UNIVERSITAT ROVIRA i VIRGILI

A3. Community detection

Complex Networks (MAI)

Professor
GÓMEZ JIMÉNEZ, Sergio

Team members
BEJARANO SEPULVEDA, Edison Jair
edison.bejarano@estudiantat.upc.edu
Walzthöny Kreutzberg, Eric
Eric.Walzthony@estudiantat.upc.edu

Universitat Rovira i Virgili
Avinguda dels Països Catalans, 26, 43007 Tarragona, Spain
Master's degree in Artificial Intelligence
Tarragona, 2021

Contents

Abstract

The detection of communities in complex networks is one of the most important methods for network analysis, as well as to better understand the behavior that they are representing in the system. The term community in complex networks refers to the group of nodes that are most likely connected to each other with nodes in other communities, in other words, communities are dense locally connected subgrahps in a network, and they play a particularly important role in social ,biological, metabolic networks and other contexts to find distribution-based explanations of grades.

While in graph partitioning the number and the size of communities is predefined, in community detection both parameters are unknown. We call a partition a division of a network into an arbitrary number of groups, such that each node belongs to one and only one group. To obtain the communities in the networks, there are different algorithms to produce the partitions. For this project, the Girvan-Newman,Greedy, and Label Propagation were used in order to generate the different communities. The hyperparameters was set to the predefined number (i.e. default values) or they were changed such that the number of communities from the original partition (.clu) file, would match with the "n-best" parameter.

The complete code can be found in our repository below:

https://github.com/EjbejaranosAI/Complex$_N etworks$

Keywords: Complex networks, community detection, optimization of modularity, network analysis, Networks representation.

1 Girvan-Newman algorithm

This is a hierarchical method used to detect communities, basically the algorithm remove edges progressively from the original network,instead of trying to build a measure that give which are the most central, is the rest of the remaining edges that are considering communities.This means that the remaining edges are the ones which are "most likely between" communities.

Ultimately, the result of the algorithm is a dendrogram, which is produced from a top-down approach. The "leaves" of the dendrogram are the individual nodes. This is one of the most classical approaches in the community detection, and one of the simplest we have seen here (in comparison to Label Propagation), which has shown to be effective and robust across all networks node sizes.

1.1 How does it work?

The algorithm proposes the following steps to identify communities:

1. Calculate betweenness for all edges in the network
2. Remove the edges with the highest betweenness
3. Re-calculate betweenness
4. Repeat from step 2 until no edges remain.

   In order to efficiently calculate betweeenness, it is recommended to use the Newman fast algorithm, which occurs in $O(mn)$ time.

1.2 Parameters

The parameters to this algorithm, as seen on NetworkX and iGraph, consist of chosing a "most-valuable edge", which in this case is a function that takes in a graph and then uses this criteria in order to remove the edges. We left this as default, which measures the "edge-betweenness-centrality".

2 Greedy Modularity algorithm

This algorithm, finds the communities by using the greedy modularity maximization. Under the hood, it uses the "Clauset-Newman-Moore" algorithm in order to find the communities with the highest modularity.

2.1 How does it work?

Each node starts in its own community (similar to agglomerative clustering), and then iteratively, joins the pairs of communities which lead to the largest modularity, until a maximum has been reached.

2.2 Parameters

The parameters in this algorithm as defined by the most-common libraries are weight, resolution, cutoff, and best-n. The weight can be passed, when the graph is weighted, which then uses these values in its calculations to know which communities to join. The resolution, defines whether to favor larger or smaller communities (greater than 1 for the latter, inverse for the first). The cutoff parameter, is the minimum number of communities, below which the merging process stops, regardless if modularity has been maximized or not. The last parameter, best-n, this is the maximum number of communities that need to remain, which forces more communities to be formed regardless if the modularity starts to decrease.

3 Label Propagation algorithm

This algorithm, is considered a semi-supervised machine learning algorithm, which assigns labels to previously unlabeled data points. These labels, which have been assigned will then be propagated throughout the entire algorithm. This algorithm is efficient and has some advantages especially the running time and the amount of a-priori information that is required about the structure of the graph. Yet, it produces no unique solution, only an aggregate of solutions.

3.1 How does it work?

An initial node,N_0, carries a corresponding label to a community, C_x. The memberships within a community change depending on the labels of the neighboring nodes. This process has 5 distinct steps:

- Initialize labels in all nodes in the network N_0 the label $C_x(0) = x$
- set $t = 1$
- Organize the nodes in the network in a random fashion, and set it to X
- For each node in X, return the label that occurs with the highest frequency amongst its neighbors. If there are multiple frequencies, select one at random.
- If every node has a label to which the maximum number of their neighbors have, stop. Else, $t = t+1$, and go to Step 3.

3.2 Parameters

The parameters for the algorithm, Label Propagation in NetworkX does not allow for any parameters to be set. While the implementation for iGraph, allows for the definition of membership, which gives the ID of the initial community that each node belongs to. Additionally, a weights parameter can be set,

which defines the weight of each edge. An optional parameter, fixed, which defines which labels should be fixed, as well as the modularity parameter, which should be a real number.

4 Methodology description

For this project, the process was structured in the following parts:

- **Problem Statement:** The problem statement lies with finding the ideal number of communities given a graph. On one hand there are baseline partitions to which one can compare to, on the other hand, some networks do not include a partition. In these cases a comparative decision should be made. Finding communities, is important for different types of networks, and it is measured by different metrics: Normalize Variance of Information (NVI), Normalized Mutual Information (NMI), and Rand Index. These metrics can either consider the original graph the partition, or the partition against the baseline / "original" partition. One later realization, is also the metric that is used when defining a community, and what are the criteria of a community? With the different algorithms: Garvin-Newman, Label-Propagation Greedy Modularity Maximization, we seek to answer which of these methods is most robust with what type of network.
- **Decisions:** The main decisions during this was to structure the code such that it allows a methodological comparison of networks and tools. In such a case we used a OOP approach, and defined a class which implements the testing algorithms for the NetworkX class as well as the iGraph class. Additionally, another point of concern was which tools to use, as some are much slower (NetworkX by about 10x) when compared to other options (iGraph,Graph-tools). Thus, an efficient way to use the slower libraries had to be implemented (via helper functions and generators), it allowed us to minimize compute time for the libraries. The last decision, was to build a common schema, in which the data will be collected. Here a schema was made which takes in the model type (model, real, toy), model-name, the ID of the partition, which method was used (Newman, Label Propagation, Greedy), whether a partition had to be created or it was provided, and then the metrics (NVI, NMI Rand Index) for both NetworkX and iGraph.
- **Software:** The software that was used was python in addition to two other libraries: NetworkX and iGraph. The main manipulation of data was done with numpy and the data wrangling with pandas. The visualizations of the networks were done with the aforementioned libraries as well as with seaborn and matplotlib.
- **Analysis:** Our analysis consisted of comparing the metrics and partitions provided by each library. First, the original partition was compared to the one generated by NetworkX, and then compared to the one by iGraph. Each of these was then measured with their respective metrics from their own modules. In addition to this, we also compare the difference in the metrics between the two modules, as well as for the different graphs (shown as barplots and heatmaps). A quick note on the presentation of the results, when using multi-plots (NxM), the first column will be the original partition, and if there is no original partition, the method implemented by NetworkX's library Community, is used (Louvain Algorithm), to find the best partition, to subsequently compare it to the rest of the partitions.
- **Coordination:** The coordination of this project was done in the following manner:
  - Initial Investigation.
  - Separation of task.
  - Initial tests.
  - For each network, perform the corresponding partition and evaluation.
  - Merge the results.

5 Results

5.1 Toy networks

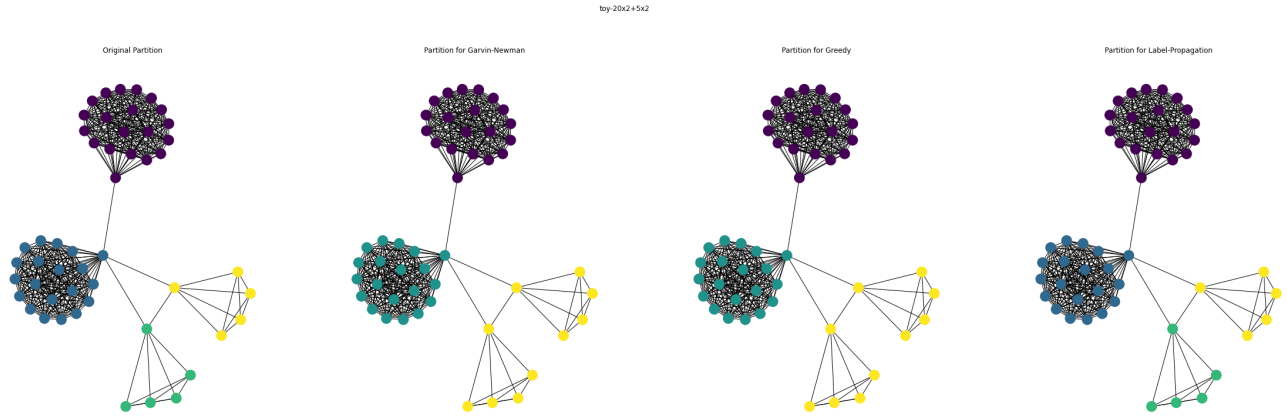

Figure 1 – Toy Network Partitions (20x2+5x2)

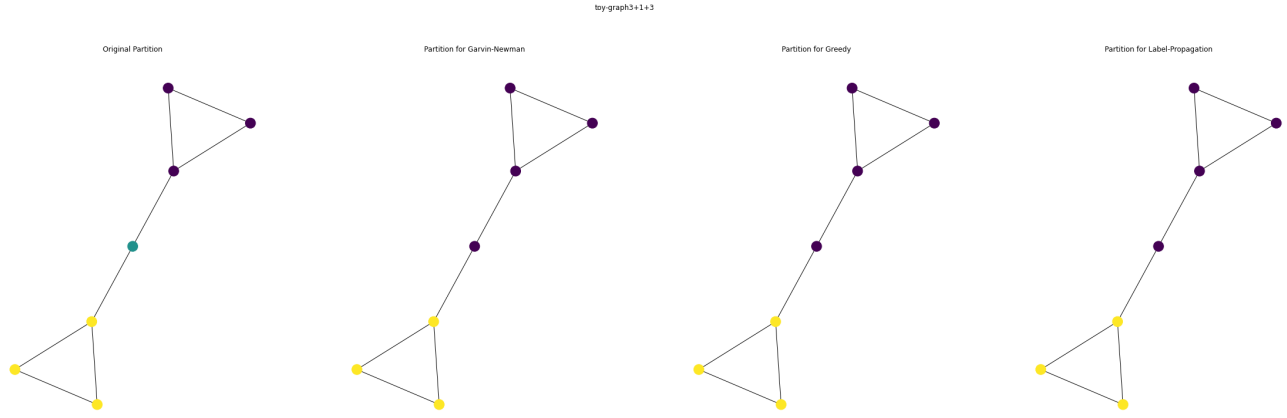

Figure 2 – Toy Network Partitions (3+1+3)

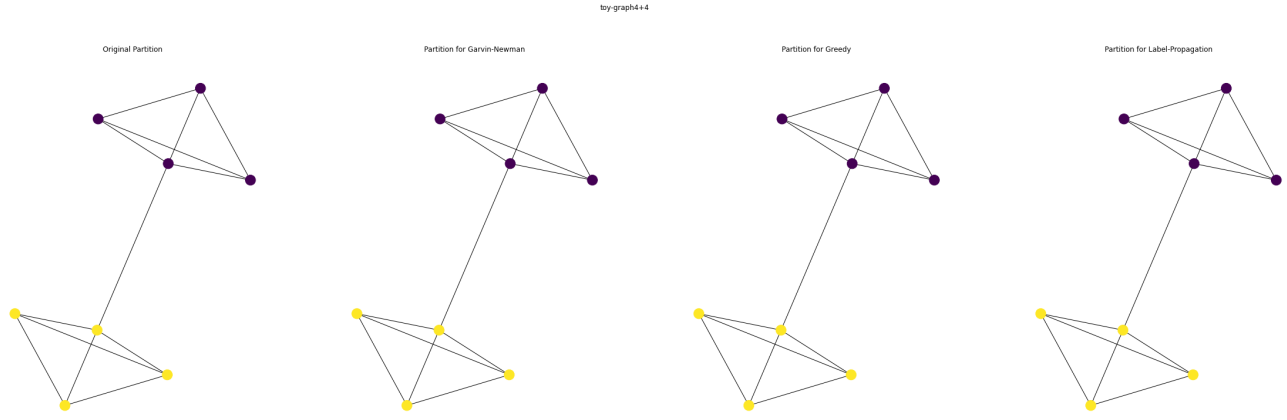

Figure 3 – Toy Network Partitions (4+4)

