HCD Simulations Write Up

Audrey Fu Lab

2024-03-07

Contents

1 Data Simulation

1.0.1 Simulating the network

We adopt a top-down approach to simulate hierarchical networks, considering various simulation parameters such as graph sparsity, noise, and the architecture of the super-level graph(s), including small-world, scale-free, and random graph networks (Watts and Strogatz 1998; Barabási and Bonabeau 2003).

Our simulations focus on basic hierarchies comprising one or two hierarchical layers. Two-layer networks mirror classical community detection on graphs, where our aim is to recover the true community labels from a given graph. Meanwhile, three-layer networks present a more intricate scenario, where the bottom layer of the hierarchy contains two levels of community structure. Here, the top level corresponds to the nodes at the uppermost layer of the hierarchy, and the middle level consists of communities nested within the top-level communities. The objective with these networks is to identify both sets of community partitions.

In each hierarchy, for fully connected networks, we initiate by simulating n_{top} top-level nodes, adhering to a directed small-world, random graph, or scale-free network architecture (Watts and Strogatz 1998; Barabási and Bonabeau 2003). In cases where the network is disconnected, we simply simulate n_{top} disconnected nodes. For networks with three hierarchical layers, we then generate a subnetwork of n_{middle} nodes from each top-layer node, adhering to the network structure utilized at the top level. If the network is fully connected, we apply a probability p_{between} to the nodes from different top-level communities being connected.

The final step in all hierarchies is to generate the nodes in the observed (bottom) layer of the hierarchy. For each top-layer or middle-layer node, we generate a subnetwork of n_{bottom} nodes under the same subnetwork structure as the previous layers, and we apply a probability p_{between} for nodes from different communities to share an edge.

1.0.2 Simulating gene expression

Once we simulate a hierarchical graph, we utilize this hierarchy to generate the node-feature matrix, which depicts the expression of N genes across p samples. Here, N denotes the number of nodes in the observed (bottom) layer of the hierarchy, and its range is governed by $a^{\ell+1} < N < a \times b^{\ell}$, where ℓ signifies the number of hierarchical layers.

We simulate the node-feature matrix using the topological order the observed level graph. We start by generating the features of nodes that have no parental input. We refer to these nodes as origin nodes. All origin nodes are simulated from a normal distribution with mean 0 and standard deviation σ. All other nodes are simulated from a normal distribution centered at the mean of their parent nodes and with standard deviation σ.

2 Datasets

We consider three sets of hierarchical networks which represent varying difficulty levels for inference:

1. **Complex networks** - used for final simulation assessment - **Table** 4 - 4 .
2. **Intermediate networks** - used for investigative model tuning and performance assessment - **Table** 4 .
3. **Simple networks** - used for code implementation and debugging - **Table** 2.

3 Application to Intermediate Networks

A comprehensive overview of the intermediate networks is presented in **Table** 4 . These networks are structured as three-layered systems, each characterized by small-world, scale-free, or random graph architectures. In contrast to the more intricate networks featured in the **Complex Networks** dataset, the intermediate networks exhibit a comparatively simpler configuration. Specifically, each network comprises 5 super layer nodes, 15 middle layer nodes, and approximately 300 bottom layer nodes. Our primary focus in utilizing this dataset is to examine the performance of the Hierarchical Community Detection (HCD) method when applied to three-layer networks. The smaller scale of these networks facilitates a more in-depth analysis of the detected communities within the middle and upper layers of their hierarchical structures.

We apply the HCD method to each network separately using three options for the input graph corresponding to the nodes at the observed layer of the hierarchy:

- The input graph is the true graph
- The input graph is the correlation matrix of the simulated gene expression
- The input graph is the correlation matrix of the simulated gene expression wherein correlations weaker than 0.2 are disregarded and set to zero
- The input graph is the correlation matrix of the simulated gene expression wherein correlations weaker than 0.5 are disregarded and set to zero
- The input graph is the correlation matrix of the simulated gene expression wherein correlations weaker than 0.7 are disregarded and set to zero

We also explore various combinations of weighting the loss function across each of the aforementioned input graphs. In all cases, we ensure that the predicted number of communities in the middle or top levels of the hierarchy aligns with the ground truth of the simulation.

3.0.1 Evaluating performance

We evaluate the performance of our HCD method using three graph-based clustering metrics:

1. **homogeneity** evaluates the degree to which each predicted community contains only data points from a single true community, indicating how well the algorithm avoids mixing different groups. Thus, homogenity tends to be high if resolved communities contain only members of the same true community.
2. **completeness** assesses the extent to which all data points that belong to the same true community are correctly assigned to a single predicted community. Thus completeness is always high if all members of the same true communities end up in the same resolved community even if several true communities are allocated together.
3. **NMI** is a weighted average of the previous two metrics.

For each simulation, we configure the number of communities in the middle and upper layers of the hierarchy to match the true count in each layer. Then, we evaluate the community predictions of the Hierarchical Community Detection (HCD) algorithm at these levels against the actual communities using three metrics. As a baseline, we employ the Louvain method, which utilizes hierarchical graph partitioning to maximize modularity, resulting in a single set of resolved communities. These resolved communities may align with the middle, upper, or a combination of both layers in the true hierarchy. Thus, we compute the performance metrics of the communities identified by the Louvain method against the true communities at both the upper and middle levels of the hierarchy.

3.0.2 Preliminary findings for applications to intermediate networks

Figures ?? - ?? give a summary of HCD and Louvain performance on predicting the middle and top layer communities across all application settings outlined in the previous section and **Table** 3 . **Figure** 1.

