XGI: A Python package for higher-order interaction networks

Nicholas W. Landry [1,2¶], **Maxime Lucas** [3], **Iacopo Iacopini** [4], **Giovanni Petri** [3], **Alice Schwarze** [5], **Alice Patania** [1,2], **and Leo Torres** [6]

1 Vermont Complex Systems Center, University of Vermont, USA **2** Department of Mathematics and Statistics, University of Vermont, USA **3** CENTAI Institute, Italy **4** Department of Network and Data Science, Central European University, Austria **5** Department of Mathematics, Dartmouth College, USA **6** Max Planck Institute for Mathematics in the Sciences, Germany ¶ Corresponding author

DOI: 10.21105/joss.05162

Software

- Review
- Repository
- Archive

Editor: Vissarion Fisikopoulos

Reviewers:

- @arashbm
- @MridulS

Submitted: 03 February 2023
Published: 17 May 2023

Summary

Comple**X** **G**roup **I**nteractions (XGI) is a library for higher-order networks, which model interactions of arbitrary size between entities in a complex system. This library provides methods for building hypergraphs and simplicial complexes; algorithms to analyze their structure, visualize them, and simulate dynamical processes on them; and a collection of higher-order datasets. XGI is implemented in pure Python and integrates with the rest of the Python scientific stack. XGI is designed and developed by network scientists with the needs of network scientists in mind.

Statement of need

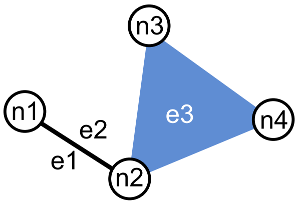
The field of network science bridges across many different disciplines, bringing together theorists, computational scientists, social scientists, and many others. To facilitate cross-disciplinary collaboration, a common tool kit is crucial. Existing packages like NetworkX (Hagberg et al., 2008), graph-tool (Peixoto, 2014), and igraph (Csardi & Nepusz, 2006) have been successful in facilitating collaboration for traditional networks, restricted to pairwise interactions. However, the rapidly growing subfield of higher-order network science (Battiston et al., 2020), which models interactions between any number of entities, requires a different approach. Higher-order interaction networks promotes rich dynamical behavior (Hickok et al., 2022; Iacopini et al., 2019; Neuhäuser et al., 2020; Skardal & Arenas, 2019), and can model some empirical interaction patterns more accurately than pairwise networks (Chodrow, 2020). We anticipate that this field will have lasting impacts on various research areas such as infectious diseases, dynamical systems, and behavioral science. To support the higher-order network science community, we have developed the Comple**X** **G**roup **I**nteractions (XGI)—an open-source solution in Python.

Related Software

There are several existing packages to represent and analyze higher-order networks: `HyperNetX` (Praggastis et al., 2023) and `Reticula` (Badie-Modiri & Kivelä, 2023) in Python, `SimpleHypergraphs.jl` (Spagnuolo et al., 2020) and `HyperGraphs.jl` (Diaz & Stumpf, 2022) in Julia, and `hyperG` in R. XGI is a valuable addition to the network science practitioner's toolbox for several reasons. First, XGI is implemented in pure Python, ensuring interoperability and easy installation across operating systems. Second, like several of the packages listed, XGI has a well-documented codebase and tutorials designed to make the learning process intuitive. Third, in contrast to existing packages, XGI contains a `stats` module enabling researchers to easily access established nodal and edge quantities, and even define custom quantities. Fourth, XGI offers data structures for hypergraphs and simplicial complexes, which allows users to explore a wider range of interaction models than comparable packages. Lastly, XGI integrates higher-order datasets with its interface, providing a standard format in which to store hypergraphs with attributes and a data repository with corresponding functions to load these datasets.

Overview

The two core classes of the library are those representing hypergraphs and simplicial complexes. The data structure (seen in Figure 1) employed by XGI for both is a bipartite graph with entities represented by one node type and relationships among entities (i.e., hyperedges or simplices) represented by a second node type.




```
>>> H._node
{
'n1': {'e1','e2'},
'n2': {'e1','e2','e3'},
'n3': {'e3'},
'n4': {'e3'}
}
```

```
>>> H._edge
{
'e1': {'n1','n2'},
'e2': {'n1','n2'},
'e3': {'n2','n3','n4'}
}
```

Figure 1: A hypergraph is internally represented as a bipartite network stored as two dictionaries, where keys are node IDs and sets specify the edges to which they belong, and vice-versa. Unique identifiers allow for multi-edges, as can be seen for edge IDs 1 and 2.