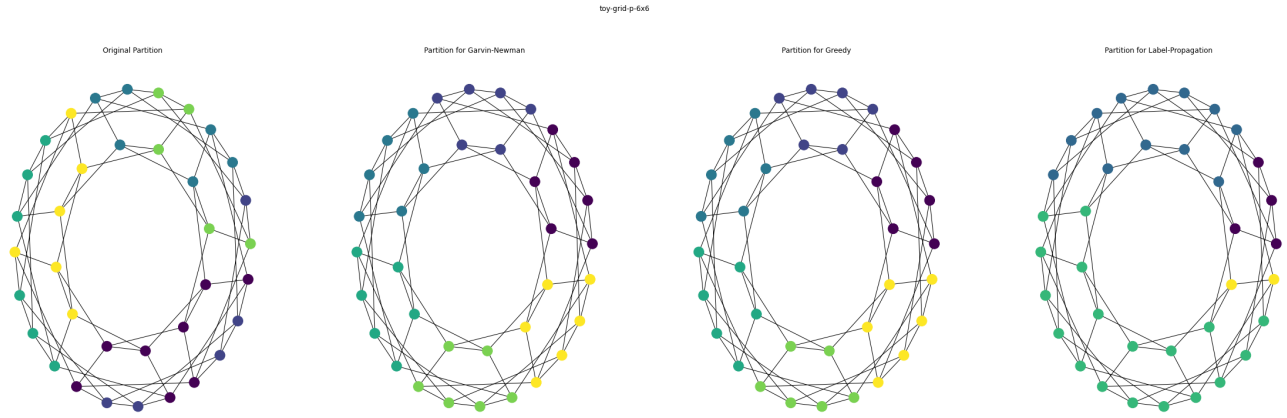

Figure 4 – Toy Network Partitions (Grid-p-6x6)

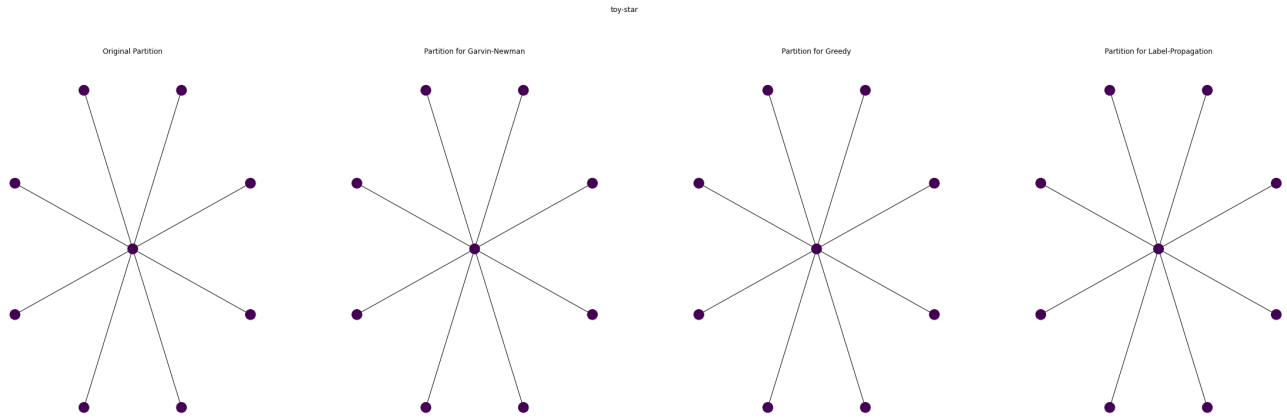

Figure 5 – Toy Network Partitions (star)

5.2 Metrics toy networks

With the toy network we have observed high values for NMI, Rand index, and low values for NVI. Here we can see that most of these are highly connected, such as the Star model for example has only one central node, and comparing it to a partition of that, is not the best comparison (hence the high values aforementioned). We can observe higher values of NVI from both iGraph and NetworkX as the connectedness of the nodes increases, for example the 6x6 grid which has average values for all and a higher value for the Rand index than all the other metrics (specifically the NetworkX variant). This and more graphical comparison between the metrics are shown in Figure 6-11.

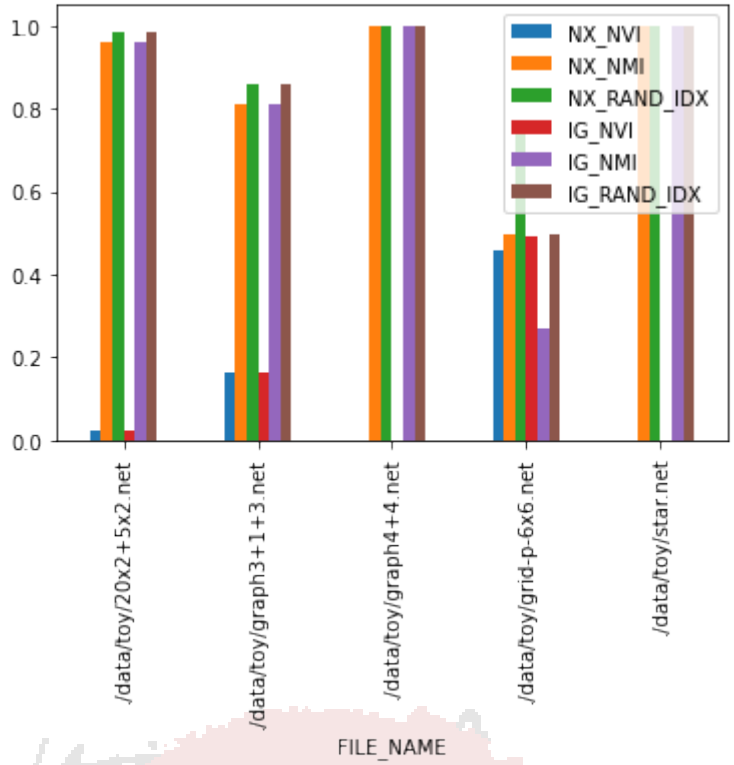

Figure 6 – Barplot - Mean Metrics per Network (file)

