Activity A3 : Community Detection  $Hajar\ Lachheb$ 

#### 1 Introduction

Through this paper, I am presenting the individual work I did. It was about the concept of Community Detection. In fact, finding clusters of nodes with strong connectedness and similarity inside a network or graph is a method known as community detection. In a number of disciplines, including social network analysis, biological network analysis, and recommendation systems, community detection is essential. For instance, using community discovery in social network analysis, it is possible to locate groups of individuals who share a common affiliation, set of values, or set of interests. This data can be used to create targeted advertising or to locate influential people inside a network. Overall, this work was quite challenging as well, but it helped me learn more about the importance of community detection as well as it importance to analyse and understand complex networks.

# 2 Work Description

### 2.1 Description of the algorithms used

In this task, I was very ambitious and I thought it will be the opportunity to learn more about the different algorithms that would help me to do community detection as best as possible. Some of the algorithms are used in igraph and will be as well helping me to plot the different partitions. An addition was to use algorithms using networkx to have a comparison table just as we did use igraph algorithms. The overall objective behind what I did is just to compare the overall performance of the algorithms and get to the conclusion "What's the best algorithm?"

- Fast Greedy: The community fastgreedy algorithm bases its modularity optimization on a greedy strategy. In order to generate clusters that maximize the modularity score, nodes or groups of nodes are gradually merged. Although this method is quick, it might not always result in the modularity function's global maximum.
- Optimal Modularity: The community optimal modularity algorithm is another that has been tested. To identify the partition that has the maximum modularity score, this algorithm does a thorough search. However, this computation may take a while for big networks.
- Label Propagation: The label propagation algorithm created by Raghavan, Albert, and Kumara serves as the foundation for the communitylabelpropagation algorithm. By using a majority vote, this technique first assigns a node an initial label before propagating that label to any nearby nodes. Up until there are no more label modifications, this process is repeated. Depending on the initial labeling, the algorithm's nondeterministic output may differ.
- Leading Eigenvector: Using the eigenvectors of the modularity matrix, the community-leading eigenvector method maximizes modularity. The pairwise connections between nodes in a network are encoded in the modularity matrix, which is a matrix. This matrix's eigenvectors can be used to determine which nodes are most crucial for maximizing modularity. Although this approach might take longer than the community fastgreedy algorithm, it might yield superior outcomes.
- Louvain: A community recognition approach called the Louvain algorithm maximizes modularity, a metric for the effectiveness of dividing a network into communities. It accomplishes this by repeatedly merging communities that raise the modularity rating until no more advancements are possible. The technique is a preferred option for extensive network research because of its speed and scalability.

For the added Networkx algorithms, we can explain them as follows:

- Greedy Modularity : The community fastgreedy algorithm in igraph is comparable to the greedy-modularity communities algorithm.
- Label Propagation : The community label propagation algorithm in igraph is comparable to the label propagation communities algorithm.

- Louvain: The louvain algorithm in igraph is comparable to the louvainnx algorithm.
- Kclique: The kcliquecommunities approach was also tested, and we had to change k in order to get good comparison results. In fact, the algorithm searches for densely linked subgraphs of a network known as k-cliques, where k is a parameter that establishes the minimum size of the subgraph. All k-cliques in the network are first identified by the algorithm, which then combines them to create larger communities. The resulting communities' sizes are not specified, and they may overlap. The network's properties and the required level of granularity in the community structure must be taken into consideration while selecting the k value.

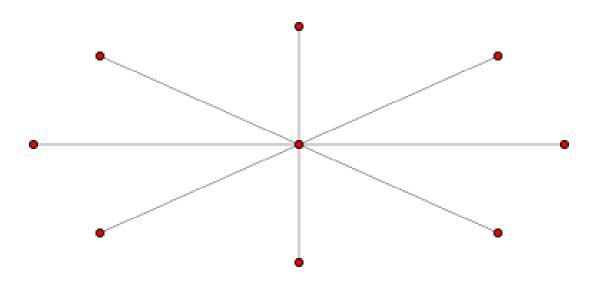
#### 2.2 Representation of the plots with color-coded communities

By giving each community a distinct hue, color-coded communities can help in the visualization of the outcomes of community detection algorithms. This makes it simpler to recognize the many communities and comprehend how they interact inside a network. The purpose of color-coded communities is to give us an intuitive visual representation of the clustering results. It might be challenging to distinguish between different cultures and comprehend their makeup without the use of color-coding. Finally, color-coded communities can also be used to compare and assess the efficacy of various community detection techniques. It is feasible to compare the color-coded communities and see which algorithms are generating results that are similar or dissimilar, as well as any patterns or anomalies in the data.

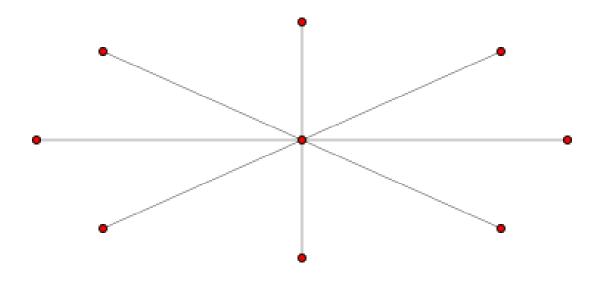
The representation in this part will be as follows: We represent each network separately and for each network, we plot the three different community detection representations following the respective algorithm.

Plots of the network: Star

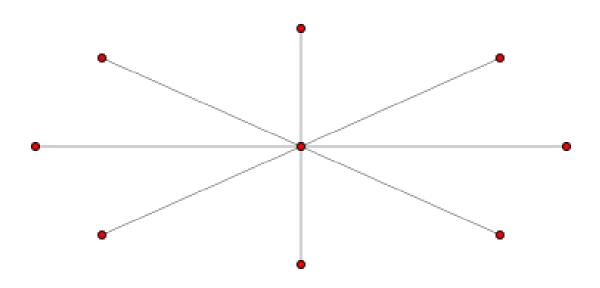
Star network plot using Label Propagation Algorithm



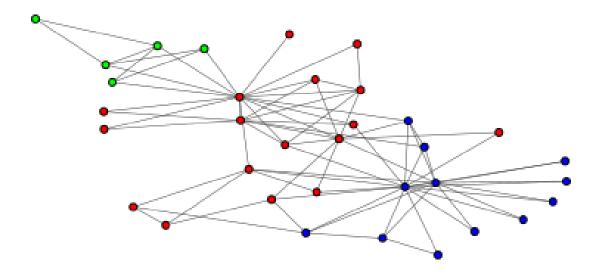
Star network plot using Louvain Algorithm



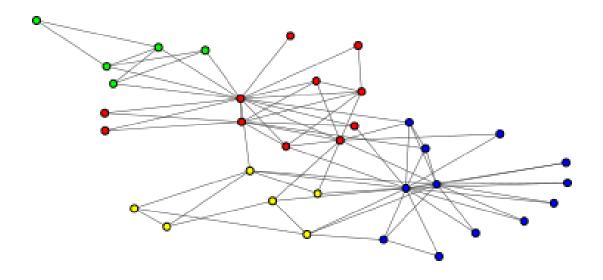
Star network plot using Fast Greedy Algorithm



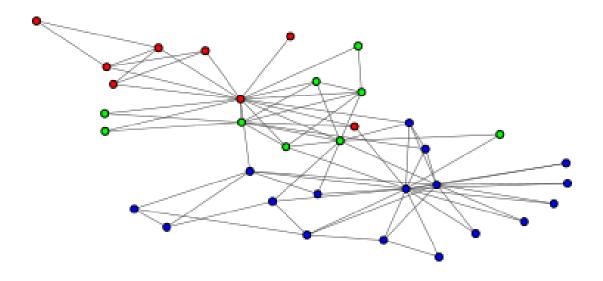
Plots of the network: Zacharyunwh Zacharyunwh network plot using Label Propagation Algorithm



