The challenges of RSE in the field of Bioinformatics

Dr William Haese-Hill

CompBio Research Software Engineering

25/04/2024

Overview

- New field - “Research Software Engineering” coined in 2012
- Established to offer career framework to software engineers in academia.
  - Existing role didn’t exist – not a RA, but also not a technician.
  - Where could universities hire developers? How to attract from industry?
  - Better training and recognition for the role – build trust.

https://groups.google.com/g/collabw12/c/xSdC0uz-IqA/m/8qhgrfceJKYJ

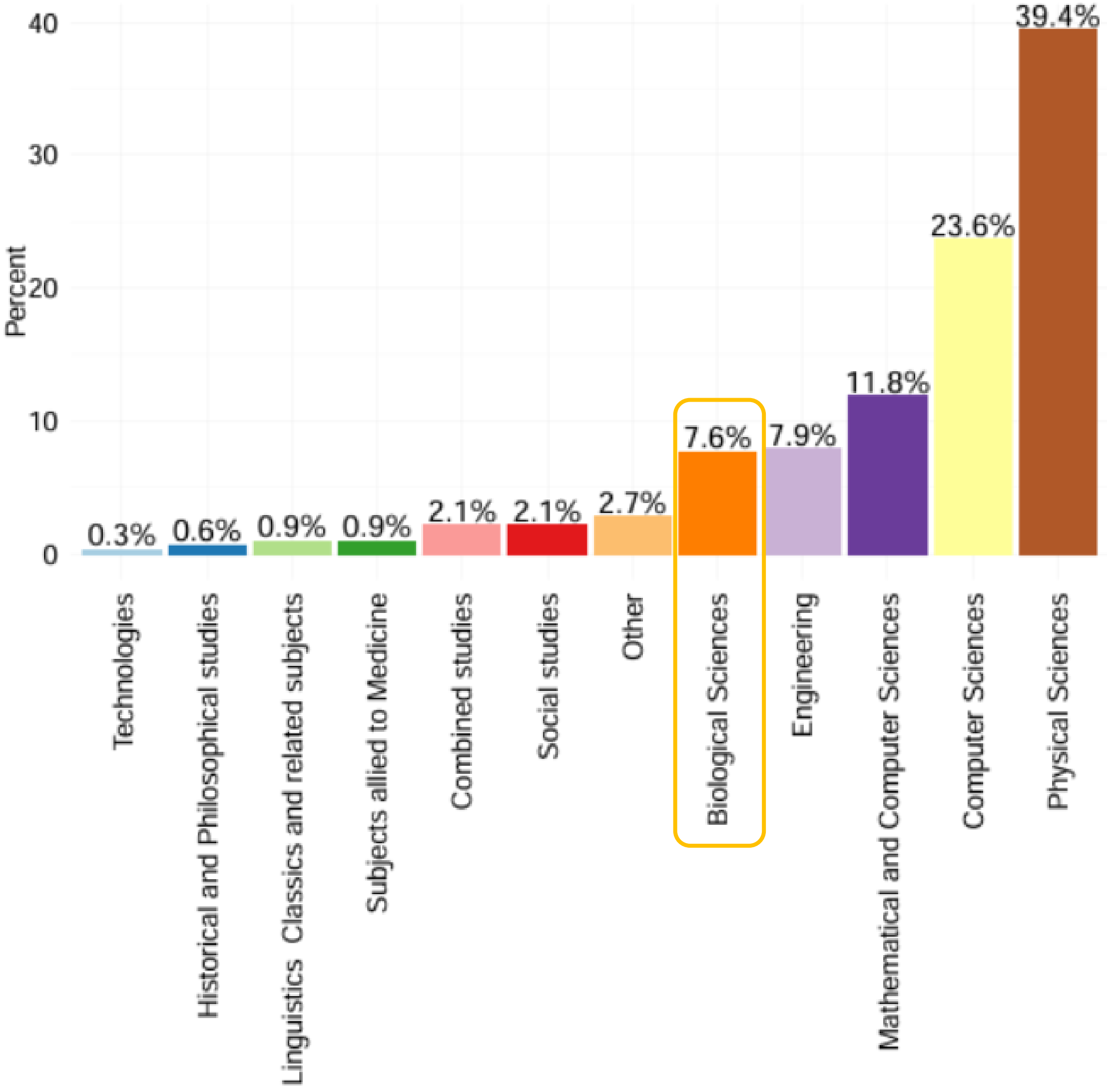
Organisations

- [Home | Software Sustainability Institute]
- [Home - Society of Research Software Engineering (society-rse.org)]
- Universities/labs across the UK (incl. UofG) are setting up “RSE Groups” (>40).

