# ISAC: An Interactive Hierarchical Interface for Efficient Structural Analysis and Vertex Search in Complex Networks

Navapat Nananukul, Khanin Sisaengsuwanchai, and Mayank Kejriwal

Information Sciences Institute. University of Southern California. CA, United states 90292.

**Abstract.** Recent years have seen increased growth and publication of graph datasets and complex networks. While the scientific study of networks is now an established area of research, information retrieval and visualization systems that address the needs of scientists and practitioners without technical expertise in network science algorithms and systems remain few and far between. Compounding the problem is that many large-scale network visualization systems are not publicly available under open licenses. As networks get larger, there is a clear need for an easy-to-use, customizable, and dynamic visualization platform that offers facilities like fast vertex search and real-time pan-and-zoom. This demonstration paper presents ISAC, a system that uses an efficient hierarchical paradigm for enabling vertex search and visualization, even in networks with hundreds of thousands of nodes. Through an evaluation, we show that it can also be used to visually find the most influential (or central) nodes in a large network that correlate positively with theoretical measures of centrality in the literature.

**Keywords:** Complex networks · hierarchical interface · clustering · local structural analysis.

## 1 Introduction

Analysis of complex networks has rapidly become an important area of scientific inquiry since the mid-1990s, with the growth of the Web and subsequent publication of networks spanning many domains, including sociology and digital humanities [3, 12]. Unfortunately, visualization of reasonably-sized complex networks on the order of a few hundred thousand nodes has not kept pace with the analytical literature. Several network visualization platforms do exist, but commonly used platforms like Gephi, Cytoscrape, and NetMiner have some limitations in efficiently and dynamically exploring large-scale networks in a customizable interface [8, 17]. Additionally, they are not easy to use for sociologists and other users without the necessary programmatic knowledge in network science. Within information retrieval, building interactive interfaces for graph data remains a challenging and specialized endeavor, and not a facility typically available to the non-programmer [21].

Recent research in graph visualization has attempted to address the problem of scale, and to some extent, user-centric needs [19,20]. For example, [20] proposes GPU-based implementations for generating fisheye views that can help users explore large networks [20]. However, even in these works, a gap remains. First, none of these platforms offer users the ability to search for nodes in the network that they have chosen to pre-index based on a node attribute (like name or email address) [1, 13]. Second, the platforms are not suited for a specific user-centric need that arises particularly often in sociology (but also in other domain-specific search applications); namely, the ability to zoom in on nodes that are *central* or influential in the network. While several theoretical definitions of centrality exist [4, 5], proposed in the network science literature over decades, and can be quantified using network science libraries like NetworkX [10], the values output by such tools are abstract. As such, they cannot be easily explained qualitatively, especially as the network grows in size. Besides, the wide variety of definitions employed in the literature suggest that, without visual exploration of centrality, it may not be possible to decide on the suitability of a specific centrality formula.

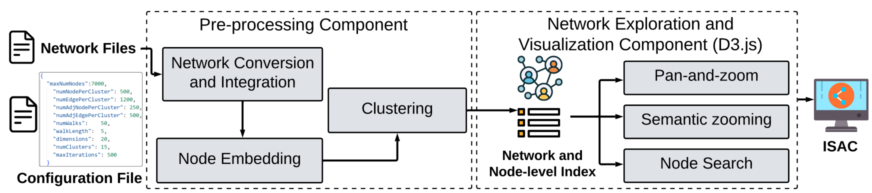


**Fig. 1.** A schematic illustration of ISAC's architecture, including components for pre-processing networks, clustering, node-level indexes, and D3-based visualization supporting features like semantic zooming and node search.

To address these needs, we present ISAC, an interactive hierarchical interface enabling structural analysis and vertex search in complex networks. Unlike many other network visualization tools, ISAC combines and simplifies multiple graph exploration techniques and is especially optimized for non-technical users who can use it as an exploratory platform for relational data without writing and debugging code using a sophisticated visualization language such as a declarative language design for interactive visualization [11], iVoLVER [14], and Superconductor [15]. We briefly describe ISAC's design, architecture, and interface, followed by an evaluation study where, on two real-world network datasets, we show that central nodes obtained visually correlate strongly with those output by a software-based tool that does not support visualization and requires code customization.

