

HyperNetX: A Python package for modeling complex network data as hypergraphs

Brenda Praggastis¹, Sinan Aksoy¹, Dustin Arendt¹, Mark Bonicillo¹,
Cliff Joslyn¹, Emilie Purvine¹, Madelyn Shapiro¹, and Ji Young Yun¹

¹ Pacific Northwest National Laboratory, USA

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

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

Editor: [Daniel S. Katz](#)

Reviewers:

- [@szhorvat](#)
- [@IvanIsCoding](#)
- [@drj11](#)

Submitted: 16 October 2023

Published: unpublished

License

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

Summary

HyperNetX (HNX) is an open source Python library for the analysis and visualization of complex network data modeled as hypergraphs. Initially released in 2019, HNX facilitates exploratory data analysis of complex networks using algebraic topology, combinatorics, and generalized hypergraph and graph theoretical methods on structured data inputs. With its 2023 release, the library supports attaching metadata, numerical and categorical, to nodes (vertices) and hyperedges, as well as to node-hyperedge pairings (incidences). HNX has a customizable Matplotlib-based visualization module as well as HyperNetX-Widget, its JavaScript addon for interactive exploration and visualization of hypergraphs within Jupyter Notebooks. Both packages are available on GitHub and PyPI. With a growing community of users and collaborators, HNX has become a preeminent tool for hypergraph analysis.

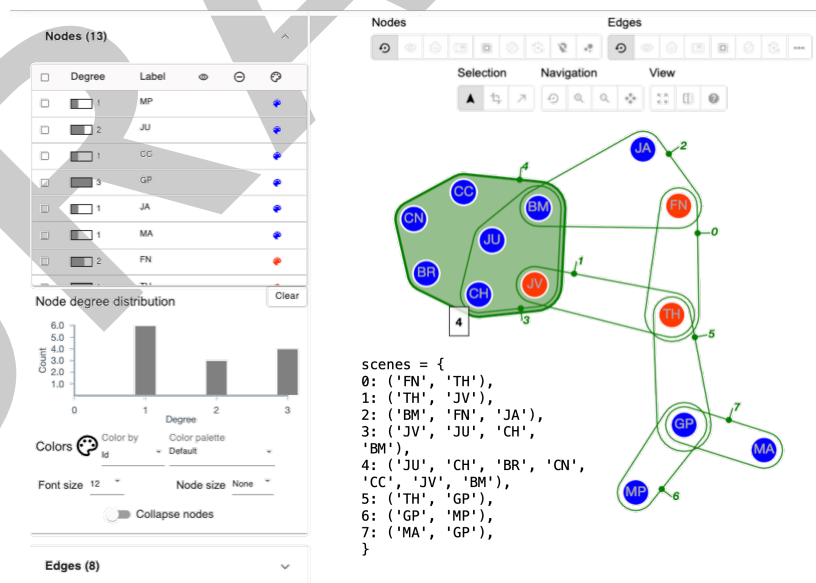


Figure 1: HNX-Widget is an add-on for the Jupyter Notebook computational environment, enabling users to view and interactively explore hypergraphs. The main features of the tool are: 1) adjustable layout 2) advanced selection and 3) visual encoding of node and edge properties. Metadata may be attached to the tool by providing tabular data via two optional data frames indexed by node and hyperedge identifiers. Above is an HNX-Widget visualization of a Scene to Character mapping from the LesMis dataset ([Knuth, 1993](#)).

Statement of need

For more than a century, graph theory has provided powerful methods for studying network relationships among abstract entities. Since the early 2000's, software packages such as NetworkX (Hagberg et al., 2008) and Igraph (Csardi et al., 2006) have made these theoretical tools available to data scientists for studying large data sets. Graphs represent pairwise interactions between entities, but for many network datasets this is a severe limitation. In 1973, hypergraphs were introduced by Claude Berge (Berge, 1973) as a strict generalization of graphs: a hyperedge in a hypergraph can contain any number of nodes, including 1, 2, or more. Hypergraphs have been used to model complex network datasets in areas such as the biological sciences, information systems, the power grid, and cyber security. Hypergraphs strictly generalize graphs (all graphs are (2-uniform) hypergraphs), and thus can represent additional data complexity and have more mathematical properties to exploit (for example, hyperedges can be contained in other hyperedges). As mathematical set systems, simplicial and homological methods from Algebraic Topology are well suited to aid in their analysis (Cliff A. Joslyn et al., 2021; Torres et al., 2021). With the development of hypergraph modeling methods, new software was required to support experimentation and exploration, which prompted the development of HyperNetX.