My route to RSE

My background:

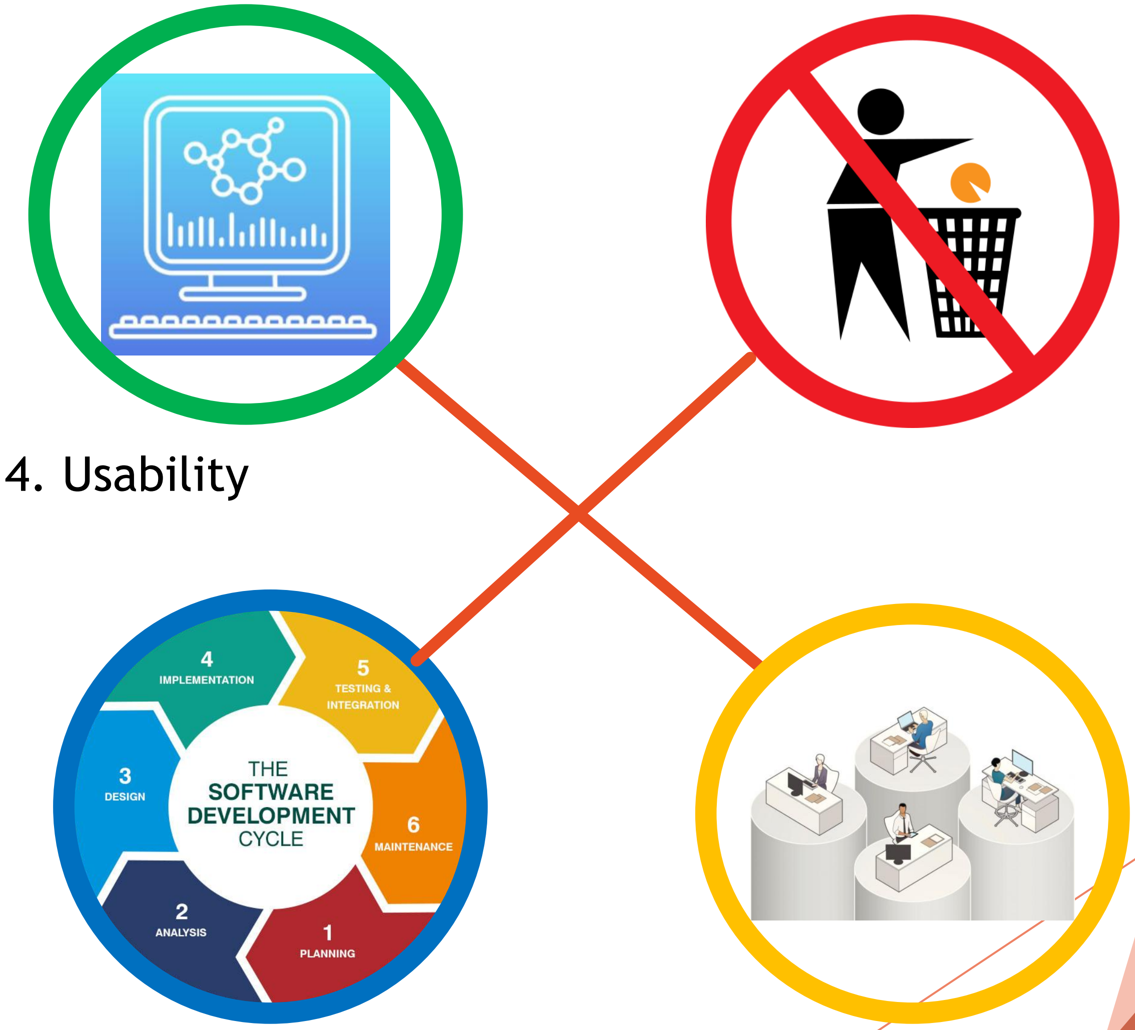
- Mathematical Physics PhD
- 6 years in industry
  - Learnt software development “on the job”, primarily Python.
- Hired in Bioinformatics lab as “Programmer/Data Analyst”
  - Last formal academic engagement with “biological science” was GCSE Biology. Was this an issue?
  - “Technician” or “Research and Teaching” grade?



Philippe, Olivier et al. “Preliminary analysis of a survey of Research Software Engineers in the UK.” (2016).

