OPINION ARTICLE

Hackathon-driven tutorial development [version 1; peer review: 1 approved, 1 approved with reservations]

Bruno M. Grande ¹, Arjun Baghela², Anna Cavalla³, Florian Privé⁴, Peter Zhang⁵, Yisong Zhen⁶

¹Department of Molecular Biology and Biochemistry, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada

²Department of Bioinformatics, University of British Columbia, Vancouver, BC, V6T 1Z4, Canada

³Eli Lilly, Windlesham, GU20 6PH, UK

⁴Laboratoire TIMC-IMAG, UMR 5525, Université Grenoble Alpes, Saint-Martin-d'Hères, 38400, France

⁵Department of Medicine, Faculty of Medicine, University of British Columbia, Vancouver, BC, V6T 1Z4, Canada

⁶State Key Laboratory of Cardiovascular Disease, Fuwai Hospital & Cardiovascular Institute, Chinese Academy of Medical Sciences, Peking Union Medical College, Beijing, 100037, China

V1 First published: 24 Dec 2018, 7:1974
<https://doi.org/10.12688/f1000research.16959.1>

Latest published: 24 Dec 2018, 7:1974
<https://doi.org/10.12688/f1000research.16959.1>

Abstract

Software is essential for data science. However, several software tools remain out of reach for many users due to a lack of documentation, thus limiting progress in the field. Tutorial development by authors and users can greatly improve a tool's accessibility and accelerate its adoption. In this article, we explore hackathons such as hackseq as a venue for authors and users to develop tutorials to address the lack of documented software. We describe four advantages of hackathon-driven tutorial development as well as three challenges that we faced. We also discuss our experience with remote participation. In short, if properly prepared, hackathons can provide a productive venue for assembling a group of passionate people, including remote participants, to develop a suite of related tutorials and address the growing need for accessible software.

Keywords

software, hackathon, documentation, tutorial, vignettes, programming, data science

This article is included in the **Hackathons** collection.

Open Peer Review

Approval Status ? ✓

	1	2
version 1	?	✓
24 Dec 2018	view	view

1. **Ming Tang** , Harvard University, Cambridge, USA
2. **Brad A. Chapman**, Harvard T.H. Chan School of Public Health, Boston, USA

Any reports and responses or comments on the article can be found at the end of the article.

Corresponding author: Bruno M. Grande (bgrande@sfu.ca)

Author roles: **Grande BM:** Conceptualization, Investigation, Supervision, Writing – Original Draft Preparation; **Baghela A:** Investigation, Writing – Review & Editing; **Cavalla A:** Investigation, Writing – Review & Editing; **Privé F:** Investigation, Writing – Review & Editing; **Zhang P:** Investigation, Writing – Review & Editing; **Zhen Y:** Investigation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: The author(s) declared that no grants were involved in supporting this work.

Copyright: © 2018 Grande BM *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Grande BM, Baghela A, Cavalla A *et al.* **Hackathon-driven tutorial development [version 1; peer review: 1 approved, 1 approved with reservations]** F1000Research 2018, 7:1974 <https://doi.org/10.12688/f1000research.16959.1>

First published: 24 Dec 2018, 7:1974 <https://doi.org/10.12688/f1000research.16959.1>

Data science is an interdisciplinary field that relies heavily on the use of software tools. These tools require advanced domain-specific knowledge, which is often difficult to acquire and keep up-to-date considering the rate at which new methods become available and best practices evolve. This difficulty primarily stems from incomplete or unclear documentation. In the case of software packages (e.g. in R and Python), minimal documentation consists of describing inputs and outputs for individual functions. The [Comprehensive R Archive Network](#) (CRAN) is the *de facto* package repository for the R programming language and requires that submitted packages at least include this degree of documentation. CRAN also offers a framework for package developers to include additional documentation in the form of vignettes, which typically demonstrate real-world use cases. However, it appears that the majority of CRAN packages do not have vignettes¹. On the other hand, vignettes are required for submission to [Bioconductor](#), a package repository geared towards computational biology. This requirement has made Bioconductor and its numerous packages much more accessible to both new and experienced users. Unfortunately, the practice of including user-friendly vignettes or tutorials remains uncommon. To address this problem, the authors experimented with tutorial development in a hackathon project at hackseq 2017². We describe our experience in this article.