Related Software

Due to the diversity of hypergraph modeling applications, hypergraph software libraries are often bootstrapped using data structures and methods most appropriate to their usage. A nice compendium of many of the hypergraph libraries created in the last decade can be found in Kurte et al. (2021). HNX leads the effort to share library capabilities by specifying a Hypergraph Interchange Format (HIF) for storing hypergraph data as a JSON object. Since hypergraphs can store metadata on its nodes, hyperedges, and incidence pairs, a standardized format makes it easy to share hypergraphs across libraries.

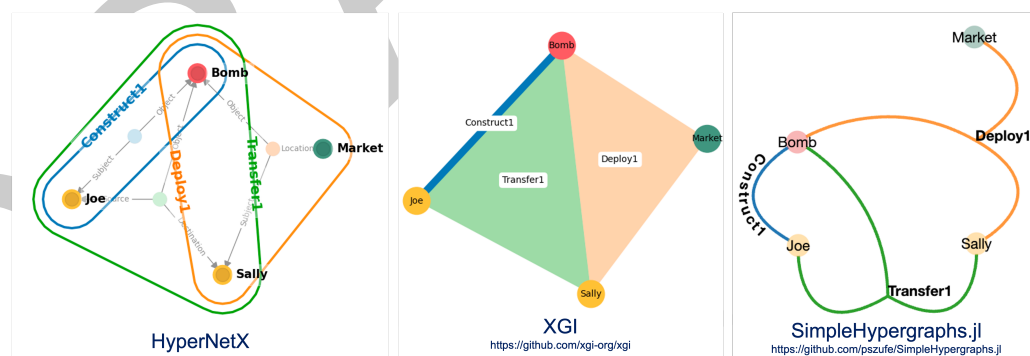


Figure 2: Visualizations from hypergraph libraries based on the bipartite graph seen in grey under the HyperNetX visualization (left side): XGI (Center), Landry et al. (2023) and SimpleHypergraph (Right), Antelmi et al. (2019).

Overview of HNX

HNX serves as a platform for the collaboration and sharing of hypergraph methods within the research community. Originally intended to generalize many of the methods from NetworkX to hypergraphs, HNX now has implementations for many hypergraph-specific metrics. While graph paths can be measured by length, hypergraph paths also have a width parameter s , given by the minimum intersection size of incident hyperedges in the path (Aksoy et al., 2020). HNX uses this s parameter in many of its core methods as well as in its s -centrality module. As set systems, hypergraphs can be viewed as subsets of abstract simplicial complexes – combinatorial

projections of geometric objects constructed from points, line segments, triangles, tetrahedrons, and their higher dimensional analogues. HNX's Simplicial Homology module identifies and computes the *voids* of different dimensions in the simplicial complexes generated by modestly sized hypergraphs. These objects, which are used for defining the *Homology Groups* studied by Algebraic Topologists, offer new metrics for exploratory data science.

As a collaborative platform, HNX contains contributed modules and tutorials in the form of Jupyter notebooks for Laplacian clustering, clustering and modularity, synthetic generation of hypergraphs, and Contagion Theory. In its latest release, HNX 2.0 uses Pandas dataframes (McKinney, 2010; The pandas development team, 2020) as its underlying data structure, making the nodes and hyperedges of a hypergraph as accessible as the cells in a dataframe. This simple design allows HNX to import data from semantically loaded graphs such as property graphs and knowledge graphs, in order to model and explore their higher order relationships. Because it is open source, HNX provides a unique opportunity for hypergraph researchers to implement their own methods built from HNX and contribute them as modules and Jupyter tutorials to the HNX user community.

Projects using HNX

HNX was created by the Pacific Northwest National Laboratory. It has provided data analysis and visualization support for academic papers in subject areas such as biological systems (Colby et al., 2023; Feng et al., 2021), cyber security (Cliff A. Joslyn et al., 2020), information systems (Molnár & Benczúr, 2022), neural networks (Praggastis et al., 2022), knowledge graphs (Cliff A. Joslyn et al., 2018), and the foundations of hypergraph theory (Vazquez, 2022).