Challenges

- from personal experience
- not necessarily Bioinformatics-specific



1. “Abandonware”

2. “Silo-fication”

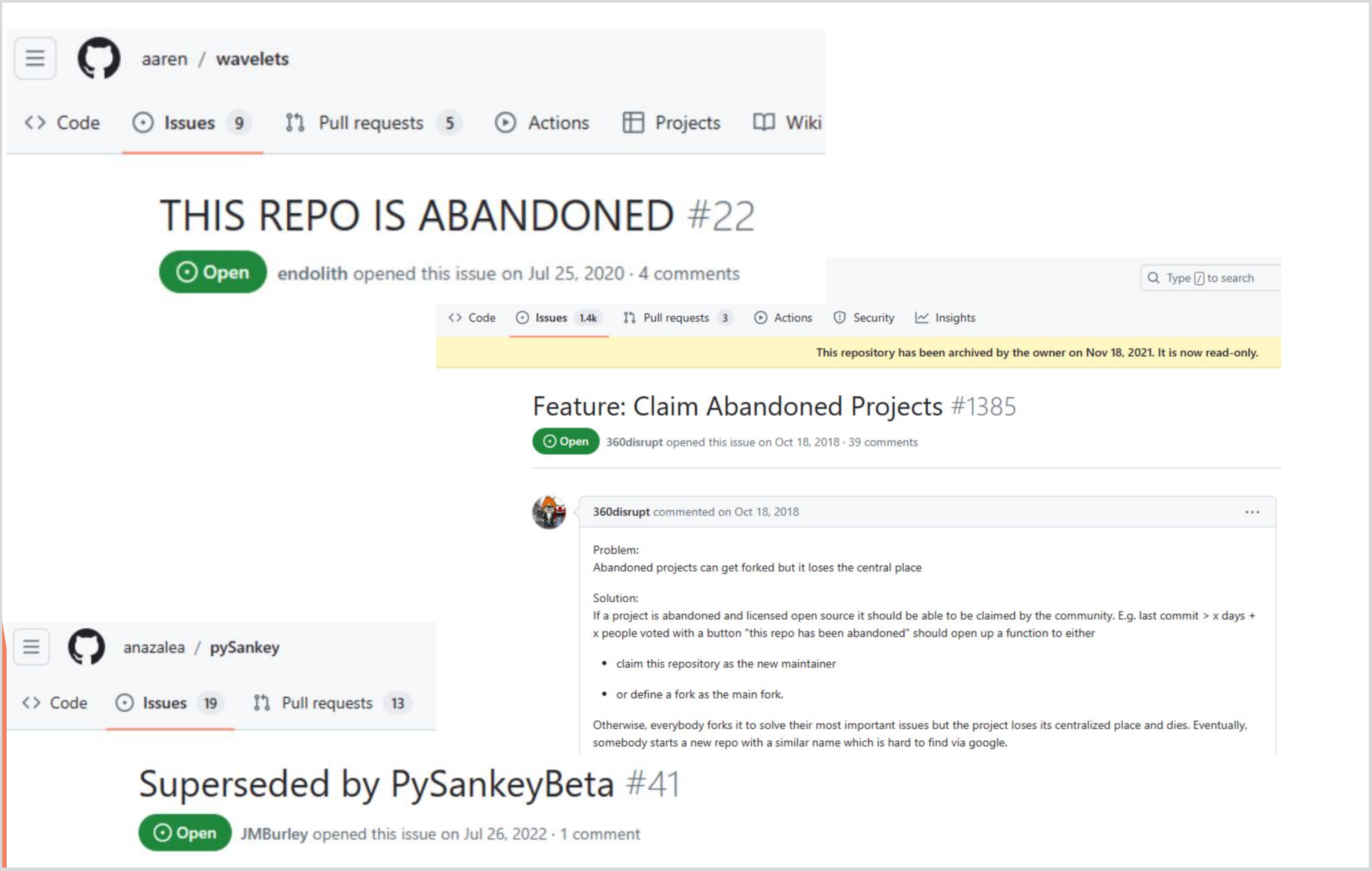
3. Standards

4. Usability

Challenge 1 – post publication “abandonware”

- Primary motivation to publish
- Once published, software tool has “achieved its purpose” -> no longer any impetus to maintain code/docs
  - Top considerations for creating bioinformatics software documentation | Briefings in Bioinformatics | Oxford Academic (oup.com)
- Nature of contracts: when grant funding runs out, who is left to maintain?
  - Frontiers | Better research software tools to elevate the rate of scientific discovery or why we need to invest in research software engineering (frontiersin.org)
- Shift focus away from publishing RSE tools in trad. journals and towards other metrics that support long term maintenance?

Challenge 1 – How can software continue to live?



aaren / wavelets

<> Code · Issues 9 · Pull requests 5 · Actions · Projects · Wiki

THIS REPO IS ABANDONED #22

Open · endolith opened this issue on Jul 25, 2020 · 4 comments

Type / to search

<> Code · Issues 1.4k · Pull requests 3 · Actions · Security · Insights

This repository has been archived by the owner on Nov 18, 2021. It is now read-only.

Feature: Claim Abandoned Projects #1385

Open · 360disrupt opened this issue on Oct 18, 2018 · 39 comments

360disrupt commented on Oct 18, 2018

Problem:
Abandoned projects can get forked but it loses the central place

Solution:
If a project is abandoned and licensed open source it should be able to be claimed by the community. E.g. last commit > x days + x people voted with a button "this repo has been abandoned" should open up a function to either

- claim this repository as the new maintainer
- or define a fork as the main fork.

