The Journal of Open Source Software

scribl: A system for the semantic capture of relationships in biological literature

Gordon D. Webster [1,2*] **and Alexander K. Lancaster** [1,2,3*]

1 Amber Biology LLC, USA **2** Ronin Institute, USA **3** Institute for Globally Distributed Open Research and Education * These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 14 March 2024
Published: unpublished

Summary

When using literature databases, researchers in systems biology need to go beyond simple keyword-based queries of biological agents (e.g., proteins, genes, compounds, receptor complexes) and processes (e.g, autophagy, cell cycle) (Krallinger et al., 2008) that return lists of articles, to extracting and visualizing relationships documented within those papers (Cary et al., 2005; Pavlopoulos et al., 2015; Suderman & Hallett, 2007). Here we describe a system that supports the annotation of scientific articles, that represent and visualize these relationships. This system, `scribl`, consists of two parts: (1) a simple syntax that can be used to curate the biological relationships described within the text of those articles, (2) a Python software API and pipeline that can transform a Zotero literature database, with entries annotated with this syntax, into a database suitable graph-based relationship queries.

The `scribl` language

The language was designed for the curation of scientific articles to document the relationships between the various biological agents and processes that they describe. Examples of relationships for each of the five basic entities for a single article are shown in Figure 1.

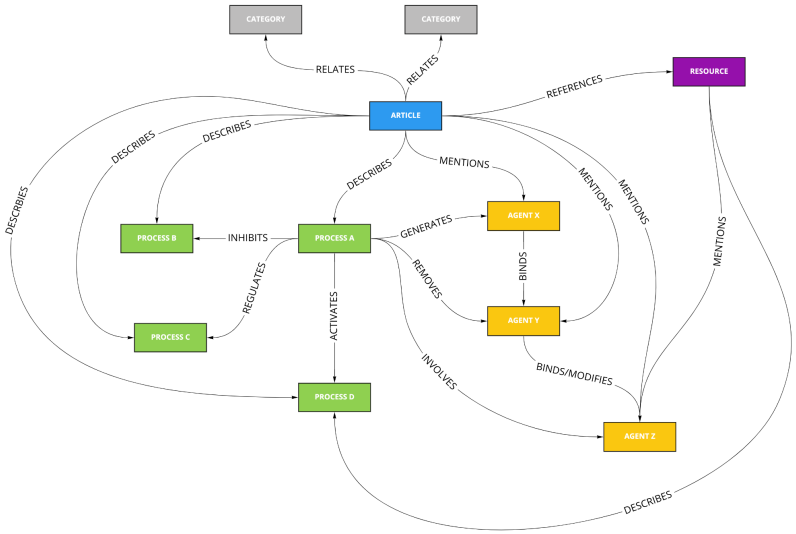

Figure 1: The scribl schema comprises a hierarchy of five basic entities: `article`, `category`, `resource`, `process`, and `agent`. Here we depict an example network of entities and possible relationships for a single `article`.

A curator can add `scribl` statements as tags to each article in a literature database, to represent aspects of the causal relationships that are described in the article.

Table 1: Example scribl statements included in Zotero tags

```
::agent c9orf72 :gene :protein :url https://www.uniprot.org/uniprot/Q96LT7
::agent gtp :tag nucleoside, purine, nucleoside triphosphate
::process exportin releases cargo into cytoplasm @ exportin-1
::process smcr8 mutation > ulk1 phosphorylation < autophagy = smcr8 expression
```

Table 1 shows two types of entities: (1) agents (`::agent`): are actual biochemical entities (e.g. proteins) described in the literature article in question, along with some metadata about the agent, (2) processes (`::process`) which represent broad mechanistic, or phenomenological biological processes (e.g., `autophagy`).

The `scribl` Python package

The `scribl` Python package provides an API to query a Zotero database where each literature record has been annotated using declarative statements in the `scribl` syntax described in Table 1. Currently, the literature source for `scribl` input can be either a remote Zotero database, or a file export from a local Zotero installation. Once the Zotero data has been parsed, the resulting graph data structure can be then be exported for use in graph database platforms. `scribl` also supports the incremental updating of the graph database as new Zotero entries come in (Figure 2).