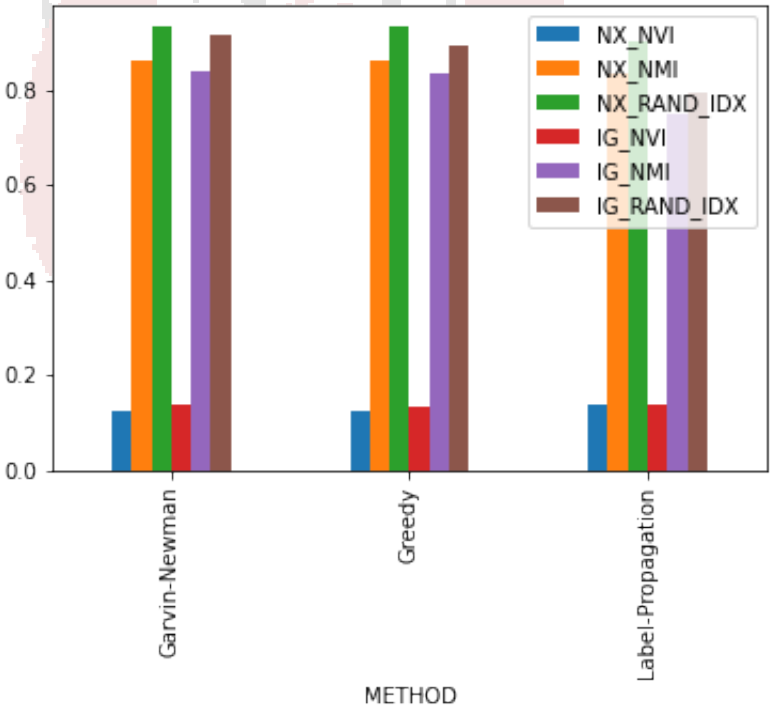

Figure 7 – Barplot - Mean Metric Score per Method (NetworkX iGraph

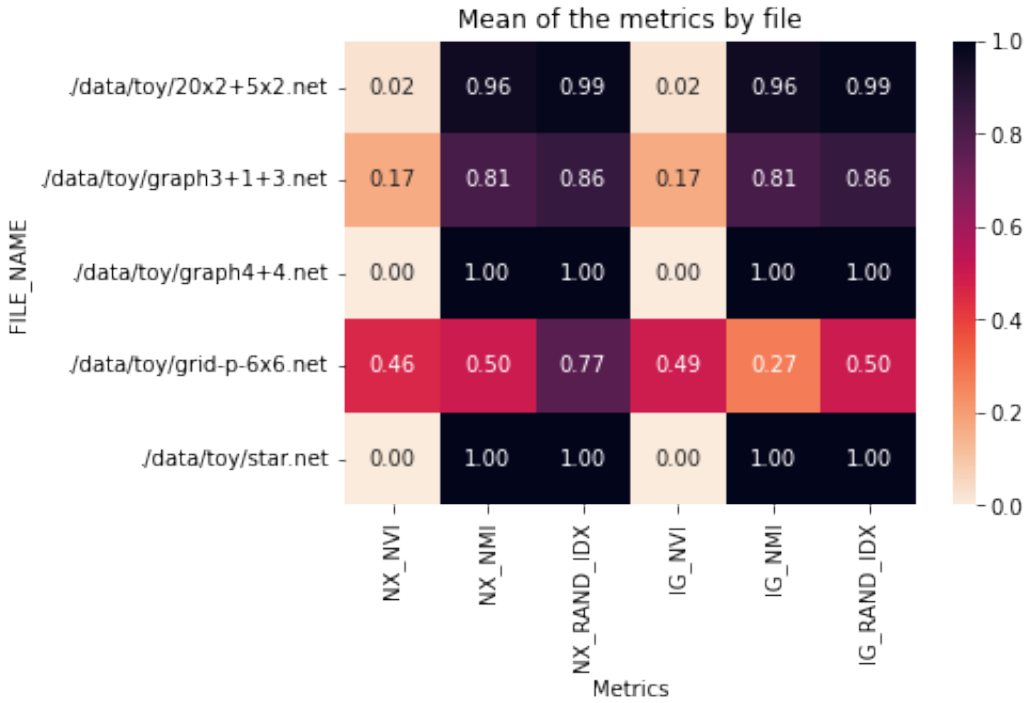

Figure 8 – Heatmap - Metrics against File name

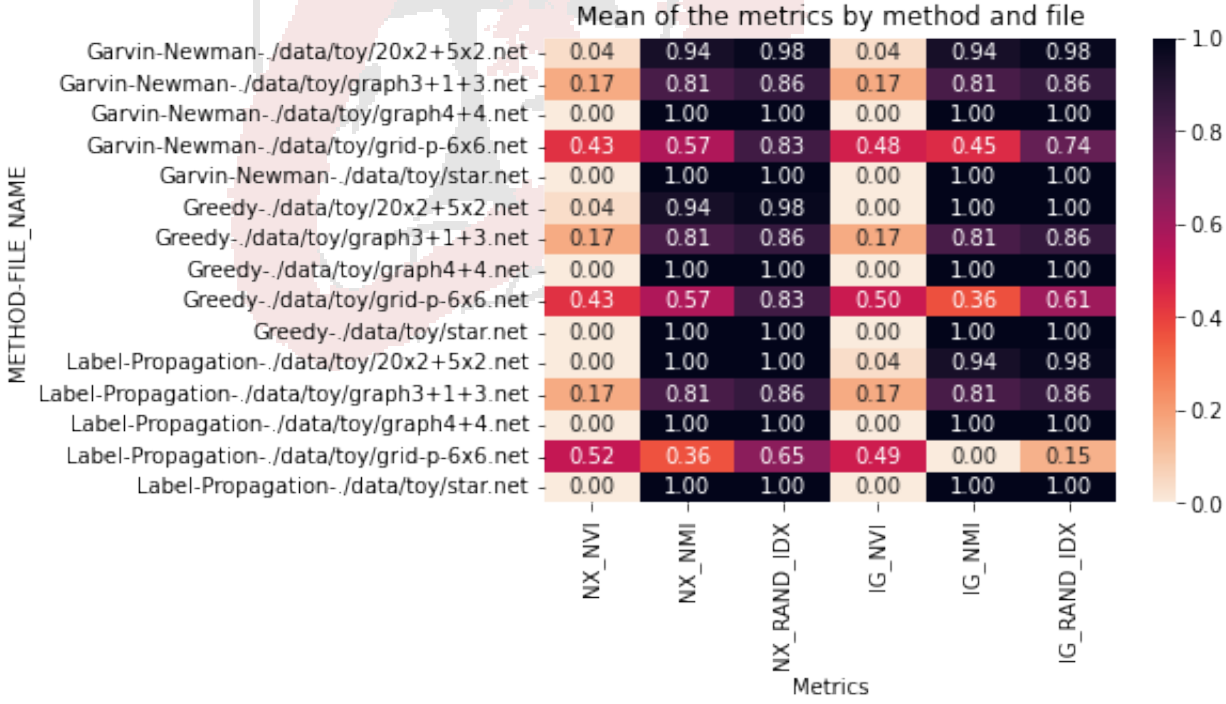

Figure 9 – Heatmap - Metrics against File name and method

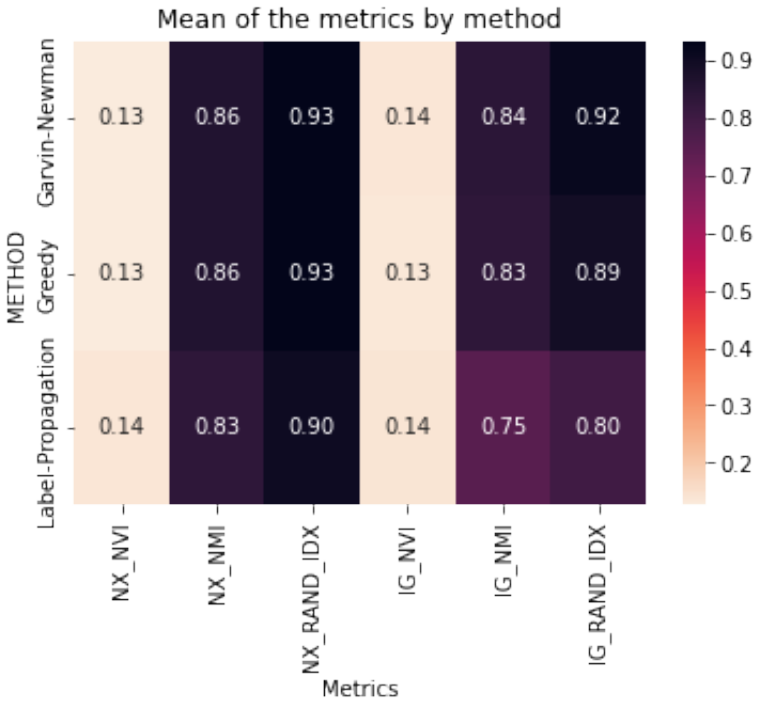

Figure 10 – Mean metrics method for real networks

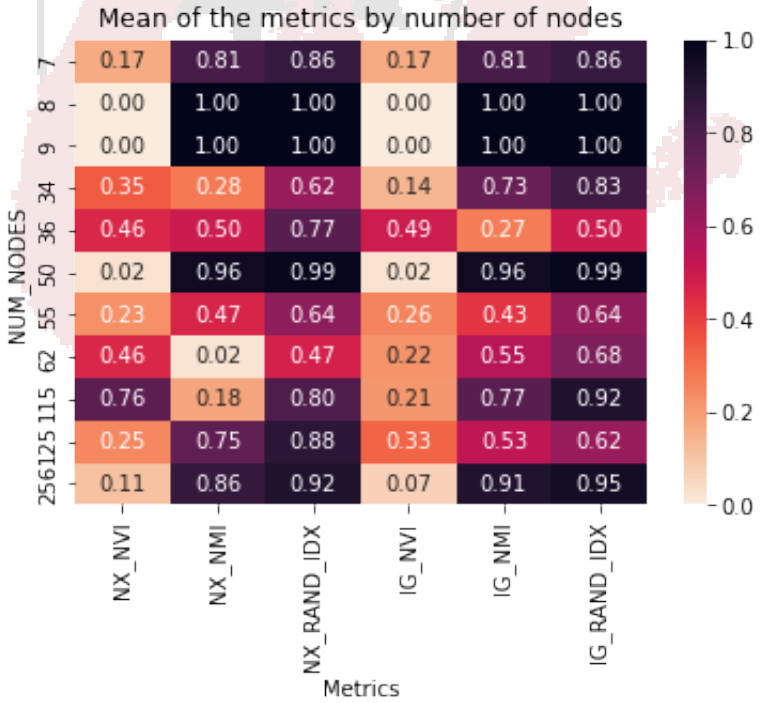

Figure 11 – Mean metrics nodes for real networks

5.3 Real networks

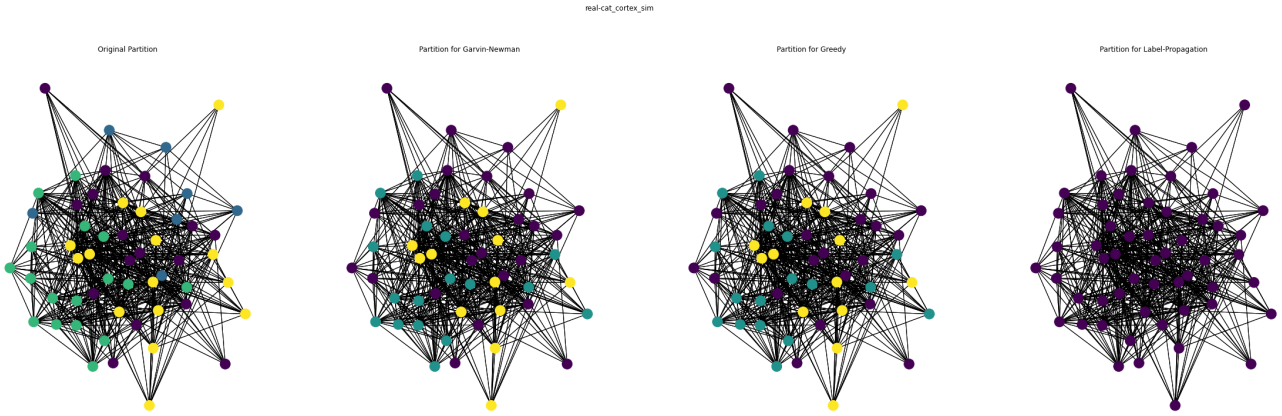

Figure 12 – Real Network Partitions (real cat cortex)

Figure 13 – Real Network Partitions (real dolphins)

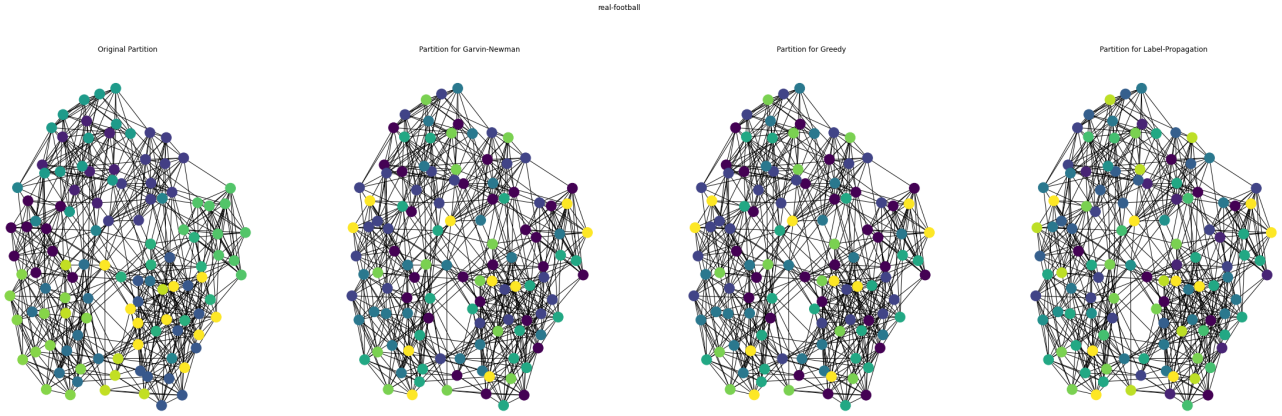

Figure 14 – Toy Network Partitions (real football)

Figure 15 – Real Network Partitions (real zachary unwh)

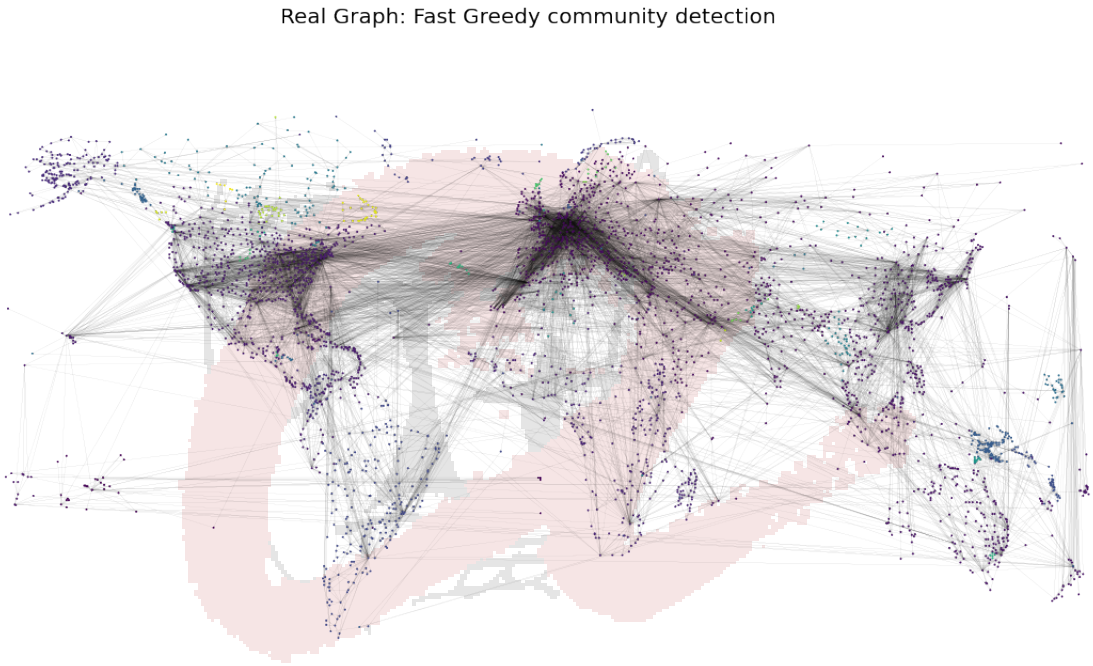

Figure 16 – Real Network Airports Partitions (Fast greedy)