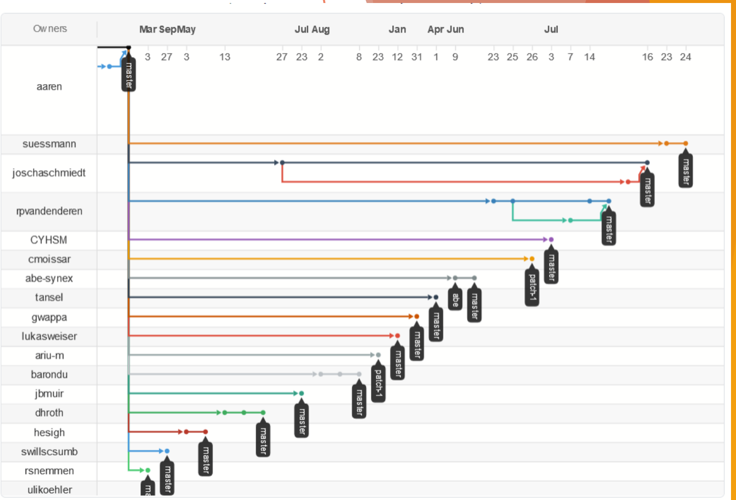
Otherwise, everybody forks it to solve their most important issues but the project loses its centralized place and dies. Eventually, somebody starts a new repo with a similar name which is hard to find via google.

anazalea / pySankey

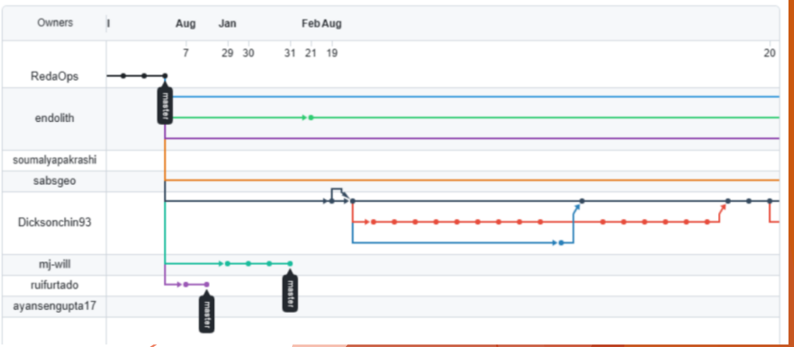
<> Code · Issues 19 · Pull requests 13

Superseded by PySankeyBeta #41

Open · JMBurley opened this issue on Jul 26, 2022 · 1 comment



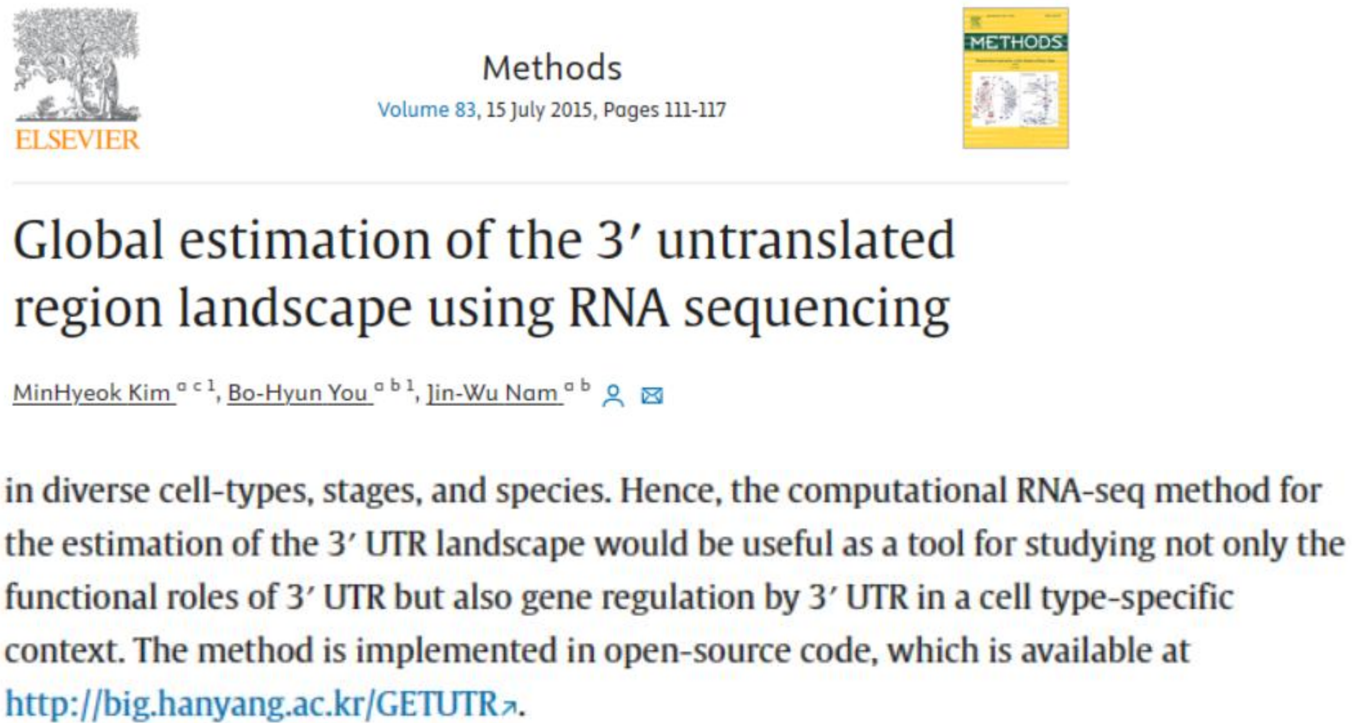
The repository network shows the 100 most recently pushed forks.