Zacharyunwh network plot using Louvain Algorithm

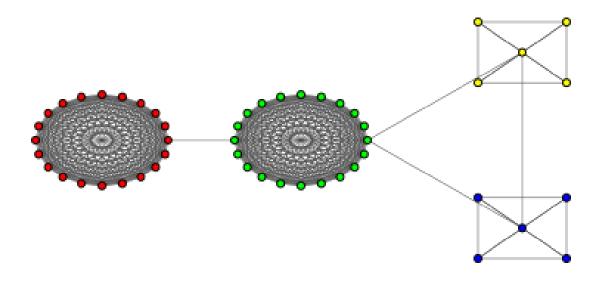


Zacharyunwh network plot using Fast Greedy Algorithm

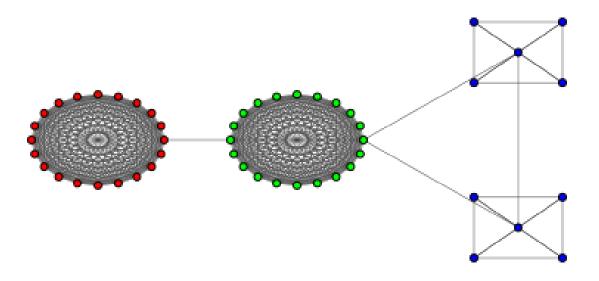


Plots of the network: 20\*2+5\*2

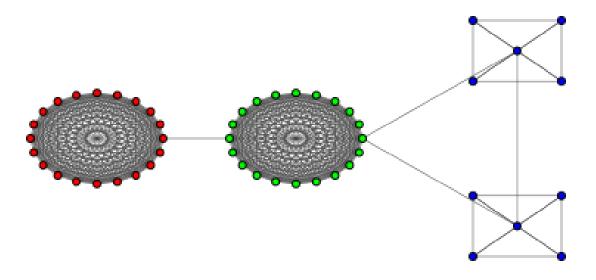
 $20^*2 + 5^*2$ network plot using Label Propagation Algorithm



 $20^*2+5^*2$  network plot using Louvain Algorithm

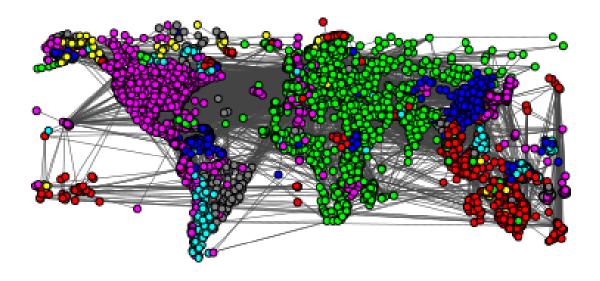


 $20^*2 + 5^*2$ network plot using Fast Greedy Algorithm

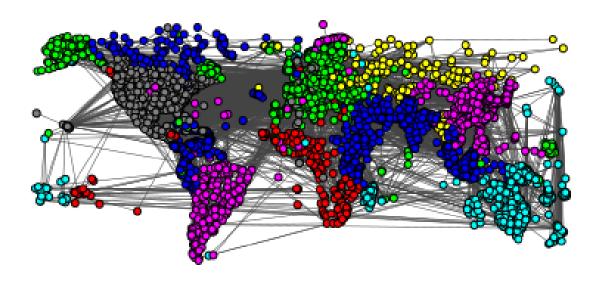


Plots of the network: Airports UW  $\,$ 

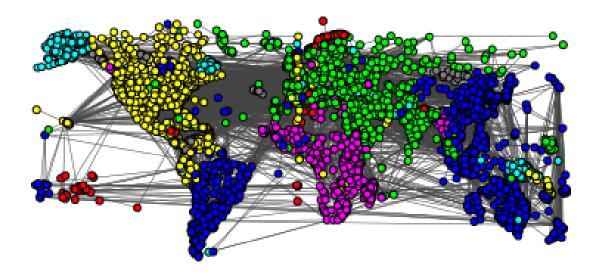
AirportsUW network plot using Label Propagation Algorithm

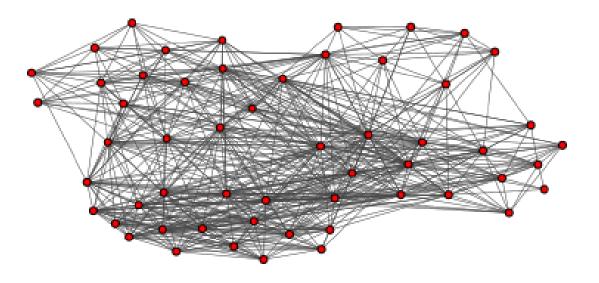


AirportsUW network plot using Louvain Algorithm

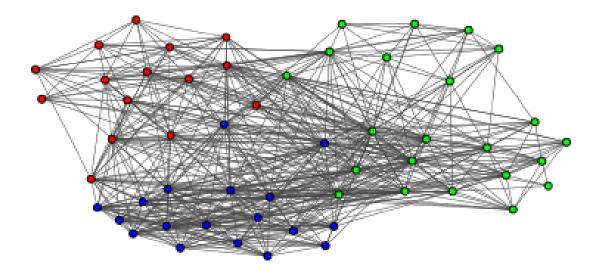


AirportsUW network plot using Fast Greedy Algorithm

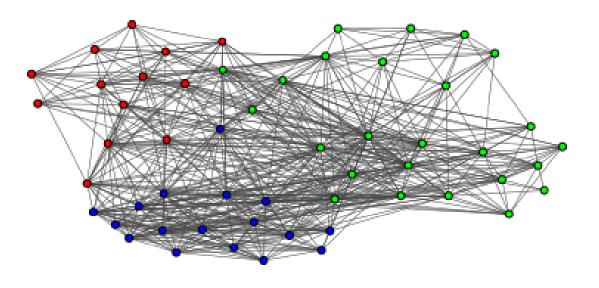




CatCortexSim network plot using Louvain Algorithm

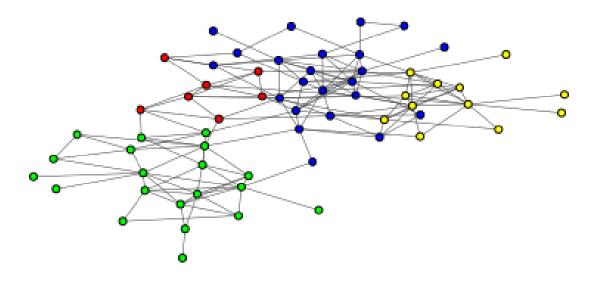


CatCortexSim network plot using Fast Greedy Algorithm

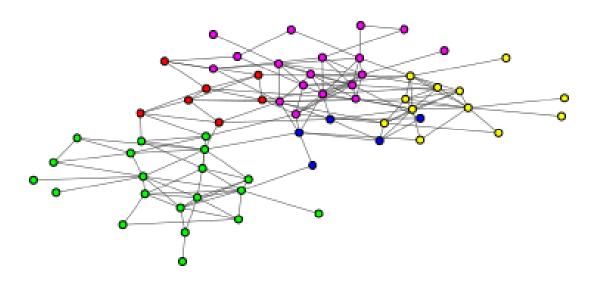


Plots of the network: Dolphins

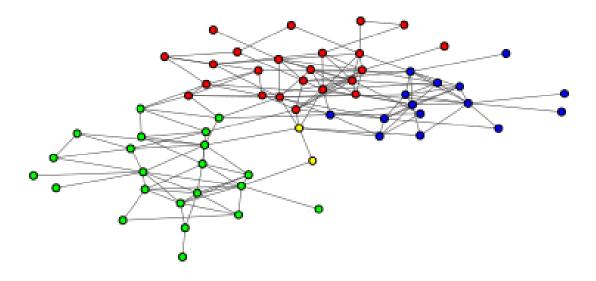
Dolphins network plot using Label Propagation Algorithm



