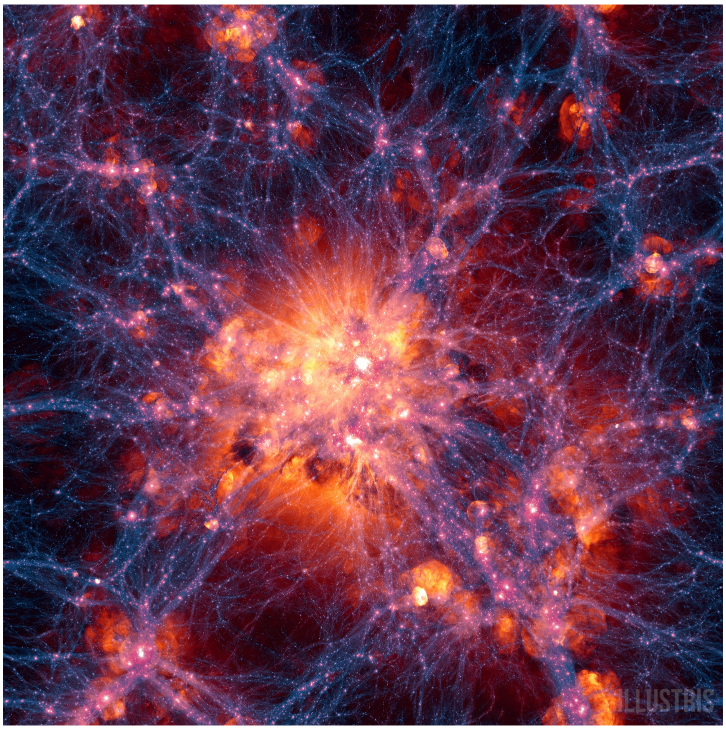
Cosmic Web as a Complex Network

Mojtaba Ali pour khoubdeh*

March 31, 2025

*Email: mojtaba.alipour@studio.unibo.it - Matricola 0001127996

The universe's large-scale structure, known as the cosmic web, is composed of galaxies, galaxy clusters, filaments, and voids. In this project, we model the cosmic web as a complex network. Galaxies, or dark matter halos, are treated as nodes within this network, with their interactions defining the edges. This approach facilitates a systematic analysis of the universe's structural properties, such as its topology, connectivity, and how it evolution over time.



Contents

1 Theoretical Fundamental

1.1 Introduction

Macroscopic patterns in nature are often due the collective action of basic, often even simple, physical processes. These may yield a surprising array of complex and genuinely unique physical manifestations. The macroscopic organization into complex spatial patterns is one of the most striking. The rich morphology of such systems and patterns represents a major source of information on the underlying physics. This has made them the subject of a major and promising area of inquiry.[1]The cosmic web, the desire to view the large-scale structure of the Universe as a network, is deeply embedded both in cosmology and in public consciousness. Sky surveys and mappings of the various wavelength bands of electromagnetic radiation (in particular 21-cm emission) have yielded much information on the content and character of the universe's structure.

1.2 Cosmic Web Formation

The cosmic web is a vast, intricate structure spanning scales of up to hundreds of megaparsecs, characterized by galaxies clustering into filaments, walls, and dense clusters surrounding near-empty void regions. This foam-like pattern was first hinted at in early redshift surveys like the CfA slice and has since been vividly revealed by large-scale surveys such as the 2dF Galaxy Redshift Survey (2dFGRS) and the Sloan Digital Sky Survey (SDSS). These surveys show galaxies arranged in striking geometric patterns, with filaments and walls forming the boundaries of enormous voids. The Local Universe, as mapped by infrared surveys like 2MASS, further illustrates our embedded position within this web, surrounded by sharply defined structures like the Perseus-Pisces supercluster and the Great Wall.