Which fork is the “de facto” continuation?

We need a way for the community to adopt abandoned repos · community · Discussion #13470 (github.com)

Challenge 1 – Case study: GETUTR



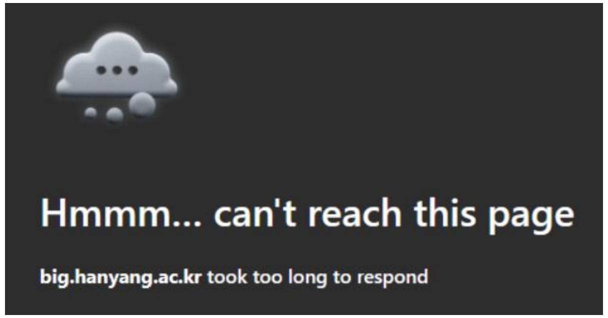
Methods

Volume 83, 15 July 2015, Pages 111-117

Global estimation of the 3′ untranslated region landscape using RNA sequencing

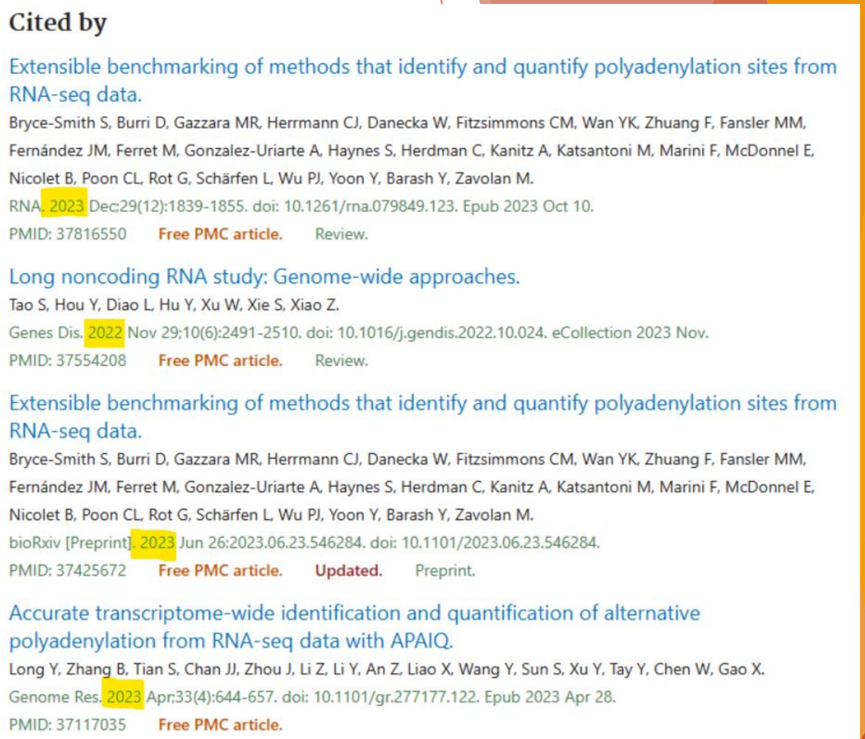
MinHyeok Kim [a c 1], Bo-Hyun You [a b 1], Jin-Wu Nam [a b]

in diverse cell-types, stages, and species. Hence, the computational RNA-seq method for the estimation of the 3′ UTR landscape would be useful as a tool for studying not only the functional roles of 3′ UTR but also gene regulation by 3′ UTR in a cell type-specific context. The method is implemented in open-source code, which is available at http://big.hanyang.ac.kr/GETUTR↗.