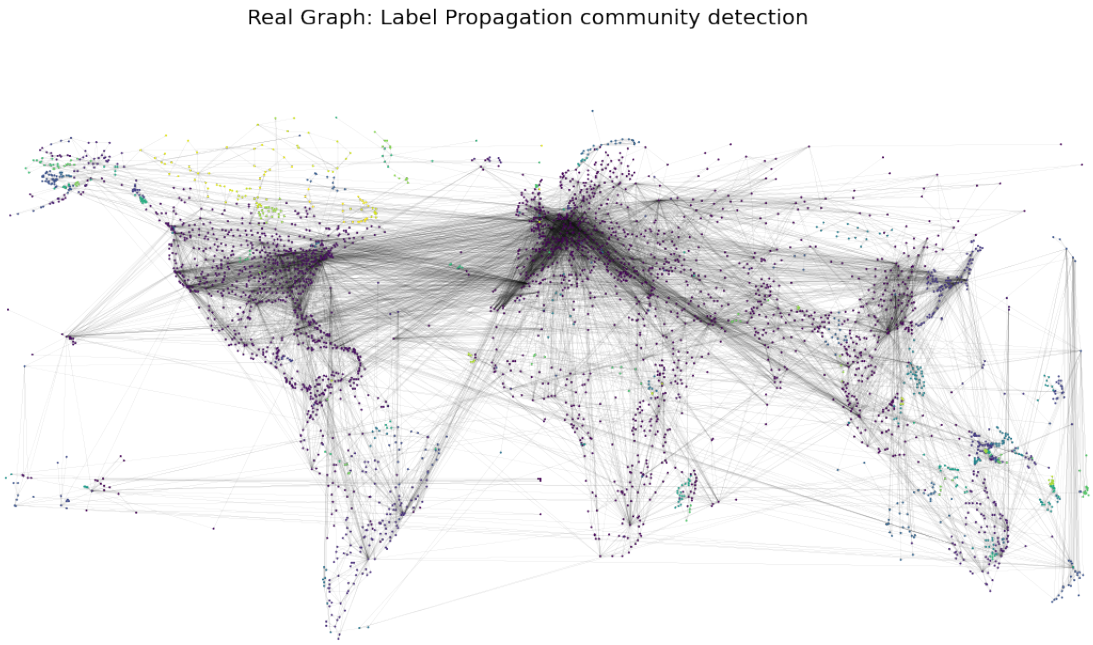

Figure 17 – Real Network Airports Partitions (Label propagation)

5.4 Metrics real networks

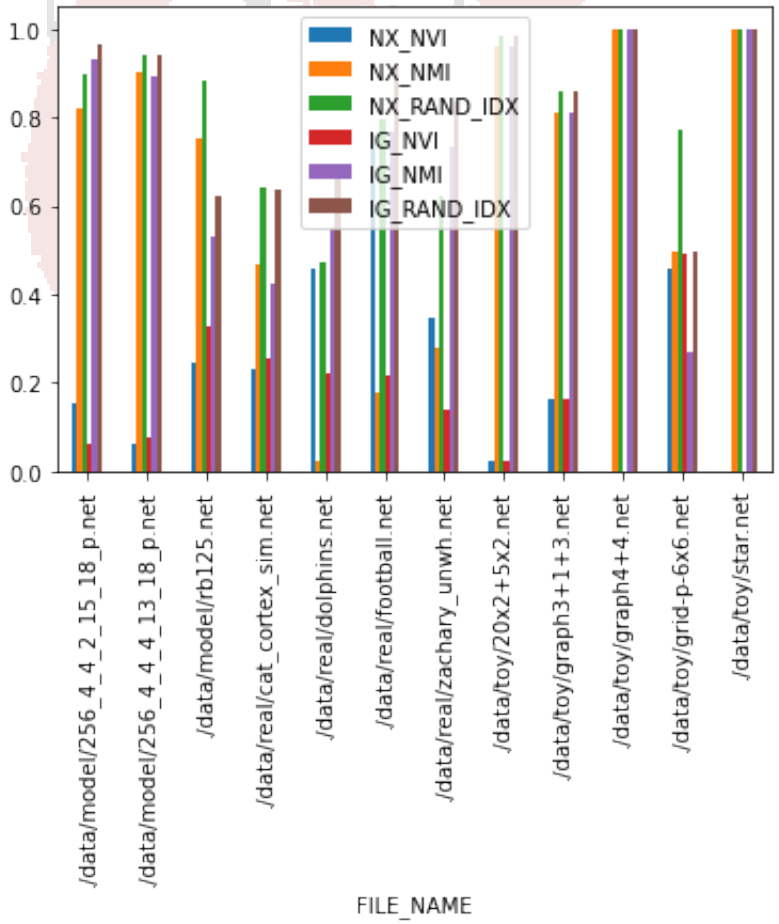

Figure 18 – Barplot - Mean Metrics per Network (file)

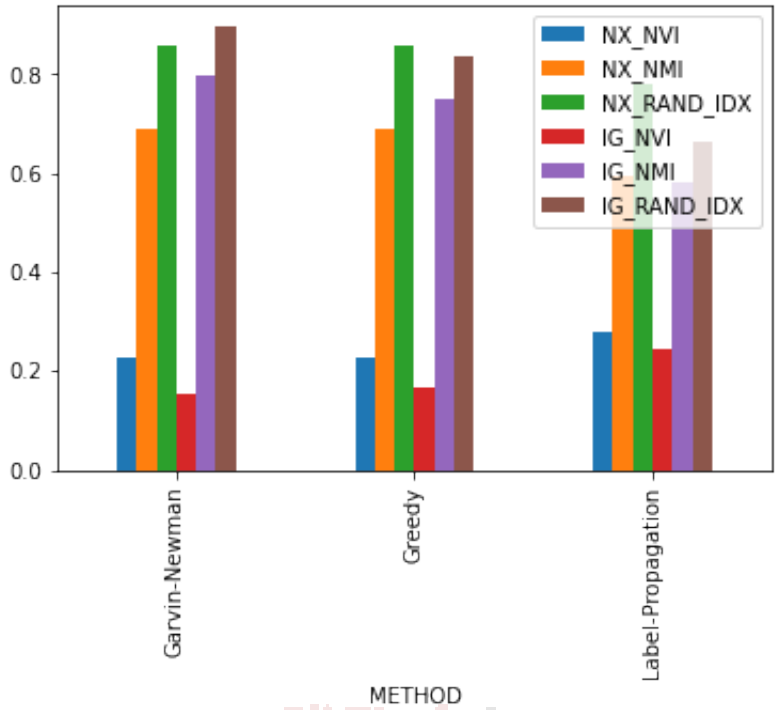

Figure 19 – Barplot - Mean Metric Score per Method (NetworkX iGraph

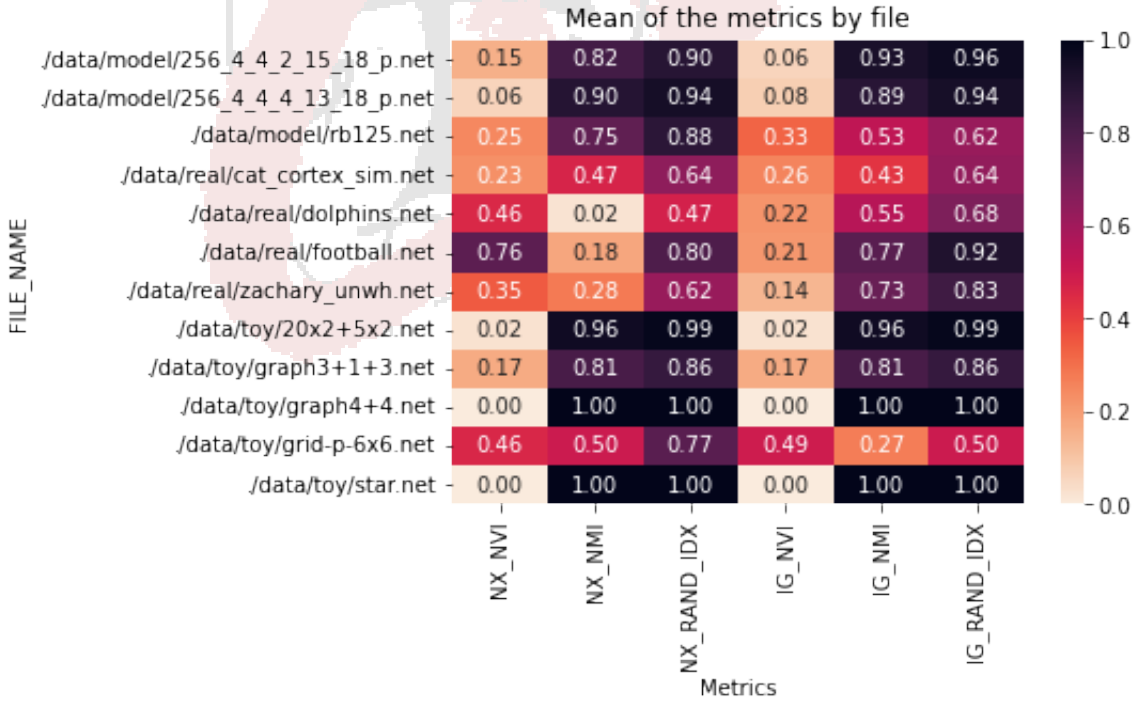

Figure 20 – Heatmap - Metrics against File name

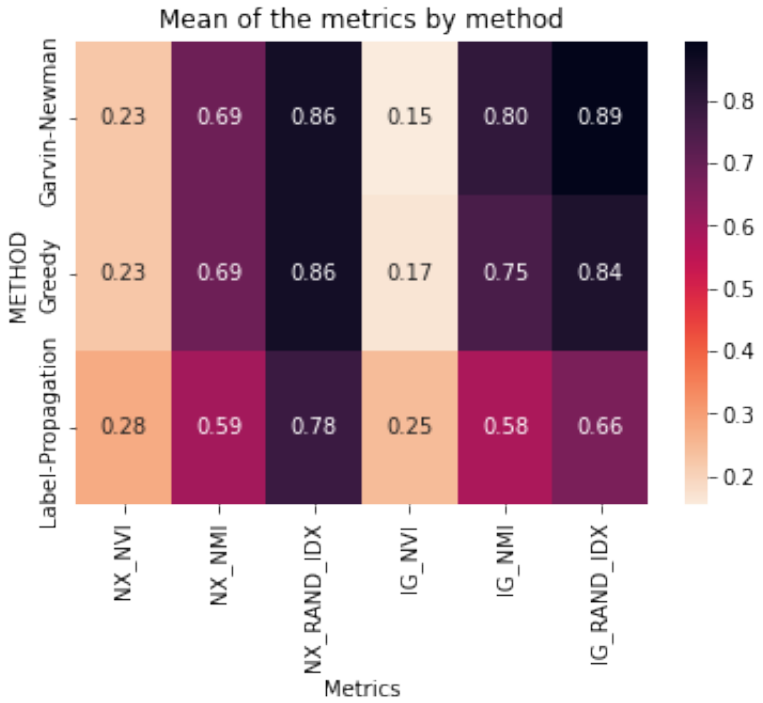

Figure 21 – Mean metrics method for real networks

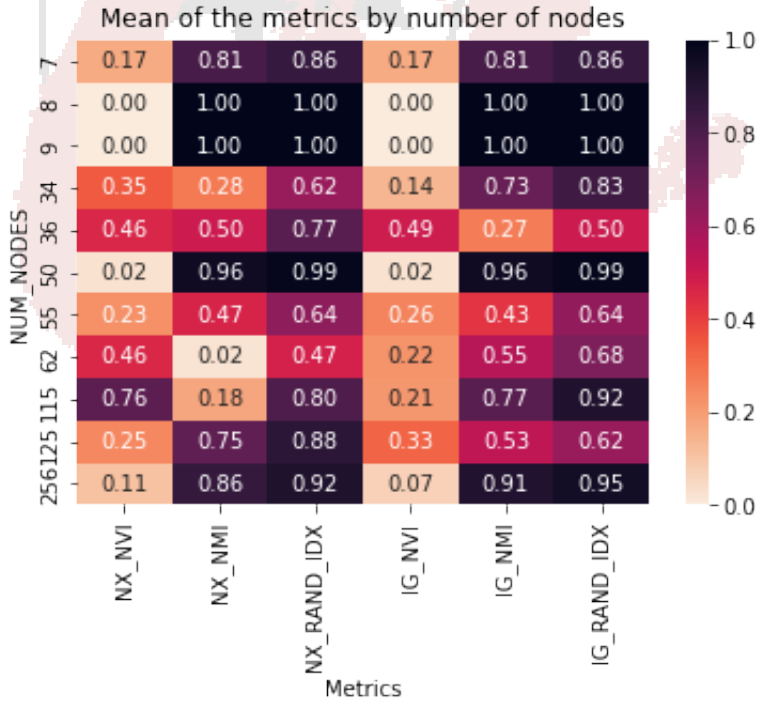

Figure 22 – Mean metrics nodes for real networks

5.5 Model networks

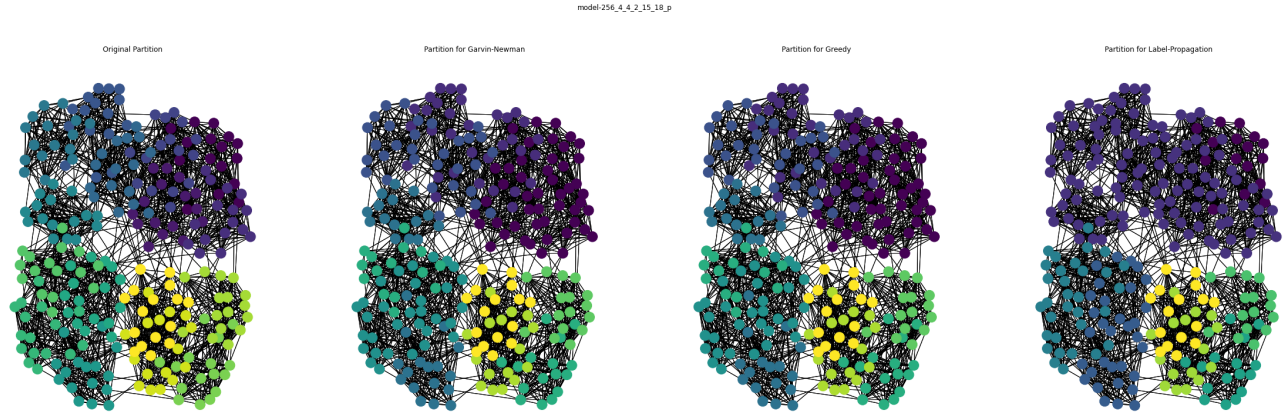

Figure 23 – Model Network Partitions (256-4-4-2-15-18)