XGI provides several ways to create hypergraphs and simplicial complexes. First, by adding or removing nodes or hyperedges (or simplices). Second, by creating generative models, which can produce datasets with desired structural characteristics. Third, by loading existing datasets. XGI allows easy and unified access to many hypergraph datasets currently existing in diverse formats (A. Benson, 2021; Clauset et al., 2016; Peixoto, 2020) in three ways: first, by implementing a standard for hypergraph data in JSON format; second, by storing datasets in this format in a single repository, XGI-DATA (Landry et al., 2023); and third, by providing file I/O for common formats. Each dataset in XGI-DATA can be easily accessed through the library's API and the repository provides a description of it.

XGI provides many standard and state-of-the-art measures such as assortativity, centralities, connectedness, and clustering. A strength of XGI is its `stats` package: it provides a convenient and unified interface for computing statistics of nodes and edges, such as degree centrality or edge order. Any measure that is a node/edge-to-quantity mapping uses the same interface. Stats can be used to filter nodes and edges and multiple stats filters can be combined. XGI provides convenient visualization functions, as illustrated in Figure 2. We support multiple layouts and allow users to control many of the drawing parameters. Finally, XGI provides functions to simulate synchronization models on hypergraphs and simplicial complexes (Adhikari et al., 2023; Arnaudon et al., 2022; Lucas et al., 2020; Millán et al., 2020).

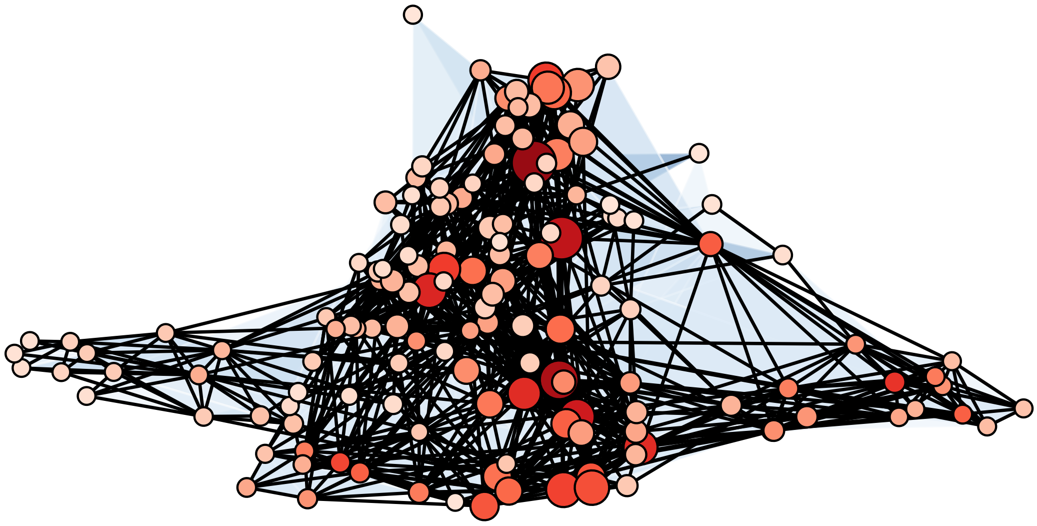


Figure 2: A visualization of the email-enron dataset (A. Benson, 2021; Landry et al., 2023) with hyperedges of sizes 2 and 3 (all isolated nodes removed). The nodes are colored by their degree and their size proportional to the Clique motif Eigenvector Centrality (A. R. Benson, 2019).

Projects using XGI

XGI has proved to be an invaluable resource for research projects (Zhang et al., 2023) on higher-order networks as well as other software projects (Landry & Miller, 2022). We expect that as this library matures, it will become a more essential part of the higher-order network science community.

Funding

N.W.L. acknowledges support from the National Science Foundation Grant 2121905, "HNDS-I: Using Hypergraphs to Study Spreading Processes in Complex Social Networks", and from the National Institutes of Health 1P20 GM125498-01 Centers of Biomedical Research Excellence Award. I.I. acknowledges support from the James S. McDonnell Foundation 21st Century Science Initiative Understanding Dynamic and Multi-scale Systems - Postdoctoral Fellowship Award.

Acknowledgements

We acknowledge contributions from Martina Contisciani, Tim LaRock, Marco Nurisso, Alexis Arnaudon, and Sabina Adhikari.