The cosmic web's components (clusters, filaments, walls, and voids) are not randomly scattered but interconnected in a highly organized, hierarchical fashion. Clusters, the densest nodes, form at the intersections of filaments, while voids occupy most of the universe's volume. The Sloan Great Wall, stretching over 400 Mpc, exemplifies the scale of these structures, though its physical coherence remains debated. High-redshift observations, such as those from the Subaru telescope, suggest similar weblike patterns existed even in the early universe, reinforcing the cosmic web's fundamental role in cosmic evolution.

Cosmic structure formation is driven by gravitational instability, where tiny primordial density and velocity perturbations grow under gravity's influence. Supported by cosmic microwave background (CMB) observations, these perturbations are Gaussian and isotropic, originating from an early inflationary phase. While linear growth is well understood, the nonlinear phase -(marked by the emergence of filaments, voids, and cluster) remain challenging to model analytically due to their complexity and nonlocal interactions.

The cosmic web emerges hierarchically, with small structures merging into larger ones, creating a multiscale tapestry. Anisotropic collapse plays a key role: overdensities flatten into sheets (walls), contract into filaments, and may further collapse into clusters. Meanwhile, voids expand faster than the universe's average, evacuating matter to their edges. This process is vividly captured in simulations like the Millennium Run, which show the web's evolution from mild fluctuations to a pronounced filamentary network.

Voids, filaments, and clusters are not isolated but dynamically linked. Voids dominate the universe's volume and shape the surrounding matter distribution, while filaments channel matter toward clusters. This interplay reflects the dual role of gravity: attracting matter into dense regions while driving the expansion of underdense ones. The resulting structure retains imprints of the initial conditions, offering clues to the universe's origins. Understanding the cosmic web requires tools that capture its hierarchical, anisotropic, and dynamic nature. Methods like the Delaunay Tessellation Field Estimator (DTFE) are designed to trace these features, revealing the web's geometry and evolution without imposing artificial symmetries or scales. This approach bridges the gap between theory and observations, illuminating how primordial noise transforms into the universe's vast, intricate web.

TABLE 1.1 Historical Timeline

Prior to 1989	It was believed that virialized galaxy clusters were the largest structures and uniformly spread across the universe. However, discoveries in the 1980s challenged this view.
1983	Adrian Webster identified a Large Quasar Group (LQG) consisting of five quasars, marking the first evidence of a large-scale structure.
1987	Robert Brent Tully discovered this galaxy filament, which includes the Milky Way.
1989	Roger G. Clowes and Luis E. Campusano found this quasar group, measuring 2 billion light-years, the largest known structure at the time.
2003	A massive cosmic structure was identified, adding to evidence of large-scale organization in the universe.
2007	A possible supervoid was detected in Eridanus, coinciding with the CMB cold spot, challenging cosmological models.
2011	A large quasar group, measuring 2.5 billion light-years across, was discovered.
2013	A large quasar group, measuring 2.5 billion light-years across, was discovered.
	In November 2013, Discovered using gamma-ray burst mapping, this structure was twice the size of previous largest structures.
2021	The American Astronomical Society announced the discovery of this crescent-shaped galaxy chain, spanning 3.3 billion light-years, located 9.2 billion light-years away in Boötes, using data from the Sloan Digital Sky Survey.

2 Network Construction

2.1 Network Construction from Galaxy Distributions

To model the cosmic web as a network, we extract real galaxy data from SDSS and construct a graph representation.

Nodes

Each galaxy in the dataset is assigned a position (α, δ, z), where:

- Right Ascension (RA) : Angular coordinate in the sky.
- Declination (Dec) δ: Vertical angular coordinate.
- Redshift z : Proportional to the galaxys distance due to cosmic expansion, given by
$$D_c(z) = \frac{c}{H_0} \int_0^z \frac{dz'}{E(z')} \tag{2.1}$$
where $E(z)$ describes the expansion rate.

Edges

Edges represent gravitational or spatial relationships between galaxies, defined using:

- Friends-of-Friends (FoF) algorithm: Connects galaxies within a linking length b,
- Voronoi tessellation: Segments space into nearest-neighbor regions,
- Tidal interaction thresholds: Links galaxies if their mutual gravitational influence is significant.