4 Tables

Summary statistics for all small world networks in the complex networks datset

1

2

3

4

5

6

7

8

9

Subgraph type

small world

small world

small world

small world

small world

small world

small world

small world

Connection type

disc

disc

disc

disc

full

full

full

full

Layers

2

2

3

3

2

2

3

3

Standard deviation

0.1

0.5

0.1

0.5

0.1

0.5

0.1

0.5

Nodes per layer

(10, 63)

(10, 63)

(10, 63, 1604)

(10, 63, 1604)

(10, 63)

(10, 63)

(10, 63, 1604)

(10, 63, 1604)

Edges per layer

(0, 63)

(0, 63)

(0, 63, 2011)

(0, 63, 2031)

(45, 115)

(45, 109)

(45, 114, 1604)

(45, 111, 1604)

Subgraph probability

0.05

0.05

0.05

0.05

0.05

0.05

0.05

0.05

Sample size

500

500

500

500

500

500

500

500

Modularity (top)

0.898

0.898

0.898

0.898

0.447

0.477

0.766

0.771

Average node degree top

1

1

1.254

1.266

1.825

1.73

1.433

1.439

Avg connections within top communities

6.3

6.3

201.1

203.1

6.3

6.3

199.3

201.3

Avg. connections between top communities

0

0

0

0

0.578

0.511

3.389

3.278

Modularity (middle)

NA

NA

0.762

0.758

NA

NA

0.667

0.663

Average node degree middle

NA

NA

1.254

1.266

NA

NA

1.433

1.439

Avg connections within middle communities

NA

NA

24.825

24.968

NA

NA

24.937

24.873

Avg connections between middle communities

NA

NA

0.114

0.117

NA

NA

0.186

0.19

Summary statistics for all scale free networks in the complex networks datset

1

10

11

12

13

14

15

16

17

Subgraph type

scale free

scale free

scale free

scale free

scale free

scale free

scale free

scale free

Connection type

disc

disc

disc

disc

full

full

full

full

Layers

2

2

3

3

2

2

3

3

Standard deviation

0.1

0.5

0.1

0.5

0.1

0.5

0.1

0.5

Nodes per layer

(10, 58)

(10, 58)

(10, 58, 1450)

(10, 58, 1450)

(10, 58)

(10, 58)

(10, 58, 1450)

(10, 58, 1450)

Edges per layer

(0, 74)

(0, 74)

(0, 74, 6700)

(0, 74, 6670)

(45, 120)

(45, 120)

(45, 123, 1450)

(45, 122, 1450)

Subgraph probability

0.05

0.05

0.05

0.05

0.05

0.05

0.05

0.05

Sample size

500

500

500

500

500

500

500

500

Modularity (top)

0.89

0.89

0.892

0.893

0.513

0.513

0.854

0.849

Average node degree top

1.276

1.276

4.621

4.6

2.069

2.069

4.781

4.843

Avg connections within top communities

7.4

7.4

670

667

7.4

7.4

665.9

671.4

Avg. connections between top communities

0

0

0

0

0.511

0.511

3.033

3.422

Modularity (middle)

NA

NA

0.906

0.91

NA

NA

0.875

0.864

Average node degree middle

NA

NA

4.621

4.6

NA

NA

4.781

4.843

Avg connections within middle communities

NA

NA

107.069

107.069

NA

NA

107.069

107.069

Avg connections between middle communities

NA

NA

0.148

0.139

NA

NA

0.218

0.246

Summary statistics for all random graph networks in the complex networks datset

1

18

19

20

21

22

23

24

25

Subgraph type

random graph

random graph

random graph

random graph

random graph

random graph

random graph

random graph

Connection type

disc

disc

disc

disc

full

full

full

full

Layers

2

2

3

3

2

2

3

3

Standard deviation

0.1

0.5

0.1

0.5

0.1

0.5

0.1

0.5

Nodes per layer

(10, 45)

(10, 45)

(10, 45, 725)

(10, 45, 725)

(10, 45)

(10, 45)

(10, 45, 725)

(10, 45, 725)

Edges per layer

(0, 32)

(0, 32)

(0, 32, 678)

(0, 32, 665)

(45, 77)

(45, 77)

(45, 78, 725)

(45, 78, 725)

Subgraph probability

0.05

0.05

0.05

0.05

0.05

0.05

0.05

0.05

Sample size

500

500

500

500

500

500

500

500

Modularity (top)

0.883

0.883

0.886

0.885

0.313

0.313

0.758

0.721

Average node degree top

0.711

0.711

0.935

0.917

1.711

1.711

1.04

1.09

Avg connections within top communities

3.2

3.2

67.8

66.5

3.2

3.2

65.5

65.7

Avg. connections between top communities

0

0

0

0

0.5

0.5

1.1

1.478

Modularity (middle)

NA

NA

0.783

0.803

NA

NA

0.703

0.669

Average node degree middle

NA

NA

0.935

0.917

NA

NA

1.04

1.09

Avg connections within middle communities

NA

NA

12.156

12.222

NA

NA

12.178

12.156

Avg connections between middle communities

NA

NA

0.066

0.058

NA

NA

0.104

0.123

Table 1: Summary statistics for intermediate difficulty simulated networks.