## 2 Design and implementation

In this section, we discuss the design and implementation of ISAC, an abstract architecture of which is shown in Figure 1. As shown therein, ISAC comprises two key modules: (1) network pre-processing and configuration; (2) graph exploration and visualization. Below we discuss these in more detail.

### 2.1 Network Pre-Processing and Configuration

ISAC simplifies user interaction by directly processing raw network data such as nodes, edges, and community files. It requires no additional input or preparatory steps from the user to generate the final network visualization. It is configured with default options to manage all pre-processing, including node embeddings and clustering, to extract and display network information to the user. At the same time, it also includes a simple configuration file for customizing these inputs for more advanced users.

**Network conversion and integration** is a module that enables ISAC to convert extensive network data from text or JSON files into a D3.js-compatible graph structure. It is equipped with specialized functions for consolidating nodes, edges, and metadata into a unified graph object that works with JavaScript libraries.

**Node embedding, clustering, and labeling**. ISAC utilizes *node2vec* as the default method for obtaining node embeddings [9]. For the more advanced user, node2vec can be 'substituted' out for other embedding algorithms, but knowledge of embeddings is not required to set up and run ISAC on a complex network dataset. The next step is clustering, where we utilize the (fairly standard) K-means clustering[1] to estimate and create a hierarchical structure of the community within the network, using the node embeddings obtained in the previous step. We also use cluster density to identify the important areas of the network for visualization purposes (i.e., which neighbors of a node to prioritize showing in the interface when too many neighbors are present). Finally, we implemented a coordination function that can take user visualization preferences into account via the configuration file. For example, if a user requests in the configuration to display only the top 10 nodes in each community, ISAC dynamically adjusts and prioritizes the top 10 nodes in the community based on the clustering result and number of neighbors.

**Configuration file** is an optional file that mainly contains visualization-dependent parameters (but also some embedding-based and low-level parameters) that users can use to customize the level of detail they see on the interface. It is meant to be useful especially when users switch from smaller screens (such as on a laptop) to bigger displays. Examples of parameters include the number of nodes per cluster, number of communities, number of edges per cluster, and so on. Users can always choose to omit the setup and use the default values of ISAC, which have been set through preliminary testing to yield a reasonable interface on most displays.

[1] Once again, the more advanced user can substitute the clustering for a different algorithm, although, in practice, we have found this to be unnecessary.

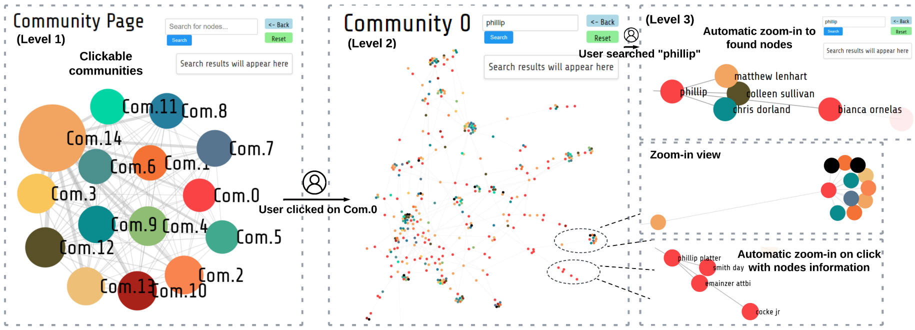


**Fig. 2.** An illustration of the front-facing interface that ISAC produces, and an illustrative workflow, given a network dataset. The 'level 1' page or home-page shows all communities in the network resulting from clustering. The user can click on the 'community node', which then shows all nodes in the community (semantic zooming), as well as neighbors of those nodes in other communities. Other facilities shown in the last column include pan-and-zoom functionality when users select a node they want to focus on, as well as when the user searches for a node (in this example, "phillip") using its indexed attribute. ISAC automatically triggers the pan-and-zoom functionality when a search is done and matched by users in a level 2 page.

