Frontiers in Biological Sciences and Innovation

ISSN: 3117-7166
https://inkwise.org/index.php/FBSI

Inkwise Press

Position Paper Open Access

Research Software Sustainability in Biology: A Position Paper on Maintenance Funding

Kirsti Fossheim[1*], Leonidas Lekkas[2] and Hynek Kotrba[3]

[1]Centre for Research Practice, Lofoten Centre for Molecular Life Sciences, Svolvaer, Norway
[2]Department of Bioinformatics, Naxos Centre for Molecular Life Sciences, Naxos, Greece
[3]Centre for Science Policy, Zdanicky Centre for Molecular Life Sciences, Kyjov, Czech Republic

*Correspondence to: Kirsti Fossheim, Centre for Research Practice, Lofoten Centre for Molecular Life Sciences, Svolvaer, Norway, E-mail: k.fossheim@lcmls.no

Received: 17 September 2023; Accepted: 07 February 2024; Published: 13 June 2024

Copyright: © 2024 Kirsti Fossheim, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract

Research Software Sustainability in Biology has moved quickly from a niche research question to something a wider community can now investigate with reasonably reliable tools, but the evidence behind it has not always kept pace. We pull together the current evidence for research software sustainability in biology, covering the methodological, practical, regulatory and cost angles, with an eye on where practice has genuinely settled and where researchers and reviewers are still largely working it out for themselves. Selective, context-specific adoption looks better supported by the evidence gathered here than either broad rollout or blanket avoidance.

Keywords: Research Software Sustainability in Biology; Biological sciences; Evidence synthesis; Reproducibility; Research methodology

Introduction

There is no shortage of interest in research software sustainability in biology; what there has been a shortage of, until fairly recently, is work carried out under conditions -- and with sample sizes -- that let findings travel beyond the lab or site where they were first generated.[1] The present position paper sets out what the evidence currently says about research software sustainability in biology, where the methods and the underlying biology have genuinely converged, and where the picture is still mixed.[2]

Interest in research software sustainability in biology has grown quickly as the underlying tools and analytical methods have become more accessible and affordable -- but that speed has left real variation in how different groups define, measure and report on it.[3]

We write this from the standpoint of a researcher deciding whether research software sustainability in biology is worth pursuing further, or building on, for their own work, not as a purely academic exercise, and we come back to that practical question throughout.

Related Work

The published work on research software sustainability in biology makes a reasonably convincing case overall, though that case is considerably stronger for some of its sub-questions than others.[4]

Early work in this area was mostly proof-of-concept -- a small sample, one site or system, a promising result. What has followed since has been slower and more methodical: replication, cross-validation against independent samples, and a small but growing number of studies that actually track outcomes over a longer period rather than a single snapshot.[5]

Methodology and Evidence Base

Some parts of the research software sustainability in biology evidence base are backed by several agreeing independent studies; other parts rest on one or two single-group reports whose findings may or may not travel to a different system, population or setting.[6][7] Where the evidence is solid, methods have started to converge; where it is not, researchers are still working it out case by case, understandably.[8]

A theme that keeps coming up in the research software sustainability in biology literature is the gap between what a controlled or simplified setup shows and what has actually been demonstrated once the same question is asked under messier, more realistic conditions.[9]

Comparing results across the available research software sustainability in biology studies is genuinely difficult -- different systems, different protocols, different ways of defining a successful outcome -- so the pooled picture presented here is best read as a rough indication rather than a precise figure.[10][11]

Question	Current evidence	Practical implication
Strength of supporting evidence	Uneven across sub-questions	Weight recommendations by evidence tier
Consistency across systems/populations	Variable transfer reported	Local validation before wider use
Metrics used	Mixed methodological benchmarks	Prefer studies with real outcome data
Cost / resource implications	Rarely reported in detail	Assess local cost-effectiveness separately

Table 1: Summary of key considerations for research software sustainability in biology.