Value	Network1	Network2	Network3	Network4	Network5	Network6
Subgraph type	small world	small world	scale free	scale free	random graph	random graph
Connection type	disc	full	disc	full	disc	full
Layers	3	3	3	3	3	3
Standard deviation	0.1	0.1	0.1	0.1	0.1	0.1
Nodes per layer	(5, 15, 300)	(5, 15, 300)	(5, 15, 300)	(5, 15, 300)	(5, 12, 167)	(5, 12, 167)
Edges per layer	(0, 15, 354)	(10, 25, 300)	(0, 10, 966)	(10, 20, 300)	(0, 7, 133)	(10, 17, 167)
Subgraph probability	0.05	0.05	0.05	0.05	0.05	0.05
Sample size	500	500	500	500	500	500
Modularity (top)	0.799	0.715	0.78	0.751	0.791	0.665
Average node degree top	1.18	1.34	3.22	3.32	0.796	0.886
Avg connections within top communities	70.8	73.6	193.2	193.2	26.6	26
Avg. connections between top communities	0	1.7	0	1.5	0	0.9
Modularity (middle)	0.781	0.679	0.873	0.845	0.787	0.696
Average node degree middle	1.18	1.34	3.22	3.32	0.796	0.886
Avg connections within middle communities	20	20	61.333	61.333	9.667	9.667
Avg connections between middle communities	0.257	0.486	0.219	0.362	0.129	0.242

Summary statistics for all random graph networks in the complex networks datset

1

NA

NA.1

NA.2

NA.3

NA.4

NA.5

NA.6

NA.7

Subgraph type

NA

NA

NA

NA

NA

NA

NA

NA

Connection type

NA

NA

NA

NA

NA

NA

NA

NA

Layers

NA

NA

NA

NA

NA

NA

NA

NA

Standard deviation

NA

NA

NA

NA

NA

NA

NA

NA

Nodes per layer

NA

NA

NA

NA

NA

NA

NA

NA

Edges per layer

NA

NA

NA

NA

NA

NA

NA

NA

Subgraph probability

NA

NA

NA

NA

NA

NA

NA

NA

Sample size

NA

NA

NA

NA

NA

NA

NA

NA

Modularity (top)

NA

NA

NA

NA

NA

NA

NA

NA

Average node degree top

NA

NA

NA

NA

NA

NA

NA

NA

Avg connections within top communities

NA

NA

NA

NA

NA

NA

NA

NA

Avg. connections between top communities

NA

NA

NA

NA

NA

NA

NA

NA

Modularity (middle)

NA

NA

NA

NA

NA

NA

NA

NA

Average node degree middle

NA

NA

NA

NA

NA

NA

NA

NA

Avg connections within middle communities

NA

NA

NA

NA

NA

NA

NA

NA

Avg connections between middle communities

NA

NA

NA

NA

NA

NA

NA

NA

Table 2: Summary statistics for simple simulated networks. These networks contain fewer than 100 nodes at the observed level and only cover small world subgraph architecture

Value	Network1	Network2	Network3	Network4
Subgraph type	small world	small world	small world	small world
Connection type	disc	disc	full	full
Layers	2	3	2	3
Standard deviation	0.1	0.1	0.1	0.1
Nodes per layer	(2, 6)	(2, 6, 18)	(2, 6)	(2, 6, 18)
Edges per layer	(0, 6)	(0, 6, 24)	(1, 7)	(1, 7, 18)
Subgraph probability	0.05	0.05	0.05	0.05
Sample size	500	500	500	500
Modularity (top)	0.5	0.5	0.357	0.46
Average node degree top	1	1.333	1.167	1.389
Avg connections within top communities	3	12	3	12
Avg. connections between top communities	0	0	0.5	0.5
Modularity (middle)	NA	0.583	NA	0.553
Average node degree middle	NA	1.333	NA	1.389
Avg connections within middle communities	NA	3	NA	3
Avg connections between middle communities	NA	0.2	NA	0.233

Table 3: Simulation settings for intermediate difficulty networks. Each row represents a single simulation scenario applied to all 6 simulated networks given in Table 1