2.2 Spectral Analysis and Cosmological Significance

The networks properties can be studied through graph spectra, specifically the eigenvalues of the Laplacian matrix. Given an adjacency matrix A , the graph Laplacian is:

$$L = D - A \tag{2.2}$$

where D is the degree matrix (diagonal matrix with node degrees). The eigenvalues λ_i of L encode key structural properties:

- Smallest nonzero eigenvalue (λ_1) measures network connectivity.
- Spectral gap ($\lambda_2 - \lambda_1$) relates to network resilience.
- Largest eigenvalue (λ_n) correlates with the network's clustering and filaments.

Interpretation of the Spectrum

- Filamentary structures exhibit a scale-free degree distribution $\sim k^{-\gamma}$ with $\gamma \approx 2$,
- High eigenvalue density at low values suggests strong local clustering,
- Eigenvalue separation provides insight into the hierarchical structure.

The computed eigenvalues and their histogram visualization help us compare different cosmic models (ΛCDM, MOND, etc.) and trace cosmic evolution across redshift bins.

2.3 Code Examples

After the theoretical explanation, we simulated our model using Python. This involved installing the necessary prerequisite tools and packages to simulate the cosmic web network.

Note: For development and writing the codes, I used the manual toturial in [4, 5] and online AI tools.

```
pip install astropy astroquery pandas networkx scipy matplotlib
python-louvain
```

Briefly, I will explain what each package does:

1. **astropy** : Provides tools for handling astronomical data, including celestial coordinates, redshifts, and cosmological calculations.
2. **astroquery** :Enables querying astronomical databases like Sloan Digital Sky Survey (SDSS), NASA Exoplanet Archive, and others.
3. **pandas** : A powerful library for data manipulation and analysis.
4. **networkx** : A Python package for creating and analyzing complex networks.
5. **scipy** :Provides scientific computing functions, including spatial algorithms and linear algebra tools.
6. **matplotlib** : A visualization library for plotting data.
7. **python-louvain** : Implements the Louvain method for community detection in graphs.

These packages together allow us to fetch real astronomical data, construct a galaxy network, analyze its structure, and visualize its spectral properties. All scripts are available in my google colab page.

2.3.1 Example A

The first python script that provides a basic framework for modeling the cosmic web as a complex network using *NetworkX*, *numpy*, and *scipy*. It includes:

- Loading mock galaxy data (RA, Dec, redshift)
- Constructing a network using the Friends-of-Friends algorithm
- Analyzing network properties (degree distribution, clustering)
- Applying community detection using the Louvain method
- visualizing the network

```
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt
from scipy.spatial import cKDTree
from community import community_louvain  # Louvain method for community
                                          detection


# Step 1: Generate mock galaxy positions (RA, Dec, redshift)
def generate_mock_data(n_galaxies=500):
    ra = np.random.uniform(0, 360, n_galaxies)  # Right Ascension (
                                                  degrees)
    dec = np.random.uniform(-90, 90, n_galaxies)  # Declination (degrees)
    redshift = np.random.uniform(0, 1, n_galaxies)  # Mock redshift (
                                                  distance proxy)
    return np.vstack((ra, dec, redshift)).T


galaxy_positions = generate_mock_data()


# Step 2: Construct Network using Friends-of-Friends (FoF) algorithm
def construct_network(positions, linking_length=5):
    G = nx.Graph()
    tree = cKDTree(positions)  # k-d tree for fast neighbor search
    for i, pos in enumerate(positions):
        indices = tree.query_ball_point(pos, linking_length)  # Find
                                                      nearby galaxies
        for j in indices:
            if i != j:
                G.add_edge(i, j)
    return G
```

```
G = construct_network(galaxy_positions)


# Step 3: Analyze network properties
def analyze_network(G):
    degrees = [d for n, d in G.degree()]
    clustering_coeffs = nx.clustering(G)
    avg_clustering = np.mean(list(clustering_coeffs.values()))
    print(f"Network Analysis:\nNodes: {G.number_of_nodes()}\nEdges: {G.
                                         number_of_edges()}\nAvg Degree:
                                         {np.mean(degrees):.2f}\nAvg
                                         Clustering: {avg_clustering:.2f}
                                         ")
    return degrees, clustering_coeffs


degrees, clustering_coeffs = analyze_network(G)


# Step 4: Community Detection using Louvain method
communities = community_louvain.best_partition(G)


# Step 5: Visualize the Network
plt.figure(figsize=(8, 6))
nx.draw(G, node_size=10, node_color=list(communities.values()), cmap=plt.
                                     cm.viridis, edge_color='gray', alpha
                                     =0.5)
plt.title("Cosmic Web Network - Community Structure")
plt.show()
```

