Research Article

Revisiting Markov Chain-Based Complex Network Analysis: A Diffusion-Geometric Toolkit with MarkovRCnet

Hiroyuki Akama[1]

1. Department of Community Medicine and Health Science, Kobe University, Japan

Beyond static topology, complex networks can be viewed as dynamical systems in which structure emerges through the flow of probability. We present MarkovRCnet, a Python package for the unified analysis of complex networks based on Markov chain dynamics. The package integrates two components derived from a common stochastic framework: the Reverse-branching Markov Clustering (RMCL) method for structural decomposition and the Markov inverse F-measure diffusion index (MiFDI) for node-level evaluation. Unlike conventional network analysis, where node importance and clustering structure are primarily interpreted through degree or density, MarkovRCnet treats networks as probability flow systems. RMCL partitions a network according to the stability of probability propagation, while MiFDI quantifies node influence from the viewpoint of broadcast diffusion. MiFDI employs a parameterized harmonic formulation that adapts the evaluation to network-specific diffusion structure rather than relying on fixed similarity measures. Because both quantities originate from the same transition process, they describe complementary aspects of a single latent geometry rather than independent statistics. Experiments on scale-free networks demonstrate that MiFDI reveals structurally meaningful nodes that are indistinguishable by degree, including late-attached peripheral nodes with high diffusion centrality. RMCL clusters consistently incorporate these nodes, indicating that the detected communities correspond to diffusion-based geodesic organization rather than topological density. This shows that even networks commonly regarded as weakly clustered possess latent cluster structure when interpreted through Markov dynamics. The package provides a reproducible and lightweight implementation designed to make stochastic network interpretation accessible as a coherent analytical workflow rather than a collection of separate metrics.

 doi.org/10.32388/95FJ91

Correspondence: papers@team.qeios.com — Qeios will forward to the authors

I. Introduction

Complex networks are typically analyzed through topological quantities such as degree, density, and modularity. These descriptors characterize how edges are arranged, but they do not directly describe how influence or information propagates across the system. As a result, structural interpretation and dynamical interpretation are often treated as separate analytical tasks.

A Markov chain representation provides an alternative viewpoint. Instead of focusing on adjacency itself, the network is regarded as a transition system in which nodes are related through probability flow. In this description, quantities derived from the same transition process naturally form a consistent family: some characterize regions of stable propagation, while others characterize node-level reachability.

This probabilistic perspective is not new. A major milestone is the Markov Clustering (MCL) algorithm proposed by Stijn van Dongen, which demonstrated that flow simulation can produce meaningful graph partitions [1]. MCL has since been widely applied in domains such as protein family detection and large-scale similarity network analysis, demonstrating the effectiveness of flow-based clustering in practical settings [2]. Despite its conceptual clarity and practical success, subsequent development has largely remained centered around the original C implementation, and lightweight, extensible implementations in high-level scientific environments remain comparatively limited. Moreover, MCL has often been treated primarily as a clustering tool rather than as part of a broader stochastic interpretive framework.

Building on this line of research, a series of studies extended the original flow-based formulation toward branching and refinement mechanisms. Branching Markov Clustering (BMCL) was proposed to address structural limitations of standard MCL [3], and various types of Recurrent Markov Clustering (RMCL) were subsequently introduced to refine semantic networks derived from large-scale linguistic resources [4][5]. RMCL extended the original flow-based intuition by incorporating recurrent and reverse-branching mechanisms that better capture latent adjacency relations. However, its application remained largely confined to language-related networks, including associative concept dictionaries and corpus-based semantic graphs. The broader implications of the approach for general complex networks were not systematically explored.

In parallel, the Markov inverse F-measure (MiF) was developed within the scope of computational neurolinguistics and applied to fMRI decoding tasks [6]. MiF reinterprets evaluation through the lens of

transition probability and broadcast reachability, introducing a parameterized harmonic formulation that allows the resulting values to adapt to network-specific diffusion characteristics rather than relying on fixed similarity coefficients. In this study, diffusion-derived graph representations were shown to support the prediction of neural activation patterns associated with lexical meaning. Although promising, MiF has not yet been widely integrated into general network analysis workflows. Meanwhile, computational neurolinguistics has gradually shifted toward deep neural architectures, leaving open the question of how Markov-based interpretive quantities might interact with contemporary graph neural network (GNN) models.

These developments suggest that flow-based clustering (MCL/RMCL) and diffusion-based evaluation (MiF) have evolved along related but largely disconnected trajectories. Both originate from the same transition dynamics, yet they have rarely been presented as components of a unified analytical system. As a result, clustering, node evaluation, and probabilistic interpretation are still frequently treated as independent layers of analysis.

We present MarkovRCnet, a lightweight Python implementation designed to reconnect these strands within a coherent framework. The package integrates Reverse-branching Markov Clustering (RMCL), which partitions networks according to propagation stability, and the Markov inverse F-measure diffusion index (MiFDI), which evaluates nodes according to broadcast reachability. Because both quantities originate from the same transition process, they can be interpreted jointly as complementary measurements of an induced stochastic geometry.

The usefulness of this perspective is illustrated through examples in which topological similarity does not imply similar diffusion behavior. In such settings, nodes indistinguishable by degree may exhibit systematically different reachability profiles, and partitions derived from transition dynamics reflect this organization. The objective of the package is therefore not merely to compute clusters but to expose the relationship between structural partition and diffusion geometry in a reproducible and extensible workflow.

Several classical stochastic approaches developed before the rise of deep graph learning were originally motivated by interpretability and structural readability rather than predictive performance alone. As research attention shifted toward neural architectures, many of these probabilistic formulations became less visible in mainstream workflows. Revisiting them is not an attempt to replace modern models but to re-express their structural insights in forms that can interact with contemporary graph representation learning.

MarkovRCnet is designed with this perspective in mind: to provide a computationally accessible bridge between earlier flow-based formulations and current graph learning paradigms. By making these transition-based quantities readily available as modular components, the package invites further experimentation in contexts that were not originally envisioned.

In short, this work does not introduce a new clustering algorithm alone; it provides a unified diffusion-based interpretive framework that connects clustering and node-level influence within a single stochastic geometry.