Input Graph	Graph Recon. Loss	Attr. Recon. Loss	Modularity Weight	Clust. Weight
A_ingraph_true	1 = on	False (on)	1 = on	1 (middle), 1 (top)
A_corr_no_cutoff	1 = on	False (on)	1 = on	1 (middle), 1 (top)
A_ingraph02	1 = on	False (on)	1 = on	1 (middle), 1 (top)
A_ingraph05	1 = on	False (on)	1 = on	1 (middle), 1 (top)
A_ingraph07	1 = on	False (on)	1 = on	1 (middle), 1 (top)
A_ingraph_true	0 = off	False (on)	1 = on	1 (middle), 1 (top)
A_corr_no_cutoff	0 = off	False (on)	1 = on	1 (middle), 1 (top)
A_ingraph02	0 = off	False (on)	1 = on	1 (middle), 1 (top)
A_ingraph05	0 = off	False (on)	1 = on	1 (middle), 1 (top)
A_ingraph07	0 = off	False (on)	1 = on	1 (middle), 1 (top)
A_ingraph_true	1 = on	True (off)	1 = on	1 (middle), 1 (top)
A_corr_no_cutoff	1 = on	True (off)	1 = on	1 (middle), 1 (top)
A_ingraph02	1 = on	True (off)	1 = on	1 (middle), 1 (top)
A_ingraph05	1 = on	True (off)	1 = on	1 (middle), 1 (top)
A_ingraph07	1 = on	True (off)	1 = on	1 (middle), 1 (top)
A_ingraph_true	0 = off	True (off)	1 = on	1 (middle), 1 (top)
A_corr_no_cutoff	0 = off	True (off)	1 = on	1 (middle), 1 (top)
A_ingraph02	0 = off	True (off)	1 = on	1 (middle), 1 (top)
A_ingraph05	0 = off	True (off)	1 = on	1 (middle), 1 (top)
A_ingraph07	0 = off	True (off)	1 = on	1 (middle), 1 (top)
A_ingraph_true	1 = on	False (on)	0 = off	1 (middle), 1 (top)
A_corr_no_cutoff	1 = on	False (on)	0 = off	1 (middle), 1 (top)
A_ingraph02	1 = on	False (on)	0 = off	1 (middle), 1 (top)
A_ingraph05	1 = on	False (on)	0 = off	1 (middle), 1 (top)
A_ingraph07	1 = on	False (on)	0 = off	1 (middle), 1 (top)
A_ingraph_true	0 = off	False (on)	0 = off	1 (middle), 1 (top)
A_corr_no_cutoff	0 = off	False (on)	0 = off	1 (middle), 1 (top)
A_ingraph02	0 = off	False (on)	0 = off	1 (middle), 1 (top)
A_ingraph05	0 = off	False (on)	0 = off	1 (middle), 1 (top)
A_ingraph07	0 = off	False (on)	0 = off	1 (middle), 1 (top)
A_ingraph_true	1 = on	True (off)	0 = off	1 (middle), 1 (top)
A_corr_no_cutoff	1 = on	True (off)	0 = off	1 (middle), 1 (top)
A_ingraph02	1 = on	True (off)	0 = off	1 (middle), 1 (top)
A_ingraph05	1 = on	True (off)	0 = off	1 (middle), 1 (top)
A_ingraph07	1 = on	True (off)	0 = off	1 (middle), 1 (top)
A_ingraph_true	0 = off	True (off)	0 = off	1 (middle), 1 (top)
A_corr_no_cutoff	0 = off	True (off)	0 = off	1 (middle), 1 (top)
A_ingraph02	0 = off	True (off)	0 = off	1 (middle), 1 (top)
A_ingraph05	0 = off	True (off)	0 = off	1 (middle), 1 (top)
A_ingraph07	0 = off	True (off)	0 = off	1 (middle), 1 (top)
A_ingraph_true	1 = on	False (on)	1 = on	0.1 (middle), 1e-4 (top)
A_corr_no_cutoff	1 = on	False (on)	1 = on	0.1 (middle), 1e-4 (top)
A_ingraph02	1 = on	False (on)	1 = on	0.1 (middle), 1e-4 (top)
A_ingraph05	1 = on	False (on)	1 = on	0.1 (middle), 1e-4 (top)
A_ingraph07	1 = on	False (on)	1 = on	0.1 (middle), 1e-4 (top)

A_ingraph_true	0 = off	False (on)	1 = on	0.1 (middle), 1e-4 (top)
A_corr_no_cutoff	0 = off	False (on)	1 = on	0.1 (middle), 1e-4 (top)
A_ingraph02	0 = off	False (on)	1 = on	0.1 (middle), 1e-4 (top)
A_ingraph05	0 = off	False (on)	1 = on	0.1 (middle), 1e-4 (top)
A_ingraph07	0 = off	False (on)	1 = on	0.1 (middle), 1e-4 (top)
A_ingraph_true	1 = on	True (off)	1 = on	0.1 (middle), 1e-4 (top)
A_corr_no_cutoff	1 = on	True (off)	1 = on	0.1 (middle), 1e-4 (top)
A_ingraph02	1 = on	True (off)	1 = on	0.1 (middle), 1e-4 (top)
A_ingraph05	1 = on	True (off)	1 = on	0.1 (middle), 1e-4 (top)
A_ingraph07	1 = on	True (off)	1 = on	0.1 (middle), 1e-4 (top)
A_ingraph_true	0 = off	True (off)	1 = on	0.1 (middle), 1e-4 (top)
A_corr_no_cutoff	0 = off	True (off)	1 = on	0.1 (middle), 1e-4 (top)
A_ingraph02	0 = off	True (off)	1 = on	0.1 (middle), 1e-4 (top)
A_ingraph05	0 = off	True (off)	1 = on	0.1 (middle), 1e-4 (top)
A_ingraph07	0 = off	True (off)	1 = on	0.1 (middle), 1e-4 (top)
A_ingraph_true	1 = on	False (on)	0 = off	0.1 (middle), 1e-4 (top)
A_corr_no_cutoff	1 = on	False (on)	0 = off	0.1 (middle), 1e-4 (top)
A_ingraph02	1 = on	False (on)	0 = off	0.1 (middle), 1e-4 (top)
A_ingraph05	1 = on	False (on)	0 = off	0.1 (middle), 1e-4 (top)
A_ingraph07	1 = on	False (on)	0 = off	0.1 (middle), 1e-4 (top)
A_ingraph_true	0 = off	False (on)	0 = off	0.1 (middle), 1e-4 (top)
A_corr_no_cutoff	0 = off	False (on)	0 = off	0.1 (middle), 1e-4 (top)
A_ingraph02	0 = off	False (on)	0 = off	0.1 (middle), 1e-4 (top)
A_ingraph05	0 = off	False (on)	0 = off	0.1 (middle), 1e-4 (top)
A_ingraph07	0 = off	False (on)	0 = off	0.1 (middle), 1e-4 (top)
A_ingraph_true	1 = on	True (off)	0 = off	0.1 (middle), 1e-4 (top)
A_corr_no_cutoff	1 = on	True (off)	0 = off	0.1 (middle), 1e-4 (top)
A_ingraph02	1 = on	True (off)	0 = off	0.1 (middle), 1e-4 (top)
A_ingraph05	1 = on	True (off)	0 = off	0.1 (middle), 1e-4 (top)
A_ingraph07	1 = on	True (off)	0 = off	0.1 (middle), 1e-4 (top)
A_ingraph_true	0 = off	True (off)	0 = off	0.1 (middle), 1e-4 (top)
A_corr_no_cutoff	0 = off	True (off)	0 = off	0.1 (middle), 1e-4 (top)
A_ingraph02	0 = off	True (off)	0 = off	0.1 (middle), 1e-4 (top)
A_ingraph05	0 = off	True (off)	0 = off	0.1 (middle), 1e-4 (top)
A_ingraph07	0 = off	True (off)	0 = off	0.1 (middle), 1e-4 (top)

