

Beyond nodegoat: a critical look at historical network research workflows

Pim Van Bree [1], Geert Kessels [1]

1: Lab1100

Van Bree Pim and Kessels Geert. 2024. "Beyond nodegoat: a critical look at historical network research workflows", *Historical Network Research* 2024, Lausanne, DOI: 10.5281/zenodo.12606364

Data exchanges between data collection and storage software on the one hand and data analysis and visualisation software on the other hand remain a largely static process.³⁸ Scholars store and work on their data in one tool, and generate an export of this data to analyse and visualise it in another tool.³⁹ While specialised tools make for a straightforward field-specific user experience, the strict divide between the working environment and the analysis software hampers many opportunities offered by exploratory data analysis and visualisation.

The open-source software nodegoat offers scholars a research environment that can be used for data modelling purposes, multi-user data collection tasks, as well as various data analysis and visualisation functionalities.⁴⁰ While this environment presents scholars with features for exploratory data analysis and visualisation, these functionalities are not exhaustive. For this reason, exports can be made of all data as CSV files⁴¹, and every nodegoat research environment is equipped with an API.⁴²

In this paper, we describe the opportunities of working with data collection and storage software that enables interactive data exchanges as opposed to working with static exports, based on the example of nodegoat. For this workflow a data collection and storage tool is required that offers any kind of web API (e.g. the nodegoat REST API, the SPARQL endpoint of Wikibase) and data analysis or visualisation software that is able to communicate with a web API. Exposing data via an API does not necessarily mean that data becomes publicly available: APIs can be configured to require authentication.

A nodegoat API allows you to expose a complete dataset or a subset of the dataset.⁴³ The exposed data can be generated based on its default configuration or with a custom configuration. This allows you to configure various 'views' on your data that are able to satisfy the needs of the analyses to be performed, and that is independent of the state of the data itself. These views allow for live preprocessing of data, which includes

³⁸ Alex Brey, "Temporal Network Analysis with R," *Programming Historian* 7 (2018), <https://doi.org/10.46430/phen0080>, Peeples, Matthew A. and Tom Brughmans (2023). *Online Companion to Network Science in Archaeology*. <https://archnetworks.net>, Accessed 2024-01-29.

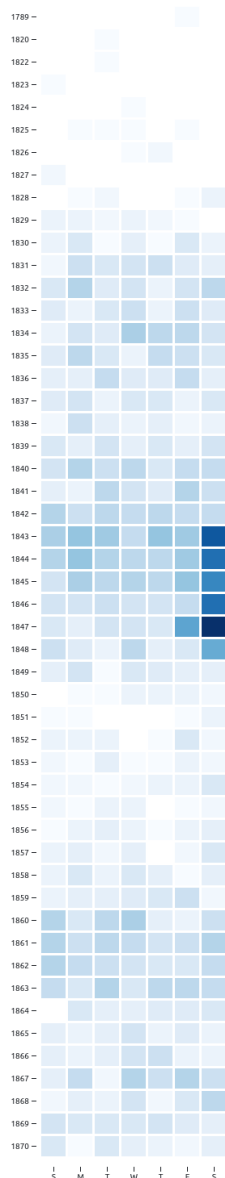
³⁹ Marten Düring, "From Hermeneutics to Data to Networks: Data Extraction and Network Visualization of Historical Sources," *Programming Historian* 4 (2015), <https://doi.org/10.46430/phen0044>.

⁴⁰ Pim van Bree, Geert Kessels (2013). nodegoat: a web-based data management, network analysis & visualisation environment, <http://nodegoat.net> from LAB1100, <http://lab1100.com>. For an introduction see <https://nodegoat.net/guide.s/112/basicprinciples>. For more advanced analysis and visualisation functionalities see, for example: Pim van Bree, Geert Kessels, 'New Analytical Features in nodegoat', 13th Workshop on Historical Network Analysis 'Networks Across Time and Space', 27-5-2019, <https://nats.hypotheses.org/> and Pim van Bree, Geert Kessels, 'Temporally-aware dynamic network analysis: traversing nodegoat graphs', 19-07-2023, <https://graphentechnologien.hypotheses.org/files/2023/05/GraphHNR-2023-30-Kessels-Temporally-Aware.pdf>.

⁴¹ See <https://nodegoat.net/guide.s/146/export-data-as-a-csv-file> and <https://nodegoat.net/guide.s/148/export-data-to-gephi>.

⁴² <https://nodegoat.net/documentation.s/98/query>.