### 2.2 Network Exploration and Visualization

ISAC incorporates multiple graph exploration techniques to enhance the user's interaction experience while navigating a complex network. Much of the visualization is implemented using the Javascript D3.js library. The visual exploration facilities implemented in ISAC are briefly described below:

**Pan-and-zoom [6]** is a fundamental interaction technique enabling users to select and zoom in on an area of interest. This functionality is embedded in ISAC and is activated when users click on a target node of interest. Additionally, the D3.js adaptive centering function is used for facilitating local or ego-centric structural analyses around the nodes of interest.

**Semantic zooming [7]** is a way to explore hierarchical networks, which is important to ISAC's core visualization. We took the clustering results (obtained through K-means by default) and show communities as clickable nodes on a home page (called the 'level 1' page), wherein users are able to select and click on clusters or communities they are interested in to explore further (taking them to 'level 2' pages).

**Node search [16]** is designed to save navigation time for users seeking specific information within a complex network. Users can enter textual information into the search bar and locate the nodes they are searching for. Depending on whether the search is conducted on the level 1 or one of the level 2 pages, the outcome is slightly different; as discussed in the next section, for the latter, ISAC automatically zooms in on the searched node when users are already on the same page, or within the same community as the searched node. This enables more rapid search. To enable search on the backend, nodes are efficiently index on an attribute specified by the user that can serve as a primary key (such as the name of the node, if the unique name assumption applies, the email address, and so on).

## 3 ISAC Demonstration

In the demonstration, we will showcase ISAC's capabilities to first-time users by using five real-world network datasets of different sizes and subject matter (ranging from publicly available social networks to co-citation and organizational networks) to instantiate ISAC in five different tabs. Users will be encouraged to pick a network of their choice and 'play' with it, including finding and exploring the local neighborhood of the most important nodes. We aim for the interface to reveal the structural, hierarchical complexities in real-world networks in an intuitive manner, allowing researchers and practitioners alike to see the scientific benefits of these powerful models.

Specifically, ISAC's utility will be demonstrated by showing its navigation on a level-by-level basis in Figure 2, starting from the left column to the right. Exploring large networks hierarchically is an obvious goal, but another important feature is the search bar on the community page, which allows users to search for information about nodes in the graph without having to guess which communities they might belong to. Upon entering a search-term, the search bar outputs the *community_id* of the matched node the user searches. This is to help users in finding their target node in the network. The result of the search bar is shown at the top right corner of the page and uses efficient node indexing data structures. During customization, users can specify which attribute of the node they want to use for indexing. As long as the attribute serves as a primary key (e.g., email address), it can be used.

The search bar on a level 2 page serves a similar purpose as that on the home page. However, instead of showing the node's community member, the level 2 search bar initiates a chain of events for users that (1) zoom in to the node they search for (as long as it is within that community) and (2) automatically highlight a node's indexed attribute and those of its neighbors in the network. In the last column in Figure 2, an example is shown of a highlighted node in the community page. At the node level (level 3), the interface supports pan-and-zoom functionality and highlights the nodes users were interested in while dimming the others. The zoom-in functionality is implemented specifically for the ego-structure of the selected node using an adaptive centering function. To visually verify that search was working properly, we sampled around 20 nodes from two reasonably-sized network datasets (described also in the next section), and verified that the search bar was able to find and match all samples to the correct community, nodes, and local neighborhood of nodes as in the original input files.

## 4 Evaluation

This section describes the evaluation measures and the result of the centrality study conducted using ISAC. We used two publicly available, real-world complex networks, both of which were configured similarly (e.g., the same number of level-1 communities). The first of these is the *email-enron* network [18] (containing 36,692 nodes and 183,831 edges) that was also used as an example in Figure 2. This network describes the email correspondence network at the (now defunct) Enron corporation. The second dataset is called *email-eu* [2] (containing 265,214 nodes and 420,045 edges) and similarly describes an EU email network. Central nodes obtained using ISAC were compared (using correlation measures) to those using NetworkX, as described next.