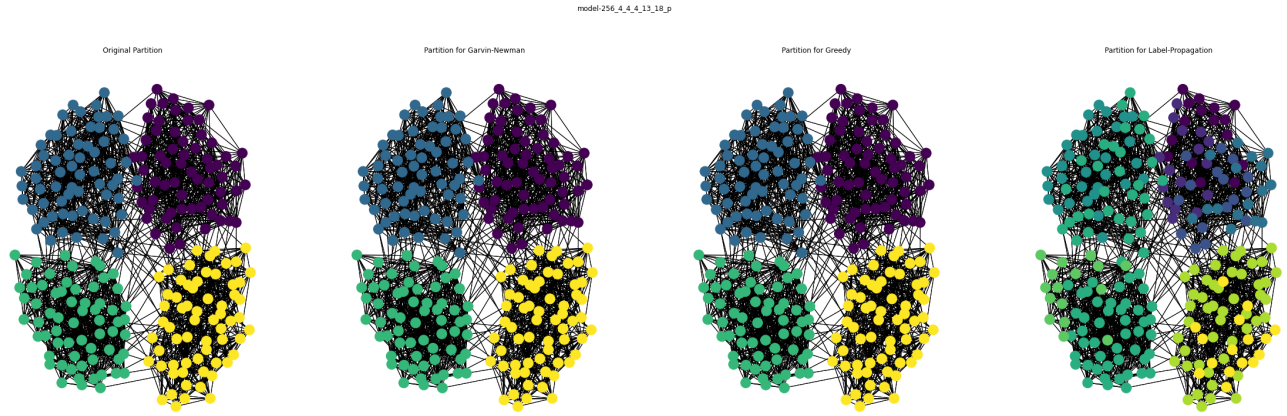

Figure 24 – Model Network Partitions (256-4-4-4-15-18)