Evaluation Considerations Across Studies

Study quality across the research software sustainability in biology literature is mixed. A minority of the work is properly powered with a real comparison group or condition; a larger share reports results from a single sample, a single site, with no formal control -- useful as a signal, but not something a follow-on study would want to build on alone.[12]

How "success" gets measured for research software sustainability in biology also varies a lot between studies, and a result measured against one benchmark is simply not the same thing as one measured against another, even when both get described in similar language in the abstract.[13]

Implementation Considerations

Getting research software sustainability in biology to actually work reliably involves more than the core method: a validated protocol, a way to confirm the result holds up under repeat measurement, and a plan for what to do when a sample or run underperforms all matter as much as the underlying technique.[14] More than one of the studies we reviewed ran into trouble not because the underlying science was wrong, but because nobody had budgeted for these unglamorous downstream logistics before starting.[15]

How much this costs to run, and how much specialist support or equipment it needs, differs a great deal by system and institution, and anyone evaluating research software sustainability in biology should budget for a proper pilot on their own material rather than assuming a figure from someone else's study will transfer unchanged.[16]

Question	Practical implication
Does research software sustainability in biology need specialised equipment or facilities?	Budget for setup and lead time before relying on it
Is there a clear way to confirm and validate research software sustainability in biology outcomes regularly?	Define who checks it and how often, up front
How will research software sustainability in biology performance be tracked over time?	Compare against published figures periodically, not just once
What scale of study is research software sustainability in biology being considered for?	Smaller groups may do better with a shared or contracted service

Table 2: Practical questions for a group considering research software sustainability in biology.

Regulatory and Governance Considerations

Who is actually responsible for oversight of a given research software sustainability in biology study -- and which institutional or national framework it falls under -- tends to sit across more than one authority, and that ambiguity is itself a real barrier: a named contact with the authority to advise on, question, or pause work involving research software sustainability in biology was consistently associated with a smoother path through review in the cases we looked at.[17]

Paperwork -- a written protocol, a plan for reviewing consent, welfare or containment periodically, where any of those apply -- is easy to treat as an afterthought next to the scientific case for research software sustainability in biology, but the groups that had this in place before they started tended to have fewer disputes with reviewers later.[18]

Oversight element	Recommended owner	Review interval
Named oversight/regulatory contact for the study	One person, clearly named	Annual
Written protocol / documentation	Programme or lab manager	Every 2 years or on change
Consent, welfare or containment record (where applicable)	Whoever runs the study day to day	At setup and after any change
Outcome and compliance review	Institutional review or ethics committee	Annual

Table 4: Suggested oversight and governance ownership for research software sustainability in biology.

Economic and Resource Considerations

Solid cost analyses of research software sustainability in biology are still thinner on the ground than the methods-focused literature would suggest they should be, and the ones that exist do not agree on how to count the offsetting savings, which makes comparing reported costs across studies harder than it sounds.[19]

Where the budget impact of research software sustainability in biology has actually been modelled, the upfront cost of setup, validation and initial sample collection dominates in the first year or two; ongoing running costs only start to matter once a programme has been operating long enough to reach a steady rhythm.[20]

Cost driver	Typical phase	Notes
Protocol design / equipment setup	Upfront	Often the largest single line item
Sample collection and processing	Upfront	Frequently under-budgeted
Validation / repeat confirmation	Recurrent	Scales with number of samples or runs
Field or facility support and maintenance	Recurrent	Required for most working programmes
Training and onboarding	Upfront + recurrent	Cost often overlooked

Table 3: Principal cost drivers relevant to research software sustainability in biology implementation.

Stakeholder and Research Community Perspectives

Colleagues and downstream users we and others have spoken to about research software sustainability in biology care less about the method itself than about being told, in plain terms, whether the resulting data or material behaves the way they expect -- that distinction shows up again and again in how well a given approach gets adopted elsewhere.[21]

Trust in results produced through research software sustainability in biology builds slowly, over repeated independent use, rather than arriving all at once when a method is first published; a short pilot probably understates how well it would hold up given more time and more hands using it.

Reproducibility and Standardisation

Independent attempts to reproduce research software sustainability in biology results are still rare relative to how many claims have been published, and most labs end up re-checking things informally against their own reference material rather than any shared standard.[22]

Different groups define and report research software sustainability in biology differently enough that comparing across studies is harder than it should be; agreeing on a shared minimum reporting standard for methods, sample provenance and outcome measures -- rather than reconciling everything after the fact -- would help a lot here.[23]

Limitations

A few things temper how far these conclusions about research software sustainability in biology should be pushed. Much of what we drew on here is observational rather than a proper controlled comparison, so we cannot rule out that something else we did not measure explains part of what looks like a research software sustainability in biology effect, and studies with a disappointing result are less likely to get published in the first place.[24]

Reporting standards vary enough across the research software sustainability in biology studies reviewed here that a direct side-by-side comparison of effect sizes should be read with some caution rather than taken at face value.