5 Figures

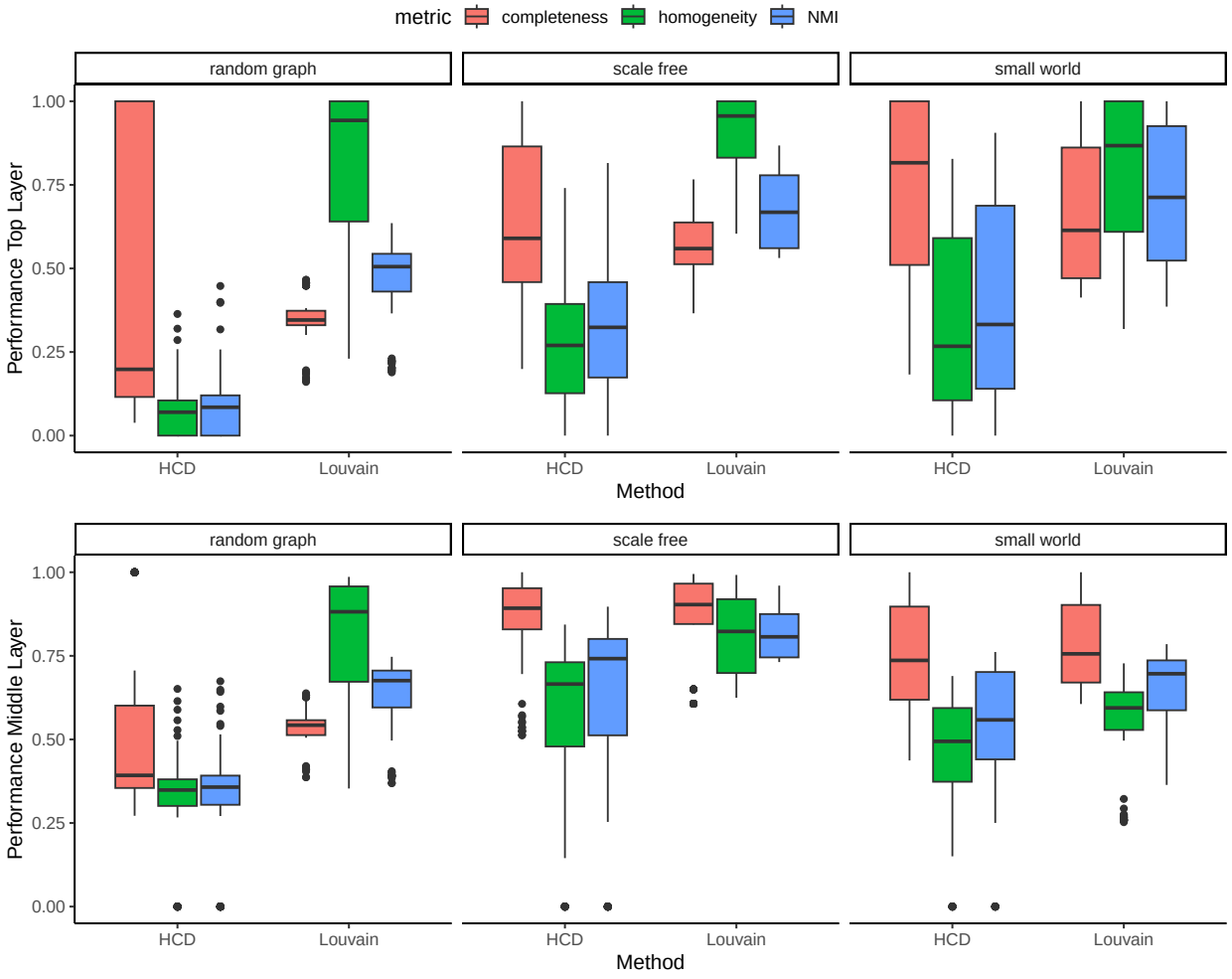

Figure 1: Performance of HCD and Louvain methods in predicting the middle and top layer communities for networks in the intermediate networks dataset. Plots are faceted by the subgraph structure. A box and whisker plot is plotted separately for each performance metric: completeness (orange) homogeneity (green) or NMI (blue) and across the two methods.