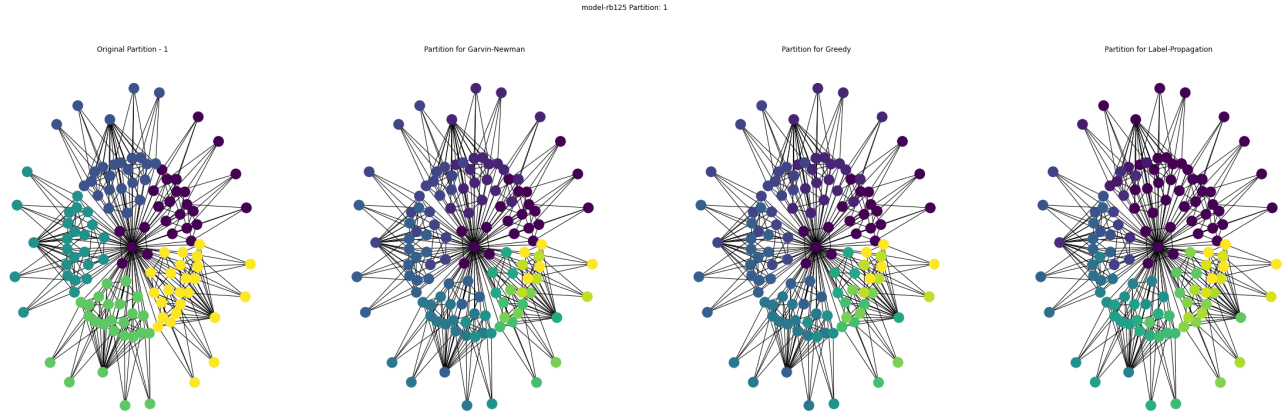

Figure 25 – Model Network Partitions (Rb125) - Partition 1

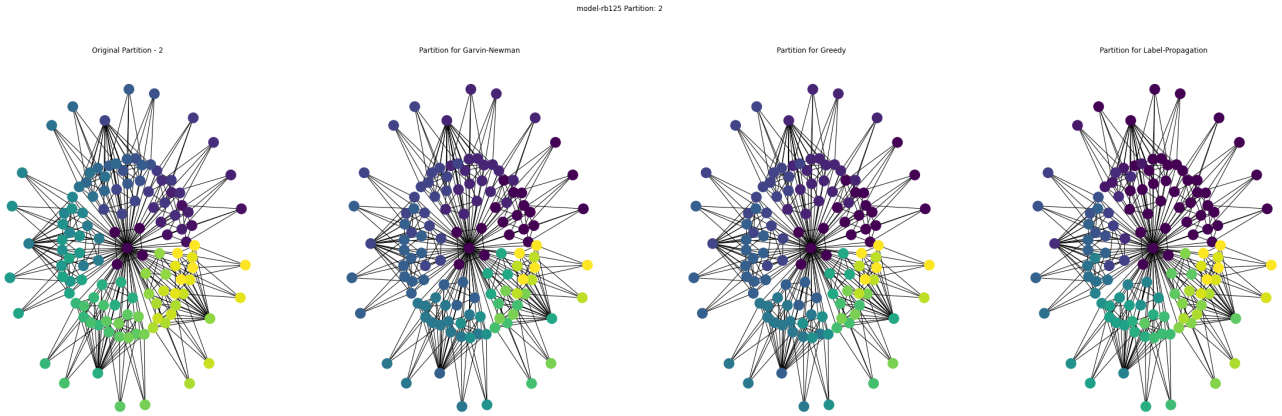

Figure 26 – Model Network Partitions (Rb125) - Partition 2

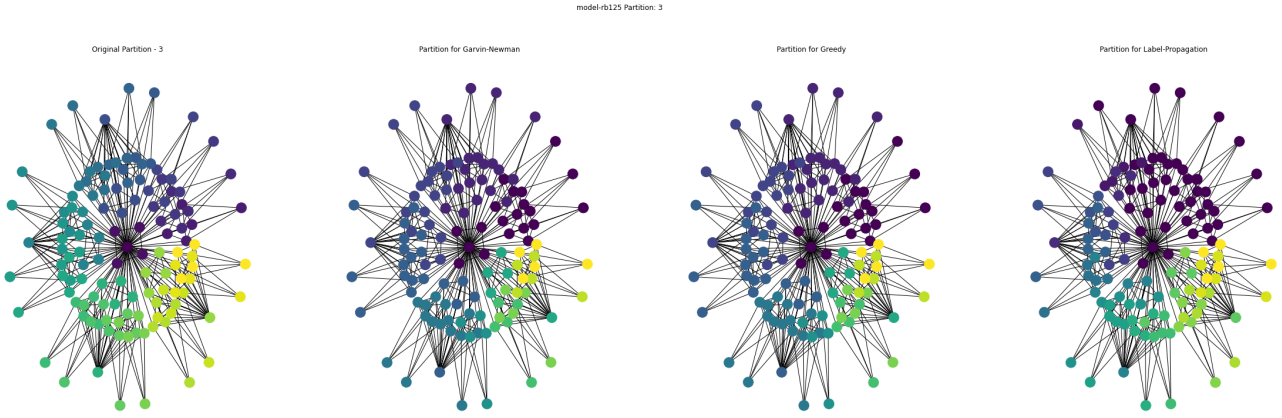

Figure 27 – Model Network Partitions (Rb125) - Partition 3

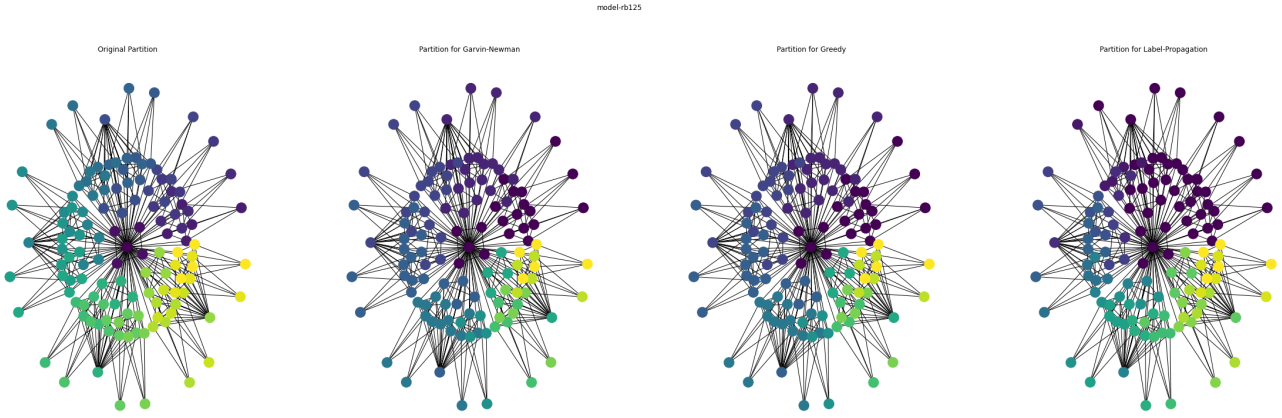

Figure 28 – Model Network Partitions (Rb125) propagation

5.6 Metrics model networks

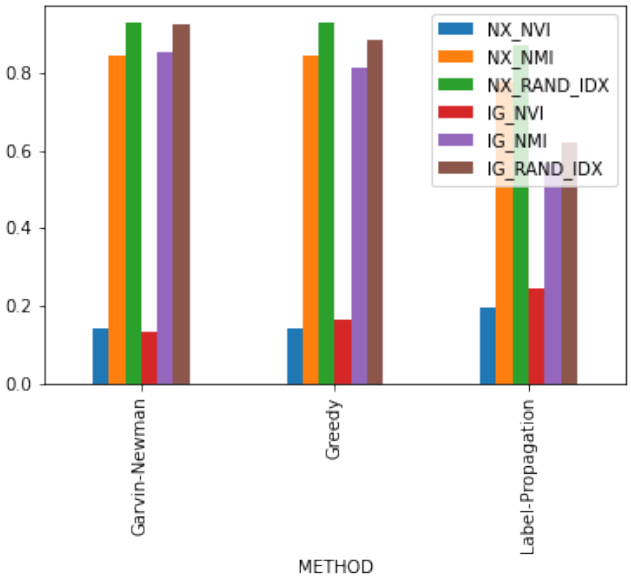

Figure 29 – Bar mean metrics nodes for model networks

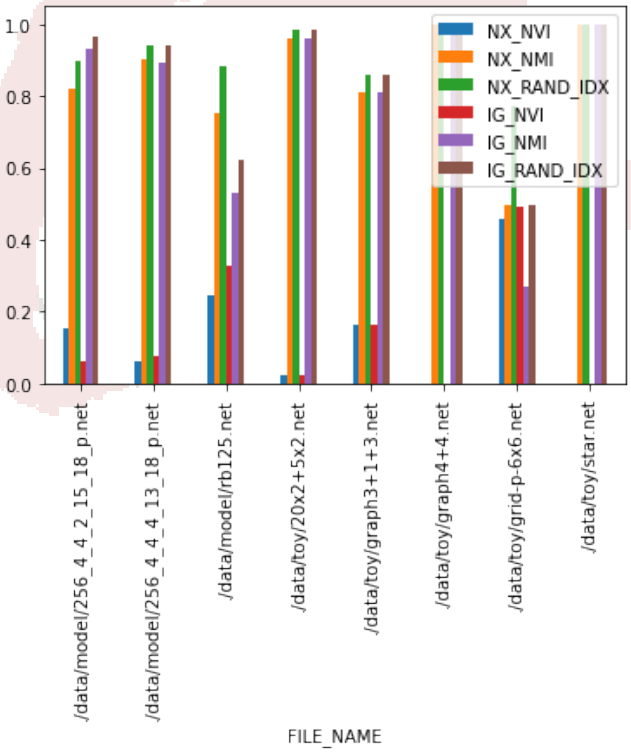

Figure 30 – Bar mean metrics nodes for model Networks 2

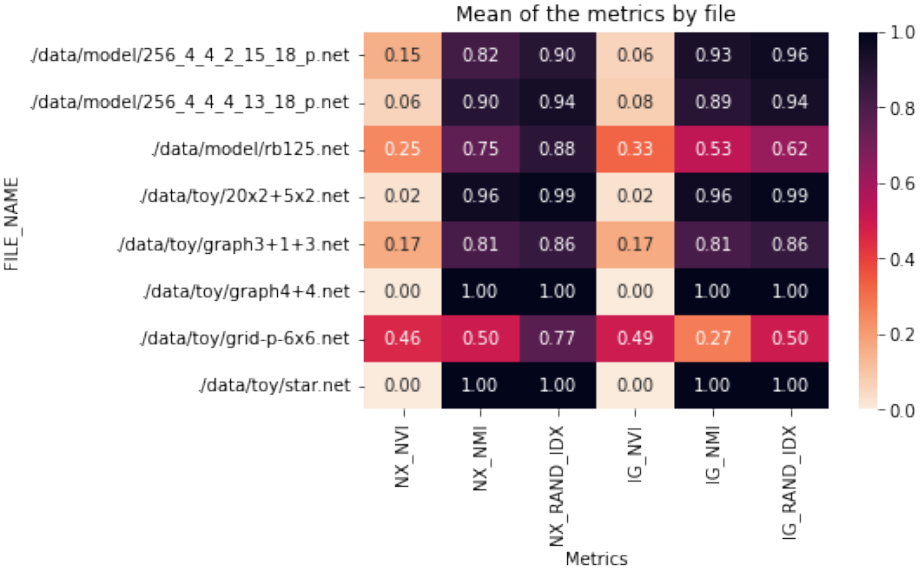

Figure 31 – Bar mean metrics file for model Networks

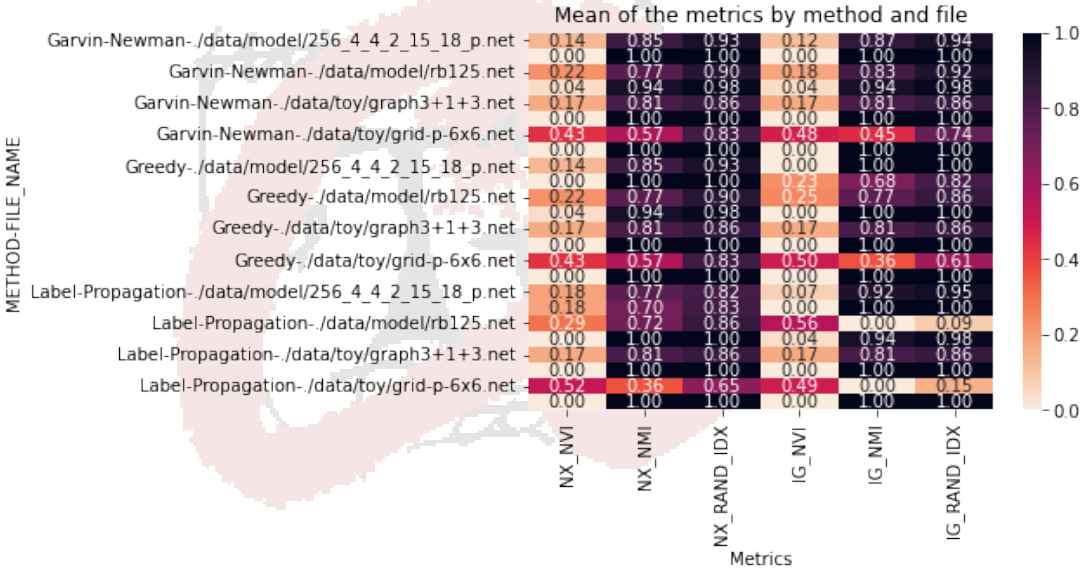

Figure 32 – Bar mean metrics method and file for model Network

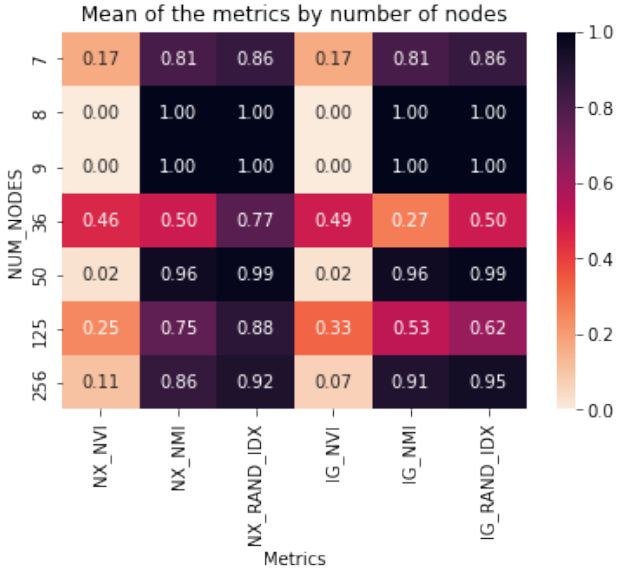

Figure 33 – Mean metrics nodes for model Network

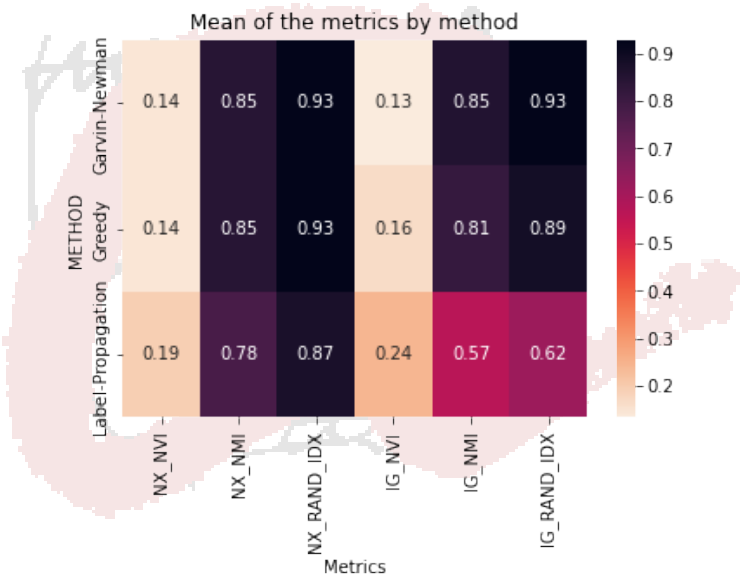

Figure 34 – Mean metrics method for model Network

6 Discussion

The experimental results, are shown from Figure 16 to 26, some above others in the Appendix. The initial problem statement, aims at finding the communities which have the best metric. In this case, the best metrics are 3 distinct ones: NVI, NMI Rand Index. The Normalized Variation of Information, is defined as ameasure of distances between two clustering elements (partitions). A higher value of NVI would correspond to a higher sharing of information, which can also be contrasted to the amount of information lost and gained while changing from one C to $C^{'}$. On the other hand NMI, Normalized Mutual Information, tells us the reduction in the entropy of class labels that we achieve, if we know the actual cluster labels. A higher value in NMI is preffered as it is an indication of a quality clustering. Additionally, it is a method that excels in the presence of class labels. Since it is a two-step approach, where we first split according to clusters and then split according to class labels and we look at the exchange of information between them or agree with each other. This metric increases as the number of clusters increases. Lastly, the Rand index is a way to compare the similarity between two different clustering methods. It compares the number of times a node corresponds to a specific cluster, the times

a node belongs to different clusters and the number of unordered pairs. A lower value indicates that the clustering methods are not aligned, and the contrary means a high alignment. We have define the metrics to be used and we have seen that a large value of NMI, NVI, and Rand index are preffered, which would mean that the clustering methods are similar to the original partitions.

We can observe a fluctuation of the shows the absolute difference between NetworkX and iGraph results for each of the metrics. We see that for NVI, across all the 3 methods there is a consistent difference (0.08 - 0.10), between the result of each of the methods. Moreover, for the NMI, we can see that there is a larger difference between the modules with the Label Propagation. It is seen that the difference (0.27) is the largest amongst all metrics. This can be attributed that Label Propagation is a stochastic process, in which each node is given a random unique label at the beginning, which is constant throughout the entire algorithm. Since within the parameters of this model, a random seed can be given, it makes sense that the values fluctuate. More experimental results would be needed (multiple iterations) in order to determine whether this value stabilizes at the given range. Lastly, the rand score is the one that is the largest for Label Propagation and has the difference amongst the methods. This can be seen that the other methods Greedy and Girvan-Newman, do not depend on stochastic processes and probably use a very similar implementation, such that their results are very close.

The bar plot below shows which method has the highest difference for each of the metrics. We can clearly see that stochastic processes such as Label Propagation would result in a higher inaccuracy between different models. This can be further optimized and reduced by using consistent random seeds during the testing environment as well as increasing the number of trials to have an average. On the other hand, we see that the Greedy Modularity Maximization algorithm is the one that performs the best in this area, such that it has the smallest difference between all the methods metrics.

Which shows the metrics per number of nodes, or the complexity of it. We can see that there is no real pattern in how the metrics perform, as there are some where the NVI does not prove to be a good metric since no result is obtained. On the other hand we can see that NMI and Rand Index, are robust ones which are able to be used across varying network sizes. This can be attributed to how they are calculated, as they are based of the Confusion Matrix (Rand Index) and NMI from the shared information between two clusters.

Lastly, in the Figure 44, we can see the final metrics, where we can see that the highest modularity is achieved of models with the largest number of nodes and being a mixture of real and model types. From this we can also infer that the lowest modularity is seen in the toy dataset, and the highest in the real networks. In the middle, lie the model networks.