the outputof script is shown in figure 2.1a. It includes the 204 galaxies as nodes, and 140 connection exist as edges, that indicating a sparsely connected network. Each node has, on average, 1.37 connections, meaning many galaxies have few connections. The clustering coefficient is relatively low, suggesting a weakly connected cosmic web structure with small-scale clustering.

Furthermore, Small, isolated galaxy groups are present instead of a single large-scale web, and the network is not strongly interconnected, meaning galaxies are primarily in small filaments or pairs, not vast structures.

Parameter	**Description**	**Effect on Network**
n_galaxies = 500	Number of galaxies	A larger sample creates a denser network with more filaments.
linking_length = 7	Distance threshold for edges	Increasing it connects more galaxies, forming larger filaments.
redshift_range (0-2)	Cosmic volume depth	A larger range simulates a deeper cosmic structure.

Community Detection Algorithm (Louvain → Infomap)	Detects filaments and clusters	Infomap finds smaller clusters, while Louvain captures broader structures.
Network Layout (spring_layout, kamada_kawai_layout)	Changes visualization	Affects how filaments appear in the graph.

2.3.2 Example B

In the next steps, we Increase linking length to test how filaments grow, and Compare networks at different redshifts to observe cosmic evolution.

First, increse redshift range from 1 to 2, for larger cosmic volume, and Increased redshift range for larger cosmic volume:

```
# Step 1: Generate mock galaxy positions (RA, Dec, redshift)
def generate_mock_data(n_galaxies=500):
    ...
    redshift = np.random.uniform(0, 2, n_galaxies)  # Increased redshift
                                                   range for larger cosmic volume
     ...

def construct_network(positions, linking_length=7):
```

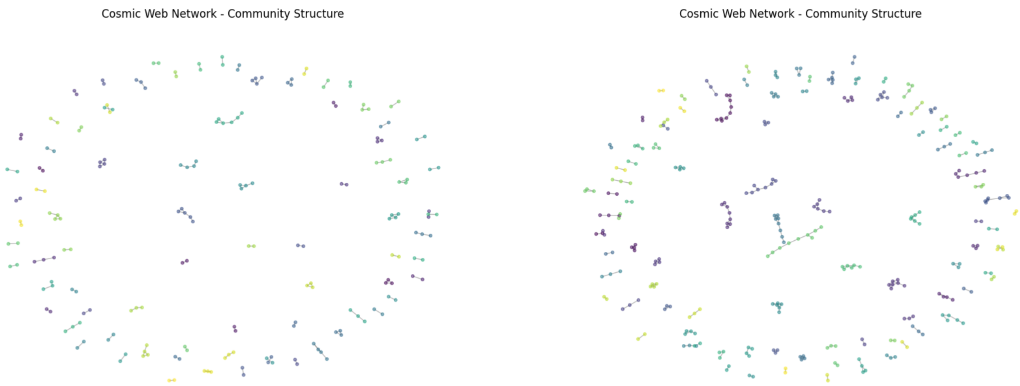
and in step 4, using Infomap,

```
pip install infomap
```