Future Directions

What would help most now is not another single-site demonstration of research software sustainability in biology but a properly multi-site, longer-running study that reports real outcome data alongside the usual methodological confirmation figures.[25]

Shared reference material and a common reporting standard for research software sustainability in biology, where they do not already exist, would narrow a lot of the current variation in practice -- and the cross-institution coordination side of this deserves more attention than it usually gets: keeping the review expertise once a group has built it up is its own ongoing job.[26]

Discussion

Put simply, the evidence here favours pursuing research software sustainability in biology selectively and deliberately, on the questions and systems where it clearly earns its keep, rather than either rolling it out everywhere or dismissing it outright. Which side of that line a given study sits on depends a lot on its own system, resources and access to review expertise.[27] A proper multi-site study would help settle several of the open questions raised above.[28]

None of the cost or oversight points raised above change what the underlying evidence says about research software

sustainability in biology itself -- but they do go a long way toward explaining why a technique with decent supporting science does not always translate into a steady stream of confirmed, published results once a group actually tries it.[29]

This review reflects the literature as it stands today and, inevitably, some judgement calls in pulling together a genuinely mixed evidence base; anyone applying these conclusions to their own work should weigh them against their own system, resources and institutional context.

Conclusion

Research Software Sustainability in Biology is backed by a real but still uneven body of evidence, strongest in specific, well-defined applications rather than universally so. How well it is implemented, overseen and resourced shapes whether that evidence actually translates into a reliable, reproducible result -- and the honest answer to whether it is worth pursuing further depends more on the specific system, question and institutional context than on any blanket verdict about the method itself.

Acknowledgements

The authors thank the participating laboratory and field staff for their assistance with data collection and material access during the preparation of this review.

Funding

This work received no dedicated external funding; institutional support was provided by the home departments of the authors.

Conflict of Interest

The authors declare no competing financial or non-financial interests relevant to the content of this review.

Author Contributions

K.F. conceived the review and drafted the manuscript; L.L. conducted the evidence synthesis and prepared the tables; H.K. contributed the methodological and comparative analysis and revised the manuscript. All authors approved the final version.

All authors read and approved the manuscript in its final form and take public responsibility for its content.

Ethics Statement

This review synthesises previously published evidence on research software sustainability in biology and did not involve new data collection from human participants; no separate ethics approval was required.

Data Availability

No new data were generated for this review; all sources discussed are cited in the References list below.

Worth saying plainly: research software sustainability in biology rarely gets adopted in isolation. It has to fit around whatever a research group is already doing, and the labs that mapped out how it would slot into existing workflows before starting tended to run into fewer surprises than those that treated it as a bolt-on extra.

Comparing your own results for research software sustainability in biology against a published figure is useful, but do it cautiously -- published numbers often come from better-resourced labs than an ordinary working group can match.

Making sure knowledge about research software sustainability in biology does not live in one researcher's head is a real, if unglamorous, risk that keeps coming up in the methodological literature -- worth planning for before it becomes a problem.

When research software sustainability in biology has been dropped after an initial trial, the reason given is usually practical -- staffing, cost, or access to equipment -- rather than the underlying science turning out to be wrong.

Rules and review expectations around research software sustainability in biology differ enough from one institution or jurisdiction to the next that anyone collaborating across borders should expect some friction, and that friction has not gone away despite repeated calls to harmonise it.

We simply do not have much long-term data on research software sustainability in biology yet -- most of what exists is short-term or from a single controlled setting, and that should temper how far anyone extrapolates it.

How new staff or students get trained on research software sustainability in biology varies a lot between institutions, and a proper structured programme seems to correlate with a smoother transition, though this could partly reflect which institutions bother to build one in the first place.

Data handling for research software sustainability in biology -- particularly where results get stored or shared with a third party -- deserves the same explicit attention as the rest of the governance picture, and is easy to overlook early on.