2. Materials and Methods

2.1. Framework Overview

MarkovRCnet is a Python toolkit for computational geometry-aware signal generation on graphs. It integrates transition-based similarity measures and Markov clustering algorithms within a unified framework. The package includes:

1. Markov Inverse F-measure (MiF)
2. MiF Degradation Index (MiFDI)
3. Markov Cluster Algorithm (MCL)
4. Recurrent/Refined MCL (RMCL)

All components are implemented as graph-derived representations computed directly from topology, without learning. They share a common transition matrix representation derived from the normalized adjacency matrix. This ensures that clustering and node-wise evaluation operate on identical stochastic foundations. Rather than computing independent metrics, the toolkit exposes multiple observables of a single transition process.

From an implementation perspective, the framework treats the transition operator as the primary computational object. Clustering (MCL/RMCL) modifies this operator iteratively, whereas MiF/MiFDI probe its propagation behavior under controlled initialization. This operator-centric design guarantees internal coherence across analytical components.

2.2. Markov Inverse F-measure (MiF)

MiF was previously introduced [6] as a similarity measure defined on graphs via Markov dynamics. A full mathematical derivation is provided in that work; here, we summarize its computational role. MiF evaluates the similarity between two vertices by combining:

1. local co-occurrence structure
2. global geodesic proximity
3. path multiplicity under stochastic transitions

The resulting value is normalized in the interval [0,1]. Two parameters govern its behavior:

1. β controls degree normalization via a harmonic-mean formulation
2. α controls the decay of longer transition paths

The decay coefficients are precomputed to satisfy a finite-sum normalization constraint, ensuring stable convergence of weighted path contributions. MiF outputs a similarity matrix usable as a distance-like representation.

2.3. MiF Degradation Index (MiFDI)

MiFDI extends MiF from pairwise similarity to a node-wise graph signal. Conceptually, MiFDI can be interpreted as a controlled broadcast of a random walker from a designated source node. While the underlying mechanism is standard Markov diffusion, the term “broadcast” emphasizes that the process expands outward from a specific origin rather than representing an equilibrium distribution. In this sense, broadcasting is treated as a special case of diffusion with an explicit initialization condition. Given a selected source vertex:

1. A Markov random walk is initiated.
2. As vertices are reached, their MiF similarity to the source is computed.
3. Log-transformed MiF values are recorded.
4. The process continues until all vertices are visited.

The resulting vector describes how similarity degrades as diffusion expands from the source.

Self-similarity is defined as zero to avoid singularity. MiFDI can optionally exclude self-loops and supports different initialization strategies (e.g., minimum-degree or maximum-degree source), allowing

sensitivity to homophily and heterophily patterns. MiFDI therefore provides a topology-derived node feature without requiring supervised learning.

2.4. Markov Cluster Algorithm (MCL)

We employ the classical Markov Cluster Algorithm proposed by Stijn van Dongen [1]. MCL simulates random walks using two alternating matrix operations:

1. expansion (matrix multiplication)
2. inflation (Hadamard power and rescaling)

These operations iteratively strengthen intra-cluster flow while weakening inter-cluster connections, yielding disjoint hard clusters.

2.5. Recurrent / Refined MCL (RMCL)

In graphs with heavy-tailed degree distributions, standard MCL may produce one dominant cluster alongside many small ones. RMCL addresses this imbalance while preserving hard clustering.

The procedure is as follows:

1. Identify oversized clusters based on cluster-size variance.
2. Construct latent adjacency relationships using representative hub nodes.
3. Reapply MCL to the reconstructed adjacency matrix.

This branching and reverse-branching mechanism allows balanced subdivision and merging under scale-free conditions [3][4][5][6].

2.6. Software Architecture

MarkovRCnet is organized into modular components:

1. `mcl/core.py` (MCL and RMCL)
2. `mif/mif.py` (MiF and MiFDI)
3. `utils/` (CSR handling, logging, PyG bridge)
4. `io/` (matrix loading/saving)
5. `data/` (toy datasets)

All adjacency matrices are internally converted to the CSR sparse format. A SafeCSR wrapper suppresses excessive intermediate logging during iterative matrix operations. Toy datasets are provided in Matrix Market format, including:

1. the Karate Club network
2. a scale-free benchmark graph

2.7. Availability and Reproducibility

The package is publicly available via the Python Package Index and can be installed using:

```
pip install markovrcnet
```

Docker images are also provided to ensure the reproducibility of the published version. They are available by executing:

```
docker pull akamahilolani/markovrcnet:latest
```

Furthermore, to enable programming within a web browser environment, another image for running Jupyter in Docker is also provisioned through the following procedure:

```
docker pull akamahilolani/markovrcnet-jupyter:latest
```

```
docker run --rm -it -p 10001:10001 akamahilolani/markovrcnet-jupyter:latest
```

2.8. Optional Interface to Graph Neural Networks

A minimal interface to PyTorch Geometric is implemented in utils/pyg.py. The provided utilities include:

1. CSR-to-edge_index conversion
2. cluster-to-label mapping
3. MiFDI-to-node-feature tensor conversion

These tools allow MarkovRCnet to function as a pre-GNN signal generator while remaining framework-agnostic at its core.

3. Results

3.1. Pairwise MiF geometry on the Karate Club network

MiF values were computed for all node pairs in the Karate Club network. The resulting similarity matrix was reordered using hierarchical clustering to reveal structural proximity.

As shown in Figure 1, bright blocks emerge near the diagonal after reordering. These blocks correspond to structurally coherent groups, even though no clustering algorithm was explicitly applied at this stage.

The reordered MiF matrix reveals concentrated similarity blocks, indicating that diffusion induces an intrinsic geometry in which community-like structures become visible without explicit clustering. The reordered node indices are shown explicitly on both axes, allowing direct inspection of structural proximity between specific vertices.