Dolphins network plot using Louvain Algorithm

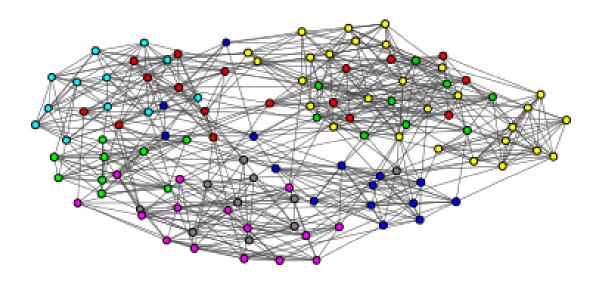


Dolphins network plot using Fast Greedy Algorithm

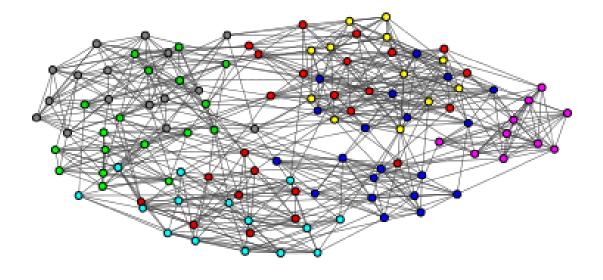


Plots of the network: Football

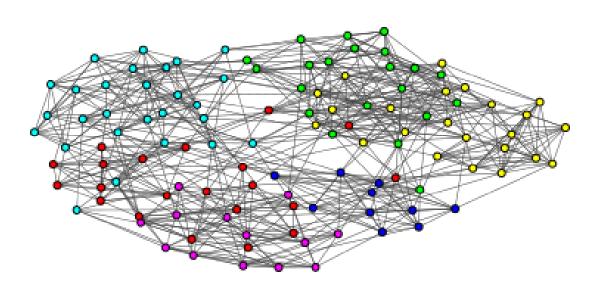
Football network plot using Label Propagation Algorithm



Football network plot using Louvain Algorithm

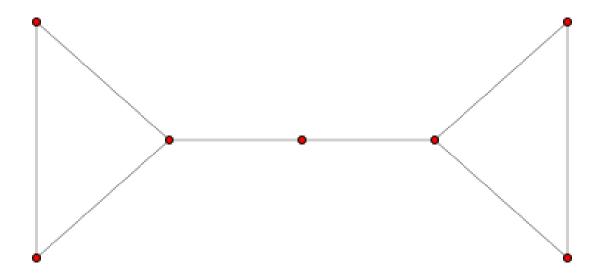


Football network plot using Fast Greedy Algorithm

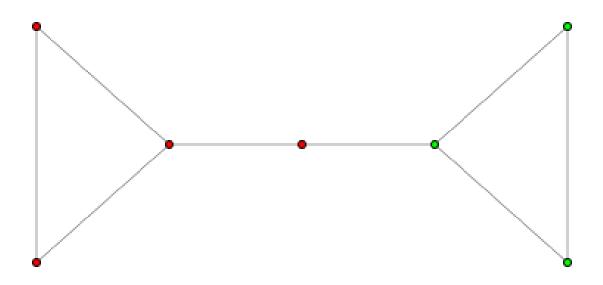


Plots of the network: Graph3+1+3

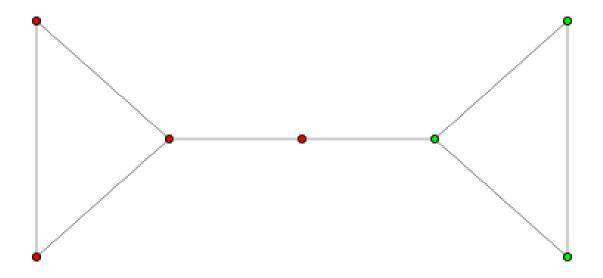
 $\operatorname{Graph} 3+1+3$ network plot using Label Propagation Algorithm



Graph3+1+3 network plot using Louvain Algorithm

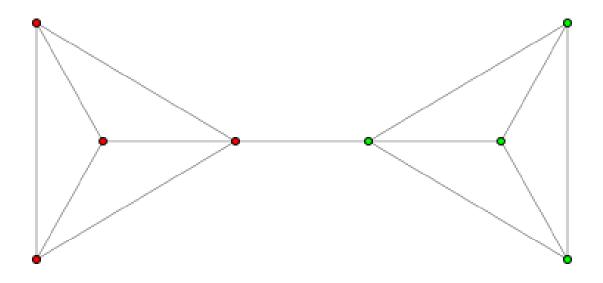


 $\operatorname{Graph} 3+1+3$  network plot using Fast Greedy Algorithm

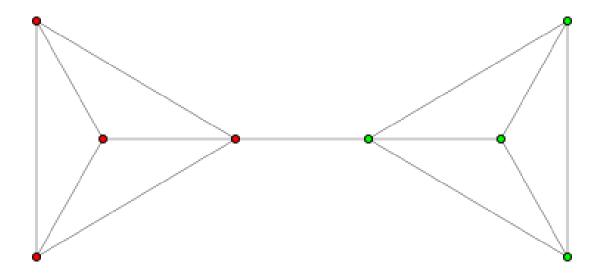


Plots of the network: Graph4+4

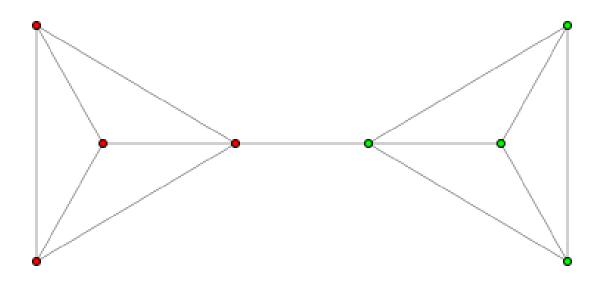
 $\operatorname{Graph4} + 4$ network plot using Label Propagation Algorithm



 $\operatorname{Graph}4+4$ network plot using Louvain Algorithm

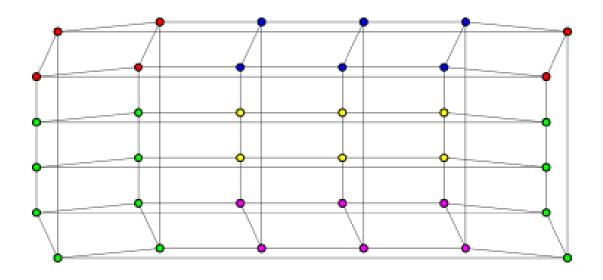


Graph4+4 network plot using Fast Greedy Algorithm

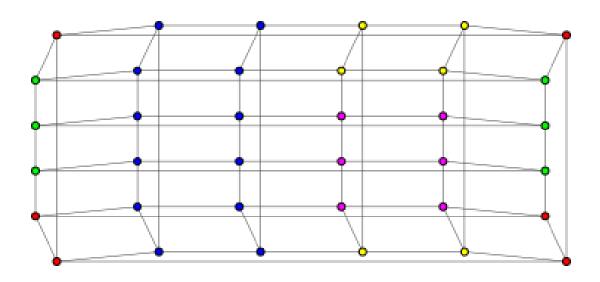


Plots of the network: Gridp 6\*6

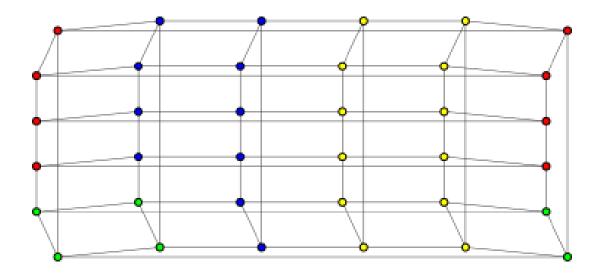
Grid<br/>p $6\!\!\!\!\!\!\!\!\!\!\!\!\!^*\!6$ network plot using Label Propagation Algorithm



Gridp 6\*6 network plot using Louvain Algorithm

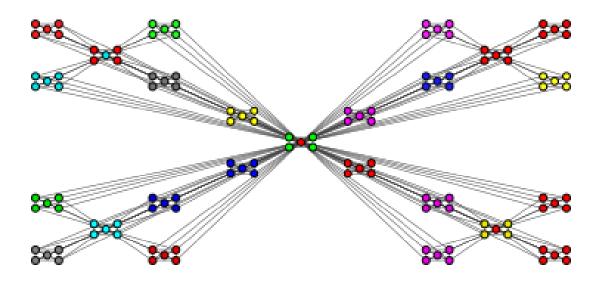


Gridp 6\*6 network plot using Fast Greedy Algorithm

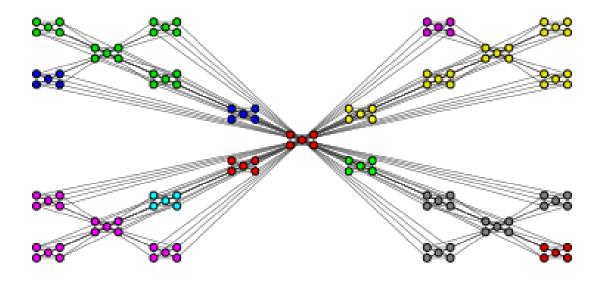


Plots of the network: Rb125

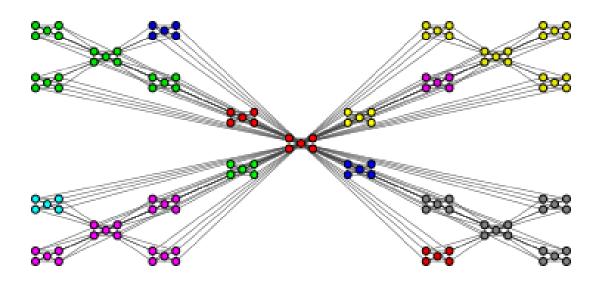
 ${\rm Rb}125$ network plot using Label Propagation Algorithm