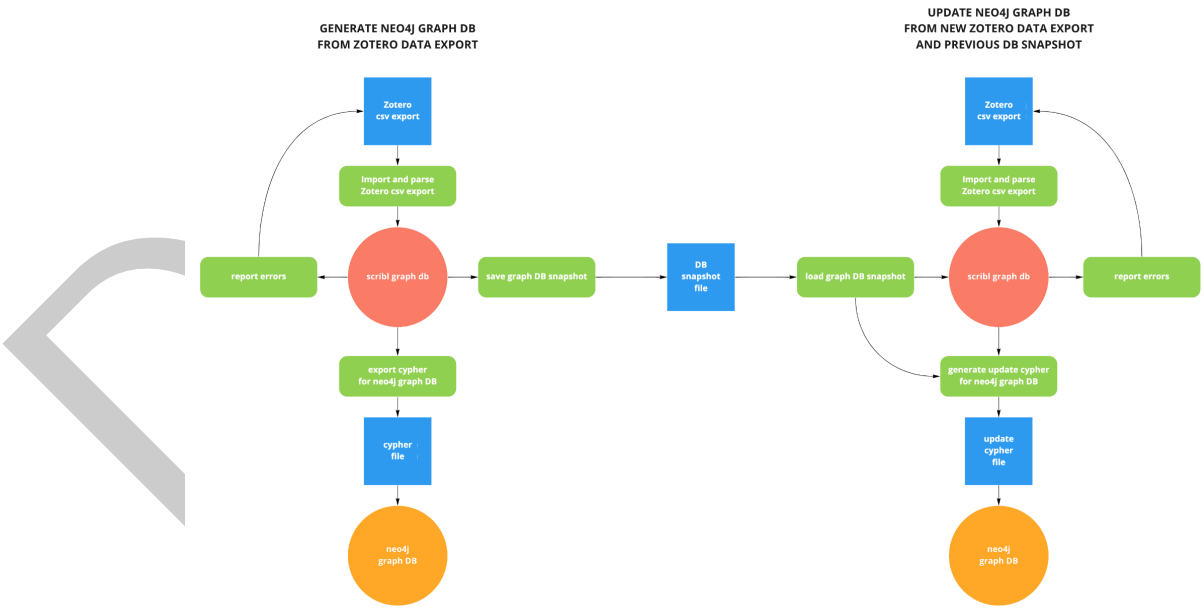

Figure 2: Two major workflows for the `scribl` software: creating a new graph database (left) and updating an existing one (right). The workflow contains a step that identifies possible syntactic errors in `scribl` statements so that they can be fixed in the Zotero database before database generation. Note that "Zotero csv export" could be replaced by a query to a remote Zotero library

`scribl` functions can be accessed programatically through writing a Python script that calls the `scribl` API, or via a command-line program `scribl` contained within the distribution. `scribl` currently supports output in one of two graph formats:

1. Cypher query language (Francis et al., 2018) used by the graph database platform neo4j. The output Cypher query text can be used directly to initialize a Neo4j database. The Neo4j setup itself is not automated by `scribl` and must be installed separately.

2. GraphML (Brandes et al., 2002) format that can be read and used for processing and visualization by packages such as Python's NetworkX (Hagberg et al., 2008). `scribl` can generate visualizations from GraphML output, directly (e.g, Figure 3).

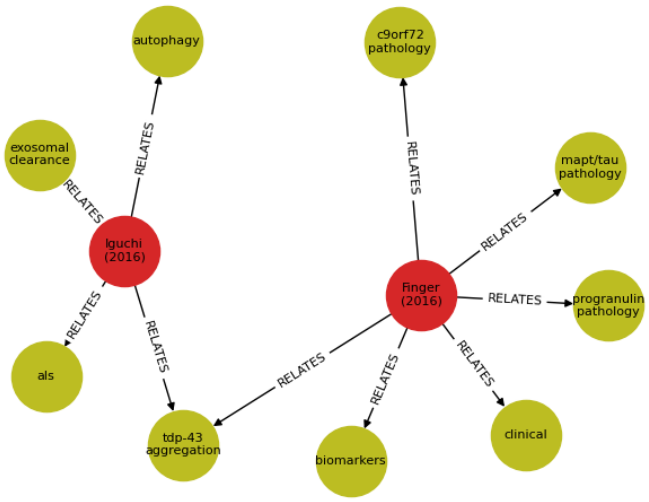

Figure 3: NetworkX visualization of a graph database exported as GraphML, generated directly by `scribl`.

Once a graph database has been created, queries can be created that are not possible with traditional keyword searching. For example, once the scribl output is loaded into a Neo4j database, it is possible to write Cypher queries of the kind: "Show me all of the agents that are involved in the process `nuclear export` along with the articles that describe them".

Statement of need

Why `scribl`?

The `scribl` platform was developed to fill a need for a simple way to enable global sharing and collaborative curation of biological relationships embedded in literature records, and to rapidly translate those relationships into queryable graph networks. The `scribl` syntax was designed to be simple to learn, but rich enough to represent important relationships relevant to molecular and systems biology.

