

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

Gordon D. Webster ^{1*} and Alexander K. Lancaster ^{1,2*}

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

DOI: [10.21105/joss.06645](https://doi.org/10.21105/joss.06645)

Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: [Ana Trisovic](#)

Reviewers:

- [@benlansdell](#)
- [@mhucka](#)

Submitted: 14 March 2024

Published: 17 July 2024

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

In systems biology research, utilizing literature databases involves more than simple keyword queries for biological agents (e.g., proteins, genes, compounds, receptor complexes) and processes (e.g. autophagy, cell cycle) (Krallinger et al., 2008), which typically only return lists of articles. Advanced methods are necessary for extracting and visualizing the relationships detailed within these documents (Cary et al., 2005; Pavlopoulos et al., 2015; Suderman & Hallett, 2007). Here, we introduce a system that supports the annotation of scientific articles and represents and visualizes 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 for graph-based relationship queries.

The scribl language

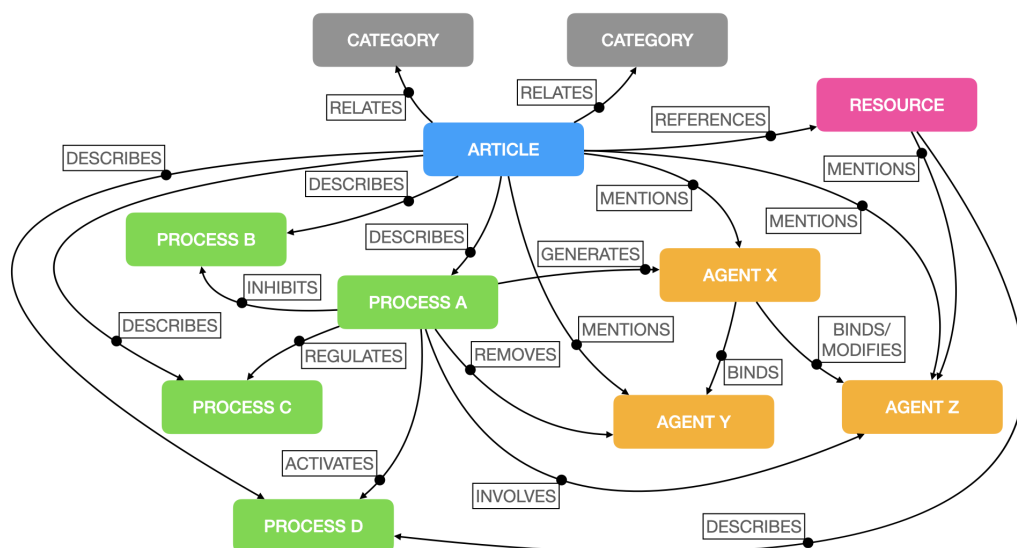


Figure 1: The scribl schema comprises five basic entities: article, category, resource, process, and agent. Here we depict an example network of entities and possible relationships for a single article.

The language was designed for the curation of scientific articles to document the relationships between 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](#).

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 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](#)).