The aim of our project was to organize the collective knowledge of a group of computational biologists into modular tutorials that leveraged the same dataset. Tutorial topics were proposed by and then assigned to members of the team. The tutorials were designed to be independent from each other, but they can readily be combined to form workshop lessons that use the same dataset. In this paper, we explore the benefits and challenges associated with hackathon-driven tutorial development, including the trade-offs of remote hackathon participation. Briefly, we believe that hackathons are an excellent venue for tutorial development and are particularly suitable for remote participation.

We found at least four benefits of collaboratively developing tutorials in a hackathon setting. First, interest in a given topic can be assessed based on the voluntary participation of hackathon attendees. The assembly of people interested in a topic can further motivate tutorial development. Second, once a common dataset is selected and processed, team members can efficiently work in parallel. Third, although team members do not have to rely on one another, they may draw on the collective knowledge of the team. The various perspectives and ideas from different research specialities can guide tutorial design, resulting in higher-quality material. In more practical terms, developing tutorials during a hackathon allows problems to be more readily resolved, and team members can perform peer review, leading to more polished tutorials. Fourth, these hackathon projects often bring together community members that have yet to work together and can thus catalyze new collaborations.

That said, there are some challenges or considerations to keep in mind when developing tutorials in the context of hackathons. First, the hackathon project should feature a theme or topic that

is focused in scope so that team members can assist one another. The skill level of the target audience should also be determined beforehand. Second, once a theme is decided, any existing tutorials should be identified beforehand to avoid repetition. As mentioned previously, vignettes exist for many software tools, and time is best invested in developing new material that is not yet available. Alternatively, one could build on the work of others by adapting or improving existing open-source tutorials. Third, we suggest that you identify a dataset that can be used in all proposed tutorials and meets the following criteria: openly accessible; properly formatted (i.e. little to no missing or malformed data); relevant to the target audience; and ideally sized (i.e. large enough to be interesting but small enough to fit on a personal computer). For example, in the case of tutorials geared towards computational biologists, there are several interesting human genomic datasets, but access is often restricted for privacy. Accordingly, it may be more practical to select a dataset first and then determine a theme and set of tutorial topics that can be developed using this dataset.

It is often cost-prohibitive to travel for short conferences, especially when travel awards do not cover non-traditional meetings such as hackathons. Fortunately, remote participation is not only possible for hackathons, but relatively straightforward. Several tools exist to support collaborative projects while eliminating the need for collocation. For instance, [GitHub](#) offers decentralized code sharing, [Skype](#) enables face-to-face team discussions, and [Slack](#) is a popular platform for asynchronous communication, which is essential when team members inhabit distant time zones. For example, we successfully managed our project despite being located in Vancouver, BC with remote participants in China and France. We argue that remote participation is especially straightforward for tutorial development because material can be developed independently and thus asynchronously. On the other hand, virtual attendance precludes any participation in social or networking events. There is also additional work involved in ensuring that every local and remote team member have assigned tasks at any given time. Overall though, we believe that allowing remote participation is a net benefit for a hackathon project, especially for tutorial development.

In conclusion, our experience with developing tutorials at a hackathon with remote participants was positive. We believe that we were able to achieve more together than separately, mostly because we gain access to immediate peer review of our tutorials. We also learned lessons that will help ensure success for future hackathon projects geared toward tutorial development. We believe this approach to be generalizable to other fields and a model for assembling passionate data scientists with similar interests and organizing their collective knowledge into modular tutorials. In turn, these tutorials can greatly benefit the field of data science by facilitating the adoption of powerful tools and accelerating the training of future data scientists.

Data availability

No data are associated with this article.

Grant information

The author(s) declared that no grants were involved in supporting this work.

Acknowledgements

The authors acknowledge the hackseq 2017 steering committee for organizing the hackathon.

References

1. Silge J: **Mining CRAN DESCRIPTION Files**. 2017; (Accessed: 26th September 2018). [Reference Source](#)
2. hackseq Organizing Committee 2016: **hackseq: Catalyzing collaboration between**

biological and computational scientists via hackathon [version 2; referees: 2 approved]. *F1000Res*. 2017; **6**: 197.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Open Peer Review

Current Peer Review Status:

Version 1

Reviewer Report 04 February 2019

<https://doi.org/10.5256/f1000research.18543.r43146>

© 2019 Chapman B. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Brad A. Chapman

Bioinformatics Core, Harvard T.H. Chan School of Public Health, Boston, MA, USA

The authors describe their experience with tutorial development in a collaborative event. These events bring together volunteer community participants, either locally or remotely, for a defined period of time to accomplish specific tasks. In this case, the motivation is to help develop and improve user facing documentation for existing software packages.