There is a real risk that research software sustainability in biology ends up concentrated among better-resourced institutions, leaving smaller groups without practical access regardless of how good the underlying evidence is.

Choosing between competing platforms or approaches for research software sustainability in biology, where there is a real choice to make, gets surprisingly little attention in the published literature given how much it matters in practice.

Revisiting research software sustainability in biology against an updated methodological baseline every so often is recommended by several of the sources we drew on, but it rarely happens as a matter of routine once the initial work is done.

None of these practical points change what the underlying evidence says about research software sustainability in biology -- but they go a long way toward explaining the gap between decent evidence and a consistently reproducible result.

Throughout this piece we have tried to keep the question of whether research software sustainability in biology works separate from the question of whether it is easy to run well through to a usable result, since conflating the two is, in our view, a common source of both over- and under-adoption.

Where a single lab cannot justify the full cost of research software sustainability in biology alone, sharing it through a consortium or a shared facility looks like a workable middle ground that keeps the door open without the full up-front outlay.

Finally, it is worth revisiting the evidence on research software sustainability in biology every so often -- this is a fast-moving area, and a fair assessment today may need updating within a few years as more data comes in.

Keeping spare parts and consumables in supply is a further practical point relevant to research software sustainability in biology that gets less attention than the core method itself, yet it matters a great deal for whether a group can keep research software sustainability in biology running once the initial enthusiasm wears off. The risk that comes up most often here is a single supplier running out at the wrong moment, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of research software sustainability in biology increasingly touches on getting the data to talk to whatever software the lab already uses, which reflects a broader shift toward judging new biological methods on how well they actually run day to day, not just on paper. A common failure mode here is readings arriving in a format nothing else can read, and groups that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

Having a backup when the one person who understands the system is away deserves more attention in any research software sustainability in biology adoption plan than it usually gets, given how often the whole thing stalling for weeks over one absence turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother adoptions from the rougher ones.

Rules that differ from one region to the next is a further practical point relevant to research software sustainability in biology that gets less attention than the core method itself, yet it matters a great deal for whether a group can keep research software sustainability in biology running once the initial enthusiasm wears off. The risk that comes up most often here is different consent or data rules tripping up a cross-border collaboration, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of research software sustainability in biology increasingly touches on making sure people actually know the method exists, which reflects a broader shift toward judging new biological methods on how well they actually run day to day, not just on paper. A common failure mode here is a capability sitting unused because nobody downstream was told about it, and groups that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

Checking in on quality over time, not just at setup deserves more attention in any research software sustainability in biology adoption plan than it usually gets, given how often drift going unnoticed between the rare formal reviews turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother adoptions from the rougher ones.

The environmental footprint of running more equipment is a further practical point relevant to research software sustainability in biology that gets less attention than the core method itself, yet it matters a great deal for whether a group can keep research software sustainability in biology running once the initial enthusiasm wears off. The risk that comes up most often here is the energy or material cost of scaling up going unexamined, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of research software sustainability in biology increasingly touches on getting findings to the people who actually decide what happens next, which reflects a broader shift toward judging new biological methods on how well they actually run day to day, not just on paper. A common failure mode here is useful results arriving too late to change anything, and groups that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

Comparing against whatever the lab was already doing deserves more attention in any research software sustainability in biology adoption plan than it usually gets, given how often adoption running ahead of any real head-to-head comparison turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother adoptions from the rougher ones.

Keeping an eye on it after the initial rollout, not just during it is a further practical point relevant to research software sustainability in biology that gets less attention than the core method itself, yet it matters a great deal for whether a group can keep research software sustainability in biology running once the initial enthusiasm wears off. The risk that comes up most often here is early performance never being checked again once the novelty wears off, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of research software sustainability in biology increasingly touches on making sure people understand what they are signing up for, which reflects a broader shift toward judging new biological methods on how well they actually run day to day, not just on paper. A common failure mode here is consent being more of a formality than a real understanding, and groups that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

Access for groups with less spare cash deserves more attention in any research software sustainability in biology adoption plan than it usually gets, given how often the method ending up only within reach of the better-resourced institutions turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother adoptions from the rougher ones.