References

Adhikari, S., Restrepo, J. G., & Skardal, P. S. (2023). Synchronization of phase oscillators on complex hypergraphs. *Chaos*, *33*(3), 033116. https://doi.org/10.1063/5.0116747

Arnaudon, A., Peach, R. L., Petri, G., & Expert, P. (2022). Connecting Hodge and Sakaguchi-Kuramoto through a mathematical framework for coupled oscillators on simplicial complexes. *Communications Physics*, *5*(1), 1–12. https://doi.org/10.1038/s42005-022-00963-7

Badie-Modiri, A., & Kivelä, M. (2023). Reticula: A temporal network and hypergraph analysis software package. *SoftwareX*, *21*, 101301. https://doi.org/10.1016/j.softx.2022.101301

Battiston, F., Cencetti, G., Iacopini, I., Latora, V., Lucas, M., Patania, A., Young, J.-G., & Petri, G. (2020). Networks beyond pairwise interactions: Structure and dynamics. *Physics Reports*, *874*, 1–92. https://doi.org/10.1016/j.physrep.2020.05.004

Benson, A. (2021). *Data!* https://www.cs.cornell.edu/%7Earb/data/

Benson, A. R. (2019). Three Hypergraph Eigenvector Centralities. *SIAM Journal on Mathematics of Data Science*, *1*(2), 293–312. https://doi.org/10.1137/18M1203031

Chodrow, P. S. (2020). Configuration models of random hypergraphs. *Journal of Complex Networks*, *8*(cnaa018). https://doi.org/10.1093/comnet/cnaa018

Clauset, A., Tucker, E., & Sainz, M. (2016). *The Colorado Index of Complex Networks*. https://icon.colorado.edu/

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal*, *Complex Systems*, 1695. https://igraph.org

Diaz, L. P. M., & Stumpf, M. P. H. (2022). HyperGraphs.jl: Representing higher-order relationships in Julia. *Bioinformatics*, *38*(14), 3660–3661. https://doi.org/10.1093/bioinformatics/btac347

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). https://www.osti.gov/biblio/960616

Hickok, A., Kureh, Y., Brooks, H. Z., Feng, M., & Porter, M. A. (2022). A Bounded-Confidence Model of Opinion Dynamics on Hypergraphs. *SIAM Journal on Applied Dynamical Systems*, 1–32. https://doi.org/10.1137/21M1399427

Iacopini, I., Petri, G., Barrat, A., & Latora, V. (2019). Simplicial models of social contagion. *Nature Communications*, *10*(1), 2485. https://doi.org/10.1038/s41467-019-10431-6

Landry, N., & Miller, J. (2022). *Hypercontagion: a Python package for contagion processes on higher-order networks*. https://github.com/nwlandry/hypercontagion

Landry, N., Torres, L., Lucas, M., Iacopini, I., Petri, G., Patania, A., & Schwarze, A. (2023). *XGI-DATA: a repository of hypergraph datasets in JSON format*. https://github.com/xgi-org/xgi-data

Lucas, M., Cencetti, G., & Battiston, F. (2020). Multiorder Laplacian for synchronization in higher-order networks. *Physical Review Research*, *2*(3), 033410. https://doi.org/10.1103/PhysRevResearch.2.033410

Millán, A. P., Torres, J. J., & Bianconi, G. (2020). Explosive Higher-Order Kuramoto Dynamics on Simplicial Complexes. *Physical Review Letters*, *124*(21), 218301. https://doi.org/10.1103/PhysRevLett.124.218301

Neuhäuser, L., Mellor, A., & Lambiotte, R. (2020). Multibody interactions and nonlinear consensus dynamics on networked systems. *Physical Review E*, *101*(3), 032310. https://doi.org/10.1103/PhysRevE.101.032310

Peixoto, T. P. (2014). The graph-tool Python library. *Figshare*. https://doi.org/10.6084/m9.figshare.1164194

Peixoto, T. P. (2020). *The Netzschleuder network catalogue and repository*. https://networks.skewed.de/

Praggastis, B., Arendt, D., Joslyn, C., Purvine, E., Aksoy, S., & Monson, K. (2023). *HyperNetX (HNX): a Python library for hypergraphs*. https://github.com/pnnl/HyperNetX

Skardal, P. S., & Arenas, A. (2019). Abrupt desynchronization and extensive multistability in globally coupled oscillator simplexes. *Physical Review Letters*, *122*(24), 248301. https://doi.org/10.1103/PhysRevLett.122.248301

Spagnuolo, C., Cordasco, G., Szufel, P., Prałat, P., Scarano, V., Kamiński, B., & Antelmi, A. (2020). Analyzing, Exploring, and Visualizing Complex Networks via Hypergraphs using SimpleHypergraphs.jl. *Internet Mathematics*, *1*(1). https://doi.org/10.24166/im.01.2020

Zhang, Y., Lucas, M., & Battiston, F. (2023). Higher-order interactions shape collective dynamics differently in hypergraphs and simplicial complexes. *Nature Communications*, *14*(1), 1605. https://doi.org/10.1038/s41467-023-37190-9