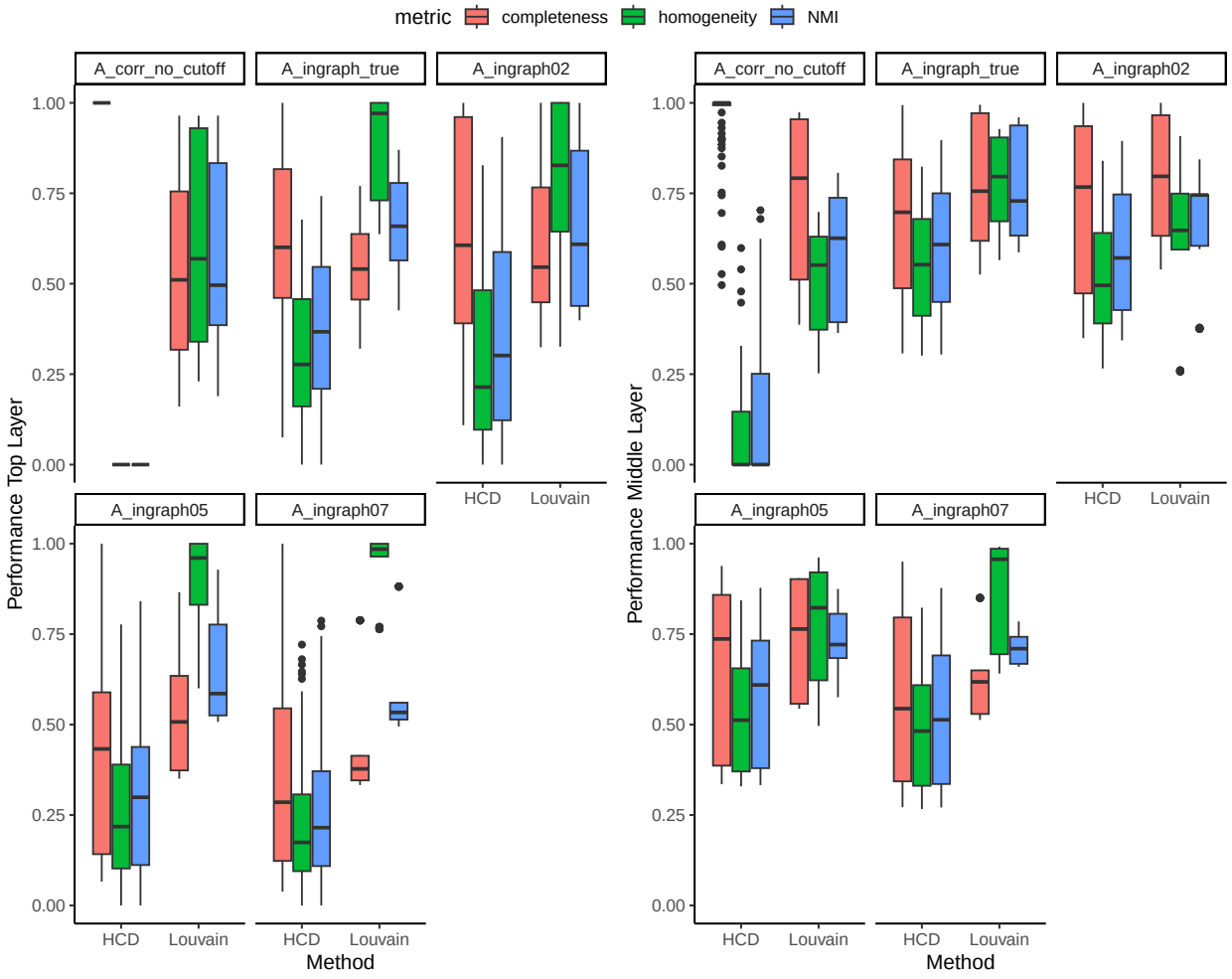

Figure 2: Performance of HCD and Louvain methods in predicting the middle and top layer communities for networks in the intermediate networks dataset. Plots are faceted according to the input graphs outlined in Section "Application to Intermediate Networks". A box and whisker plot is plotted separately for each performance metric: completeness (orange) homogeneity (green) or NMI (blue) and across the two methods.