 ${\rm Rb}125$ network plot using Louvain Algorithm

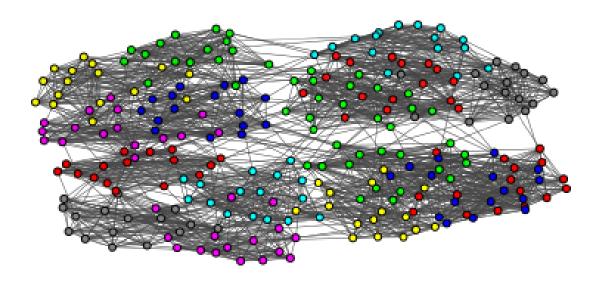


Rb125 network plot using Fast Greedy Algorithm

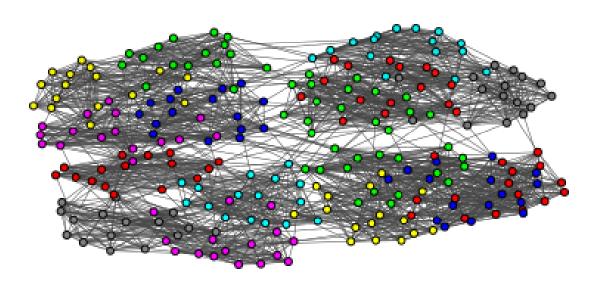


Plots of the network: 256 4421518 p

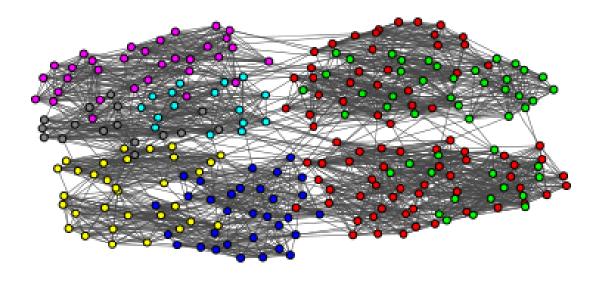
 $256\ 4421518$ p network plot using Label Propagation Algorithm



 $256\ 4421518$ p network plot using Louvain Algorithm

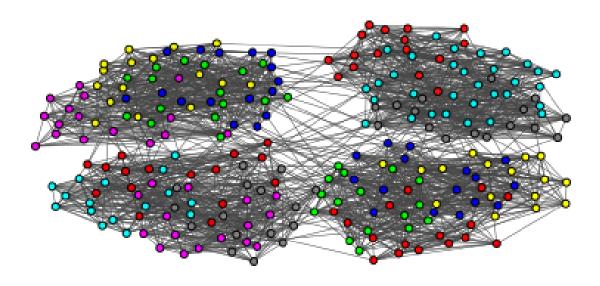


 $256\ 4421518$ p network plot using Fast Greedy Algorithm

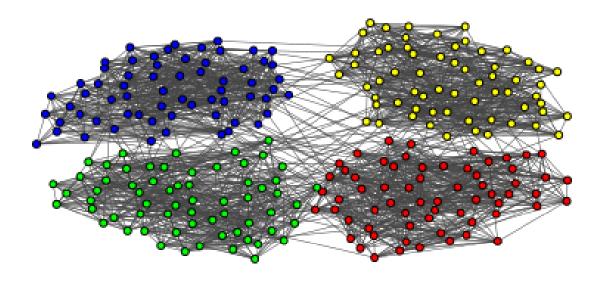


Plots of the network: 256 4441318 p

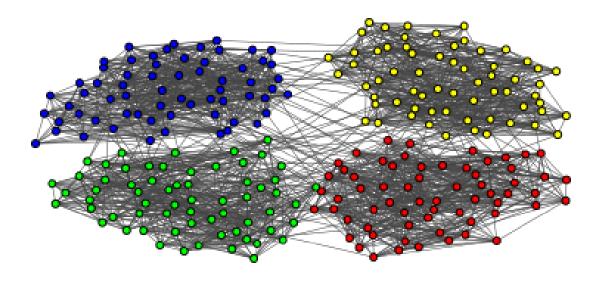
 $256\ 4441318$ p network plot using Label Propagation Algorithm



 $256\ 4441318$ p network plot using Louvain Algorithm



256 4441318 p network plot using Fast Greedy Algorithm



## 2.3 Comparison measures

This section's goal is to compare the partitions produced by various community detection techniques to reference partitions, which are thought to represent the network's nodes' actual clustering in reality. In the field of community detection, comparison measures like the Jaccard index, NMI, and NVI are frequently employed to assess the quality of a partition.

We can see how each algorithm performs on each network by grouping the comparison measurements by network. This enables us to examine how well various algorithms perform across various networks and determine which algorithms function well uniformly across all networks and which ones could be better suited for particular types of networks.

Following we will share the tables representing the comparison using Jaccard, NVI and NMI. We will also share the networkx results as well.