Hmmm... can't reach this page

big.hanyang.ac.kr took too long to respond



Cited by

Extensible benchmarking of methods that identify and quantify polyadenylation sites from RNA-seq data.
Bryce-Smith S, Burri D, Gazzara MR, Herrmann CJ, Danecka W, Fitzsimmons CM, Wan YK, Zhuang F, Fansler MM, Fernández JM, Ferret M, Gonzalez-Uriarte A, Haynes S, Herdman C, Kanitz A, Katsantoni M, Marini F, McDonnel E, Nicolet B, Poon CL, Rot G, Schärfen L, Wu PJ, Yoon Y, Barash Y, Zavolan M.
RNA. 2023 Dec;29(12):1839-1855. doi: 10.1261/rna.079849.123. Epub 2023 Oct 10.
PMID: 37816550 **Free PMC article.** Review.

Long noncoding RNA study: Genome-wide approaches.
Tao S, Hou Y, Diao L, Hu Y, Xu W, Xie S, Xiao Z.
Genes Dis. 2022 Nov 29;10(6):2491-2510. doi: 10.1016/j.gendis.2022.10.024. eCollection 2023 Nov.
PMID: 37554208 **Free PMC article.** Review.

Extensible benchmarking of methods that identify and quantify polyadenylation sites from RNA-seq data.
Bryce-Smith S, Burri D, Gazzara MR, Herrmann CJ, Danecka W, Fitzsimmons CM, Wan YK, Zhuang F, Fansler MM, Fernández JM, Ferret M, Gonzalez-Uriarte A, Haynes S, Herdman C, Kanitz A, Katsantoni M, Marini F, McDonnel E, Nicolet B, Poon CL, Rot G, Schärfen L, Wu PJ, Yoon Y, Barash Y, Zavolan M.
bioRxiv [Preprint]. 2023 Jun 26:2023.06.23.546284. doi: 10.1101/2023.06.23.546284.
PMID: 37425672 **Free PMC article.** **Updated.** Preprint.

Accurate transcriptome-wide identification and quantification of alternative polyadenylation from RNA-seq data with APAIQ.
Long Y, Zhang B, Tian S, Chan JJ, Zhou J, Li Z, Li Y, An Z, Liao X, Wang Y, Sun S, Xu Y, Tay Y, Chen W, Gao X.
Genome Res. 2023 Apr;33(4):644-657. doi: 10.1101/gr.277177.122. Epub 2023 Apr 28.
PMID: 37117035 **Free PMC article.**

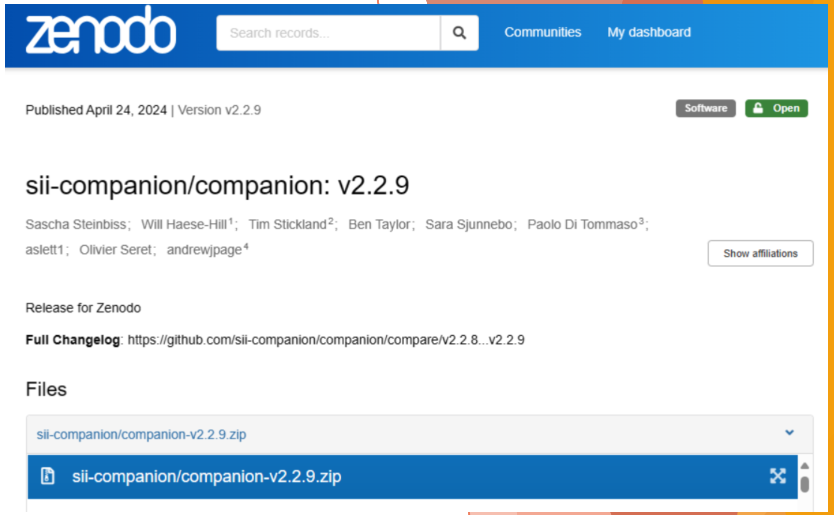
- Python 2 (end-of-life 2020)
- Source code unreachable

Global estimation of the 3' untranslated region landscape using RNA sequencing - PubMed (nih.gov)

Challenge 1 – Reasons to hope?

- Repositories like Zenodo offering DOI
  - Integration with GitHub to sync releases

Automated testing passing | License ISC | NAR 10.1093/nar.gkw292 | DOI 10.5281/zenodo.11059829



Published April 24, 2024 | Version v2.2.9

sii-companion/companion: v2.2.9

Sascha Steinbiss; Will Haese-Hill[1]; Tim Stickland[2]; Ben Taylor; Sara Sjunnebo; Paolo Di Tommaso[3]; aslett1; Olivier Seret; andrewjpage[4]

Release for Zenodo

Full Changelog: https://github.com/sii-companion/companion/compare/v2.2.8...v2.2.9

Files

sii-companion/companion-v2.2.9.zip

sii-companion/companion-v2.2.9.zip

- Journals such as NAR stipulating minimum software availability of 2 years
  - How is this enforced?
  - Could/should it be longer?