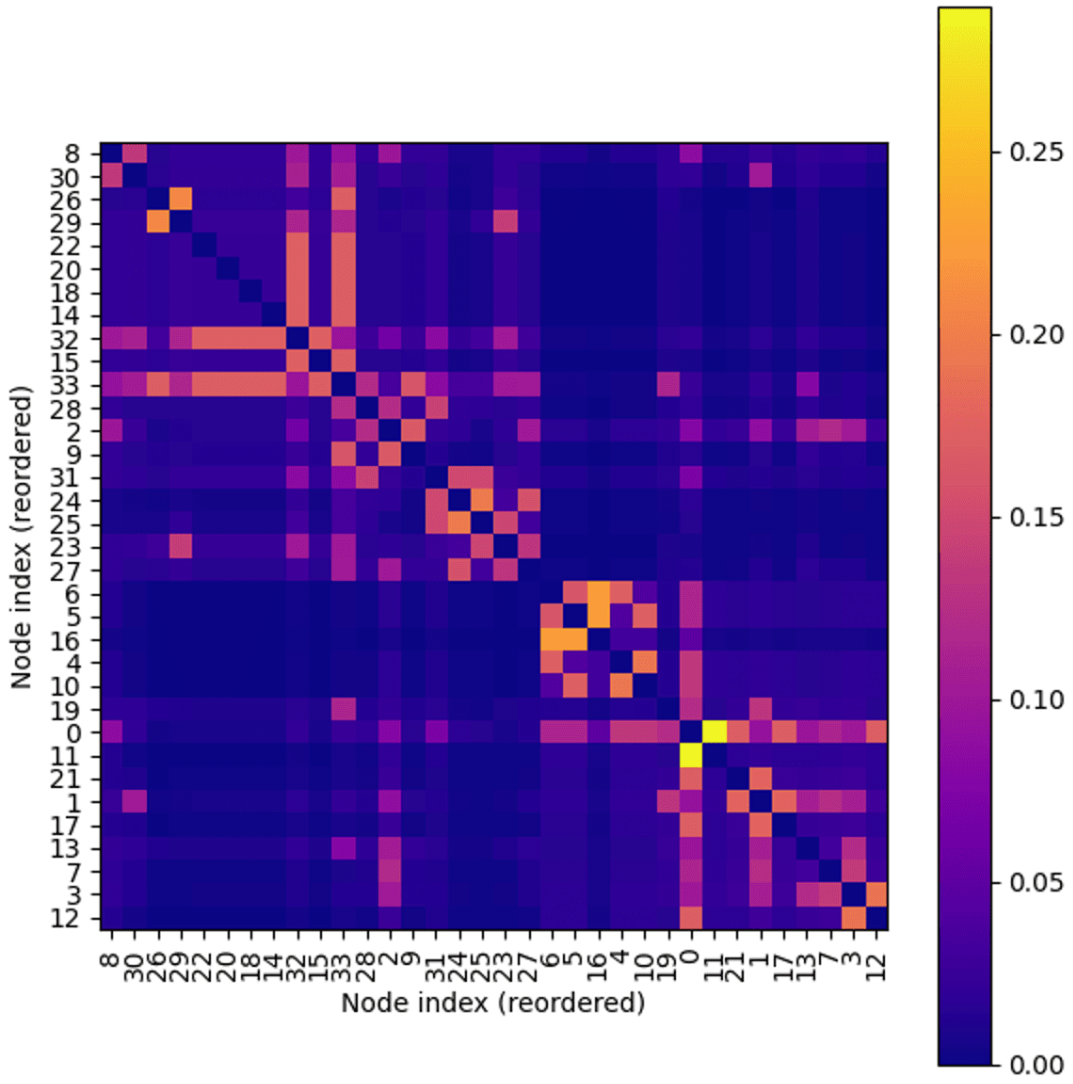


Figure 1. MiF Heatmap (Karate Club). The MiF values for all combinations of nodes *i* and *j* in the Karate Club have been calculated, and the structure is arranged so that combinations with larger values cluster around the diagonal. Therefore, the node numbers on the vertical and horizontal axes have been reordered for this purpose (note that numbering is zero-indexed). The reordered node numbers are explicitly shown as axis labels. Rows and columns are simultaneously sorted in the same order, appropriately arranged such that larger values represent structurally "closer" or "stronger influence," while smaller values represent structurally "farther" relationships. Bright cells (plasma yellow to white) cluster around the diagonal or in block-like formations, visually revealing the cluster and quasi-cluster structures. The program for generating the figure is included in the Appendix in the form of a Python file reference.

doi.org/10.32388/95FJ91

3.2. MiFDI as a node-wise graph signal

MiFDI was computed by initiating a Markov diffusion process from a dangling node. The resulting values were embedded directly onto the graph layout as node-wise signals in Figure 2. Unlike the grid-based matrix representation in Figure 1, this visualization places the signal onto the native graph geometry.

Each node is colored using a continuous heat colormap according to the relative magnitude of its MiFDI value, emphasizing how influence divergence propagates across the network topology. The starting node 11 is marked in black, while node size corresponds to degree, allowing structural prominence and Markov-derived signal intensity to be compared directly.

Nodes belonging to the same MCL clusters exhibit similar MiFDI magnitudes, despite MiFDI not being a clustering algorithm. This suggests that MiFDI captures global structural roles rather than merely local connectivity. The signal gradient reflects diffusion-constrained reachability rather than local degree, illustrating how broadcast dynamics translate topology into measurable geometry.

This example illustrates how MarkovRCnet produces graph-native signals that can be used independently for interpretation or combined with downstream pipelines such as graph neural networks. Node outlines indicate MCL cluster membership, enabling visual comparison between diffusion-derived signals and clustering structure.