Table representing the comparison using igraph

gra	aph3+1+3	3	IMM	NVI
a	algorithm fastgreedy	Jaccard 0.666667		0.238237
1	leading_eigenvector	0.666667	0.80954	0.238237
2	label_propagation	0.666667		0.238237
3	louvain	0.666667	0.80954	0.238237
20:	x2+5×2			
0	algorithm	Jaccard 0.941176	NMI 0.938345	NVI 0.051124
1	fastgreedy leading_eigenvector	0.941176	0.938345	0.051124
2	label_propagation	1.000000	1.000000	0.000000
3	louvain	0.941176	0.938345	0.051124
gra	aph4+4			
	algorithm		NMI NVI	
0	fastgreedy	1.0	1.0 0.0	
2	leading_eigenvector label_propagation	1.0	1.0 0.0	
3	louvain	1.0	1.0 0.0	
sta	ar algorithm	Jaccard	NMI NVI	
0	fastgreedy		1.0 0.0	
1	leading_eigenvector	1.0	1.0 0.0	
2	label_propagation	1.0	1.0 0.0	
3	louvain	1.0	1.0 0.0	
cat	t_cortex_sim			
	algorithm	Jaccard	NMI	NVI
0	fastgreedy	0.542169	0.656873	0.296602
1	leading_eigenvector	0.547872	0.618651	0.332598 0.481994
3	label_propagation louvain	0.257239 0.589416	0.672641	0.285823
zac	chary_unwh	3	NMT	NVT
0	algorithm fastgreedy	Jaccard 0.683274	0.692467	0.217697
1	leading_eigenvector	0.505495	0.677092	0.269804
2	label_propagation	0.758741	0.723881	0.182728
3	louvain	0.450355	0.581858	0.350338
do	olphins algorithm	Jaccard	i NMI	NVI
do Ø	algorithm			
		0.504125	0.572700	0.271613
0	algorithm fastgreedy	0.504125 0.329314	0.572700 0.448914	0.271613 0.423804
0	algorithm fastgreedy leading_eigenvector label_propagation	0.504125 0.329314 1.000000	0.572700 0.448914	0.271613 0.423804 0.000000
0 1 2 3	algorithm fastgreedy leading_eigenvector label_propagation louvain	0.504125 0.329314 1.000000	0.572700 0.448914 0.000000	0.271613 0.423804 0.000000
0 1 2 3	algorithm fastgreedy leading_eigenvector label_propagation louvain	0.504125 0.329314 1.000000 0.304798	0.572700 0.448914 0.1000000 0.474253	0.271613 0.423804 0.000000 0.431107
0 1 2 3	algorithm fastgreedy leading_eigenvector label_propagation louvain ootball algorithm	0.504125 0.329314 1.000000 0.304798	0.572700 0.448914 0.1.000000 0.474253	0.271613 0.423804 0.000000 0.431107
0 1 2 3 fo	algorithm fastgreedy leading_eigenvector label_propagation louvain ootball algorithm fastgreedy	0.504125 0.329314 1.000000 0.304798 Jaccard	0.572700 0.448914 0.1.000000 0.474253	0.271613 0.423804 0.000000 0.431107 NVI 0.385871
0 1 2 3 fo	algorithm fastgreedy leading_eigenvector label_propagation louvain obtball algorithm fastgreedy leading_eigenvector	0.504125 0.329314 1.000000 0.304798 Jaccard 0.362153 0.350324	0.572700 0.448914 1.000000 0.474253 1 NMI 3 0.697732 0.698670	0.271613 0.423804 0.000000 0.431107 NVI 0.385871 0.407185
0 1 2 3 fo 0 1 2	algorithm fastgreedy leading_eigenvector label_propagation louvain  ootball algorithm fastgreedy leading_eigenvector label_propagation	0.504125 0.329314 1.000000 0.304798 Jaccard 0.362153 0.350324 0.763200	0.572700 0.448914 1.000000 0.474253 1 NMI 3 0.697732 4 0.698670 0 0.910680	0.271613 0.423804 0.000000 0.431107 NVI 0.385871 0.407185 0.131033
0 1 2 3 fo	algorithm fastgreedy leading_eigenvector label_propagation louvain obtball algorithm fastgreedy leading_eigenvector	0.504125 0.329314 1.000000 0.304798 Jaccard 0.362153 0.350324	0.572700 0.448914 1.000000 0.474253 1 NMI 3 0.697732 4 0.698670 0 0.910680	0.271613 0.423804 0.000000 0.431107 NVI 0.385871 0.407185 0.131033
0 1 2 3 fo 0 1 2 3	algorithm fastgreedy leading_eigenvector label_propagation louvain  ootball algorithm fastgreedy leading_eigenvector label_propagation louvain	0.504125 0.329314 1.000000 0.304798 Jaccard 0.362153 0.350324 0.763200	0.572700 0.448914 1.000000 0.474253 1 NMI 3 0.697732 4 0.698670 0 0.910680	0.271613 0.423804 0.000000 0.431107 NVI 0.385871 0.407185 0.131033
0 1 2 3 fo 0 1 2 3	algorithm fastgreedy leading_eigenvector label_propagation louvain  botball algorithm fastgreedy leading_eigenvector label_propagation louvain  66_4_4_13_18_p	0.504125 0.329314 1.000000 0.304798 Jaccarc 0.362153 0.350324 0.763200 0.695906	0.572700 4 0.448914 1.000000 0.474253 MMI 3 0.697732 0.698670 0.910680 0.884962	0.271613 0.423804 0.00000 0.431107 NVI 0.385871 0.407185 0.131033 0.165626
0 1 2 3 fo 0 1 2 3	algorithm fastgreedy leading_eigenvector label_propagation louvain  ootball algorithm fastgreedy leading_eigenvector label_propagation louvain  66_4_4_4_13_18_p algorithm	0.504125 0.329314 1.000000 0.304798 Jaccarc 0.362155 0.350324 0.763200 0.695906	0.572700 0.448914 1.000000 0.474253 1 NMI 3 0.697732 4 0.69873 0.910680 0.884962	0.271613 0.423804 0.000000 0.431107 NVI 0.385871 0.407185 0.131033 0.165626
0 1 2 3 fo 1 2 3	algorithm fastgreedy leading_eigenvector label_propagation louvain  botball algorithm fastgreedy leading_eigenvector label_propagation louvain  66_4_4_13_18_p	0.504125 0.329314 1.000000 0.304798 Jaccarc 0.362153 0.350324 0.763200 0.695906 Jaccarc	0.572700 4 0.448914 1.000000 3 0.474253 MMI 8 0.697732 4 0.698670 0 9.910680 0 .884962 MMI 1.000000	0.271613 0.423804 0.000000 0.431107 NVI 0.385871 0.407185 0.131033 0.165626
0 1 2 3 fo 0 1 2 3	algorithm fastgreedy leading_eigenvector label_propagation louvain  sotball algorithm fastgreedy leading_eigenvector label_propagation louvain  66_4_4_4_13_18_p algorithm fastgreedy leading_eigenvector	0.504125 0.329314 1.000000 0.304798 Jaccarc 0.362153 0.350324 0.695906 Jaccarc 1.000000	0.572700 4 0.448914 1.000000 0.474253 MMI 3 0.697732 0.998670 0.910680 0.884962 MMI 1.000000 1.000000	0.271613 0.423804 0.000000 0.431107 NVI 0.385871 0.407185 0.131033 0.165626 NVI 0.000000 0.000000
0 1 2 3 fc 0 1 2 3 2 3	algorithm fastgreedy leading_eigenvector label_propagation louvain  ootball algorithm fastgreedy leading_eigenvector label_propagation louvain  66_4_4_4_13_18_p algorithm fastgreedy leading_eigenvector label_propagation	0.504125 0.329314 1.000000 0.304798 Jaccarc 0.362153 0.350324 0.695906 Jaccarc 1.000000	0.572700 0.448914 1.000000 0.474253 0.698670 0.910680 0.884962 0.884962 0.1000000 1.0000000 0.688851	0.271613 0.423804 0.000000 0.431107 NVI 0.385871 0.407185 0.131033 0.165626 NVI 0.000000 0.338132
0 1 2 3 fc 0 1 2 3 2 5 0 1 2 3	algorithm fastgreedy leading_eigenvector label_propagation louvain  sotball algorithm fastgreedy leading_eigenvector label_propagation louvain  66_4_4_4_13_18_p algorithm fastgreedy leading_eigenvector label_propagation louvain	0.504125 0.329314 1.000006 0.304798 Jaccarc 0.36215 0.350324 0.763200 0.695906 Jaccarc 1.000000 0.269841	0.572700 0.448914 1.000000 0.474253 0.698670 0.910680 0.884962 0.884962 0.1000000 1.0000000 0.688851	0.271613 0.423804 0.000000 0.431107 NVI 0.385871 0.407185 0.131033 0.165626 NVI 0.000000 0.338132
0 1 2 3 fc 0 1 2 3 2 5 0 1 2 3	algorithm fastgreedy leading_eigenvector label_propagation louvain  ootball algorithm fastgreedy leading_eigenvector label_propagation louvain  66_4_4_4_13_18_p algorithm fastgreedy leading_eigenvector label_propagation louvain fastgreedy leading_eigenvector label_propagation louvain	0.504125 0.329314 1.000006 0.304798 Jaccarc 0.36215 0.350324 0.763200 0.695906 Jaccarc 1.000000 0.269841 0.904762	0.572700 0.448914 1.000000 0.474253 0.474253 0.698670 0.910680 0.884962 1.000000 1.000000 0.68851 0.951742	0.271613 0.423804 0.000000 0.431107 NVI 0.385871 0.407185 0.131033 0.165626 NVI 0.000000 0.338132 0.036576
0 1 2 3 6 0 1 2 3 2 9 0 1 2 3 0 1 2 3 7	algorithm fastgreedy leading_eigenvector label_propagation louvain  botball algorithm fastgreedy leading_eigenvector label_propagation louvain  66_4_4_13_18_p algorithm fastgreedy leading_eigenvector label_propagation louvain  61_10_10_10_10_10_10_10_10_10_10_10_10_10	0.504125 0.329314 1.000000 0.304798 Jaccarc 0.36215 0.36215 0.36225 0.763200 0.695906 Jaccarc 1.000000 0.269841 0.904762	0.572700 4 0.448914 1.000000 3 0.474253 MMI 8 0.697732 4 0.698670 0.910680 0.884962 MMI 1.000000 1.000000 1.000000 0.680851 0.951742	0.271613 0.423804 0.000000 0.431107 NVI 0.385871 0.407185 0.131033 0.165626 NVI 0.000000 0.000000 0.338132 0.036576