### 4.1 Evaluation Measures

We employed four common centrality implementations[2] in Networkx to generate a ranked list (per centrality measure). Because the number of communities was set to fifteen in the configuration, fifteen community-specific ranked lists per centrality measure were generated across the two datasets.

**Table 1.** A comparison of centrality of a visually produced ranking (using ISAC) vs. NetworkX ranking on retrieved entities using the Spearman's rank correlation (r), standard deviation (SD), and 95% confident interval (CI) based on four centrality measures: (1) Degree centrality (DC); (2) Eigenvector centrality (EC); (3) Closeness centrality (CC); and (4) Betweenness centrality (BC). Two network datasets were used for the study: *email-enron* and *email-eu*.

| Dataset | email-enron | | | email-eu | | |
|---|---|---|---|---|---|---|
| | r | SD | 95% CI | r | SD | 95% CI |
| DC | 0.799 | 0.236 | [0.68,0.92] | 0.886 | 0.115 | [0.83,0.94] |
| EC | 0.494 | 0.433 | [0.28,0.71] | 0.746 | 0.312 | [0.59,0.90] |
| CC | 0.541 | 0.369 | [0.35,0.73] | 0.746 | 0.265 | [0.61,0.88] |
| BC | 0.766 | 0.185 | [0.67,0.86] | 0.600 | 0.460 | [0.37,0.83] |

To assess ISAC's utility in visually finding central nodes, we used it to find the five most central nodes (based only on the visual inspection facilities offered by ISAC) within each of the 15 communities, for each dataset. We then used NetworkX to provide four sets of ranked lists using the centrality measures defined earlier. Subsequently, we calculated the Spearman's rank correlation between the ranked list produced visually and that generated by NetworkX. This process yielded four distinct correlation measures per community. To summarize the result, we average these correlations for all communities within the dataset and also calculate the 95% confidence interval (CI).

[2] Specifically, degree, closeness, eigenvector, and betweenness centrality measures.

As shown in Table 1, there are consistently positive and significant correlations across all centrality measures for both datasets, confirming ISAC's capability to guide the user in creating accurate node centrality rankings. The CIs are narrow, but in line with sociological theory, we find that correlations are not equally strong across all centrality measures, lending credence to our earlier claim that a visual inspection can help elucidate which centrality measure is most suitable for studying the network dataset. For example, the betweenness centrality (BC) is less suitable for analyzing *email-eu* than *email-enron*, while the opposite holds for EC.

We also conducted a detailed analysis of correlations at the community level and found that a majority of communities exhibited high positive correlations between the visual and theoretical methods at 99% significance. When using degree centrality, we found that 13 out of 14 communities show a strong positive correlation (0.5 and above), with correlations in 9 out of 14 communities statistically significant at the 99 percent confidence level or above on both datasets. In the case of eigenvector centrality, we found that 10 out of 14 communities from the *email-enron* dataset have a strong positive correlation with 5 communities at the 99 percent confidence level or above; on *email-eu*, 11 out of 14 communities have a strong positive correlation with 8 communities at the 99 percent confidence level or above.