This is an important problem since documentation writing is an undervalued but important part of building useful scientific software. Some suggestions to help improve the manuscript:

- Could you provide specific examples of materials developed during your collaborative events? Seeing outcomes would help motivate readers of the paper.
- Do you have specific approaches for identifying existing tutorials and datasets, and connecting with other communities doing similar work? One downside of time-bounded events is that it can be hard to have enough time to become familiar with previous work and finish tutorials in the time frame of the event. How do you help make it easier to get started, and ensure that work is in a completed state at the end of the event?
- How do you maintain documentation following the event? Is it integrated back into the projects? Do subsequent events follow up and continue the work? Out of date and obsolete documentation can be as problematic as absent or terse documentation. Strategies to help mitigate this problem and ensure the produced documentation evolves over time would be helpful.
- What are your strategies for motivating attendees to attend a documentation based event? It's always challenging to get researchers to write docs, so it might also be hard to convince people to devote dedicated time over multiple days for this.
- It would be worth mentioning and coordinating with Bérénice Batut, who organizes similar events for the [Galaxy Training Materials](#).

Is the topic of the opinion article discussed accurately in the context of the current literature?

Yes

Are all factual statements correct and adequately supported by citations?

Yes

Are arguments sufficiently supported by evidence from the published literature?

Yes

Are the conclusions drawn balanced and justified on the basis of the presented arguments?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: bioinformatics, community development

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Reviewer Report 11 January 2019

<https://doi.org/10.5256/f1000research.18543.r42213>

© 2019 Tang M. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ming Tang

Harvard FAS informatics (Faculty of Arts and Sciences), Harvard University, Cambridge, MA, USA

Software is the driving force for data science. However, lacking documentation hinders the adoption and usage of certain software tools. In this article, Bruno et.al described the advantages and challenges of hackathon-based tutorial development to promote software usage. Additionally, the authors shared their experience with remote participation of hackathons. Overall, the laid-out points are practical and useful to the community. I have several comments below.

I just learned that documentation has four different functions: tutorials, how-to-guides, explanation and technical references¹. They are distinct from each other and good documentation needs to be structured around them and be separated from each other. Specifically, a tutorial is learning-oriented; a how-to-guide is goal-oriented; an explanation is understanding-oriented and a reference guide is information-oriented. That being said, it would be good for the authors to make it clear that a tutorial is only one part of the function of documentation.

1. The authors did not provide any details of the hackathon. It will be good to have examples. E.g. what tutorials were developed for what tools?
2. If hackathon-driven tutorial development is successful, the authors may need to provide what they have accomplished by developing the tutorials and provide some feedback on the user-end of that tool.
3. Are there any other challenges besides choosing topics, identifying existing tutorials and selecting a dataset? Are there any challenges in organizing such a hackathon?

4. Are there any challenges for remote attendance? E.g. if the attendants are in a different time-zone, how the peer-review was conducted in a timely manner?
5. The sentence: "minimal documentation consists of describing inputs and outputs for individual functions" should contain "arguments/parameters" as well.
6. If one shortens this sentence "These tools require advanced domain-specific knowledge, which is often difficult to acquire and keep up-to-date considering the rate at which new methods become available and best practices evolve. This difficulty primarily stems from incomplete or unclear documentation." to "Difficulties in acquiring domain-specific knowledge primary stems from incomplete or unclear documentation." This is not true. Please re-structure the sentence.

References

1. What nobody tells you about documentation. [accessed 11th Jan 2019]. [Reference Source](#)

Is the topic of the opinion article discussed accurately in the context of the current literature?

Yes

Are all factual statements correct and adequately supported by citations?

Yes

Are arguments sufficiently supported by evidence from the published literature?

Yes

Are the conclusions drawn balanced and justified on the basis of the presented arguments?

Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: bioinformatics, data science

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

The benefits of publishing with F1000Research:

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

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

F1000Research