



King's Research Portal

Document Version
Peer reviewed version

[Link to publication record in King's Research Portal](#)

Citation for published version (APA):

Chapman, M., Rasmussen, L., Pacheco, J., & Curcin, V. (Accepted/In press). Using Version Control Systems to Support High-Quality Phenotype Definitions. In *American Medical Informatics Association (AMIA) Informatics Summit*

Citing this paper

Please note that where the full-text provided on King's Research Portal is the Author Accepted Manuscript or Post-Print version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version for pagination, volume/issue, and date of publication details. And where the final published version is provided on the Research Portal, if citing you are again advised to check the publisher's website for any subsequent corrections.

General rights

Copyright and moral rights for the publications made accessible in the Research Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the Research Portal

Take down policy

If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Using Version Control Systems to Support High-Quality Phenotype Definitions

Martin Chapman¹, Luke V. Rasmussen², Jennifer A. Pacheco², Vasa Curcin¹

¹King's College London, London, UK; ²Northwestern University, Chicago, Illinois, USA

Background and Problem Statement

High-quality phenotype definitions—which are reproducible, portable and valid—are essential in providing access to accurate disease cohorts for research and, going forward, will support clinical care. It was recently identified that the *libraries* in which definitions are stored are capable of contributing to their quality; so-called *next-generation* libraries¹. A library can contribute to the quality of its definitions by, for example, recording how they evolve over time (including the extension of existing definitions), increasing reproducibility. However, building these features into a library, or adding them to an existing one, is not a straightforward task.

Building a VCS-based Phenotype Library

A version control system (VCS) can support many of the features associated with a next-generation phenotype library (Table 1). However, to date, VCSs have mostly been used for storing and disseminating phenotype definitions—typically in individual repositories associated with a publication—with the most robust example of this being the OHDSI Phenotype Library².

We have updated an existing library, Phenoflow (<https://kclhi.org/phenoflow>), to better leverage VCS features by using GitHub as a remote VCS backend. Phenoflow uses the Common Workflow Language (CWL) to support multiple programming language implementations. Upon importing a new definition into the library, its CWL implementation is generated and stored in a GitHub repository, with subsequent updates tracking its version history.

One of Phenoflow's novel contributions is that it is built to support a canonical phenotype definition that can be connected to different data sources (e.g., OMOP, FHIR, or a local data warehouse)³. As such, permutations of the definitions (based on a target data source) are versioned along with it as separate *branches*. These branches retain a relationship between these versions, as they often have the same core logic. An abstract view of phenotype logic is retained by using *CWL Viewer* (<https://view.commonwl.org>). Definitions that target the same disease yet have entirely different logic (an increasingly common occurrence as definitions are produced for a wider variety of use cases) can also be compared and understood by computing the difference between multiple repositories.

Conclusions

We have extended the Phenoflow phenotype repository to leverage a VCS (GitHub) as its backend, demonstrating the benefits of VCSs over and above the simple versioning of phenotype definitions. In doing so, we are, in effect, building a 'GitHub for phenotypes', which is an essential step towards promoting phenotype definitions as first-class research objects which can be published, assigned DOIs, reused, and referenced, thereby improving the reproducibility and transparency of health data research.

References

- [1] Chapman M, Mumtaz S, Rasmussen L, Karwath A, Gkoutos GV, Thayer D, et al. Desiderata for the development of next-generation phenotype libraries. *Gigascience*. 2021;10(9):1-13.
- [2] Rao, Gowtham. The OHDSI Phenotype Library; 2022. <https://ohdsi.github.io/PhenotypeLibrary>, Last accessed on 2022-09-13.
- [3] Chapman M, Rasmussen LV, Pacheco JA, Curcin V. Connecting computable phenotypes with multiple Health IT Standards using the Phenoflow library. In: *AMIA Clinical Informatics Conference*; 2022. .

Table 1: VCS support for phenotype libraries

VCS feature	Library feature
Versioning	View definition evolution
Branching	Access multiple definition implementations (for multiple data sources).
Cross-repository <i>diffs</i>	View definition intersection (for the same disease).
Repository visualisation	View an <i>abstract</i> representation of a definition.
Code attribution	Understand contributions and author roles.
Submodules	View parent/child definition relationships.
Continuous Integration (CI)	Determine <i>technical validity</i> using unit tests; automate gold standard comparisons.