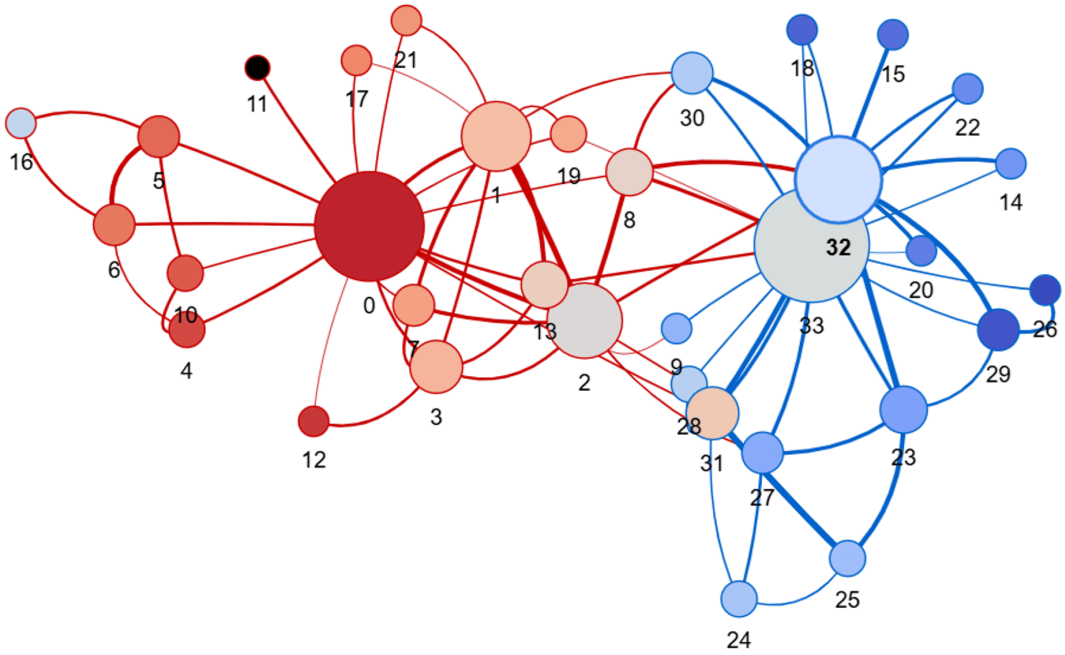


Figure 2. MiFDI as a Signal (Karate Club). MiFDI as a node-wise graph signal on the Karate Club network. This is an example of "mapping" MiFDI onto phase space as a graph signal. It demonstrates that MiFDI can be treated as a signal, not as a label or a cluster. Nodes are colored according to their MiFDI values computed from a dangling node (node 11) as the starting vertex. Warmer colors indicate larger influence divergence, while cooler colors represent smaller values. The starting node 11 is highlighted in black. Node size reflects degree. There are two Markov Clusters, distinguished by node outline color, and this diagram itself functions as an MCL cluster-colored graph. When the MiFDI values from the dangling node are heat-colored, these outline colors nearly match the background colors, demonstrating that MiFDI can also serve as an auxiliary metric for graph clustering. The program for generating the figure is included in the Appendix in the form of a Python file reference.

3.3. MCL behavior on a scale-free network

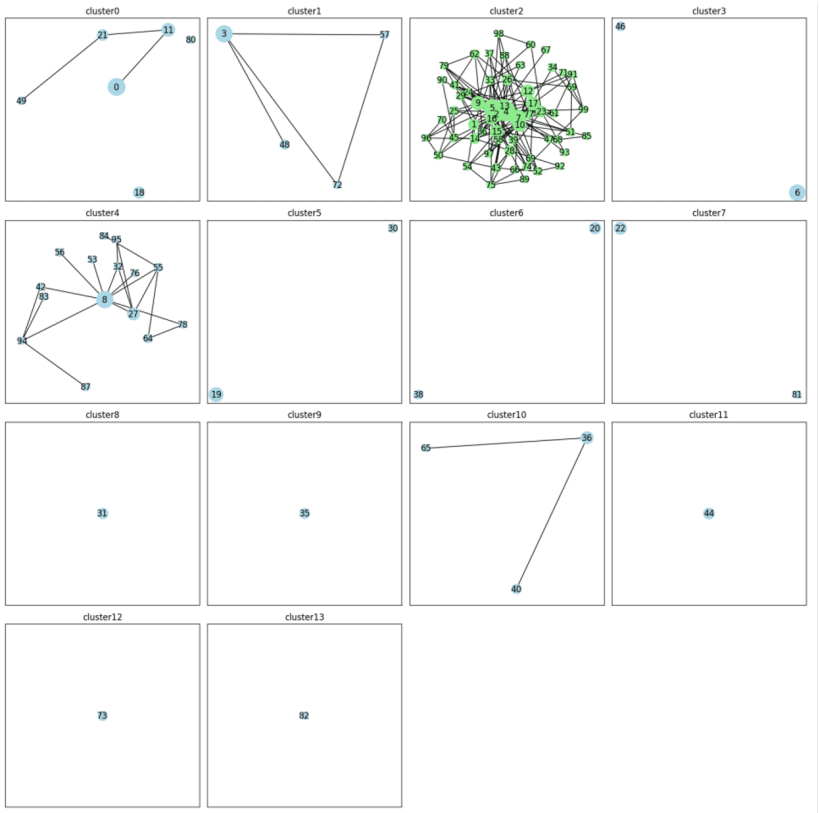
We next applied MCL to a scale-free graph generated by the Barabási–Albert model. As shown in the upper panel of Figure 3, MCL produces one dominant core cluster containing 52 nodes (more than half of the nodes), while the remaining nodes are fragmented into many small clusters, including several singletons. This imbalance reflects the heavy-tailed degree distribution typical of scale-free networks.

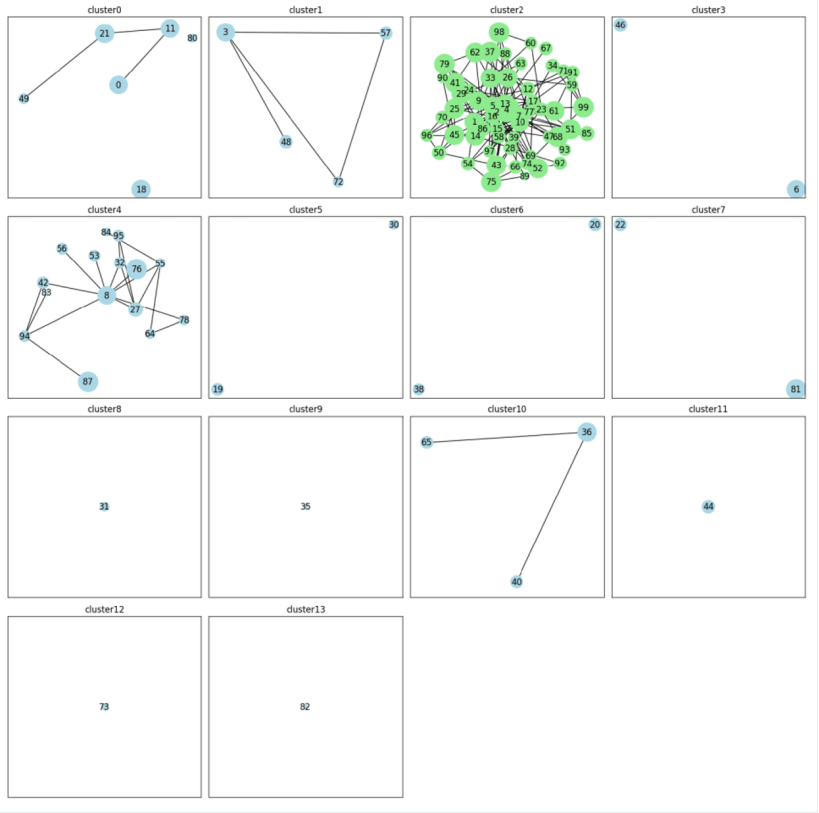
When node size is determined by MiFDI (lower panel) rather than degree (upper panel), a different pattern emerges: nodes within the core cluster exhibit remarkably similar MiFDI magnitudes, whereas

non-core clusters display greater internal variability. This phenomenon suggests the presence of a “core homogeneity effect” under Markov diffusion dynamics.