0 1 2 3 6 0 1 2 3 2 5 0 1 2 3 7 0 1 2 3 7 0 1 1 2 1 2 1 2 1 2 3 7 1 1 2 1 2 1 1 2 1 2 1 1 2 1 2 1 1 2 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 1 1 1 2 1	algorithm fastgreedy leading_eigenvector label_propagation louvain  sotball algorithm fastgreedy leading_eigenvector label_propagation louvain  66_4_4_4_13_18_p algorithm fastgreedy leading_eigenvector label_propagation louvain  1025 algorithm fastgreedy	0.504125 0.329314 1.000006 0.304798 Jaccarc 0.36215 0.350324 0.695906 Jaccarc 1.000000 0.269844 0.904762 Jaccarc	0.572700 0.448914 0.448914 1.000000 0.474253  MMI 3.0.697732 0.998670 0.910680 0.884962  MMI 1.000000 1.000000 1.000000 0.680851 0.951742	0.271613 0.423804 0.00000 0.431107 NVI 0.385871 0.407185 0.131033 0.165626 NVI 0.000000 0.338132 0.036576
0 1 2 3 fc 0 1 2 3 7 tc 0 1 2 3 7 tc 0 1	algorithm fastgreedy leading_eigenvector label_propagation louvain  algorithm fastgreedy leading_eigenvector label_propagation louvain  66_4_4_4_13_18_p algorithm fastgreedy leading_eigenvector label_propagation louvain  2125 algorithm fastgreedy leading_eigenvector	0.504125 0.329314 1.000000 0.304798 Jaccarc 0.36215 0.350324 0.763200 0.695906 Jaccarc 1.000000 0.269841 0.904762	0.572700 0.448514 1.000000 0.474253 0.474253 0.697732 0.698670 0.910680 0.884962 0.1000000 0.688551 0.951742 0.951742	0.271613 0.423804 0.000000 0.431107 0.385871 0.407185 0.131033 0.165626 NVI 0.000000 0.338132 0.036576
0 1 2 3 fc 0 1 2 3 rt 0 1 2	algorithm fastgreedy leading_eigenvector label_propagation louvain  sotball algorithm fastgreedy leading_eigenvector label_propagation louvain  66_4_4_13_18_p algorithm fastgreedy leading_eigenvector label_propagation louvain  125 algorithm fastgreedy leading_eigenvector label_propagation  125 algorithm fastgreedy leading_eigenvector label_propagation	0.504125 0.329314 1.000000 0.304798 0.36215: 0.350322 0.763200 0.695906 1.000000 1.000000 0.269841 0.904762	0.572700 0.448914 0.448914 1.000000 0.474253  MMI 0.697732 0.698670 0.910680 0.884962  MMI 0.000000 0.680851 0.951742  MMI 0.680851 0.951742	0.271613 0.423804 0.000000 0.431107 NVI 0.385871 0.407185 0.131033 0.165626 NVI 0.000000 0.000000 0.338132 0.036576 NVI 0.287653 0.347461 0.157606
0 1 2 3 fc 0 1 2 3 7 tc 0 1 2 3 7 tc 0 1	algorithm fastgreedy leading_eigenvector label_propagation louvain  sotball algorithm fastgreedy leading_eigenvector label_propagation louvain  66_4_4_13_18_p algorithm fastgreedy leading_eigenvector label_propagation louvain  125 algorithm fastgreedy leading_eigenvector label_propagation  125 algorithm fastgreedy leading_eigenvector label_propagation	0.504125 0.329314 1.000000 0.304798 Jaccarc 0.36215 0.350324 0.763200 0.695906 Jaccarc 1.000000 0.269841 0.904762	0.572700 0.448914 0.448914 1.000000 0.474253  MMI 0.697732 0.698670 0.910680 0.884962  MMI 0.000000 0.680851 0.951742  MMI 0.680851 0.951742	0.271613 0.423804 0.000000 0.431107 NVI 0.385871 0.407185 0.131033 0.165626 NVI 0.000000 0.000000 0.338132 0.036576 NVI 0.287653 0.347461 0.157606
0 1 2 3 fc 0 1 2 3 rt 0 1 2 3	algorithm fastgreedy leading_eigenvector label_propagation louvain  algorithm fastgreedy leading_eigenvector label_propagation louvain  66_4_4_4_13_18_p algorithm fastgreedy leading_eigenvector label_propagation louvain  2125 algorithm fastgreedy leading_eigenvector label_propagation louvain  2125 algorithm fastgreedy leading_eigenvector label_propagation louvain	0.504125 0.329314 1.000000 0.304798 0.36215: 0.350322 0.763200 0.695906 1.000000 1.000000 0.269841 0.904762	0.572700 0.448914 0.448914 1.000000 0.474253  MMI 0.697732 0.698670 0.910680 0.884962  MMI 0.000000 0.680851 0.951742  MMI 0.680851 0.951742	0.271613 0.423804 0.000000 0.431107 NVI 0.385871 0.407185 0.131033 0.165626 NVI 0.000000 0.000000 0.338132 0.036576 NVI 0.287653 0.347461 0.157606
0 1 2 3 fc 0 1 2 3 rt 0 1 2 3	algorithm fastgreedy leading_eigenvector label_propagation louvain  sotball algorithm fastgreedy leading_eigenvector label_propagation louvain  66_4_4_13_18_p algorithm fastgreedy leading_eigenvector label_propagation louvain  5125 algorithm fastgreedy leading_eigenvector label_propagation louvain  5125 algorithm fastgreedy leading_eigenvector label_propagation louvain  56_4_4_2_15_18_p	0.504125 0.329314 1.000000 0.304798 0.36215: 0.350322 0.763200 0.695906 1.000000 1.000000 0.269841 0.904762 Jaccarc 0.28114: 0.22883; 0.429358	0.572700 0.572700 0.448914 0.00000 0.474253 0.474253 0.697732 0.698670 0.910680 0.884962 0.910680 0.1.000000 0.680851 0.951742 0.825443 0.825443	0.271613 0.423804 0.000000 0.431107 NVI 0.385871 0.407185 0.131033 0.165626 NVI 0.000000 0.000000 0.338132 0.036576 NVI 0.287653 0.347461 0.157606 0.287653
0 1 2 3 fc 0 1 2 3 rt 0 1 2 3	algorithm fastgreedy leading_eigenvector label_propagation louvain  ootball algorithm fastgreedy leading_eigenvector label_propagation louvain  66_4_4_4_13_18_p algorithm fastgreedy leading_eigenvector label_propagation louvain  0125 algorithm fastgreedy leading_eigenvector label_propagation louvain  0125 algorithm fastgreedy leading_eigenvector label_propagation louvain  0126 algorithm fastgreedy leading_eigenvector label_propagation louvain	0.504125 0.329314 1.000006 0.304798 Jaccarc 0.36215 0.350324 0.695906 Jaccarc 1.000000 0.269841 0.904762 Jaccarc 0.281144 0.22883 0.429358 0.281145	0.572700 0.448914 1.000000 0.474253 0.474253 0.698670 0.910680 0.910680 0.884962 0.884962 0.910680 0.910680 0.884962 0.884962 0.884962	0.271613 0.423804 0.000000 0.431107 NVI 0.385871 0.407185 0.131033 0.165626 NVI 0.000000 0.000000 0.338132 0.036576 NVI 0.287653 0.347461 0.157606 0.287653
0 1 2 3 fc 0 1 2 3 rt 0 1 2 3 3 25 3	algorithm fastgreedy leading_eigenvector label_propagation louvain  algorithm fastgreedy leading_eigenvector label_propagation louvain  66_4_4_4_13_18_p algorithm fastgreedy leading_eigenvector label_propagation louvain  2125 algorithm fastgreedy leading_eigenvector label_propagation louvain  2125 algorithm fastgreedy leading_eigenvector label_propagation louvain  2126 algorithm fastgreedy leading_eigenvector label_propagation louvain  2127 algorithm fastgreedy	0.504125 0.329314 1.000006 0.304798 Jaccarc 0.36215 0.350324 0.695906 Jaccarc 1.000000 0.269841 0.904762 Jaccarc 0.281144 0.22883 0.429358 0.281145	0.572700 0.448514 1.000000 0.474253 0.474253 0.474253 0.697732 0.697832 0.991680 0.884962 0.1000000 0.0680851 0.951742 0.951742 0.951742 0.951742 0.88543	0.271613 0.423804 0.000000 0.431107 0.385871 0.407185 0.131033 0.165626 NVI 0.000000 0.338132 0.036576 NVI 0.287653 0.347461 0.157606 0.287653
0 1 2 3 fc 0 1 2 3 rt 0 1 2 3 2 5 0	algorithm fastgreedy leading_eigenvector label_propagation louvain  ootball algorithm fastgreedy leading_eigenvector label_propagation louvain  66_4_4_4_13_18_p algorithm fastgreedy leading_eigenvector label_propagation louvain  0125 algorithm fastgreedy leading_eigenvector label_propagation louvain  0125 algorithm fastgreedy leading_eigenvector label_propagation louvain  0126 algorithm fastgreedy leading_eigenvector label_propagation louvain	0.504125 0.329314 1.000000 0.304798 0.36215 0.350324 0.763200 0.695906 1.000000 0.269841 0.22813 0.281143 0.22883 0.281143	0.572700 0.448914 0.448914 1.000000 0.474253  MMI 0.697732 0.998670 0.910680 0.1000000 0.680851 0.951742  MMI 0.825443 0.951742	0.271613 0.423804 0.000000 0.431107 NVI 0.385871 0.407185 0.131033 0.165626 NVI 0.000000 0.338132 0.036576 NVI 0.287653 0.347461 0.157606 0.287653 0.347461 0.157606 0.287653