Nucleic Acids Research

Challenge 2 – “Silo-fication”

- [How To Identify And Break Down Tech Silos In IT (advsyscon.com)]
- [What is Silo Mentality? How Working in Silos is Dangerous | Miro]

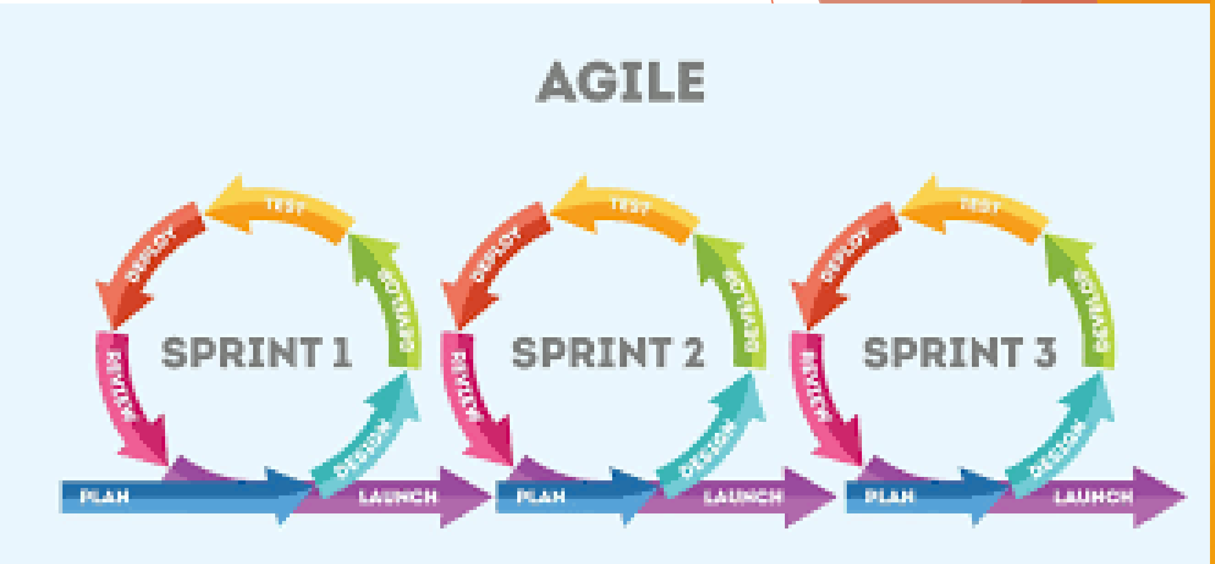
- [Why scientists like to work in silos | World Economic Forum (weforum.org)]
- Researchers building tools for individual problems

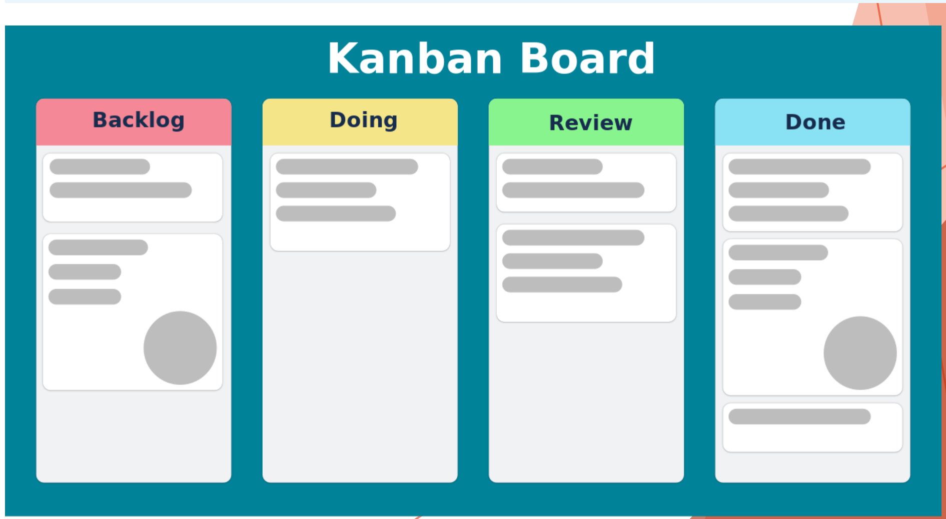
In practice:

- ➢ Pushing code without review
- ➢ Lack of oversight
- ➢ Duplicated effort

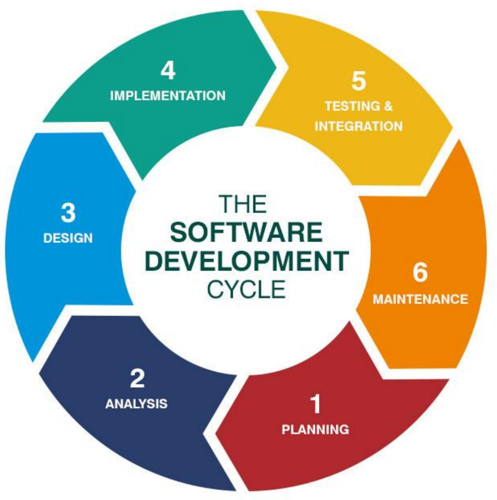
Challenge 2 – Lessons from industry

- “Agile”
  - Sprints
  - Stand-ups
  - Scrums
- Pair-programming
- Code reviews (require before merge)
- Kanban / Task management
  - E.g. Trello/JIRA





