A tool for qualitative data analysis designed to support computational thinking

Chris Proctor [1]

1 Graduate School of Education, University at Buffalo (SUNY), United States

Summary

qc is a free, open-source command-line-based tool for qualitative data analysis designed to support computational thinking. In addition to making the qualitative data analysis process more robust and efficient, computational thinking can contribute to the richness of subjective interpretation. The typical workflow in qualitative research is an iterative cycle of "notice things," "think about things," and "collect things" (Seidel, 1998, p. 2). qc provides computational affordances for each of these practices, including the ability to integrate manual coding with automated coding, a tree-based hierarchy of codes stored in a YAML file, allowing versioning of thematic analysis, and a powerful query interface for viewing code statistics and snippets of coded documents.

Background

qc is designed to support the application of computational thinking (CT) to qualitative data analysis (QDA). In the social sciences, QDA is a method of applying codes to text, images, video, and other artifacts, then analyzing the resulting patterns of codes and using the codes to more deeply understand the text. When QDA is used in quantitative or mixed-methods research, it is typically used to transform loosely-structured data such as an interview transcript into categories or codes which can then be used in downstream quantitative analysis answering predefined research questions. In contrast, when QDA is used in qualitative research, it is typically part of an interpretive sensemaking process. These two uses of QDA have been referred to as *little-q* ("looking for answers") and *big-Q* ("looking for questions") qualitative research (Kidder & Fine, 1987).

The central design hypothesis of qc is that a closer partnership between the researcher and the computational tool can enhance the quality of QDA. This partnership, which could be characterized as augmented (Engelbart, 1962) or distributed cognition (Pea, 1997), depends on the researcher's ability to conceptualize the data and the process in computational terms, becoming immersed in the matrices, trees, and other computational structures inherent to QDA rather than remaining "outside" at the level of user interface. Such practices can be identified as *computational thinking* (CT), "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can effectively be carried out by an information-processing agent" (Wing, 2011). The application of CT to QDA would mean conceptualizing the goal and the process of QDA in computational terms, keeping a mental model of the work the computer is doing for you.

Statement of need

Although there are numerous well-known commercial QDA software packages such as NVivo (Dhakal, 2022), Dedoose (Salmona et al., 2019), ATLAS.ti (Smit, 2002), and MAXQDA

(Kuckartz, 2010), they do not provide affordances for users desiring more active engagement with the data and processes underlying QDA. qc better-supports such users, providing a scriptable command-line interface with powerful and flexible queries, what data stored in simple and standardized formats. qc adopts the "unix philosophy" (McIlroy et al., 1978) of building tools which do one thing well while being composable into flexible workflows, and the values of "plain-text social science" (Healy, 2020), emphasizing reproducability, transparency, and collaborative open science.

qc was used in (Proctor et al., 2019) (described but not cited) and the author's doctoral dissertation; qc is currently a core tool supporting a large NSF-funded Delphi study (Ogbeifun et al., 2016) involving multiple interviews with forty participant experts, open coding with over a thousand distinct codes, four separate coders, and several custom machine learning tools supporting the research team with clustering and synthesizing emergent themes.

Acknowledgements

References

Dhakal, K. (2022). NVivo. *Journal of the Medical Library Association: JMLA*, *110*(2), 270.

Engelbart, D. C. (1962). *Augmenting human intellect: A conceptual framework*. 64–90. https://doi.org/10.21236/ad0289565

Healy, K. (2020). *The Plain Person's Guide to Plain Text Social Science*.

Kidder, L. H., & Fine, M. (1987). Qualitative and quantitative methods: When stories converge. *New Directions for Program Evaluation*, *1987*(35), 57–75. https://doi.org/10.1002/ev.1459

Kuckartz, U. (2010). Realizing mixed-methods approaches with MAXQDA. *Philipps-Universität, Marburg*.

McIlroy, D., PInson, E., & Tague, B. (1978). UNIX time-sharing system. *The Bell System Technical Journal*, 1902–1903.

Ogbeifun, E., Agwa-Ejon, J., Mbohwa, C., & Pretorius, J. (2016). *The Delphi technique: A credible research methodology*.

Pea, R. D. (1997). Practices of distributed intelligence and designs for education. In G. Salomon (Ed.), *Distributed cognitions: Psychological and educational considerations* (pp. 47–87). Cambridge University Press. ISBN: 978-0-521-41406-7

Proctor, C., Bigman, M., & Blikstein, P. (2019). Defining and designing computer science education in a K12 public school district. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 314–320. https://doi.org/10.1145/3287324.3287440

Salmona, M., Lieber, E., & Kaczynski, D. (2019). *Qualitative and mixed methods data analysis using Dedoose: A practical approach for research across the social sciences*. Sage Publications.

Seidel, J. V. (1998). *Qualitative data analysis*.

Smit, B. (2002). ATLAS.ti for qualitative data analysis. *Perspectives in Education*, *20*(3), 65–75.

Wing, J. (2011). Research notebook: Computational thinking—What and why. *The Link Magazine*, *6*, 20–23.