⁴³ A Clariana-Rodagut, A Cardillo, Quantifying women marginalisation in Ibero-American film culture during the first half of XX century: a quantitative proposal based on network science, arXiv preprint, <https://doi.org/10.48550/arXiv.2307.13137>. APIs also allow for the dynamic integration of research data in external applications, see: <https://streetlife.amsterdamtimemachine.nl/>.



edge generation based on multi-modal graphs and the ability to apply dynamic conditions.⁴⁴ Data entry and curation continues while forms of analysis and visualisation are being tested and explored.

By using query parameters, any data selection can be transformed from the calling analysis and visualisation software. Filters and scopes that can be configured within the nodegoat working environment, can also be configured by means of the API. This essentially bridges the gap between the data storage and data analysis tool as the latter is able to communicate with the former. By using the PATCH method of the nodegoat API, results of the analysis can also be sent back to the data store for filtering, weighting, and conditioning purposes.⁴⁵

Programming languages that are well equipped to perform data analysis and data visualisation operations like R, Python, and JavaScript come with built in modules to interactively query web APIs.⁴⁶ Development environments like RStudio⁴⁷, Jupyter Notebook⁴⁸, and Observable⁴⁹ provide scholars with entry level examples and rich documentation, see figure 1.⁵⁰ A specialised tool like QGIS is a good example of visualisation software that can communicate interactively with an API.⁵¹

The open-endedness of this approach is also reflected in the tools that can be used for the storage of research data. Any database that offers a web API has the ability to be used in an integrated workflow. Other examples of database software that can be used in this manner include Wikibase⁵² and Numishare.⁵³ If the use of spreadsheet software is unavoidable, the Google Sheets API allows for an integration of the gathered research data into a more dynamic workflow.⁵⁴

A workflow in which research data and exploratory modes of analysis and visualisation are closely integrated brings many advantages to methods associated with historical network research. First of all, it gives scholars the opportunity to think about actionable data in the course of their research project. This contrasts with the unwelcome realisation at the end of a project that a date statement such as 'around 1680' or a location statement formulated as 'Springfield' are computationally unsound.

Figure 1. This visualisation has been generated in an interactive notebook of the data exploration platform Observable and runs on a request to a nodegoat API. The visualisation displays the amount of letters sent by the French writer Prosper Mérimée per year per day of the week and shows that he wrote, dated, or posted his letters mostly on Saturdays, see: <https://observablehq.com/@lab1100/nodegoat-demo-daily-intensity-of-letters-interactive>.

⁴⁴ Pim van Bree, Geert Kessels, 'Temporally-aware dynamic network analysis: traversing nodegoat graphs', 19-07-2023, <https://graphentechnologien.hypotheses.org/files/2023/05/GrapHNR-2023-30-Kessels-Temporally-Aware.pdf>.

⁴⁵ <https://nodegoat.net/documentation.s/103/store#patch>.

⁴⁶ For R see e.g. <https://httr2.r-lib.org/>, for Python see e.g. <https://pypi.org/project/requests/>, for JavaScript see e.g. https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API.

⁴⁷ <https://posit.co/download/rstudio-desktop/>.

⁴⁸ <https://jupyter.org/>.

⁴⁹ <https://observablehq.com/>.

⁵⁰ For R see e.g. <https://www.tidyverse.org/blog/2023/11/httr2-1-0-0/>, for Python see e.g. <https://medium.com/swlh/using-and-calling-an-api-with-python-494a18cb1f44>, for JavaScript see e.g. <https://nodegoat.net/guide.s/150/export-data-to-observable>.

⁵¹ <https://nodegoat.net/guide.s/149/export-data-to-qgis>.

⁵² <https://www.mediawiki.org/wiki/Wikibase/Installation>

⁵³ <https://github.com/ewg118/numishare>

⁵⁴ <https://developers.google.com/sheets/api/guides/concepts>

Next, a more dynamic workflow allows for the testing of hypotheses or case studies early on, without the need to invest time in data export and data cleaning processes. Integrated workflows can also signify new questions that result from exploratory data visualisations during the course of a research project. Unexpected 'gaps' or clusters can be identified quickly and can offer new directions of research as the project progresses. Unsound data models, incorrectly configured attributes, and missing relationships come to light at a much earlier stage in an integrated workflow. These advantages improve multiple aspects of any historical network research project, and allow for a general application of source criticism and digital hermeneutics.⁵⁵

This paper steers away from a workflow that rigidly separates data from exploratory modes of analysis and visualisation. Many questions, pitfalls, failures, and perhaps even insights can be discovered if workflows are implemented with a closer integration between data collection and data analysis practices.

⁵⁵ Fickers, Andreas, Juliane Tatarinov, and Tim van der Heijden. "Digital history and hermeneutics—between theory and practice: An introduction." *Digital History and Hermeneutics Between Theory and Practice*. Berlin, Boston: De Gruyter (2022): 1-22, p. 8.