References

- Aksoy, S. G., Joslyn, C., Marrero, C. O., Praggastis, B., & Purvine, E. (2020). Hypernetwork science via high-order hypergraph walks. *EPJ Data Science*, 9(1), 16. <https://doi.org/10.1140/epjds/s13688-020-00231-0>
- Antelmi, A., Cordasco, G., Kamiński, B., Prałat, P., Scarano, V., Spagnuolo, C., & Szufel, P. (2019). SimpleHypergraphs. JI—novel software framework for modelling and analysis of hypergraphs. *Algorithms and Models for the Web Graph: 16th International Workshop, WAW 2019, Brisbane, QLD, Australia, July 6–7, 2019, Proceedings 16*, 115–129. https://doi.org/10.1007/978-3-030-25070-6_9
- Berge, C. (1973). *Graphs and Hypergraphs*, (translated by Edward Minieka). American Elsevier, New York. [https://doi.org/10.1016/s0924-6509\(09\)x7013-3](https://doi.org/10.1016/s0924-6509(09)x7013-3)
- Colby, S. M., Shapiro, M. R., Lin, A., Bilbao, A., Broeckling, C. D., Purvine, E., & Joslyn, C. A. (2023). Introducing molecular hypernetworks for discovery in multidimensional metabolomics data. *bioRxiv*. <https://doi.org/10.1101/2023.09.29.560191>
- Csardi, G., Nepusz, T., & others. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695(5), 1–9.
- Feng, S., Heath, E., Jefferson, B., Joslyn, C., Kvinge, H., Mitchell, H. D., Praggastis, B., Einfeld, A. J., Sims, A. C., Thackray, L. B., & others. (2021). Hypergraph models of biological networks to identify genes critical to pathogenic viral response. *BMC Bioinformatics*, 22(1), 1–21. <https://doi.org/10.1186/s12859-021-04197-2>
- Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring network structure, dynamics, and function using NetworkX. In G. Varoquaux, T. Vaught, & J. Millman (Eds.), *Proceedings of the 7th python in science conference* (pp. 11–15).
- Joslyn, Cliff A., Aksoy, S. G., Callahan, T. J., Hunter, L. E., Jefferson, B., Praggastis, B., Purvine, E., & Tripodi, I. J. (2021). Hypernetwork science: From multidimensional

- 96 networks to computational topology. In D. Braha, M. A. M. de Aguiar, C. Gershenson,
97 A. J. Morales, L. Kaufman, E. N. Naumova, A. A. Minai, & Y. Bar-Yam (Eds.), *Unifying*
98 *themes in complex systems x* (pp. 377–392). Springer International Publishing. https://doi.org/10.1007/978-3-030-67318-5_25
99
- 100 Joslyn, Cliff A., Aksoy, S., Arendt, D., Firoz, J., Jenkins, L., Praggastis, B., Purvine, E., &
101 Zalewski, M. (2020). Hypergraph analytics of domain name system relationships. In B.
102 Kamiński, P. Prafat, & P. Szufel (Eds.), *Algorithms and models for the web graph* (pp.
103 1–15). Springer International Publishing. https://doi.org/10.1007/978-3-030-48478-1_1
- 104 Joslyn, Cliff A., Robinson, M., Smart, J., Agarwal, K., Bridgeland, D., Brown, A., Choudhury, S.,
105 Jefferson, B. A., Praggastis, B., Purvine, E., & others. (2018). HyperThesis: Topological
106 hypothesis management in a hypergraph knowledgebase. *TAC*.
- 107 Knuth, D. E. (1993). *The stanford GraphBase: A platform for combinatorial computing* (Vol.
108 1). AcM Press New York.
- 109 Kurte, K., Imam, N., Hasan, S. M. S., & Kannan, R. (2021). *Phoenix: A scalable streaming*
110 *hypergraph analysis framework*. https://doi.org/10.1007/978-3-030-71704-9_1
- 111 Landry, N. W., Lucas, M., Iacopini, I., Petri, G., Schwarze, A., Patania, A., & Torres, L. (2023).
112 XGI: A python package for higher-order interaction networks. *Journal of Open Source*
113 *Software*, 8(85), 5162. <https://doi.org/10.21105/joss.05162>
- 114 McKinney, Wes. (2010). Data Structures for Statistical Computing in Python. In Stéfan van
115 der Walt & Jarrod Millman (Eds.), *Proceedings of the 9th Python in Science Conference*
116 (pp. 56–61). <https://doi.org/10.25080/Majors-92bf1922-00a>
- 117 Molnár, B., & Benczúr, A. (2022). The application of directed hyper-graphs for analysis
118 of models of information systems. *Mathematics*, 10(5), 759. <https://doi.org/10.3390/math10050759>
119
- 120 Praggastis, B., Brown, D., Marrero, C. O., Purvine, E., Shapiro, M., & Wang, B. (2022).
121 The SVD of convolutional weights: A CNN interpretability framework. *arXiv Preprint*
122 *arXiv:2208.06894*. <https://doi.org/10.48550/arXiv.2208.06894>
- 123 The pandas development team. (2020). *pandas-dev/pandas: Pandas* (latest). Zenodo.
124 <https://doi.org/10.5281/zenodo.3509134>
- 125 Torres, L., Blevins, A. S., Bassett, D., & Eliassi-Rad, T. (2021). The why, how, and
126 when of representations for complex systems. *SIAM Review*, 63(3), 435–485. <https://doi.org/10.1137/20M1355896>
127
- 128 Vazquez, A. (2022). Growth principles of natural hypergraphs. *arXiv Preprint arXiv:2208.03103*.
129 <https://doi.org/10.48550/arXiv.2208.03103>