Zotero was chosen as the initial backend, because it is simple to install and run, as well as supporting the tagging of literature records and web-based curation. `scribl` allows a researcher or group of researchers, to rapidly build and visualize important relationships useful for understanding the celluar and systems biology within a chosen subdomain. In fact our main use-case for `scribl` was the building of a relationship database of neurodegenerative disease pathways for the frontotemporal degeneration (FTD) research community.

What `scribl` is not

`scribl` is not primarily intended for the construction of formal, kinetic models of biological systems in the way that modeling languages such as Kappa (Boutillier et al., 2020) and SBML

(Keating et al., 2020) are. However, these networks can be considered a coarse-grained model of biological systems that sit somewhere between low resolution, keyword-based representations; and high resolution, formal, kinetic models. `scribl`-enabled networks may also help researchers identify interactions or parameters that require measurement in order to build those detailed models.

`scribl` is also not intended to be a replacement for biological graph databases such as Reactome (Gillespie et al., 2022). The Reactome database is actually based upon the same Neo4j graph database engine supported by scribl, so `scribl` could actually help facilitate the curation of biological pathways from newly-published literature, in a format that is ready for graph data repositories like Reactome.

Availability

`scribl` is available as a package on PyPI with the source code and documentation available at https://github.com/amberbiology/scribl.

Acknowledgements

The development of the scribl platform was made possible with the support of the Association for Frontotemporal Degeneration (AFTD). We are grateful to AFTD members Debra Niehoff and Penny Dacks for their support.

References

Boutillier, P., Feret, J., Krivine, J., & Fontana, W. (2020). *The Kappa Language and Kappa Tools*. https://kappalanguage.org/

Brandes, U., Eiglsperger, M., Herman, I., Himsolt, M., & Marshall, M. S. (2002). GraphML Progress Report Structural Layer Proposal. In P. Mutzel, M. Jünger, & S. Leipert (Eds.), *Graph Drawing* (pp. 501–512). Springer. https://doi.org/10.1007/3-540-45848-4_59

Cary, M. P., Bader, G. D., & Sander, C. (2005). Pathway information for systems biology. *FEBS Letters*, *579*(8), 1815–1820. https://doi.org/10.1016/j.febslet.2005.02.005

Francis, N., Green, A., Guagliardo, P., Libkin, L., Lindaaker, T., Marsault, V., Plantikow, S., Rydberg, M., Selmer, P., & Taylor, A. (2018). Cypher: An Evolving Query Language for Property Graphs. *Proceedings of the 2018 International Conference on Management of Data*, 1433–1445. https://doi.org/10.1145/3183713.3190657

Gillespie, M., Jassal, B., Stephan, R., Milacic, M., Rothfels, K., Senff-Ribeiro, A., Griss, J., Sevilla, C., Matthews, L., Gong, C., Deng, C., Varusai, T., Ragueneau, E., Haider, Y., May, B., Shamovsky, V., Weiser, J., Brunson, T., Sanati, N., … D'Eustachio, P. (2022). The reactome pathway knowledgebase 2022. *Nucleic Acids Research*, *50*(D1), D687–D692. https://doi.org/10.1093/nar/gkab1028

Hagberg, A., Swart, P. J., & Schult, D. A. (2008). *Exploring network structure, dynamics, and function using NetworkX* (LA-UR-08-05495; LA-UR-08-5495). Los Alamos National Laboratory (LANL), Los Alamos, NM (United States). https://www.osti.gov/biblio/960616

Keating, S. M., Waltemath, D., König, M., Zhang, F., Dräger, A., Chaouiya, C., Bergmann, F. T., Finney, A., Gillespie, C. S., Helikar, T., Hoops, S., Malik-Sheriff, R. S., Moodie, S. L., Moraru, I. I., Myers, C. J., Naldi, A., Olivier, B. G., Sahle, S., Schaff, J. C., … Zucker, J. (2020). SBML Level 3: An extensible format for the exchange and reuse of biological models. *Molecular Systems Biology*, *16*(8), e9110. https://doi.org/10.15252/msb.20199110

Krallinger, M., Valencia, A., & Hirschman, L. (2008). Linking genes to literature: Text mining, information extraction, and retrieval applications for biology. *Genome Biology*, *9*(2), S8. https://doi.org/10.1186/gb-2008-9-s2-s8

Pavlopoulos, G. A., Malliarakis, D., Papanikolaou, N., Theodosiou, T., Enright, A. J., & Iliopoulos, I. (2015). Visualizing genome and systems biology: Technologies, tools, implementation techniques and trends, past, present and future. *GigaScience*, *4*(1), s13742-015-0077-2. https://doi.org/10.1186/s13742-015-0077-2

Suderman, M., & Hallett, M. (2007). Tools for visually exploring biological networks. *Bioinformatics*, *23*(20), 2651–2659. https://doi.org/10.1093/bioinformatics/btm401