Interestingly, when node size is determined by MiFDI, the visual dominance of early high-degree hubs diminishes. Instead, nodes within the core cluster—including late-attached vertices with modest degree—exhibit uniformly high MiFDI values. This suggests the presence of a diffusion-defined elite layer that is not reducible to degree centrality. In this sense, the phenomenon resembles aspects of the VIP-club effect described by Masuda and Konno [7], where influential nodes are not necessarily characterized by maximal degree. However, unlike the original VIP-club model based on fitness and homophily, the structure observed here emerges purely from diffusion geometry.

In short, focusing on MiFDI rather than degree through node size, hub dominance visually diminishes, revealing a diffusion-defined core whose internal homogeneity is not predictable from the degree distribution alone.





doi.org/10.32388/95FJ91

Figure 3. Clusters by MCL applied to the scale-free graph. The original graph used for MCL/RMCL ("scalefree.mtx") is designed based on the Barabási-Albert model: it starts with a complete graph consisting of 5 nodes. Nodes with 5 edges are then successively connected to this graph according to preferential attachment, where the probability of connection depends on the degree of the target node. This process ultimately forms a scale-free network with 100 nodes as a target of MCL. When presenting the MCL results, we visualized the clusters such that the original adjacency relationships within each cluster are preserved by retaining edges. In this figure, the node size in the upper panel is determined by degree, while in the lower panel it is determined by the MiFDI value (setting the starting node to the hub with the maximum degree: node 4). Upper panel: Nodes with higher degrees have larger radii. Nodes within the core cluster (cluster2) are massive, numbering 52, and are depicted with a green background color. Nodes not belonging to the core cluster (light blue) are scattered across 13 small clusters, with sizes [6, 4, 2, 15, 2, 2, 2, 1, 1, 3, 1, 1, 1], forming a pattern of dispersed small clusters. Among these, 10 clusters are disconnected (cluster numbers 0, 3, 5, 6, 7, 8, 9, 11, 12, 13), and within them, there are 5 singleton clusters (cluster numbers 8, 9, 10, 12, 13). The subcluster information for non-core nodes is scattered and completely disconnected. Only the core clusters can be identified from the adjacency relationships, making it impossible to interpret the overall clustering results of MCL. Lower panel: This part shows the MCL results under the same conditions as above, but with node sizes determined by MiFDI values instead of degree. The configuration itself is identical to the panel above. Since MiFDI values are logarithmically transformed MiF values of 1.0 or less, they become negative. To reflect them in a node size that is more visually appealing, we adjusted them as follows: we took the absolute value of the minimum value, added 0.1 to make it slightly larger, added this to each original (negative) MiFDI magnitude, and then multiplied it by 200 during rendering. Interestingly, all members of cluster2 corresponding to the core cluster exhibit similarly large MiFDI values, and their node sizes are depicted correspondingly large. Unlike the upper panel diagram, which represents node characteristics by degree, no size irregularities are observed here. In contrast, the MiFDI values of the non-core clusters show significant variation in magnitude within each cluster. This suggests that a quasi-VIP club phenomenon emerges in the core cluster generated by MCL on scale-free graphs. The program for generating these figures is included in the Appendix in the form of a Python file reference.

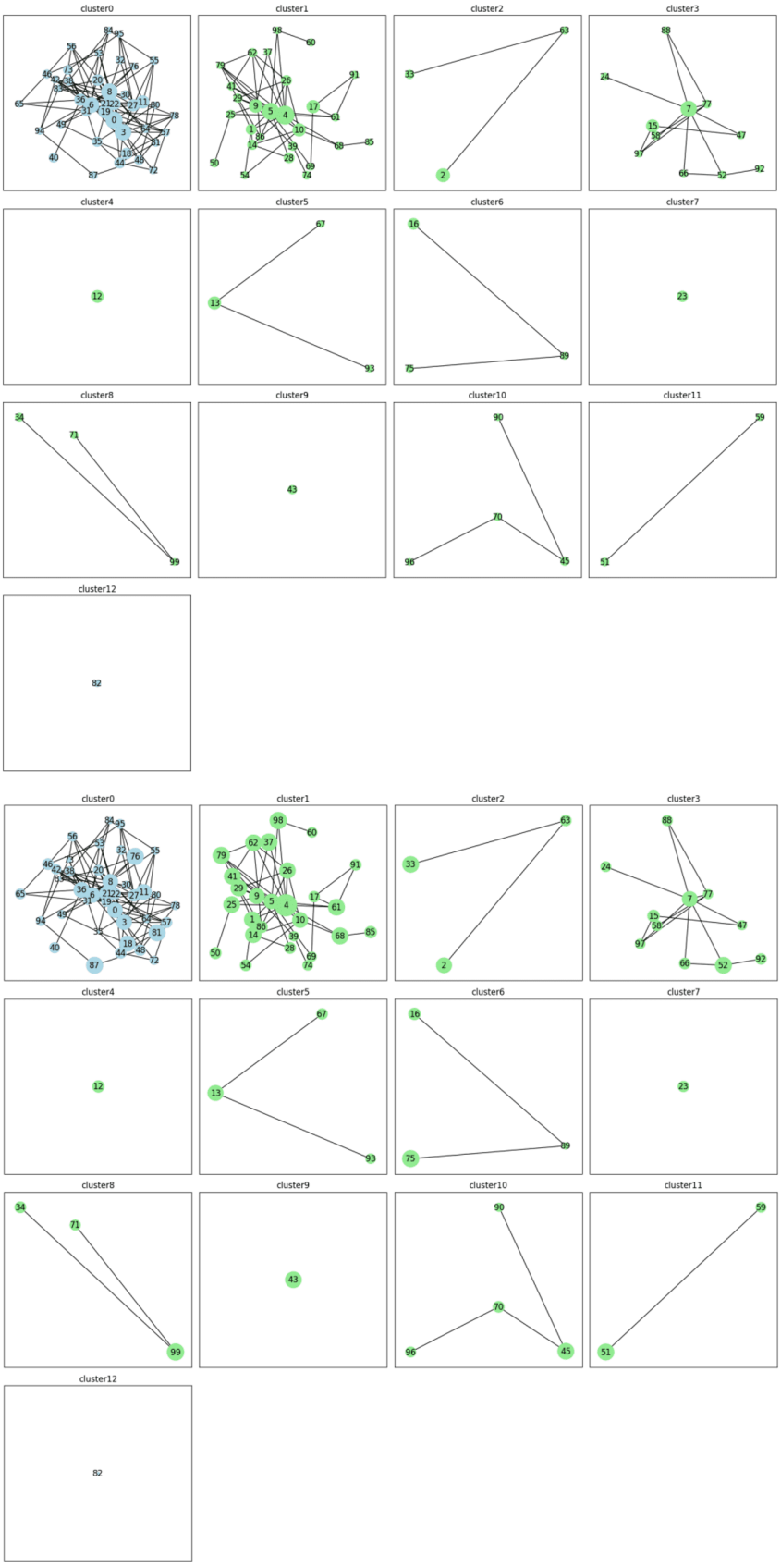
3.4. RMCL and latent adjacency reconstruction