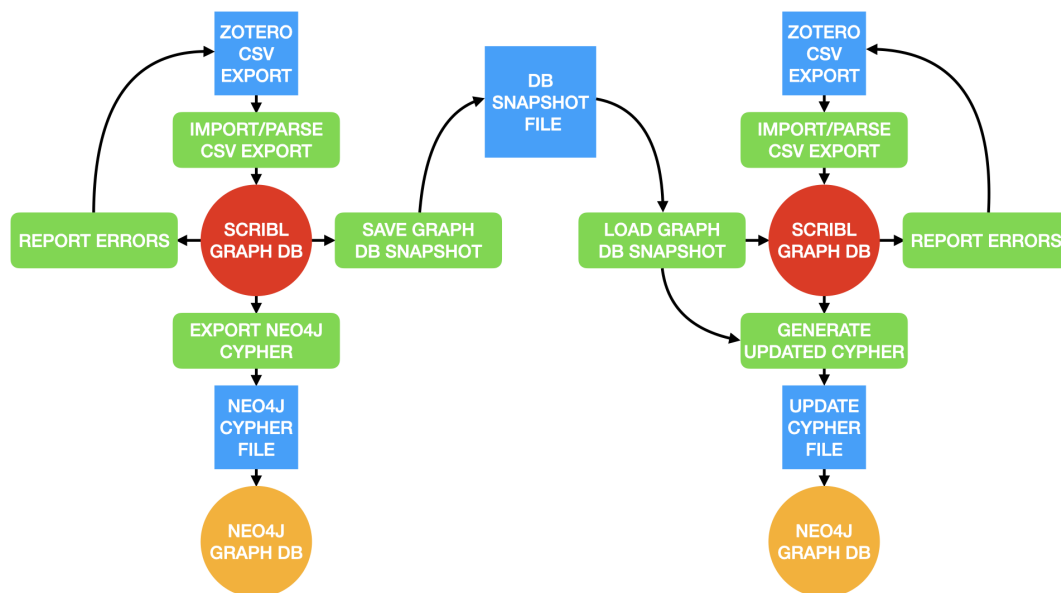


Figure 2: Two major workflows for the scribl software: creating a new graph database (left) and updating an existing one (right). The workflow contains a step that identifies possible syntactic errors in scribl statements so that they can be fixed in the Zotero database before database generation. Note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

scribl functions can be accessed programmatically by writing a Python script that calls the scribl API, or via the included command-line program scribl.

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). The scribl command-line program can generate visualizations (e.g, [Figure 3](#)) from GraphML output, and from an input Zotero file in CSV format directly, a basic example of which is shown below:

```
scribl -g new_graphdb -z zotero.csv --networkx-fig graphdb-visual.png
```

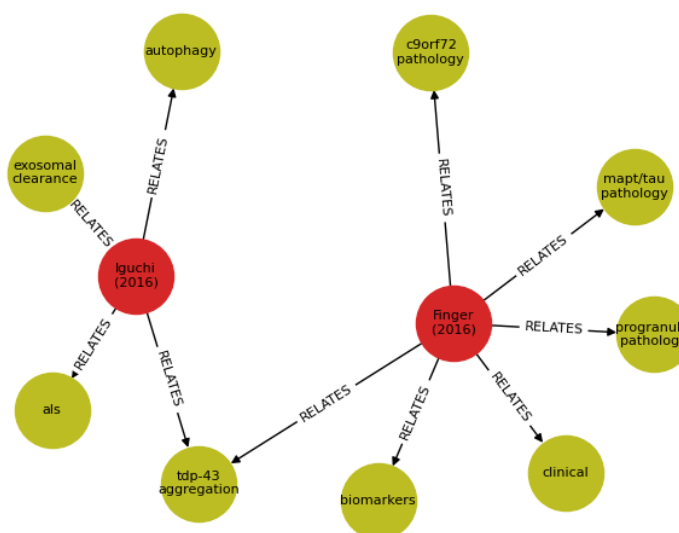


Figure 3: NetworkX visualization of a graph database exported as GraphML, generated directly by scribl.

Once a graph database has been created, it can be queried with prompts that go beyond the capabilities of traditional keyword searches. For example, once the scribl output is loaded into a Neo4j database, it is possible to write Cypher queries such as: “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 cellular 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, and in-principle, scribl could be extended to directly generate models in Kappa or SBML format for the subset of entries with sufficient kinetic annotations to form a self-contained network.

scribl is also not intended to be a replacement for an interactive visualization engine such as Cytoscape (Shannon et al., 2003), in fact there are plugins to Cytoscape that allow the import of both the Neo4j and GraphML formats that scribl produces. Nor is it a substitute for biological graph databases such as Reactome (Gillespie et al., 2022). The Reactome database is actually based upon the Neo4j graph database engine, 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>
- Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., & Ideker, T. (2003). Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research*, *13*(11), 2498–2504. <https://doi.org/10.1101/gr.1239303>
- Suderman, M., & Hallett, M. (2007). Tools for visually exploring biological networks. *Bioinformatics*, *23*(20), 2651–2659. <https://doi.org/10.1093/bioinformatics/btm401>