The infomap package implements the Infomap algorithm, a community detection method based on information theory. It is used for detecting clusters, filaments, and voids in complex networks, such as the cosmic web. It views the network as a random walker moving along the graph. It tries to minimize the description length of the walkers path, grouping nodes into communities where movement is frequent.

```
# Step 4: Community Detection using Infomap (alternative to Louvain)
try:
    import infomap
    im = infomap.Infomap()
    for node in G.nodes():
        im.add_node(node)
    for edge in G.edges():
        im.add_link(*edge)
    im.run()
    communities = {node: im.get_modules()[node] for node in G.nodes()}
except ImportError:
    print("Infomap not installed. Using Louvain instead.")
    communities = community_louvain.best_partition(G)
```

the output of this script is shown in the figure 2.1b, and 2.2



(a) 204 Nodes, 140 Edges, Avg. Degree=1.37 and Avg. Clustering= 0.22

(b) 328 Nodes, 285 Edges, Avg. Degree: 1.74, and Avg. Clustering: 0.31

Figure 2.1: Network Analysis with Louvin method

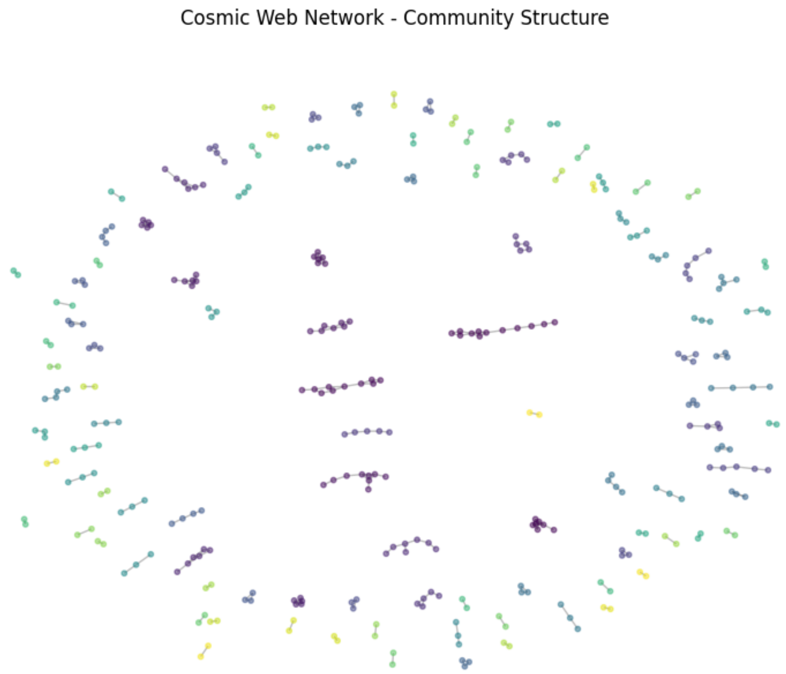


Figure 2.2: Network Analysis with Infomap method for 353 Nodes, 312 Edges, Avg. Degree: 1.77, and Avg. Clustering: 0.28

