1113

Gownaris NJ, Vermeir K, Bittner MI, Gunawardena L, Kaur-Ghumaan S, Lepenies R, Ntsefong GN, Zakari IS (2022) Barriers to Full Participation in the Open Science Life Cycle among Early Career Researchers. *Data Science Journal*, **21**, 2. https://doi.org/10.5334/DSJ-2022-002

Green AJ, Figuerola J (2005) Recent advances in the study of long-distance dispersal of aquatic invertebrates via birds. *Diversity and Distributions*, **11**, 149–156. https://doi.org/10.1111/j.1366-9516.2005.00147.x

Grüning B, Chilton J, Köster J, Dale R, Soranzo N, van den Beek M, Goecks J, Backofen R, Nekrutenko A, Taylor J (2018) Practical Computational Reproducibility in the Life Sciences. *Cell Systems*, **6**, 631–635. https://doi.org/10.1016/j.cels.2018.03.014

Hampton SE, Jones MB, Wasser LA, Schildhauer MP, Supp SR, Brun J, Hernandez RR, Boettiger C, Collins SL, Gross LJ, Fernández DS, Budden A, White EP, Teal TK, Labou SG, Aukema JE (2017) Skills and Knowledge for Data-Intensive Environmental Research. *BioScience*, **67**, 546–557. https://doi.org/10.1093/BIOSCI/BIX025

Hardisty AR, Bacall F, Beard N, Balcázar-Vargas MP, Balech B, Barcza Z, Bourlat SJ, Giovanni R, Jong Y, Leo F, Dobor L, Donvito G, Fellows D, Guerra AF, Ferreira N, Fetyukova Y, Fosso B, Giddy J, Goble C, Güntsch A, Haines R, Ernst VH, Hettling H, Hidy D, Horváth F, Ittzés D, Ittzés P, Jones A, Kottmann R, Kulawik R, Leidenberger S, Lyytikäinen-Saarenmaa P, Mathew C, Morrison N, Nenadic A, Hidalga AN, Obst M, Oostermeijer G, Paymal E, Pesole G, Pinto S, Poigné A, Fernandez FQ, Santamaria M, Saarenmaa H, Sipos G, Sylla KH, Tähtinen M, Vicario S, Vos RA, Williams AR, Yilmaz P (2016) BioVeL: A virtual laboratory for data analysis and modelling in biodiversity science and ecology. *BMC Ecology*, **16**, 49. https://doi.org/10.1186/S12898-016-0103-Y

Hiltemann S, Rasche H, Gladman S, Hotz HR, Larivière D, Blankenberg D, Jagtap PD, Wollmann T, Bretaudeau A, Goué N, Griffin TJ, Royaux C, Bras Y Le, Mehta S, Syme A, Coppens F, Droesbeke B, Soranzo N, Bacon W, Psomopoulos F, Gallardo-Alba C, Davis J, Föll MC, Fahrner M, Doyle MA, Serrano-Solano B, Fouilloux AC, van Heusden P, Maier W, Clements D, Heyl F, Grüning B, Batut B (2023) Galaxy Training: A powerful framework for teaching! *PLOS Computational Biology*, **19**, e1010752. https://doi.org/10.1371/JOURNAL.PCBI.1010752

Ioannidis JPA (2022) Correction: Why Most Published Research Findings Are False. *Plos Medicine*, **39**, e1004085. https://doi.org/10.1371/JOURNAL.PMED.1004085

Ivimey-Cook ER, Pick JL, Bairos-Novak K, Culina A, Gould E, Grainger M, Marshall B, Moreau D, Paquet M, Royauté R, Sanchez-Tojar A, Silva I, Windecker S (2023) Implementing Code Review in the Scientific Workflow: Insights from Ecology and Evolutionary Biology. *EcoEvoRxiv*. https://doi.org/10.32942/X2CG64

Jenkins GB, Beckerman AP, Bellard C, Benítez-López A, Ellison AM, Foote CG, Hufton AL, Lashley MA, Lortie CJ, Ma Z, Moore AJ, Narum SR, Nilsson J, O'Boyle B, Provete DB, Razgour O, Rieseberg L, Riginos C, Santini L, Sibbett B, Peres-Neto PR (2023) Reproducibility in ecology and evolution: Minimum standards for data and code. *Ecology and Evolution*, **13**, e9961. https://doi.org/10.1002/ECE3.9961

Keenan M, Cutler P, Marks J, Meylan R, Smith C, Koivisto E (2012) Orienting international science cooperation to meet global "grand challenges." *Science and Public Policy*, **39**, 166–177. https://doi.org/10.1093/SCIPOL/SCS019

Knijn A, Michelacci V, Orsini M, Morabito S (2020) Advanced Research Infrastructure for Experimentation in genomicS (ARIES): a lustrum of Galaxy experience. *bioRxiv*. https://doi.org/10.1101/2020.05.14.095901

Köster J, Rahmann S (2012) Snakemake—a scalable bioinformatics workflow engine. *Bioinformatics*, **28**, 2520–2522. https://doi.org/10.1093/bioinformatics/bts480

Lai J, Lortie CJ, Muenchen RA, Yang J, Ma K (2019) Evaluating the popularity of R in ecology. *Ecosphere*, **10**, e02567. https://doi.org/10.1002/ECS2.2567

Lamprecht A-L, Garcia L, Kuzak M, Martinez C, Arcila R, Martin Del Pico E, Dominguez Del Angel V, van de Sandt S, Ison J, Martinez PA, McQuilton P, Valencia A, Harrow J, Psomopoulos F, Gelpi JL, Chue Hong N, Goble C, Capella-Gutierrez S (2019) Towards FAIR principles for research software. *Data Science*, **3**, 37–59. https://doi.org/10.3233/ds-190026