RMCL was applied to address the imbalance observed under standard MCL. The upper panel of Figure 4 shows that the oversized core cluster is subdivided into more interpretable substructures while preserving hard clustering. Simultaneously, reverse branching merges previously fragmented non-core clusters into structurally coherent groups. The reverse-branching threshold was set to 2 for the

configuration shown. When visualized using MiFDI-based node sizes (lower panel), these refined clusters reveal latent structural organization that is not captured by degree alone. Notably, some low-degree nodes exhibit high MiFDI values, indicating that geodesic influence—rather than local degree—governs their structural importance. This demonstrates that latent adjacency reconstruction enables the recovery of hidden geometric structure even in scale-free graphs.

Across these experiments, a consistent pattern emerges: Markov diffusion transforms network topology into a geometry that reveals structural organization at multiple scales. Pairwise MiF exposes intrinsic similarity structure without explicit clustering, MiFDI provides a graph-native node signal reflecting global structural roles, and RMCL recovers interpretable cluster organization through latent adjacency reconstruction.

Taken together, these results indicate that MarkovRCnet functions not merely as a clustering toolkit but as a diffusion-based geometric framework for analyzing complex networks.



doi.org/10.32388/95FJ91

Figure 4. Clusters by RMCL applied to the scale-free graph. This figure shows the results of applying RMCL to the MCL results (depicted in Figure 3) of the scale-free graph. RMCL further partitioned the core clusters and merged the non-core clusters to obtain a more robust clustering result. Similar to Figure 3, the results are divided into an upper panel and a lower panel, differing only in the factor determining node size: degree in the upper panel and MiFDI value in the lower panel. The core cluster was appropriately resegmented, highlighting the largest sub-core cluster (cluster 1), which could be termed the secondary core, and the remaining sub-clusters (depicted in green). Among the sub-clusters originating from the core cluster, eight sub-clusters form connected components and restore an interpretable structure, except for three singleton clusters (numbers 4, 7, and 9). On the other hand, nodes not belonging to the core cluster (light blue) clearly show their structure in cluster 0, except for one singleton cluster (number 12), when the latent adjacency threshold in reverse branching MCL is set to 2, as shown in the figure. That is, both core and non-core components become interpretable from the connections, demonstrating the advantage of RMCL. Note that at threshold 1, the relationships among nodes not belonging to the core cluster can be aggregated into a single graph representing one cluster (cluster 0). Furthermore, as the threshold for identifying latent adjacency increases, the number of singleton clusters correspondingly grows. Upper panel: The size of a node is determined by its degree, and nodes with higher degrees have larger node radii. The diversity in the degrees of member nodes belonging to the same cluster can be observed. RMCL via reverse branching MCL incorporates the initial nodes of preferential attachment—namely, nodes with already high degrees and large MiFDI values (nodes 0, 3, 6, and 8)—as internal hubs when merging split non-core clusters (cluster 0). However, it simultaneously incorporates late peripheral nodes (nodes 76, 81, and 87) with low degrees but high MiFDI values. (In preferential attachment, later numbers are added later, hence the low degree.) Lower panel: The RMCL results, under the same conditions as above, show a diagram where node sizes are determined by MiFDI values rather than degree. The visualization method for MiFDI values is the same as that for the lower panel in Figure 3. The configuration is identical. Interestingly, the node sizes revealed meaningful features not apparent from degree alone. Specifically, among the decomposed core clusters, the largest secondary core (cluster 1) spans from lower to higher numbers. Although degrees vary, MiFDI generally shows large values. This indicates that geodesic information, not solely dependent on degree, is also correctly absorbed at the appropriate positions.

4. Discussion

4.1. Summary of Findings

The results consistently showed that controlled Markov diffusion induces an intrinsic geometric structure that makes latent organization observable across multiple scales. The present study introduced a computational geometry-aware signal toolkit grounded in controlled diffusion dynamics on complex networks. Across synthetic and scale-free graphs, three principal observations emerged.

First, diffusion-based signal propagation revealed structural regularities not directly apparent from static adjacency matrices. Cluster boundaries emerged as geometric phenomena induced by stochastic flow rather than as purely topological partitions.

Second, diffusion-informed metrics such as MiFDI exhibited value distributions that were sensitive to mesoscale organization while remaining robust to superficial variations in degree heterogeneity.

Third, although standard MCL applied to scale-free networks produced highly imbalanced clustering dominated by a large core component, RMCL partially restored structural heterogeneity through latent adjacency reconstruction, revealing organization not directly predictable from the degree distribution alone.

Finally, RMCL, as an extension of MCL, demonstrated the methodological validity of reverse-informed diffusion within the broader class of Markov chain–based approaches to complex networks, yielding partition structures more consistent with the value distribution of MiFDI. This suggests that controlled diffusion is not merely a clustering heuristic but a geometric probe capable of recovering latent organization.