2.3.3 Example C

In this script we import the SDSS library and that sends query the SDSS database for reach the real galaxy positions (α , δ ,and z redshift)., the script is

```
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt
from scipy.spatial import cKDTree
from community import community_louvain  # Louvain method for community
                                          detection
from astroquery.sdss import SDSS
from astropy.table import Table
from scipy.linalg import eigh


# Step 1: Query SDSS for real galaxy positions
def query_sdss(n_galaxies=500):
    query = """
        SELECT TOP {}
            ra, dec, z
        FROM SpecObj
        WHERE class = 'GALAXY' AND z BETWEEN 0 AND 2
    """.format(n_galaxies)
    result = SDSS.query_sql(query)
    return np.array([result['ra'], result['dec'], result['z']]).T


galaxy_positions = query_sdss()


# Step 2: Construct Network using Friends-of-Friends (FoF) algorithm
def construct_network(positions, linking_length=7):
    G = nx.Graph()
    tree = cKDTree(positions)  # k-d tree for fast neighbor search
    for i, pos in enumerate(positions):
        indices = tree.query_ball_point(pos, linking_length)  # Find
                                                 nearby galaxies
        for j in indices:
            if i != j:
                G.add_edge(i, j)
    return G


G = construct_network(galaxy_positions)


# Step 3: Analyze network properties
def analyze_network(G):
    degrees = [d for n, d in G.degree()]
    clustering_coeffs = nx.clustering(G)
    avg_clustering = np.mean(list(clustering_coeffs.values()))
    print(f"Network Analysis:\nNodes: {G.number_of_nodes()}\nEdges: {G.
                                      number_of_edges()}\nAvg Degree:
```

```
                                              {np.mean(degrees):.2f}\nAvg
                                              Clustering: {avg_clustering:.2f}
                                              ")
    return degrees, clustering_coeffs


degrees, clustering_coeffs = analyze_network(G)


# Step 4: Community Detection using Infomap (alternative to Louvain)
try:
    import infomap
    im = infomap.Infomap()
    for node in G.nodes():
        im.add_node(node)
    for edge in G.edges():
        im.add_link(*edge)
    im.run()
    communities = {node: im.get_modules()[node] for node in G.nodes()}
except ImportError:
    print("Infomap not installed. Using Louvain instead.")
    communities = community_louvain.best_partition(G)


# Step 5: Compute Spectral Properties
def spectral_analysis(G):
    L = nx.laplacian_matrix(G).toarray()
    eigenvalues, _ = eigh(L)
    return eigenvalues


eigenvalues = spectral_analysis(G)


# Step 6: Visualize the Eigenvalue Distribution
plt.figure(figsize=(8, 6))
plt.hist(eigenvalues, bins=50, color='blue', alpha=0.7)
plt.xlabel("Eigenvalue")
plt.ylabel("Frequency")
plt.title("Spectral Analysis: Laplacian Eigenvalue Distribution")
plt.show()


# Step 7: Visualize the Network
plt.figure(figsize=(8, 6))
nx.draw(G, node_size=10, node_color=list(communities.values()), cmap=plt.
                                    cm.viridis, edge_color='gray', alpha
                                    =0.5)
plt.title("Cosmic Web Network - Community Structure")
plt.show()
```

the results in figure 2.3, with the SDSS data are shown. This analysis constructs a network model of galaxies using a Friends-of-Friends (FoF) algorithm , On that represents