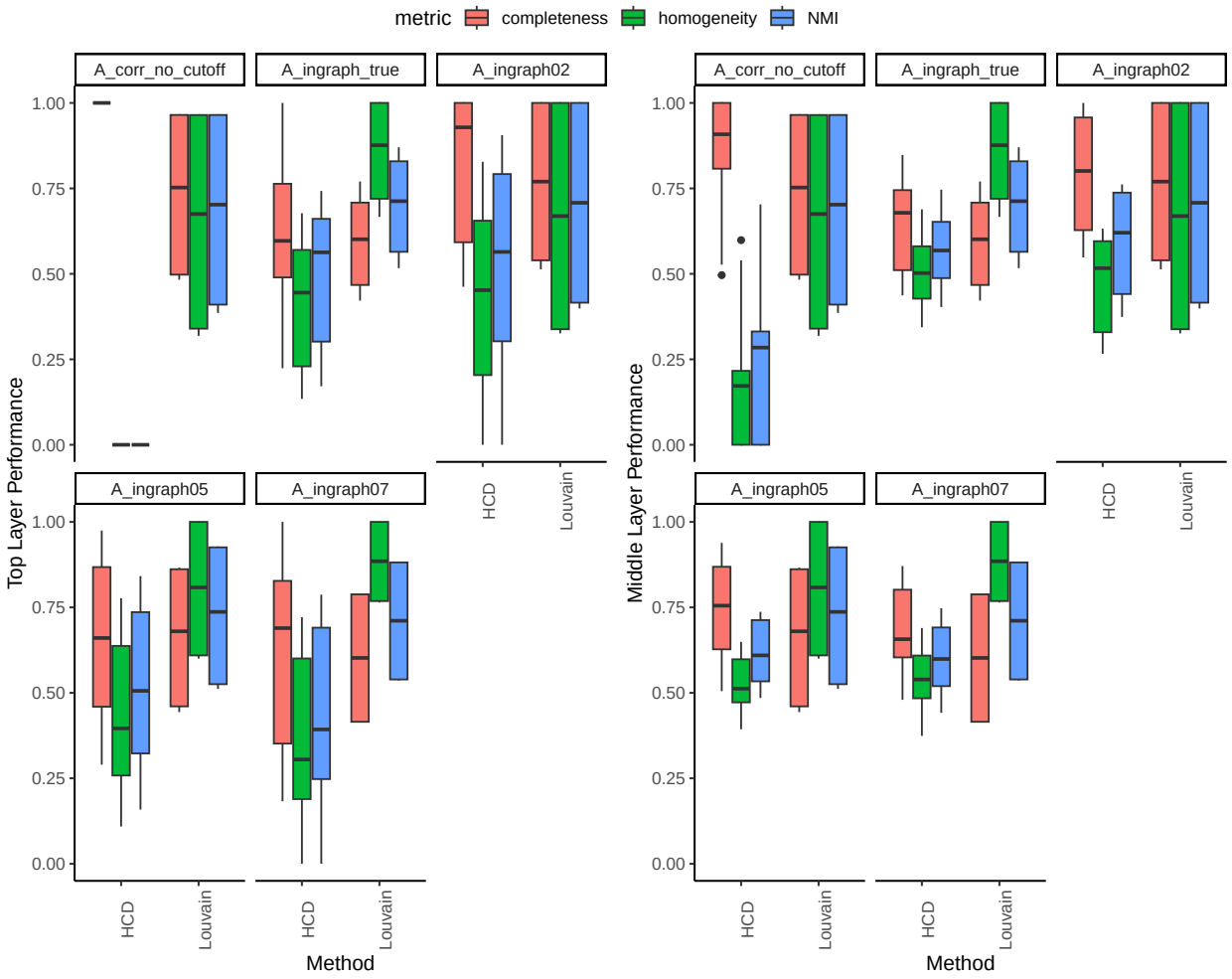

Figure 3: Performance of HCD and Louvain methods in predicting the middle and top layer communities. Results are shown for only intermediate networks generated under the small world architectures. Plots are faceted according to the input graphs outlined in Section "Application to Intermediate Networks". A box and whisker plot is plotted separately for each performance metric: completeness (orange) homogeneity (green) or NMI (blue) and across the two methods.