4.2. Diffusion-Induced Geometry

The central conceptual claim of this work is that diffusion induces geometry. A graph, when viewed statically, is a combinatorial object defined by edges and degrees. However, once stochastic flow is introduced, the graph acquires an effective geometry defined by transition probabilities, flow retention, and information dissipation. Distances and boundaries become emergent properties of dynamic interaction rather than direct consequences of adjacency alone.

In this sense, diffusion does not simply traverse topology—it reorganizes it into a geometric space in which mesoscale structure becomes visible. The toolkit presented here operationalizes this perspective

by coupling diffusion processes with controlled branching mechanisms, thereby amplifying latent structural asymmetries.

4.3. Signal Without Embedding

A distinctive feature of the present framework is that it extracts a signal without requiring explicit low-dimensional embedding. Many contemporary graph analyses rely on spectral embeddings, manifold learning, or neural representation learning. While powerful, these approaches introduce additional representational layers that may obscure the relationship between structure and signal.

In contrast, diffusion-based reconstruction operates directly on transition dynamics. The signal emerges from flow constraints rather than from coordinate assignment. This makes the approach particularly suitable for systems where embedding assumptions are either theoretically unjustified or empirically unstable.

4.4. Latent Structural Reconstruction

Scale-free networks pose a known challenge for clustering algorithms, as hub-dominated cores tend to absorb peripheral nodes, resulting in highly imbalanced partitions. RMCL mitigated this imbalance by reconstructing latent adjacency through branching. By selectively reintroducing alternative diffusion pathways, the method exposed structural heterogeneity that remained hidden under conventional expansion–inflation dynamics.

Importantly, this reconstruction does not fabricate structure; rather, it reveals alternative connectivity patterns already implicit in the transition landscape. In this sense, RMCL functions as a controlled perturbation of diffusion geometry, allowing suppressed mesoscale patterns to surface.

The homogeneous MiFDI profiles observed within the core cluster—and within the largest secondary core revealed by RMCL—indicate the emergence of a diffusion-defined elite layer. This layer is not strictly determined by degree magnitude, as it includes both early high-degree hubs and later-attached nodes with comparatively modest connectivity. In this respect, the structure partially resonates with the VIP-club phenomenon reported by Masuda and Konno [7], where influence is not reducible to degree centrality alone. The present results do not reproduce the fitness-based VIP mechanism; rather, they suggest that diffusion geometry itself can generate quasi-elite structures under scale-free topology.

4.5. Conceptual Lineage: From Semantic Networks to Neural Signals

Diffusion-based geometry has previously been explored in semantic networks constructed from associative concept dictionaries or from document corpora [8]. In that context, stochastic transition processes were used to integrate geodesic information with co-occurrence structure.

In prior work, diffusion-derived graph components and the Markov inverse F-measure (MiF) were applied to predict fMRI neural activation patterns associated with noun semantics [6]. There, diffusion-informed distances improved predictive performance despite relatively small linguistic datasets. These findings suggested that diffusion-induced geometry can bridge symbolic relational structure and biological signal patterns. The present study extends this conceptual trajectory by demonstrating that diffusion-based reconstruction can uncover latent organization even in structurally heterogeneous networks.

4.6. Diffusion as a Conceptual Lens

Diffusion should not be regarded merely as an algorithmic technique. It is better understood as a conceptual lens. By allowing stochastic flow to probe structural constraints, one converts discrete connectivity into a dynamic geometric object. From this perspective, clusters are not simply dense subgraphs; they are regions of flow retention. Boundaries are not edge cuts; they are diffusion bottlenecks.

Future applications may include large-scale biological systems. In functional neuroimaging, resting-state and task-based connectivity graphs often suffer from sparsity or binarization effects [9]. Diffusion-based reconstruction may help reveal latent mesoscale organization without relying on dense weighted matrices. Similarly, gene regulatory networks, protein–protein interaction systems, and molecular interaction networks frequently exhibit core–periphery organization and heavy-tailed degree distributions [10]. Controlled diffusion processes could provide a principled mechanism for extracting structure beyond degree effects.

Social and communication networks may likewise benefit from being interpreted through diffusion-induced geometry rather than static adjacency. In such domains, the toolkit introduced here may serve as a computational scaffold for exploring how complex systems self-organize under stochastic constraints.

Whether diffusion-induced geometries can function as stable computational primitives for large-scale systems remains an open question. Nevertheless, the results presented here suggest that treating graphs

as dynamic geometric entities—rather than static combinatorial objects—opens a methodological direction that is both mathematically coherent and empirically promising.

In closing, the availability of accessible implementations is essential for bringing diffusion-based perspectives into everyday research practice. We hope that MarkovRCnet will promote exploratory use across diverse network domains and help reveal latent structural organization that is not apparent from topology alone.

Appendix A. Data, Code, and Reproducibility

This appendix is designed to make the experiments fully reproducible and easily extensible. All figures in the paper can be regenerated using a single command, and the provided scripts are intended as practical starting points for readers who wish to explore or adapt the methods to their own networks.

A.1. Software Availability

The methods described in this study are implemented in the Python package MarkovRCnet, which provides an integrated toolkit for Markov chain–based analysis of complex networks, including:

- Markov Clustering (MCL)
- Recurrent Markov Clustering (RMCL)
- Markov inverse F-measure (MiF)
- MiF-based diffusion indices (MiFDI)

The package is available via:

- PyPI
  - https://pypi.org/project/markovrcnet
- Docker environment
  - Images to be pulled:
    - akamahilolani/markovrcnet:latest,
    - akamahilolani/markovrcnet-jupyter:latest

All analyses reported in this paper were conducted using the publicly released version of the package.

Detailed package information can be found at the following URL:

- https://sites.google.com/site/akamatitechlab/markovrcnet

A.2. Reproduction of Figures

All figures presented in the main text can be reproduced using the script: data/generate_figures.py

Execution:

```
pip install markovrcnet
```

Generate all figures

- `python -m markovrcnet.data.generate_figures`

The above command reproduces all figures reported in the Results section.

The script automatically performs the following steps:

- Computes pairwise MiF values for the Karate Club network
- Generates MiFDI-based node signals
- Performs MCL and RMCL on the scale-free network
- Produces all visualizations used in the Results section

No parameter tuning or manual interaction is required.

