Improving software quality in bioinformatics through teamwork

Katalin Ferenc[1], Ieva Rauluseviciute[1], Ladislav Hovan[1], Vipin Kumar[1], Mariekie Kuijjer[1], and Anthony Mathelier[1,2,✉]

[1] Centre for Molecular Medicine Norway (NCMM), Nordic EMBL Partnership, University of Oslo, 0318 Oslo, Norway
[2] Department of Medical Genetics, Institute of Clinical Medicine, University of Oslo and Oslo University Hospital, Oslo, Norway

✉ Correspondence: Anthony Mathelier <anthony.mathelier@ncmm.uio.no>

SUPPLEMENTARY FILE

SUPPLEMENTARY METHODS

Following the standard methods of literature review, here we list the phrases and platforms of search. The literature search was performed in multiple iterations using Google (to include grey literature), PubMed and Google Scholar based on phrases “guidelines for bioinformatics software”, “rules for biologists learning bioinformatics”, “scientific software development”, “software engineering bioinformatics” and “bioinformatics software recommendations” throughout 2023. Additionally, relevant articles were selected based on the snowball effect from the references of the initial publications.

SUPPLEMENTARY FIGURES

Dependency management

There are two angles of dependency management we give example to here. First, we share a previous and current version of a code where the placing of the package imports are improved. This code also can be seen as an example for modularization with the rearrangement of the linear script to setup and functions. Furthermore, we improved the documentation and usability with using named arguments instead of positional ones. Second, we share an example of documenting the requirements where the responsibility of installing the software is moved from the user to the developer.

Modularization

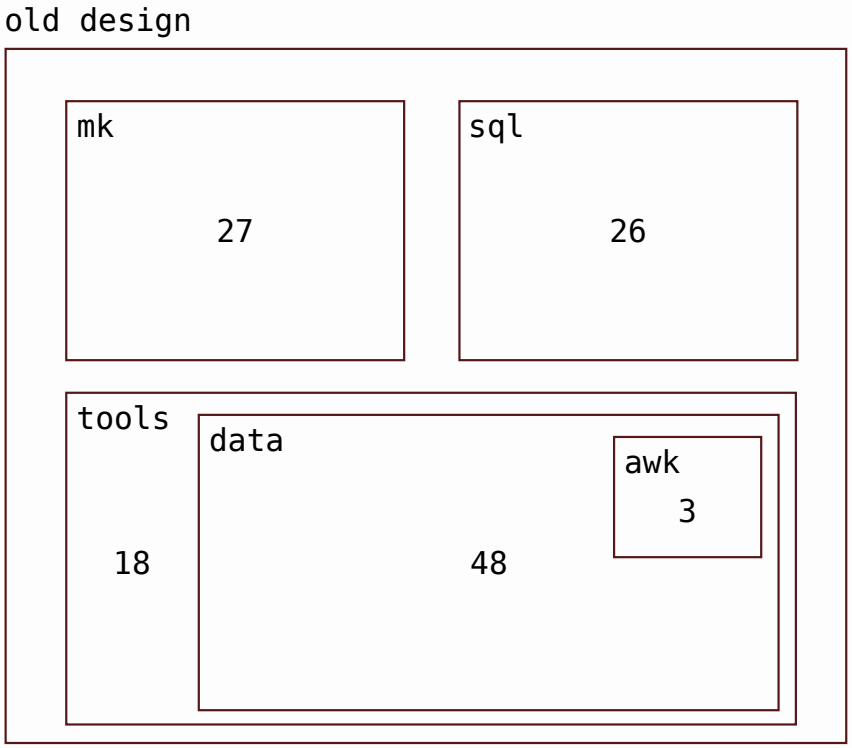

Figure 6: **Improving the modularization of a large codebase: previous.** In the previous design the files were arranged by on their type. The numbers denote the number of files in each directory represented by the rectangle. mk: makefile