## References

1. Allegri, S.A., McCoy, K., Mitchell, C.S.: Compositeview: A network-based visualization tool. Big data and cognitive computing **6**(2), 66 (2022)
2. Benson, A.R., Abebe, R., Schaub, M.T., Jadbabaie, A., Kleinberg, J.: Simplicial closure and higher-order link prediction. Proceedings of the National Academy of Sciences (2018). https://doi.org/10.1073/pnas.1800683115
3. Chung, D., Sohn, I.: Neural network optimization based on complex network theory: A survey. Mathematics **11**(2) (2023). https://doi.org/10.3390/math11020321, `https://www.mdpi.com/2227-7390/11/2/321`
4. Everett, M.G., Borgatti, S.P.: The centrality of groups and classes. The Journal of mathematical sociology **23**(3), 181–201 (1999)
5. Friedkin, N.E.: Theoretical foundations for centrality measures. American journal of Sociology **96**(6), 1478–1504 (1991)
6. Furnas, G.W.: Generalized fisheye views. Acm Sigchi Bulletin **17**(4), 16–23 (1986)
7. Furnas, G.W., Bederson, B.B.: Space-scale diagrams: Understanding multiscale interfaces. In: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 234–241 (1995)
8. Ghim, G.H., Kim, K., Ko, Y., Bae, S., Choi, W.: NetMiner, pp. 1–25. Springer New York, New York, NY (2018). https://doi.org/10.1007/978-1-4614-7163-9_305-2, `https://doi.org/10.1007/978-1-4614-7163-9_305-2`
9. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864 (2016)
10. Hagberg, A.A., Schult, D.A., Swart, P.J.: Exploring network structure, dynamics, and function using networkx. In: Varoquaux, G., Vaught, T., Millman, J. (eds.)

Proceedings of the 7th Python in Science Conference. pp. 11 – 15. Pasadena, CA USA (2008)
11. Heer, J., Bostock, M.: Declarative language design for interactive visualization. IEEE Transactions on Visualization and Computer Graphics **16**(6), 1149–1156 (2010)
12. Jiang, H., Liu, Z., Liu, C., Su, Y., Zhang, X.: Community detection in complex networks with an ambiguous structure using central node based link prediction. Knowledge-Based Systems **195**, 105626 (2020). https://doi.org/https://doi.org/10.1016/j.knosys.2020.105626, https://www.sciencedirect.com/science/article/pii/S0950705120300897
13. Majeed, S., Uzair, M., Qamar, U., Farooq, A.: Social network analysis visualization tools: A comparative review. In: 2020 IEEE 23rd International Multitopic Conference (INMIC). pp. 1–6 (2020). https://doi.org/10.1109/INMIC50486.2020.9318162
14. Méndez, G.G., Nacenta, M.A., Vandenheste, S.: ivolver: Interactive visual language for visualization extraction and reconstruction. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. pp. 4073–4085 (2016)
15. Meyerovich, L.A., Torok, M.E., Atkinson, E., Bodık, R.: Superconductor: A language for big data visualization. In: Workshop on Leveraging Abstractions and Semantics in High-Performance Computing. pp. 1–2 (2013)
16. Scheffler, P.: Node ranking and searching on graphs. In: 3rd Twente Workshop on Graphs and Combinatorial Optimization, Memorandum. vol. 1132 (1993)
17. Shannon, P., Markiel, A., Ozier, O., Baliga, N.S., Wang, J.T., Ramage, D., Amin, N., Schwikowski, B., Ideker, T.: Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome research **13**(11), 2498–2504 (2003)
18. Shetty, J., Adibi, J.: The enron email dataset database schema and brief statistical report. Information sciences institute technical report, University of Southern California **4**(1), 120–128 (2004)
19. Wang, Y., Wang, Y., Sun, Y., Zhu, L., Lu, K., Fu, C.W., Sedlmair, M., Deussen, O., Chen, B.: Revisiting stress majorization as a unified framework for interactive constrained graph visualization. IEEE Transactions on Visualization and Computer Graphics **24**(1), 489–499 (2018). https://doi.org/10.1109/TVCG.2017.2745919
20. Wang, Y., Wang, Y., Zhang, H., Sun, Y., Fu, C.W., Sedlmair, M., Chen, B., Deussen, O.: Structure-aware fisheye views for efficient large graph exploration. IEEE Transactions on Visualization and Computer Graphics **25**(1), 566–575 (2019). https://doi.org/10.1109/TVCG.2018.2864911
21. Zhu, Y., Yan, E., Song, I.Y.: A natural language interface to a graph-based bibliographic information retrieval system. Data & Knowledge Engineering **111**, 73–89 (2017). https://doi.org/https://doi.org/10.1016/j.datak.2017.06.006, https://www.sciencedirect.com/science/article/pii/S0169023X17302823