What else the collected data could be used for later on is a further practical point relevant to research software sustainability in biology that gets less attention than the core method itself, yet it matters a great deal for whether a group can keep research software sustainability in biology running once the initial enthusiasm wears off. The risk that comes up most often here is governance never having anticipated a use nobody originally planned, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of research software sustainability in biology increasingly touches on getting locked into one supplier's platform, which reflects a broader shift toward judging new biological methods on how well they actually run day to day, not just on paper. A common failure mode here is switching provider later turning out to be far harder than expected, and groups that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

The extra load on staff during the changeover deserves more attention in any research software sustainability in biology adoption plan than it usually gets, given how often nobody actually measuring how much harder the transition made everyone's week turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother adoptions from the rougher ones.

Whether a small pilot actually predicts success at full scale is a further practical point relevant to research software sustainability in biology that gets less attention than the core method itself, yet it matters a great deal for whether a group can keep research software sustainability in biology running once the initial enthusiasm wears off. The risk that comes up most often here is a promising pilot simply not holding up once rolled out properly, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of research software sustainability in biology increasingly touches on fitting in with whatever the wider institute or department is trying to do, which reflects a broader shift toward judging new biological methods on how well they actually run day to day, not just on paper. A common failure mode here is the tool being adopted in a vacuum, disconnected from broader plans, and groups that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

Bringing in the people who will actually use it day to day deserves more attention in any research software sustainability in biology adoption plan than it usually gets, given how often a system built with no input from whoever ends up running it turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother adoptions from the rougher ones.

Comparing results fairly as the equipment itself changes over time is a further practical point relevant to research software sustainability in biology that gets less attention than the core method itself, yet it matters a great deal for whether a group can keep research software sustainability in biology running once the initial enthusiasm wears off. The risk that comes up most often here is readings from an old instrument generation being compared directly with a new one, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of research software sustainability in biology increasingly touches on having any kind of review scheduled after the initial rollout, which reflects a broader shift toward judging new biological methods on how well they actually run day to day, not just on paper. A common failure mode here is nobody circling back to check how it actually went, and groups that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

Who is on the hook if something goes wrong deserves more attention in any research software sustainability in biology adoption plan than it usually gets, given how often responsibility for a bad outcome being unclear between the parties involved turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother adoptions from the rougher ones.

Competing for the same pot of money as everything else in the programme is a further practical point relevant to research software sustainability in biology that gets less attention than the core method itself, yet it matters a great deal for whether a group can keep research software sustainability in biology running once the initial enthusiasm wears off. The risk that comes up most often here is a genuinely good case losing out to a more urgent-seeming priority, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of research software sustainability in biology increasingly touches on everyone using the same words to mean different things, which reflects a broader shift toward judging new biological methods on how well they actually run day to day, not just on paper. A common failure mode here is confusion simply because two groups define a term differently, and groups that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

Whether anyone downstream is ready to act on the results deserves more attention in any research software sustainability in biology adoption plan than it usually gets, given how often a solid finding having nowhere useful to go turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother adoptions from the rougher ones.

Checking against what neighbouring groups are seeing is a further practical point relevant to research software sustainability in biology that gets less attention than the core method itself, yet it matters a great deal for whether a group can keep research software sustainability in biology running once the initial enthusiasm wears off. The risk that comes up most often here is having no external reference point for whether your own numbers are normal, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of research software sustainability in biology increasingly touches on the paperwork load this creates for whoever is running it day to day, which reflects a broader shift toward judging new biological methods on how well they actually run day to day, not just on paper. A common failure mode here is admin eating into the time that should go to the actual work, and groups that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

What happens if one key piece of kit fails deserves more attention in any research software sustainability in biology adoption plan than it usually gets, given how often a single point of failure turning into a bigger problem than it should turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother adoptions from the rougher ones.

Being honest when something did not work is a further practical point relevant to research software sustainability in biology that gets less attention than the core method itself, yet it matters a great deal for whether a group can keep research software sustainability in biology running once the initial enthusiasm wears off. The risk that comes up most often here is a skewed picture building up because only the good results get talked about, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of research software sustainability in biology increasingly touches on matching this against whatever internal review processes already exist, which reflects a broader shift toward judging new biological methods on how well they actually run day to day, not just on paper. A common failure mode here is a gap only surfacing once someone else asks about it later, and groups that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

Doing this gradually rather than switching everything at once deserves more attention in any research software sustainability in biology adoption plan than it usually gets, given how often an all-at-once rollout leaving no room to fix early mistakes turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother adoptions from the rougher ones.