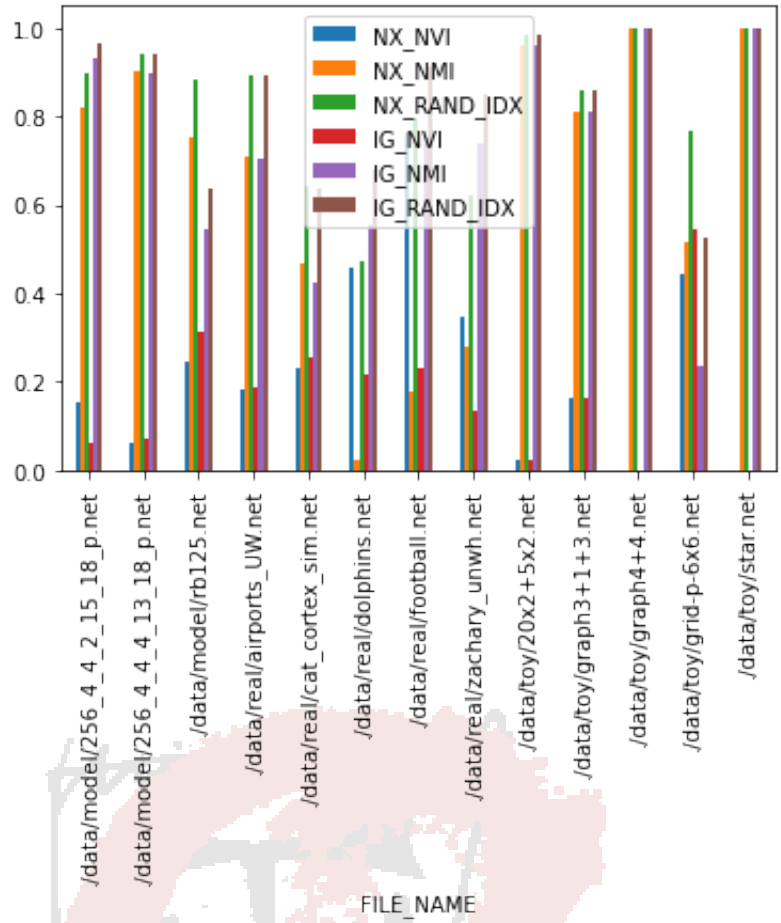

Figure 35 – Bar mean metrics

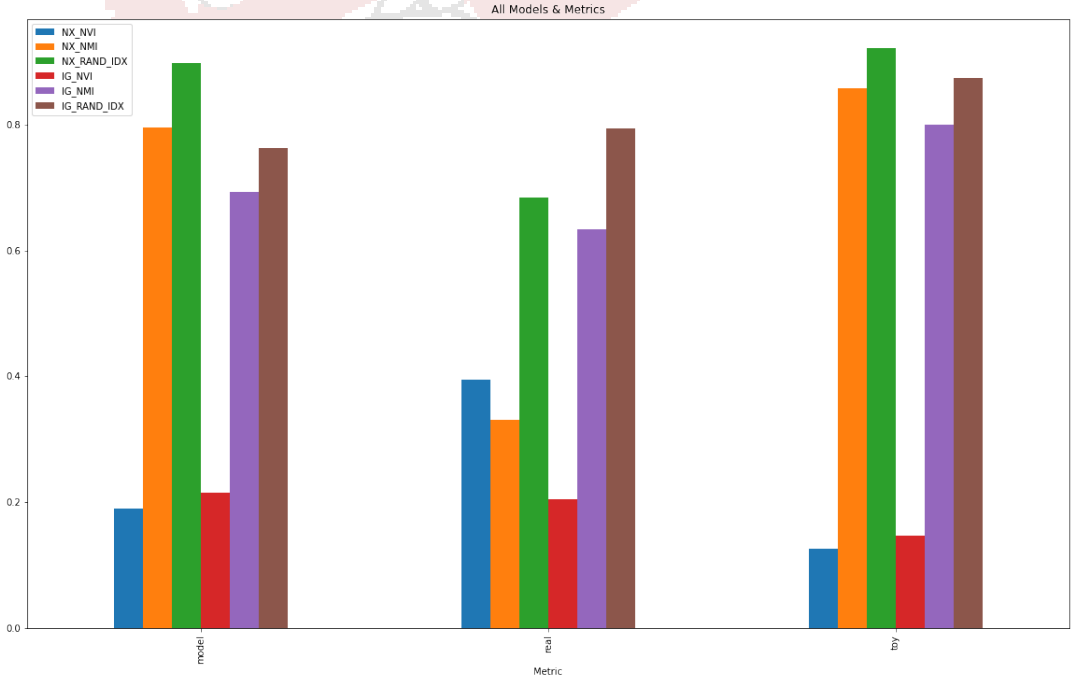

Figure 36 – Barall model type metrics

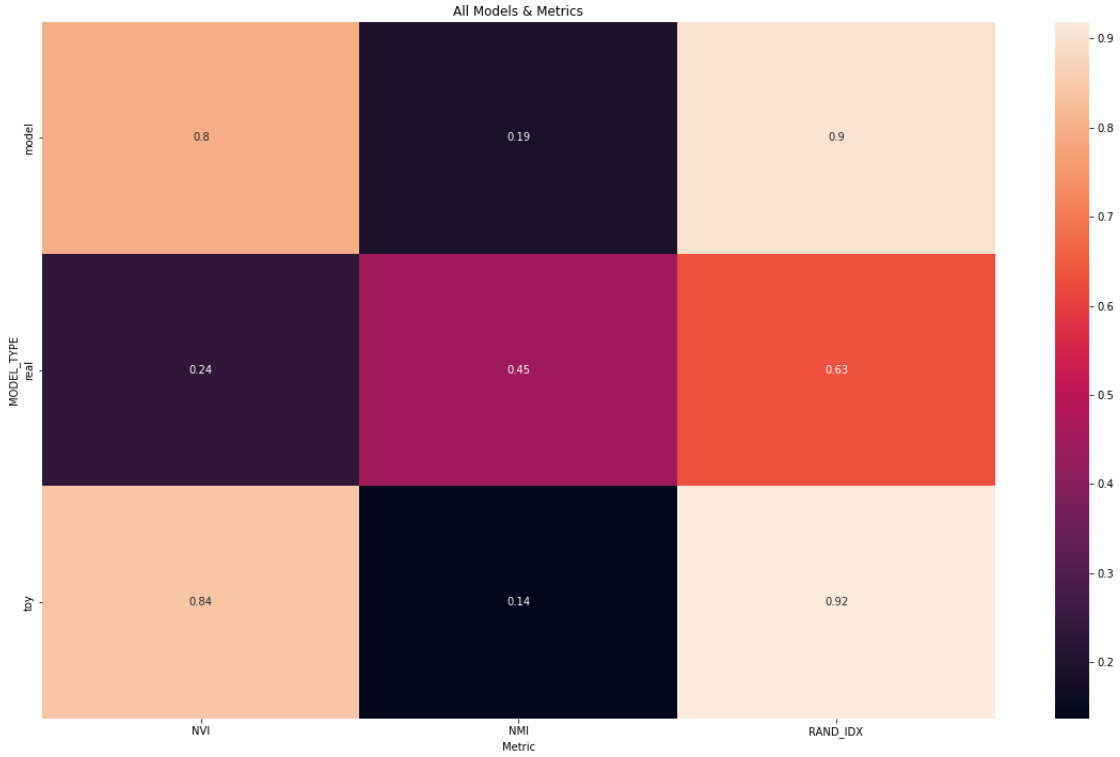

Figure 37 – All model type metrics

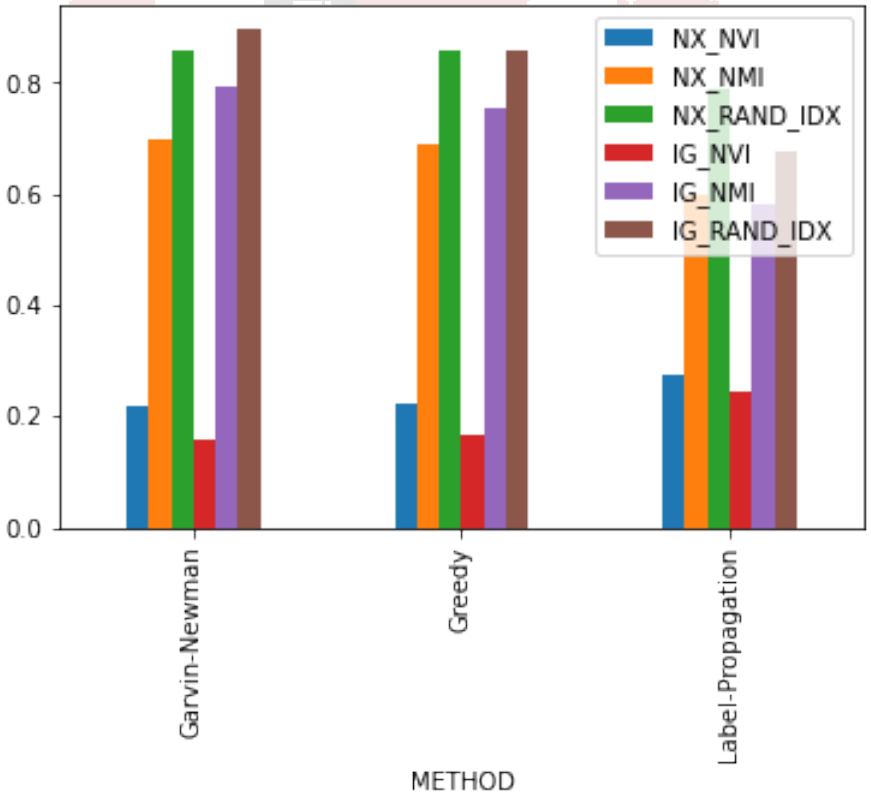

Figure 38 – Bar mean metrics nodes

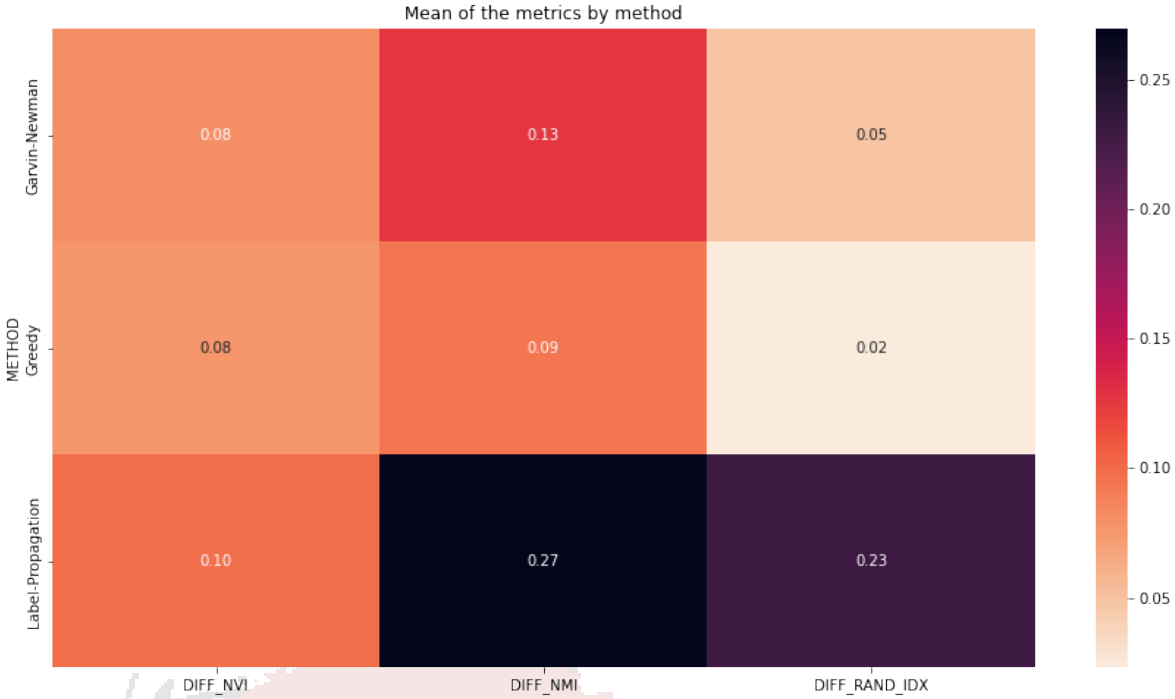

Figure 39 – Metrics by method

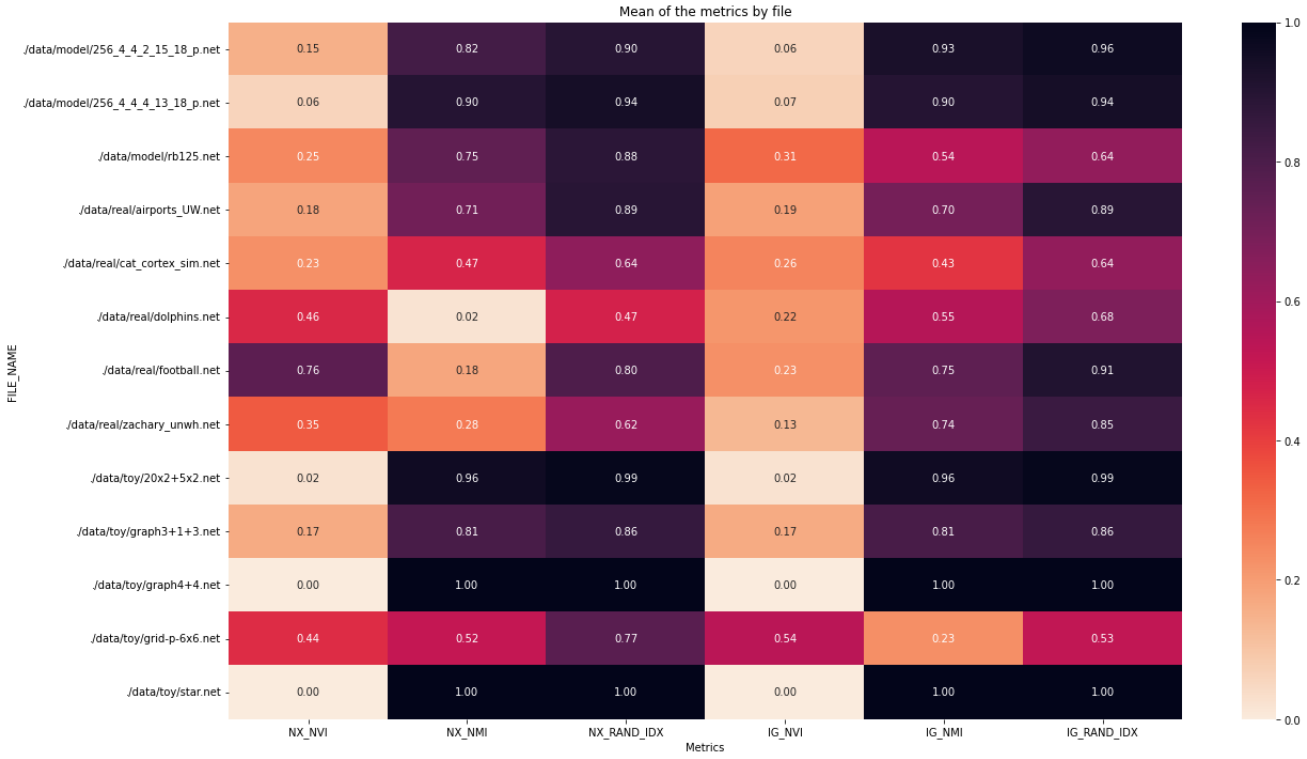

Figure 40 – Mean metrics by file

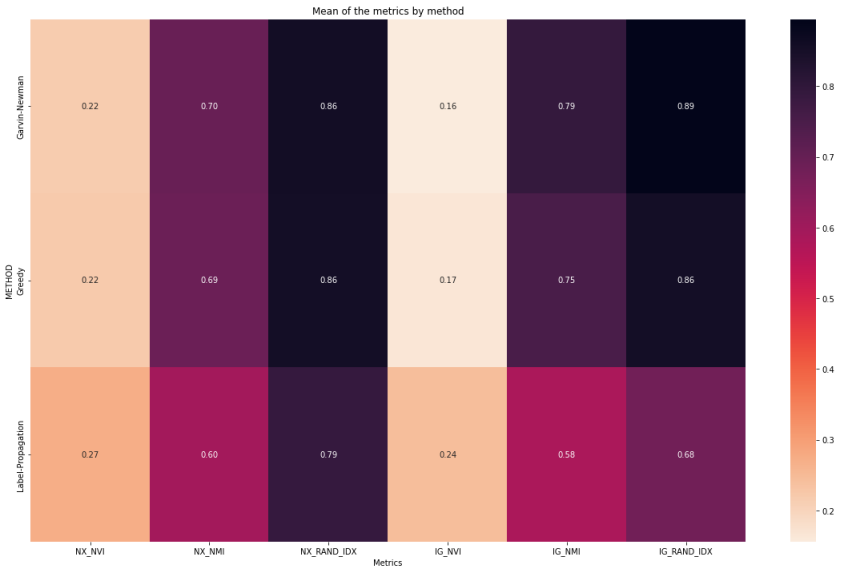

Figure 41 – Mean metrics by method

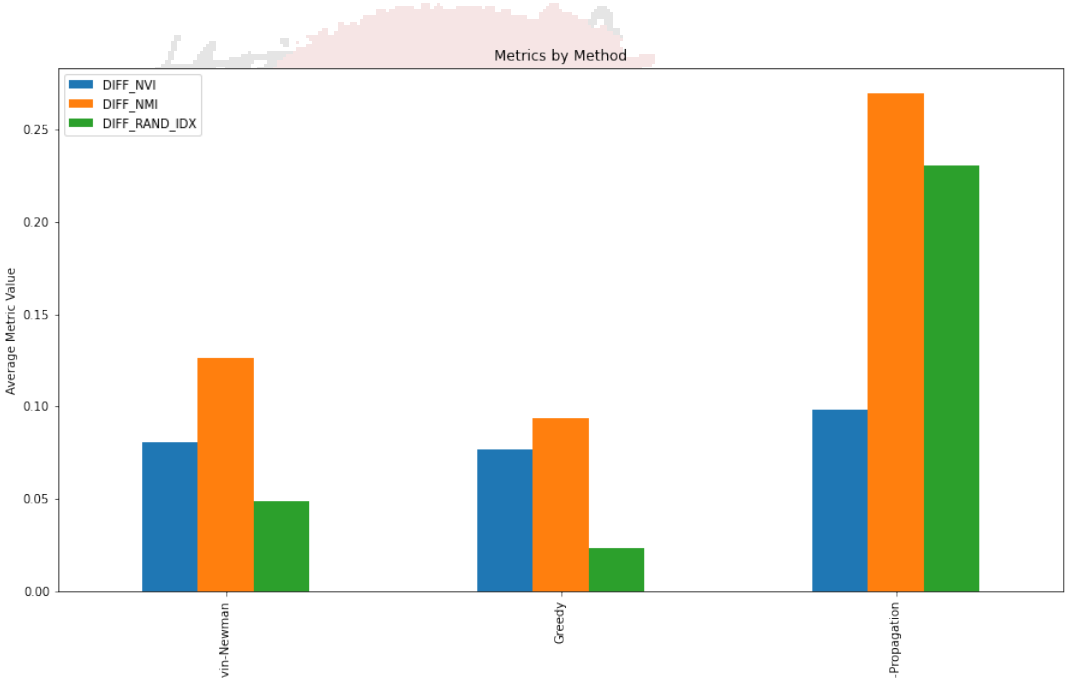

Figure 42 – Barplot metrics by method

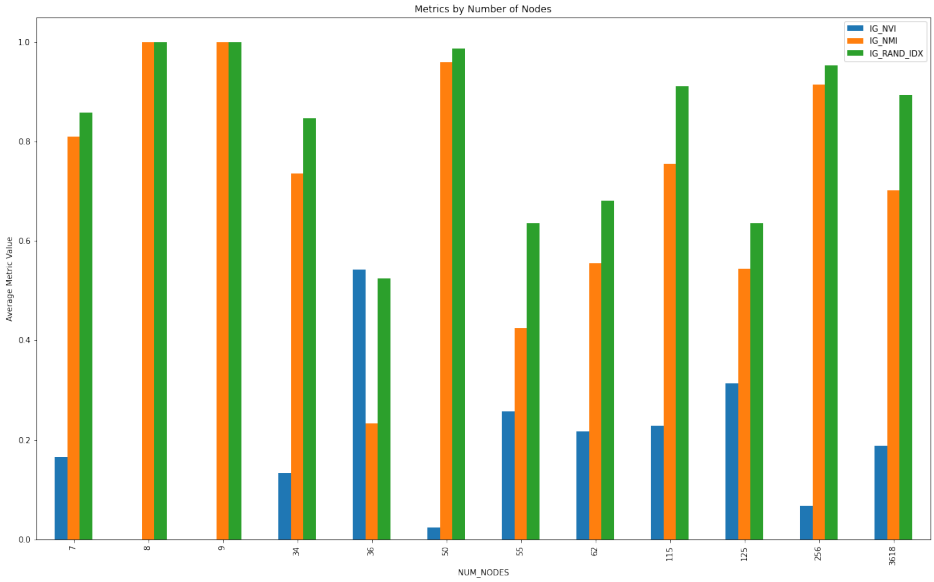

Figure 43 – Metrics by method

References

[1] Reka Albert, Albert-Laszlo Barabasi. "Statistical mechanics of complex networks".(2001).

[2] Newman, Mark. 'Networks'. *Second Editon: chapter 1 to 8.* (Newman, Mark. 2018): 13–262.

[3] Haykin, Simon. "Community structure". *Neural networks and learning machines.*(Prentice Hall/Pearson): 11–17.(2009)

[4] Documentation igraph-Python– for version 0.9.9 https://igraph.org/python/

Appendix A. Implementation details

MODEL_TYPE	name	METHOD	NUM_NODES	NX_NVI	NX_NMI	NX_RAND_IDX	IG_NVI	IG_NMI	IG_RAND_IDX	MOD
model	256_4_4_2_15_18_p	Garvin-Newman	256.0	0.14	0.85	0.93	0.12	0.87	0.94	0.76
		Greedy	256.0	0.14	0.85	0.93	0.02	0.98	0.99	0.76
		Label-Propagation	256.0	0.18	0.77	0.82	0.07	0.92	0.95	0.65
	256_4_4_4_13_18_p	Garvin-Newman	256.0	0.00	1.00	1.00	0.00	1.00	1.00	0.70
		Greedy	256.0	0.00	1.00	1.00	0.23	0.68	0.82	0.70
		Label-Propagation	256.0	0.18	0.70	0.83	0.00	1.00	1.00	0.64
	rb125	Garvin-Newman	125.0	0.22	0.77	0.90	0.18	0.83	0.92	0.62
		Greedy	125.0	0.22	0.77	0.90	0.21	0.79	0.89	0.62
		Label-Propagation	125.0	0.29	0.72	0.86	0.56	0.00	0.09	0.52