Challenge 3 – Software development standards



- Research software often built by researchers whose route to programming is building scripts to perform some analysis
  - Less emphasis on robustness, more on just getting the job done
    - Lacking tests or CI/CD.
  - Lacking version control.
- Attempts to right this:
  - Introducing the FAIR Principles for research software | Scientific Data (nature.com)
  - Proposed Standards For Public Health Bioinformatics Software (pha4ge.org)

Challenge 4 – Usability

- Users often have little computational background (e.g. wet lab scientists)
- Bioinformatics (particularly genomics) tools often pipelines:
  - Lots of parameters for each component – how do you present this to a user in interface (and remain user-friendly)?
  - Many components mean many dependencies -> containerisation or web interface (centralise dependency handling)

Personal experience 1 – legacy system (Companion)

- Came in to upgrade system that had been running more or less unchanged for a number of years.
- Upgrading dependencies (some now obsolete)
- Unfamiliar languages to learn (Ruby, Perl)

+ Improving standards (C3): versioned releases, more robust deployment
+ Using Trello for task handling (C2)

- Still working in a “silo” (C2)

https://companion.ac.uk/

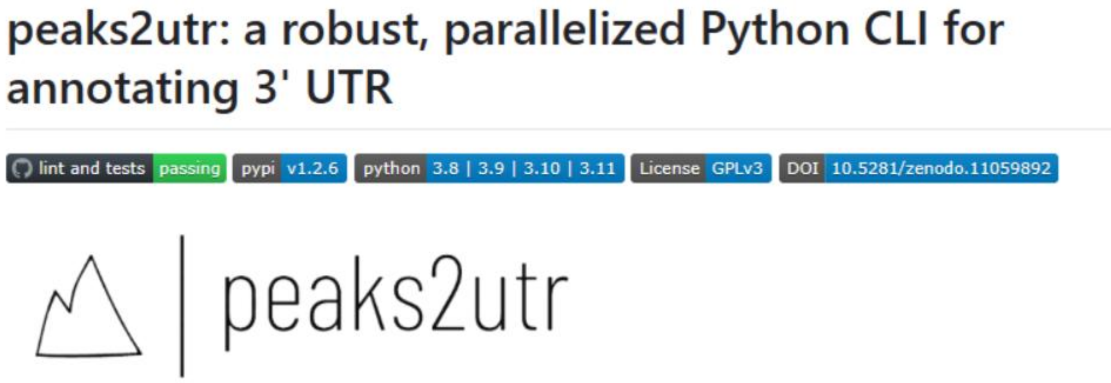
https://github.com/iii-companion/companion

COMPANION

Easy and reliable genome annotation.

Personal experience 2 – “greenfield” project (peaks2utr)

- C4: Wanted to make installation as simple as possible. However, CLI so might still be off-putting (no GUI)
- C1: Published but still compelled to maintain
  - Thanks to “GitHub portfolio” effect (i.e. increase “stars”/engagement), respond to issues
- C2: Worked on it more-or-less independently, although
  - A post-doc reached out and collaborated with me on a PR for a desired feature (post-publication). Made possible by “readable” source code and version control (C3): unit tests, code style



peaks2utr: a robust, parallelized Python CLI for annotating 3' UTR

https://github.com/haessar/peaks2utr

Personal experience 3 – plugin development (paraCell, Apollo3)

CZ CELL × GENE

- Identified existing tool “cellxgene” and plugin “cellxgene_VIP” that could meet needs. Worked on fork of VIP
- C3: VIP source code was a mess (e.g. zero organisation)
  - But didn’t want to diverge too much in case we needed to pull from upstream
  - Didn’t have time/resource to rebuild from scratch
  - C1: higher risk of abandonment due to difficulty to maintain
- Apollo3: example of RSE project using principles from commercial software development (e.g. “Agile”; C2) – active development

https://github.com/sii-cell-atlas/paraCell

https://github.com/GMOD/Apollo3

Resources

- Improving bioinformatics software quality through incorporation of software engineering practices - PMC (nih.gov)
- Why are so many bioinformatic tools so infuriating to use? : r/bioinformatics (reddit.com)
- madhadron - A farewell to bioinformatics
- Why science needs more research software engineers (nature.com)
- Breakout: career track for software developers (google.com)