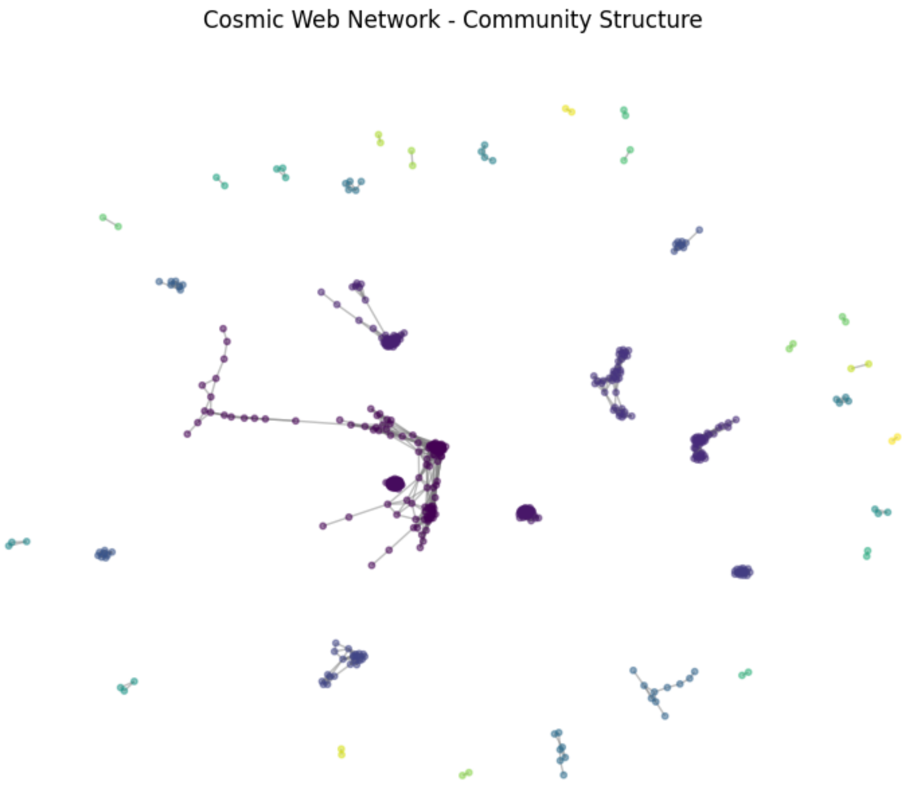


Figure 2.3: Network Analysis with Infomap method for 464 Nodes , 4356 Edges, Avg. Degree: 18.78, and Avg. Clustering: 0.75

464 galaxies in the dataset as nodes, 4356 Connections between galaxies, indicating their proximity based on the linking length. The average Degree is 18.78 it means each galaxy is, on average, connected to 19 others, which suggests a highly connected cosmic web and finally, the average clustering coefficient is 0.75 ,that shows the strong clustering suggests dense structures, similar to galaxy filaments and clusters.

Also this script Compute Spectral Properties (Eigenvalues) of the graph Laplacian, and Visualize the Eigenvalue Distribution to analyze network structure. the result is shown in figure 2.4. The computed eigenvalues and their histogram visualization help us compare different cosmic models (CDM, MOND, etc.) and trace cosmic evolution across redshift bins. Capture the connectivity and modularity of the network. The peaks in the eigenvalue distribution indicate hierarchical clustering, which may correspond to large-scale structures observed in SDSS, the low ei). This figure eigenvalues suggest presence of large, well-connected structures (filaments or superclusters) and the highrt eigenvalues Represent isolated or peripheral galaxies.

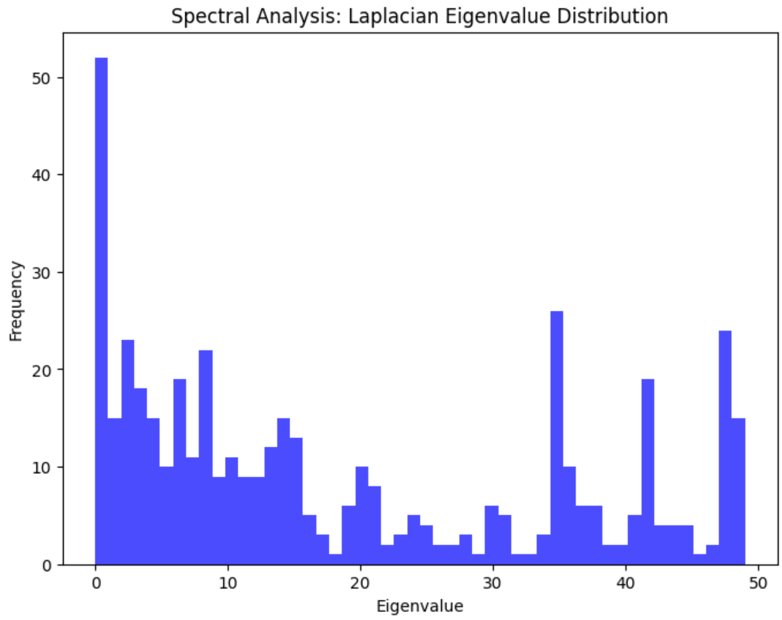


Figure 2.4: Spectral Analysis, Laplacian Eigenvalue Distribution

The adjacency matrix of the cosmic web represents the connectivity between galaxies, where each entry A_{ij} denotes the presence or absence of a direct link between two nodes. The resulting matrix visualization reveals distinct block-like structures along the diagonal, indicating highly interconnected communities, likely corresponding to galaxy clusters and filaments. The sparse background suggests a network with weakly connected or isolated galaxies, reinforcing the notion that the cosmic web is not fully connected. The prominence of strong diagonal elements highlights the dominance of local interactions, where galaxies are more likely to be linked to their nearest neighbors.