real	airports_UW	Garvin-Newman	3618.0	0.13	0.80	0.91	0.13	0.80	0.91	0.66
		Greedy	3618.0	0.18	0.74	0.88	0.19	0.74	0.91	0.66
		Label-Propagation	3618.0	0.24	0.61	0.89	0.24	0.61	0.89	0.64
	cat_cortex_sim	Garvin-Newman	55.0	0.18	0.70	0.84	0.21	0.66	0.82	0.35
		Greedy	55.0	0.18	0.70	0.84	0.33	0.00	0.26	0.35
		Label-Propagation	55.0	0.33	0.00	0.26	0.23	0.62	0.83	0.00
	dolphins	Garvin-Newman	62.0	0.44	0.01	0.47	0.19	0.57	0.71	0.50
		Greedy	62.0	0.44	0.01	0.47	0.16	0.65	0.72	0.50
		Label-Propagation	62.0	0.50	0.04	0.47	0.29	0.45	0.62	0.50
	football	Garvin-Newman	115.0	0.75	0.15	0.77	0.27	0.70	0.88	0.56
		Greedy	115.0	0.75	0.15	0.77	0.10	0.90	0.98	0.56
		Label-Propagation	115.0	0.77	0.24	0.84	0.28	0.70	0.89	0.58
	zachary_unwh	Garvin-Newman	34.0	0.35	0.29	0.62	0.15	0.69	0.84	0.38
		Greedy	34.0	0.35	0.29	0.62	0.06	0.84	0.94	0.38
		Label-Propagation	34.0	0.34	0.25	0.62	0.19	0.68	0.76	0.33
toy	20x2+5x2	Garvin-Newman	50.0	0.04	0.94	0.98	0.04	0.94	0.98	0.54
		Greedy	50.0	0.04	0.94	0.98	0.00	1.00	1.00	0.54
		Label-Propagation	50.0	0.00	1.00	1.00	0.04	0.94	0.98	0.54
	graph3+1+3	Garvin-Newman	7.0	0.17	0.81	0.86	0.17	0.81	0.86	0.37
		Greedy	7.0	0.17	0.81	0.86	0.17	0.81	0.86	0.37
		Label-Propagation	7.0	0.17	0.81	0.86	0.17	0.81	0.86	0.37
	graph4+4	Garvin-Newman	8.0	0.00	1.00	1.00	0.00	1.00	1.00	0.44
		Greedy	8.0	0.00	1.00	1.00	0.00	1.00	1.00	0.44
		Label-Propagation	8.0	0.00	1.00	1.00	0.00	1.00	1.00	0.44
	grid-p-6x6	Garvin-Newman	36.0	0.52	0.44	0.76	0.49	0.40	0.73	0.42
		Greedy	36.0	0.52	0.44	0.76	0.57	0.26	0.65	0.42
		Label-Propagation	36.0	0.47	0.37	0.67	0.44	0.00	0.20	0.35
	star	Garvin-Newman	9.0	0.00	1.00	1.00	0.00	1.00	1.00	0.00
		Greedy	9.0	0.00	1.00	1.00	0.00	1.00	1.00	0.00
		Label-Propagation	9.0	0.00	1.00	1.00	0.00	1.00	1.00	0.00

Figure 44 – Final Metrics per Model, name and Method