Larcombe L, Hendricusdottir R, Attwood T, Bacall F, Beard N, Bellis L, Dunn W, Hancock J, Nenadic A, Orengo C, Overduin B, Sansone S, Thurston M, Viant M, Winder C, Goble C, Ponting C, Rustici G (2017) ELIXIR-UK role in bioinformatics training at the national level and across ELIXIR. *F1000Research*, **6**, 952. https://doi.org/10.12688/f1000research.11837.1

Lin D, Crabtree J, Dillo I, Downs RR, Edmunds R, Giaretta D, De Giusti M, L'hours H, Hugo W, Jenkyns R, Khodiyar V, Martone ME, Mokrane M, Navale V, Petters J, Sierman B, Sokolova D V, Stockhause M, Westbrook J (2020) the TRUST Principles for digital repositories. *Scientific Data,* **7**, 144. https://doi.org/10.1038/s41597-020-0486-7

Lortie CJ (2021) The early bird gets the return: The benefits of publishing your data sooner. *Ecology and Evolution*, **11**, 10736–10740. https://doi.org/10.1002/ECE3.7853

McIntire EJB, Chubaty AM, Cumming SG, Andison D, Barros C, Boisvenue C, Haché S, Luo Y, Micheletti T, Stewart FEC (2022) PERFICT: A Re-imagined foundation for predictive ecology. *Ecology Letters*, **25**, 1345–1351. https://doi.org/10.1111/ELE.13994

Michener WK (2015) Ten Simple Rules for Creating a Good Data Management Plan. *PLOS Computational Biology*, **11**, e1004525. https://doi.org/10.1371/JOURNAL.PCBI.1004525

Michener WK, Jones MB (2012) Ecoinformatics: Supporting ecology as a data-intensive science. *Trends in Ecology and Evolution*, **27**, 85–93. https://doi.org/10.1016/j.tree.2011.11.016

Minocher R, Atmaca S, Bavero C, McElreath R, Beheim B (2021) Estimating the reproducibility of social learning research published between 1955 and 2018. *Royal Society Open Science*, **8**, 210450. https://doi.org/10.1098/RSOS.210450

Munafò MR, Nosek BA, Bishop DVM, Button KS, Chambers CD, Percie Du Sert N, Simonsohn U, Wagenmakers EJ, Ware JJ, Ioannidis JPA (2017) A manifesto for reproducible science. *Nature Human Behaviour*, **1**, 0021. https://doi.org/10.1038/s41562-016-0021

Natural Environment Research Council (2010, 2012) Most Wanted: Postgraduate Skills Needs in the Environment Sector.

Plesser HE (2018) Reproducibility vs. Replicability: A brief history of a confused terminology. *Frontiers in Neuroinformatics*, **11**, 76. https://doi.org/10.3389/FNINF.2017.00076

Powers SM, Hampton SE (2019) Open science, reproducibility, and transparency in ecology. *Ecological applications*, **29**, e01822. https://doi.org/10.1002/eap.1822

Samota EK, Davey RP (2021) Knowledge and Attitudes Among Life Scientists Toward Reproducibility Within Journal Articles: A Research Survey. *Frontiers in Research Metrics and Analytics*, **6**, 678554. https://doi.org/10.3389/FRMA.2021.678554

Serrano-Solano B, Fouilloux A, Eguinoa I, Kalaš M, Grüning B, Coppens F (2022) Galaxy: A Decade of Realising CWFR Concepts. *Data Intelligence*, **4**, 358–371. https://doi.org/10.1162/dint_a_00136

Soiland-Reyes S, Sefton P, Crosas M, Castro LJ, Coppens F, Fernández JM, Garijo D, Grüning B, La Rosa M, Leo S, Ó Carragáin E, Portier M, Trisovic A, Community R-C, Groth P, Goble C (2022) Packaging research artefacts with RO-Crate. *Data Science*, **5**, 97–138. https://doi.org/10.3233/DS-210053

Strijkers R, Cushing R, Vasyunin D, De Laat C, Belloum ASZ, Meijer R (2011) Toward executable scientific publications. *Procedia Computer Science*, **4**, 707–715. https://doi.org/10.1016/J.PROCS.2011.04.074

The Galaxy Community (2022) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update. *Nucleic acids research*, **50**, W345–W351. https://doi.org/10.1093/NAR/GKAC247

Touchon JC, McCoy MW (2016) The mismatch between current statistical practice and doctoral training in ecology. *Ecosphere*, **7**, e01394. https://doi.org/10.1002/ECS2.1394

Wilkinson MD, Dumontier M, Aalbersberg IjJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten JW, da Silva Santos LB, Bourne PE, Bouwman J, Brookes AJ, Clark T, Crosas M, Dillo I, Dumon O, Edmunds S, Evelo CT, Finkers R, Gonzalez-Beltran A, Gray AJG, Groth P, Goble C, Grethe JS, Heringa J, t Hoen PAC, Hooft R, Kuhn T, Kok R, Kok J, Lusher SJ, Martone ME, Mons A, Packer AL, Persson B, Rocca-Serra P, Roos M, van Schaik R, Sansone SA, Schultes E, Sengstag T, Slater T, Strawn G, Swertz MA, Thompson M, Van Der Lei J, Van Mulligen E, Velterop J, Waagmeester A, Wittenburg P, Wolstencroft K, Zhao J, Mons B (2016) Comment: The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, **3**, 1–9. https://doi.org/10.1038/sdata.2016.18

Williams JJ, Teal TK (2017) A vision for collaborative training infrastructure for bioinformatics. *Annals of the New York Academy of Sciences*, **1387**, 54–60. https://doi.org/10.1111/NYAS.13207