Output files

All generated files are saved in:

- `data/figures/`

The outputs include:

File	Description
fig1_mif_heatmap.png	MiF similarity heatmap
fig2_karate_colormap_mifdi.html	Interactive MiFDI visualization on the Karate graph
fig3_mcl_degree.png	MCL clusters with degree-based node size
fig3_mcl_mifdi.png	MCL clusters with MiFDI-based node size
fig4_rmcl_degree.png	RMCL clusters with degree-based node size
fig4_rmcl_mifdi.png	RMCL clusters with MiFDI-based node size

The HTML file can be opened in any web browser.

Requirements

The script requires a standard Python environment with the dependencies listed in the MarkovRCnet package documentation. All networks used in the figures are included in the package via:

- `karateclub`
- `scalefree`

and are loaded automatically during execution.

Reproducibility Note

The figure-generation workflow is fully deterministic except for layout initialization in network visualization, where a fixed random seed is used to ensure consistent results.

The figure-generation scripts are intentionally written in a transparent and modular style so that individual components (MiF, MiFDI, MCL/RMCL visualization) can be reused independently in other workflows.

A.3. Datasets and Scale-Free Network Generation

Karate Club Network

The Karate Club network is provided as

`data/karateclub.mtx`

through the MarkovRCnet dataset loader:

```
from markovrcnet.datasets import load_all_adjmats
```

This corresponds to the standard network introduced by Wayne W. Zachary. The adjacency matrix is loaded automatically and requires no preprocessing.

Scale-Free Network

The scale-free network used in this study is distributed as:

- `data/scalefree.mtx`

This file contains the adjacency matrix used for all experiments and figures.

Regeneration (Optional)

The network can be regenerated using:

Generate the scale-free network

- `python -m markovrcnet.data.generate_scalefree`

The command can be executed from any working directory.

The script implements a degree-proportional preferential attachment process with fixed parameters:

- Number of nodes: 100
- Initial complete graph size: 5
- Edges per new node: 5
- **SEED = 42**

A fixed seed ensures reproducibility within the Python implementation.

Relation to the Original Network

The scale-free network originally used in this study was generated using a Wolfram Mathematica implementation of a preferential attachment algorithm. The Python script included in this repository reproduces the same generative principle, but:

- exact edge configurations may differ,
- node identities may not match,
- small structural variations

may occur, due to differences in:

- random number generation,
- sampling procedures,
- implementation details across environments.

Despite these differences, regenerated networks exhibit comparable scale-free structural properties and produce consistent experimental behavior.

Reproducibility Recommendation

For exact reproduction of the results reported in the paper, users are encouraged to use the distributed file:

- `data/scalefree.mtx`

The regeneration script is provided for transparency and methodological reproducibility.

A.4. Computational Environment

The analyses were conducted using:

- Python 3.x
- NumPy
- SciPy
- NetworkX
- Matplotlib
- PyVis

For full environment reproducibility, a Docker configuration is provided with the MarkovRCnet distribution.

A.5. Reproducibility Philosophy

The objective of this work is not to reproduce a single fixed network instance but to provide reproducible computational procedures for exploring diffusion-induced structure in complex networks.

Because diffusion-based analyses reflect structural tendencies rather than exact graph realizations, networks generated from the same stochastic mechanism are expected to produce qualitatively consistent geometric and clustering patterns. This perspective aligns with the conceptual role of MarkovRCnet as a practical toolkit for exploring networks as dynamic diffusion systems.

The provided scripts are intended not only for reproducing the results but also as a lightweight toolkit for exploratory analysis.

Users may apply the workflow to their own adjacency matrices or integrate MiF/MiFDI as graph-derived signals in downstream pipelines such as clustering or graph learning.

Readers are encouraged to experiment with different networks, parameters, or starting nodes. We welcome feedback, application reports, and methodological suggestions from the community to further refine and extend the practical use of this framework.

Statements and Declarations

Funding

No specific funding was received for this work.

Potential Competing Interests

No potential competing interests to declare.

Acknowledgments

The authors would like to thank ChatGPT (OpenAI) for support for programming and academic writing.

References

1. ^a, ^b*Van Dongen S (2000). "Graph Clustering by Flow Simulation." Utrecht University. https://dspace.library.uu.nl/bitstream/handle/1874/848/full.pdf?sequence=1&isAllowed=y.*
2. ^*Enright AJ, Van Dongen S, Ouzounis CA (2002). "An Efficient Algorithm for Large-Scale Detection of Protein Families." Nucleic Acids Res. **30**(7):1575–1584. doi:10.1093/nar/30.7.1575.*
3. ^a, ^b*Jung J, Miyake M, Akama H (2006). "Recurrent Markov Cluster (RMCL) Algorithm for the Refinement of the Semantic Network." LREC-2006. p.1428–1431. http://www.lrec-conf.org/proceedings/lrec2006/.*
4. ^a, ^b*Akama H, Miyake M, Jung J (2008). "How to Take Advantage of the Limitations with Markov Clustering?—The Foundations of Branching Markov Clustering (BMCL)." IJCNLP-2008. p.901–906. https://aclanthology.org/I08-2129.pdf.*
5. ^a, ^b*Jung J, Akama H (2008). "Employing Latent Adjacency for Appropriate Clusteringof Semantic Networks." New Trends in Psychometrics. p.131–140.*
6. ^a, ^b, ^c, ^d*Akama H, Miyake M, Jung J, Murphy B (2015). "Using Graph Components Derived from an Associative Concept Dictionary to Predict fMRI Neural Activation Patterns that Represent the Meaning of Nouns." PLoS ONE. doi:10.1371/journal.pone.0125725.*

7. ^a, ^b *Masuda N, Konno N (2005). "VIP-Club Phenomenon: Emergence of Elites and Masterminds in Social Networks." Soc Netw. **28**(4):297–309. doi:10.1016/j.socnet.2005.04.002.*
8. ^ *Miyake M, Joyce T, Jung J, Akama H (2007). "Hierarchical Structure in Semantic Networks of Japanese Word Associations." Proceedings of the 21st Pacific Asia Conference on Language, Information and Computation. p.321–329. https://aclanthology.org/Y07-1033/.*
9. ^ *Bullmore E, Sporns O (2009). "Complex Brain Networks: Graph Theoretical Analysis of Structural and Functional Systems." Nat Rev Neurosci. **10**(3):186–198. doi:10.1038/nrn2575.*
10. ^ *Jeong H, Mason S, Barabási AL, Oltvai Z (2001). "Lethality and Centrality in Protein Networks." Nature. **411**(6833):41–42. doi:10.1038/35075138.*

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.