2.3.4 conslusion

This framework can be extended with: Graph Neural Networks (GNNs) for automatic classification of filaments and voids, Multiplex networks (e.g., layering dark matter, gas, and galaxy distributions), and Anomaly detection for identifying unexpected cosmic voids or overdensities. This study offers a network-theoretic perspective on the cosmic web, bridging cosmology, graph theory, and spectral analysis. By integrating SDSS data and spectral techniques, we gain deeper insights into the geometry, connectivity, and evolution of the universes largest structures.

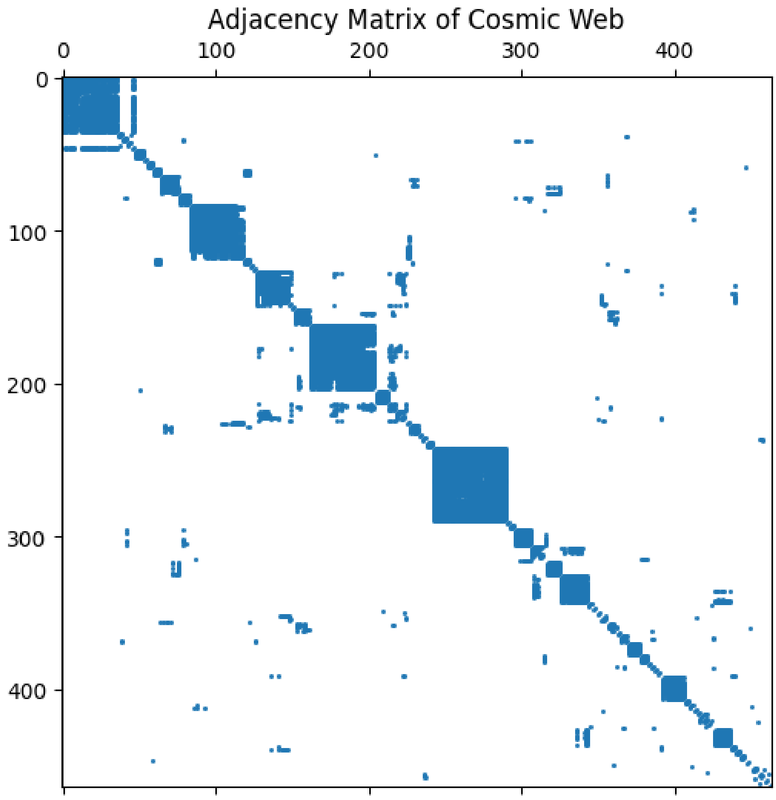


Figure 2.5: Adjacency matrix representation of the cosmic web network. The block-like structures along the diagonal indicate highly interconnected galaxy clusters and filaments, while the sparse regions represent weakly connected or isolated galaxies. This visualization highlights the hierarchical organization of cosmic structures within the network.

Bibliography

[1] Rien van de Weygaert Willem Schaap , "The Cosmic Web: Geometric Analysis",Lectures at Summerschool, 2004, Spain .

[2] B. C. Coutinho and etal., "The Network Behind the Cosmic Web", https://arxiv.org/abs/1604.03236/,2016.

[3] Zehavi, and etal. Galaxy Clustering in the Completed SDSS Redshift Survey: The Dependence on Color and Luminosity, arXiv:1005.2413, 2010.

[4] SDSS tutorial Skyserver

[5] NetworkX is a Python package official website

[6] Mapping the Cosmic Web