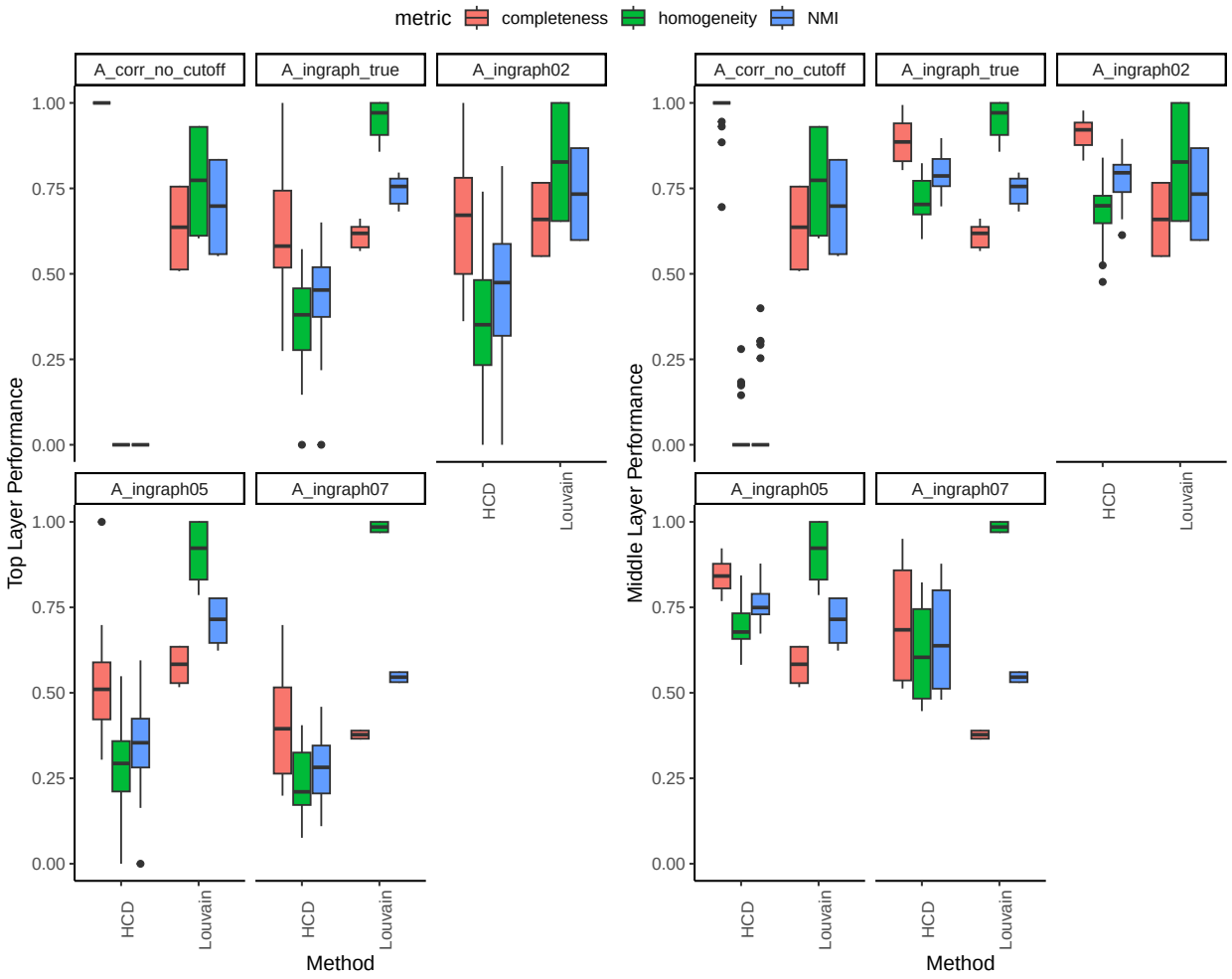

Figure 4: Performance of HCD and Louvain methods in predicting the middle and top layer communities. Results are shown for only intermediate networks generated under the scale free architecture. Plots are faceted according to the input graphs outlined in Section "Application to Intermediate Networks". A box and whisker plot is plotted separately for each performance metric: completeness (orange) homogeneity (green) or NMI (blue) and across the two methods.

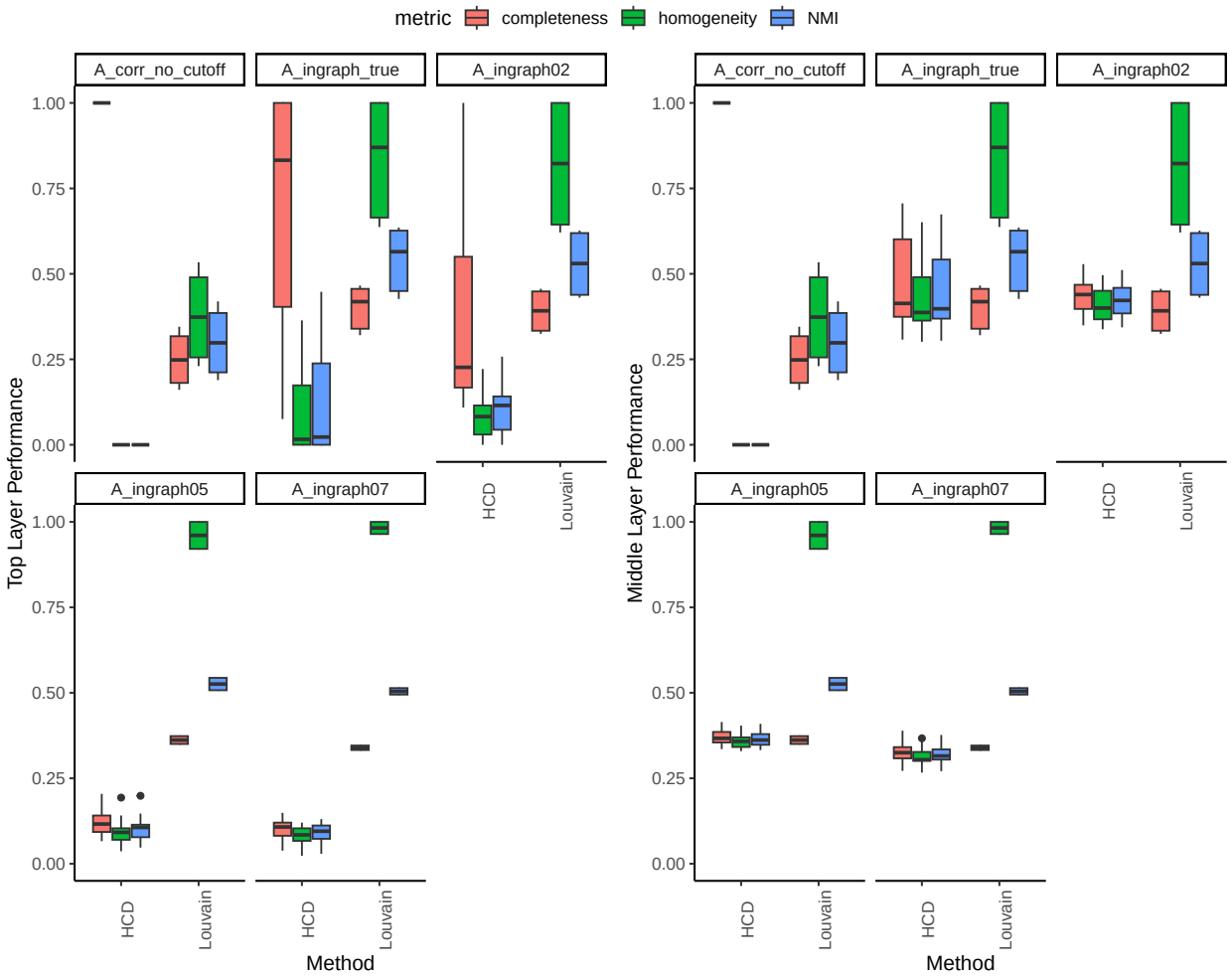

Figure 5: Performance of HCD and Louvain methods in predicting the middle and top layer communities. Results are shown for only intermediate networks generated under the random graph architecture. Plots are faceted according to the input graphs outlined in Section "Application to Intermediate Networks". A box and whisker plot is plotted separately for each performance metric: completeness (orange) homogeneity (green) or NMI (blue) and across the two methods.

References

Barabási, Albert-László, and Eric Bonabeau. 2003. "Scale-Free Networks." *Scientific American* 288 (5): 60–69.

Watts, Duncan J, and Steven H Strogatz. 1998. "Collective Dynamics of 'Small-World'networks." *Nature* 393 (6684): 440–42.