Table representing the comparison using networkx

gra	aph3+1+3 algorithm	Jaccard	NMI	NVI
0	greedy_modularity	0.666667	0.80954	0.238237
2	louvain	0.666667	0.80954	0.238237 0.744544
3	k-clique label_propagation	0.285714 0.666667	0.80954	0.238237
20×	(2+5×2 algorithm	Jaccard	NMI	NVI
0	greedy_modularity	0.941176	0.938345	0.051124
1	louvain	0.941176	0.938345	0.051124
2	k-clique label_propagation	0.326531	0.000000	0.440163
_	Tabel_pi opagacion	1.000000	1.000000	0.000000
gra	aph4+4			
а	algorithm greedy_modularity	Jaccard 1.000000	NMI 1.000000	NVI 0.000000
1	louvain	0.461538	0.585645	0.510367
2	k-clique	0.428571	0.000000	0.480898
3	label_propagation	1.000000	1.000000	0.000000
sta				
_	algorithm	Jaccard	NMI NVI	
0	greedy_modularity louvain	1.0	1.0 0.0	
2	k-clique	1.0	1.0 0.0	
3	label_propagation	1.0	1.0 0.0	
cat	_cortex_sim			
	algorithm	Jaccard	NMI	NVI
0	greedy_modularity	0.571930	0.702341	0.257299
2	louvain k-clique	0.637218	0.729666	0.236034 0.481994
3	label_propagation	0.257239	0.000000	0.481994
zac	hary_unwh: algorithm	Jaccard	NMI	NVI
0	greedy_modularity	0.379009	0.289122	0.503219
1	louvain	0.465950	0.592773	0.341193
2	k-clique label_propagation	0.486631 0.403361	0.000000 0.254134	0.282870 0.494754
_	Tabel_propagation	0.403301	0.234134	0.454/34
do	lphins			
	algorithm			MI NVI
0	algorithm greedy_modularity	0.24506	8 0.0109	11 0.628715
0	algorithm greedy_modularity louvain	0.24506 0.37911	8 0.0109 6 0.5162	11 0.628715 34 0.364424
0 1 2	algorithm greedy_modularity louvain k-clique	0.24506 0.37911 0.55579	0.0109 6 0.5162 1 0.0000	11 0.628715 34 0.364424 00 0.219805
0	algorithm greedy_modularity louvain	0.24506 0.37911 0.55579	0.0109 6 0.5162 1 0.0000	11 0.628715 34 0.364424 00 0.219805
0 1 2 3	algorithm greedy_modularity louvain k-clique label_propagation	0.24506 0.37911 0.55579	0.0109 6 0.5162 1 0.0000	11 0.628715 34 0.364424 00 0.219805
0 1 2 3	algorithm greedy_modularity louvain k-clique	0.24506 0.37911 0.55579 0.20336	68 0.0109 16 0.5162 01 0.0000 63 0.0411	11 0.628715 34 0.364424 00 0.219805
0 1 2 3	algorithm greedy_modularity louvain k-clique label_propagation otball	0.24506 0.37911 0.55579 0.20336	58 0.0109 16 0.5162 01 0.0000 53 0.0411	11 0.628715 34 0.364424 00 0.219805 41 0.717362 MI NVI
0 1 2 3	algorithm greedy_modularity louvain k-clique label_propagation otball algorithm	0.24506 0.37911 0.55579 0.20336 Jaccar 0.05867	0.0109 0.5162 0.0000 0.00411 0.0000 0.0411	11 0.628715 34 0.364424 00 0.219805 41 0.717362 MI NVI 75 1.086858
0 1 2 3 fo 0 1 2	algorithm greedy_modularity louvain k-clique label_propagation otball algorithm greedy_modularity louvain k-clique	0.24506 0.37911 0.55579 0.20336 Jaccar 0.05867 0.71300 0.07978	68 0.0109 60 0.5162 91 0.0000 63 0.0411 7d N 75 0.1479 94 0.8922 96 0.0000	11 0.628715 34 0.364424 00 0.219805 41 0.717362 MI NVI 75 1.086858 96 0.155381 00 0.746947
0 1 2 3 fo	algorithm greedy_modularity louvain k-clique label_propagation otball algorithm greedy_modularity louvain	0.24506 0.37911 0.55579 0.20336 Jaccar 0.05867 0.71300 0.07978	68 0.0109 60 0.5162 91 0.0000 63 0.0411 7d N 75 0.1479 94 0.8922 96 0.0000	11 0.628715 34 0.364424 00 0.219805 41 0.717362 MI NVI 75 1.086858 96 0.155381 00 0.746947
0 1 2 3 fo 0 1 2	algorithm greedy_modularity louvain k-clique label_propagation  otball algorithm greedy_modularity louvain k-clique label_propagation	0.24506 0.37911 0.55579 0.20336 Jaccar 0.05867 0.71300 0.07978	68 0.0109 60 0.5162 91 0.0000 63 0.0411 7d N 75 0.1479 94 0.8922 96 0.0000	11 0.628715 34 0.364424 00 0.219805 41 0.717362 MI NVI 75 1.086858 96 0.155381 00 0.746947
0 1 2 3 fo 0 1 2	algorithm greedy_modularity louvain k-clique label_propagation otball algorithm greedy_modularity louvain k-clique label_propagation	0.24506 0.37911 0.55579 0.20336 Jaccar 0.05867 0.71306 0.07978 0.04461	68 0.0109 66 0.5162 01 0.0000 63 0.0411 6d N 75 0.1479 94 0.8922 66 0.0000 0.2387	11 0.628715 34 0.364424 00 0.219805 41 0.717362 MI NVI 75 1.086858 96 0.155381 00 0.746947 07 1.113253
0 1 2 3 fo 0 1 2 3	algorithm greedy_modularity louvain k-clique label_propagation otball algorithm greedy_modularity louvain k-clique label_propagation 6_4_4_4_13_18_p algorithm	0.24506 0.37911 0.55579 0.20336 Jaccar 0.05867 0.71306 0.07978 0.04461	88 0.0109 66 0.5162 01 0.0000 33 0.0411 rd N 75 0.1479 44 0.8922 66 0.0000 0.2387	11 0.628715 0.364424 00 0.219805 41 0.717362 MI NVI 75 1.086858 96 0.155381 00 0.746947 07 1.113253
0 1 2 3 fo 0 1 2 3	algorithm greedy_modularity louvain k-clique label_propagation otball algorithm greedy_modularity louvain k-clique label_propagation 6_4_4_13_18_p algorithm greedy_modularity	0.24506 0.37911 0.55579 0.20336 Jaccar 0.05867 0.71300 0.07978 0.04461	88 0.0109 66 0.5162 10 0.0000 63 0.0411 75 0.1479 144 0.8922 66 0.0000 0.2387	11 0.628715 34 0.364424 00 0.219805 41 0.717362 MI NVI 75 1.086858 96 0.155381 00 0.746947 1.113253  NVI 0.0000000
0 1 2 3 fo 0 1 2 3 25	algorithm greedy_modularity louvain k-clique label_propagation otball algorithm greedy_modularity louvain k-clique label_propagation 6_4_4_4_13_18_p algorithm greedy_modularity louvain louvain	0.24506 0.37911 0.55579 0.20336 Jaccar 0.05867 0.71300 0.07978 0.04461	68 0.0109 66 0.5162 10 0.0000 63 0.0411 64 N 75 0.1479 49 0.8922 66 0.0000 80 0.2387 64 NMI 10000 10000 1000000 100000 1000000 1000000 1000000 100000 100000 100000 100000 1000000 10000	11 0.628715 34 0.364424 00 0.219805 41 0.717362 MI NVI 75 1.0868558 96 0.155381 00 0.746947 07 1.113253  NVI 0.000000 0.000000