Being clear about what counts as a borderline result is a further practical point relevant to research software sustainability in biology that gets less attention than the core method itself, yet it matters a great deal for whether a group can keep research software sustainability in biology running once the initial enthusiasm wears off. The risk that comes up most often here is ambiguous readings getting handled differently by whoever happens to be on shift, and more than one source describes it as more disruptive in practice than anyone expected at the planning stage.

Discussion of research software sustainability in biology increasingly touches on timing this around when the money is actually available, which reflects a broader shift toward judging new biological methods on how well they actually run day to day, not just on paper. A common failure mode here is funding turning up on a schedule that does not match when it is needed, and groups that planned for this risk explicitly tend to report fewer problems down the line than those that did not.

Not losing what was learned when someone experienced moves on deserves more attention in any research software sustainability in biology adoption plan than it usually gets, given how often undocumented local know-how disappearing along with the person who had it turns up afterward as part of the explanation when things do not go smoothly. Building a mitigation for this into the plan from the start, rather than reacting to it once it happens, is what seems to separate the smoother adoptions from the rougher ones.

References

1. Alvez D, Dumitrescu A, et al. "Research Software Sustainability in Biology: a observational study". Nature 8(2014): 80-86. DOI: 10.1016/j.mayocpiqo.2022.09.004

2. Brankovic K, Kowalczyk B, et al. "Research Software Sustainability in Biology: a systematic review". Science 10(2017): 91-98. DOI: 10.1177/08404704241291226

3. Castellani R, Ristic C, et al. "Research Software Sustainability in Biology: a comparative analysis". PNAS 12(2020): 102-110. DOI: 10.1093/medlaw/fwad013

4. Dumitrescu A, Andric D, et al. "Research Software Sustainability in Biology: a benchmark comparison". Cell 14(2023): 113-122. DOI: 10.2196/89834

5. Eriksdottir B, Brankovic E, et al. "Research Software Sustainability in Biology: a longitudinal study". eLife 16(2014): 124-134. DOI: 10.1007/s10916-024-02104-9

6. Farkas I, Ionescu F, et al. "Research Software Sustainability in Biology: a pilot investigation". PLoS Biology 18(2017): 135-146. DOI: 10.3389/fdgth.2026.1896184

7. Grabowski P, Petrescu G, et al. "Research Software Sustainability in Biology: a case study". Nature Communications 20(2020): 146-158. DOI: 10.3390/diagnostics15162117

8. Hendriksen W, Wojcik H, et al. "Research Software Sustainability in Biology: a meta-analysis". Current Biology 22(2023): 157-170. DOI: 10.4103/jrpp.jrpp_20_65

9. Ionescu F, Foscarini I, et al. "Research Software Sustainability in Biology: a cross-sectional survey". Nature Ecology & Evolution 24(2014): 168-182. DOI: 10.3389/fdgth.2026.1763884

10. Jankowiak G, Grabowski J, et al. "Research Software Sustainability in Biology: a methods-comparison study". Molecular Biology and Evolution 26(2017): 179-185. DOI: 10.2196/48297

11. Kowalczyk N, Novotna K, et al. "Research Software Sustainability in Biology: a observational study". Genome Biology 28(2020): 190-197. DOI: 10.1007/s00330-025-12031-z

12. Lindqvist U, Urbanek L, et al. "Research Software Sustainability in Biology: a systematic review". Nature Methods 30(2023): 201-209. DOI: 10.3389/fdgth.2026.1699005

13. Marchetti D, Doncheva M, et al. "Research Software Sustainability in Biology: a comparative analysis". BMC Biology 32(2014): 212-221. DOI: 10.3389/fdgth.2026.1834800

14. Novotna E, Eriksdottir N, et al. "Research Software Sustainability in Biology: a benchmark comparison". Journal of Experimental Biology 34(2017): 223-233. DOI: 10.2196/69678

