Meta Rabbit

Rabbit goes meta(genomics)

Thoughts on “Revisiting authorship, and JOSS software publications”

□ luispedro □ Science □ January 19, 2019 □ 3 Minutes

This is a direct response to Titus’ post: [Revisiting authorship, and JOSS software publications (http://ivory.idyll.org/blog/2019-authorship-revisiting.html)](http://ivory.idyll.org/blog/2019-authorship-revisiting.html). In fact, I started writing it as a comment there and it became so long that I decided it was better as its own post.

I appreciate the tone of Titus’ post, more asking questions than answering them, so here are my two or three cent:

There is nothing special about software papers. Different fields have different criteria for what consistutes authorship. I believe that particle physics has an approach close to “everyone that committed to the git repo gets to be an author”, which leads to papers with >1000 authors (note that I am not myself a particle physicist or anything close to it). At that point, the currency of authorship and citations is dilluted to the point where I seriously don’t know what the point is. What is maybe special about software papers is that, because they are newer, there hasn’t been time for a rough-consensus to emerge on what should be the criteria (I guess this is done in part in discussions like these). Having said that, even in well-established fields, you still have issues that are never resolved (in molecular biology: the question of whether technicians should be listed as authors brings in strong opinions on both sides).

My position is against *every positive contribution deserves authorship*. Some positive contributions are significant enough that they deserve authorship, others acknowledgements, others can even go unmentioned. Yes, it’s a judgement call what is “significant” and what is not, but I think someone who, for example, reports a bug that leads to a bugfix is a clear NO (even if it’s a good bug report). Even small code contributions should not lead to authorship (perhaps an acknowledgement is where my indecision boundary is at that point). People who proof read a manuscript also don’t get authorship even if they do find a few typos or suggest a few wording changes.

Of course, contribution need not be code. Tutorials, design, &c, all count. But they should be significant. I also would not consider adding as an author an individual that asked a good question during a seminar on the work. Those sometimes turn out to be like good bug reports in that you improve the work based on them. The fact that significance is a judgement call does not imply that we should drop significance as a criterion.

I think authorship is also about responsibility. If you are an author, then you must take responsibility for some part of the work (naturally, not all of it, but some of it). If there are issues later, it is your responsibility to, at the very least, explain what you did or even to fix it, &c. You should be involved in the paper writing and if, for example, some work is need for revision on that particular aspect of the code, you need to do it.

From my side, I have submitted several patches to projects which were best-efforts at the time, but I don't want to take any responsibility for the project beyond that. If the author of one of those projects now told me that I needed to redo that to work on Mac OS X because one of the reviewers complained about it, I'd tell them "sorry, I cannot help you". I don't think authors should get to do that.

I would get off-topic here, but I also wish there was more of an explicit expectation that *if you publish a software paper, you shall provide minimal maintenance for 5-10 years*. Too often software papers are the obituary of the project rather than the announcement.

My answer to "Another question: does authorship keep accruing over versions? Should all the authors on sourmash 2.0 be authors on sourmash 3.0?" is a strong NO. You don't double dip. If anything, I think it's generally the authors of version 2 that often lose out as people benefit from their work and keep citing version 1.

Finally, a question of my own: what if someone does something outside the project that clearly benefits the project, should they be authors? For example, what if someone does a bioconda package for your code? Or writes an excellent blogpost/tutorial that brings you a very large number of users (or runs a tutorial on it)? Should they be authors? My answer is that it first needs to be significant, and it should not be automatic. It may be appropriate to invite them for authorship, but they should then commit to (at the very least) reading the manuscript and keeping up with the development of the project over the medium-term.

Tagged:
authorship,
science,
scientific-software

Published by luispedro

Luis Pedro Coelho is a computational biologist at EMBL. View all posts by luispedro

This site uses Akismet to reduce spam. Learn how your comment data is processed.

Create a free website or blog at WordPress.com.