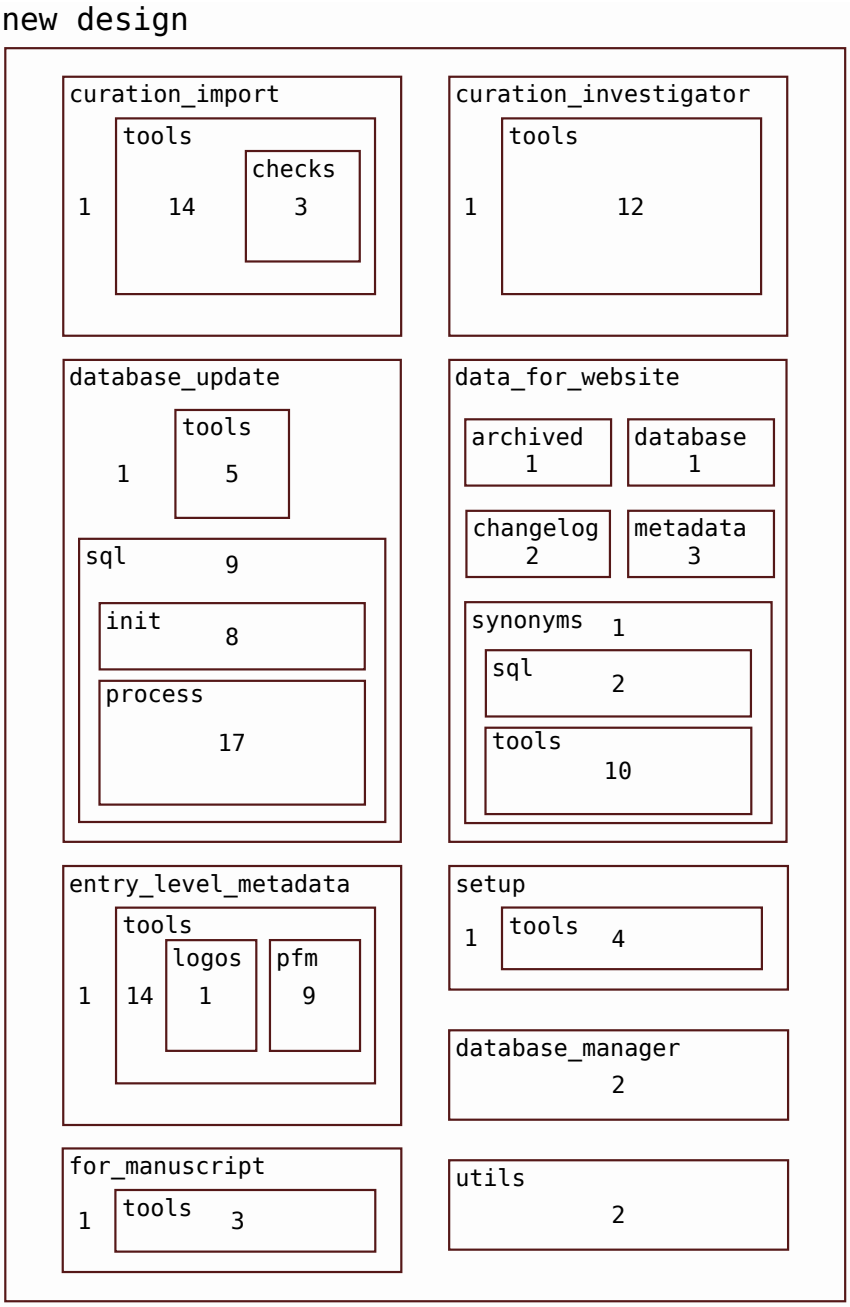

Figure 7: **Improving the modularization of a large codebase: current.** In the current design the files are arranged by their function. The numbers denote the number of files in each directory represented by the rectangle. The number of files are different due to added features and changes beyond the organization. pfm: position frequency matrix

SUPPLEMENTARY TABLES

TODO: SQ attributes description

Supplementary Table 2: **Examples of software quality meeting topics** This table contains examples of the topics of past software quality meetings. It has been organise to follow the same categories as **Table 1**.

Category	Title	Description
Software development 101	To be identified	To be filled
Advanced software development	Design patterns	To be filled
Software development process	Code review	To be filled
Testing and validation	Why testing?	To be filled
Reproducibility	Dependency management	To be filled
Documentation	On Pages and Reports	To be filled
Community effort	To be identified / we never covered it probably	To be filled