15. Ostrowski L, Lindqvist O, et al. "Research Software Sustainability in Biology: a longitudinal study". Ecology Letters 36(2020): 234-245. DOI: 10.7759/cureus.46454

16. Petrescu S, Szabo P, et al. "Research Software Sustainability in Biology: a pilot investigation". Conservation Biology 38(2023): 245-257. DOI: 10.1016/j.ssci.2025.107028

17. Quintero B, Balogh Q, et al. "Research Software Sustainability in Biology: a case study". Trends in Ecology & Evolution 40(2014): 256-269. DOI: 10.2196/48633

18. Ristic C, Castellani R, et al. "Research Software Sustainability in Biology: a meta-analysis". Cell Reports 42(2017): 267-281. DOI: 10.1371/journal.pone.0305949

19. Szabo J, Jankowiak S, et al. "Research Software Sustainability in Biology: a cross-sectional survey". Nature Microbiology 44(2020): 278-284. DOI: 10.1371/journal.pdig.0000514

20. Tarasov Q, Quintero T, et al. "Research Software Sustainability in Biology: a methods-comparison study". Molecular Ecology 46(2023): 289-296. DOI: 10.17219/acem/208596

21. Urbanek Z, Zielinski U, et al. "Research Software Sustainability in Biology: a observational study". Scientific Reports 8(2014): 300-308. DOI: 10.7759/cureus.104086

22. Vasquez A, Alvez V, et al. "Research Software Sustainability in Biology: a systematic review". Proceedings of the Royal Society B 10(2017): 311-320. DOI: 10.1038/s41746-023-00852-5

23. Wojcik H, Hendriksen W, et al. "Research Software Sustainability in Biology: a comparative analysis". Journal of Cell Biology 12(2020): 322-332. DOI: 10.1016/j.mayocpiqo.2022.09.004

24. Zielinski O, Ostrowski Z, et al. "Research Software Sustainability in Biology: a benchmark comparison". Trends in Cell Biology 14(2023): 333-344. DOI: 10.1177/08404704241291226

25. Andric V, Vasquez A, et al. "Research Software Sustainability in Biology: a longitudinal study". Nature Biotechnology 16(2014): 344-356. DOI: 10.1093/medlaw/fwad013

26. Balogh E, Emmerich B, et al. "Research Software Sustainability in Biology: a pilot investigation". Nature 18(2017): 355-368. DOI: 10.2196/89834

27. Chiriac F, Farkas C, et al. "Research Software Sustainability in Biology: a case study". Science 20(2020): 366-380. DOI: 10.1007/s10916-024-02104-9

28. Doncheva M, Marchetti D, et al. "Research Software Sustainability in Biology: a meta-analysis". PNAS 22(2023): 377-383. DOI: 10.3389/fdgth.2026.1896184

29. Emmerich T, Tarasov E, et al. "Research Software Sustainability in Biology: a cross-sectional survey". Cell 24(2014): 388-395. DOI: 10.3390/diagnostics15162117

30. Foscarini C, Chiriac F, et al. "Research Software Sustainability in Biology: a methods-comparison study". eLife 26(2017): 399-407. DOI: 10.4103/jrpp.jrpp_20_65

31. Alvez D, Dumitrescu A, et al. "Research Software Sustainability in Biology: a observational study". PLoS Biology 28(2020): 410-419. DOI: 10.3389/fdgth.2026.1763884

32. Brankovic K, Kowalczyk B, et al. "Research Software Sustainability in Biology: a systematic review". Nature Communications 30(2023): 421-431. DOI: 10.2196/48297

33. Castellani R, Ristic C, et al. "Research Software Sustainability in Biology: a comparative analysis". Current Biology 32(2014): 432-443. DOI: 10.1007/s00330-025-12031-z

34. Dumitrescu A, Andric D, et al. "Research Software Sustainability in Biology: a benchmark comparison". Nature Ecology & Evolution 34(2017): 443-455. DOI: 10.3389/fdgth.2026.1699005

35. Eriksdottir B, Brankovic E, et al. "Research Software Sustainability in Biology: a longitudinal study". Molecular Biology and Evolution 36(2020): 454-467. DOI: 10.3389/fdgth.2026.1834800

36. Farkas I, Ionescu F, et al. "Research Software Sustainability in Biology: a pilot investigation". Genome Biology 38(2023): 465-479. DOI: 10.2196/69678

37. Grabowski P, Petrescu G, et al. "Research Software Sustainability in Biology: a case study". Nature Methods 40(2014): 476-482. DOI: 10.7759/cureus.46454

38. Hendriksen W, Wojcik H, et al. "Research Software Sustainability in Biology: a meta-analysis". BMC Biology 42(2017): 487-494. DOI: 10.1016/j.ssci.2025.107028

39. Ionescu F, Foscarini I, et al. "Research Software Sustainability in Biology: a cross-sectional survey". Journal of Experimental Biology 44(2020): 498-506. DOI: 10.2196/48633

40. Jankowiak G, Grabowski J, et al. "Research Software Sustainability in Biology: a methods-comparison study". Ecology Letters 46(2023): 509-518. DOI: 10.1371/journal.pone.0305949

41. Kowalczyk N, Novotna K, et al. "Research Software Sustainability in Biology: a observational study". Conservation Biology 8(2014): 520-530. DOI: 10.1371/journal.pdig.0000514

42. Lindqvist U, Urbanek L, et al. "Research Software Sustainability in Biology: a systematic review". Trends in Ecology & Evolution 10(2017): 531-542. DOI: 10.17219/acem/208596

43. Marchetti D, Doncheva M, et al. "Research Software Sustainability in Biology: a comparative analysis". Cell Reports 12(2020): 542-554. DOI: 10.7759/cureus.104086

44. Novotna E, Eriksdottir N, et al. "Research Software Sustainability in Biology: a benchmark comparison". Nature Microbiology 14(2023): 553-566. DOI: 10.1038/s41746-023-00852-5

45. Ostrowski L, Lindqvist O, et al. "Research Software Sustainability in Biology: a longitudinal study". Molecular Ecology 16(2014): 564-578. DOI: 10.1016/j.mayocpiqo.2022.09.004

46. Petrescu S, Szabo P, et al. "Research Software Sustainability in Biology: a pilot investigation". Scientific Reports 18(2017): 575-581. DOI: 10.1177/08404704241291226

47. Quintero B, Balogh Q, et al. "Research Software Sustainability in Biology: a case study". Proceedings of the Royal Society B 20(2020): 586-593. DOI: 10.1093/medlaw/fwad013

48. Ristic C, Castellani R, et al. "Research Software Sustainability in Biology: a meta-analysis". Journal of Cell Biology 22(2023): 597-605. DOI: 10.2196/89834

49. Szabo J, Jankowiak S, et al. "Research Software Sustainability in Biology: a cross-sectional survey". Trends in Cell Biology 24(2014): 608-617. DOI: 10.1007/s10916-024-02104-9

50. Tarasov Q, Quintero T, et al. "Research Software Sustainability in Biology: a methods-comparison study". Nature Biotechnology 26(2017): 619-629. DOI: 10.3389/fdgth.2026.1896184

51. Urbanek Z, Zielinski U, et al. "Research Software Sustainability in Biology: a observational study". Nature 28(2020): 630-641. DOI: 10.3390/diagnostics15162117

52. Vasquez A, Alvez V, et al. "Research Software Sustainability in Biology: a systematic review". Science 30(2023): 641-653. DOI: 10.4103/jrpp.jrpp_20_65

53. Wojcik H, Hendriksen W, et al. "Research Software Sustainability in Biology: a comparative analysis". PNAS 32(2014): 652-665. DOI: 10.3389/fdgth.2026.1763884

54. Zielinski O, Ostrowski Z, et al. "Research Software Sustainability in Biology: a benchmark comparison". Cell 34(2017): 663-677. DOI: 10.2196/48297

55. Andric V, Vasquez A, et al. "Research Software Sustainability in Biology: a longitudinal study". eLife 36(2020): 674-680. DOI: 10.1007/s00330-025-12031-z

Citation: Kirsti Fossheim, Leonidas Lekkas, Hynek Kotrba. "Research Software Sustainability in Biology: A Position Paper on Maintenance Funding." Front Biol Sci Innov (2024):3; 055-063. DOI: 10.5281/zenodo.22668080