0 1 2 3 fo 0 1 2 3	algorithm greedy_modularity louvain k-clique label_propagation otball algorithm greedy_modularity louvain k-clique label_propagation 6_4_4_13_18_p algorithm greedy_modularity	0.24506 0.37911 0.55579 0.20336 Jaccar 0.05867 0.71306 0.07978 0.04461 Jaccar 1.00006 1.00006	88 0.0109 66 0.5162 10.0000 63 0.0411 75 0.1479 94 0.8922 66 0.0000 0.2387 74 NMI 1.0000 99 0.0000	11 0.628715 0.364424 00 0.219805 41 0.717362 MI NVI 75 1.086858 96 0.155381 00 0.746947 07 1.113253 NVI 0.000000 0.000000 0.360674
0 1 2 3 fo 0 1 2 3 25 0 1 2	algorithm greedy_modularity louvain k-clique label_propagation otball algorithm greedy_modularity louvain k-clique label_propagation 6_4_4_4_13_18_p algorithm greedy_modularity louvain k-clique	0.24506 0.37911 0.55579 0.20336 Jaccar 0.05867 0.71306 0.07978 0.04461 Jaccar 1.00006 1.00006	88 0.0109 66 0.5162 10.0000 63 0.0411 75 0.1479 94 0.8922 66 0.0000 0.2387 74 NMI 1.0000 99 0.0000	11 0.628715 0.364424 00 0.219805 41 0.717362 MI NVI 75 1.086858 96 0.155381 00 0.746947 07 1.113253 NVI 0.000000 0.000000 0.360674
0 1 2 3 fo 0 1 2 3 25 0 1 2 3	algorithm greedy_modularity louvain k-clique label_propagation otball algorithm greedy_modularity louvain k-clique label_propagation 6_4_4_4_13_18_p algorithm greedy_modularity louvain k-clique	0.24506 0.37911 0.55579 0.20336 Jaccar 0.05867 0.71306 0.07978 0.04461 Jaccar 1.00006 1.00006	88 0.0109 66 0.5162 10.0000 63 0.0411 75 0.1479 94 0.8922 86 0.0000 0.2387 74 NMI 1.0000 10 1.0000 10 0.6993	11 0.628715 34 0.364424 00 0.219805 41 0.717362 MI NVI 75 1.086858 96 0.155381 00 0.746947 07 1.113253  NVI 0.000000 0.000000 0.360674 0.263123
0 1 2 3 fo 0 1 2 3 25 0 1 2 3 rb	algorithm greedy_modularity louvain k-clique label_propagation  otball algorithm greedy_modularity louvain k-clique label_propagation  6_4_4_13_18_p algorithm greedy_modularity louvain k-clique label_propagation  125 algorithm	0.24506 0.37911 0.55579 0.20336 Jaccar 0.05867 0.71300 0.07978 0.04461 Jaccar 1.00000 0.24705 0.41333	88 0.0109 66 0.5162 10 0.0000 63 0.0411 64 N 75 0.1479 14 0.8922 86 0.0000 80 0.2387 64 NMI 100 1.0000 100 1.0000 100 0.0000 100 0.0000	11 0.628715 34 0.364424 00 0.219805 41 0.717362 MI NVI 75 1.086858 96 0.155381 00 0.746947 07 1.113253  NVI 0.000000 0.000000 0.360674 0.263123
0 1 2 3 fo 0 1 2 3 25 0 1 2 3 rb	algorithm greedy_modularity louvain k-clique label_propagation  otball algorithm greedy_modularity louvain k-clique label_propagation  6_4_4_4_13_18_p algorithm greedy_modularity louvain k-clique label_propagation  125 algorithm greedy_modularity	0.24596 0.37911 0.55579 0.20336 Jaccar 0.05867 0.71396 0.07978 0.04461 Jaccar 1.00008 0.24705 0.41333	88 0.0109 66 0.5162 61 0.0000 63 0.0411 64 N 75 0.1479 64 0.8922 66 0.0000 60 0.2387 64 NMI 60 1.0000 60 0.0000 61 0.0000 63 0.6993	11 0.628715 0.364424 00 0.219805 41 0.717362  MI NVI 75 1.086858 96 0.155381 00 0.746947 07 1.113253  NVI 0.000000 0.360674 0.263123  MI NVI 43 0.287653
0 1 2 3 fo 0 1 2 3 25 0 1 2 3 rb	algorithm greedy_modularity louvain k-clique label_propagation otball algorithm greedy_modularity louvain k-clique label_propagation 6_4_4_4_13_18_p algorithm greedy_modularity louvain k-clique label_propagation 125 algorithm greedy_modularity louvain label_propagation 125 algorithm greedy_modularity louvain	0.24506 0.37911 0.55579 0.20336 Jaccar 0.05867 0.71306 0.07978 0.04461 Jaccar 1.00006 0.24705 0.41333 Jaccar 0.28114	88 0.0109 60 0.5162 61 0.0000 63 0.0411 64 N 75 0.1479 74 0.8922 75 0.0000 76 0.0000 77 0.0000 78 0.0000 7	11 0.628715 34 0.36442 00 0.219805 41 0.717362 MI NVI 75 1.086858 96 0.155381 00 0.746947 07 1.113253  NVI 0.000000 0.000000 0.360674 0.263123  MI NVI 43 0.287653 43 0.287653
0 1 2 3 fo 0 1 2 3 25 0 1 2 3 rb	algorithm greedy_modularity louvain k-clique label_propagation  otball algorithm greedy_modularity louvain k-clique label_propagation  6_4_4_13_18_p algorithm greedy_modularity louvain k-clique label_propagation  125 algorithm greedy_modularity louvain k-clique label_propagation  125 algorithm greedy_modularity louvain k-clique louvain k-clique	0.24596 0.37911 0.55579 0.20336 Jaccar 0.05867 0.71306 0.07978 0.04461 Jaccar 1.00006 0.24705 0.41333 Jaccar 0.28114 0.28114	88 0.0109 60 0.5162 10 0.0000 63 0.0411 MMI 10 0.2387 MMI 10 1.0000 10 1.0000 10 0.0000 10 0.0000 1	11 0.628715 34 0.364424 0 0.219805 41 0.717362 MI NVI 75 1.086858 96 0.155381 00 0.746947 1.113253  NVI 0.000000 0.000000 0.360674 0.263123  MI NVI 43 0.287653 00 0.967777
0 1 2 3 fo 0 1 2 3 25 0 1 2 3 rb	algorithm greedy_modularity louvain k-clique label_propagation otball algorithm greedy_modularity louvain k-clique label_propagation 6_4_4_4_13_18_p algorithm greedy_modularity louvain k-clique label_propagation 125 algorithm greedy_modularity louvain label_propagation 125 algorithm greedy_modularity louvain	0.24596 0.37911 0.55579 0.20336 Jaccar 0.05867 0.71306 0.07978 0.04461 Jaccar 1.00006 0.24705 0.41333 Jaccar 0.28114 0.28114	88 0.0109 60 0.5162 10 0.0000 63 0.0411 MMI 10 0.2387 MMI 10 1.0000 10 1.0000 10 0.0000 10 0.0000 1	11 0.628715 34 0.364424 0 0.219805 41 0.717362 MI NVI 75 1.086858 96 0.155381 00 0.746947 1.113253  NVI 0.000000 0.000000 0.360674 0.263123  MI NVI 43 0.287653 00 0.967777
0 1 2 3 fo 0 1 2 3 25 0 1 2 3 rb	algorithm greedy_modularity louvain k-clique label_propagation  otball algorithm greedy_modularity louvain k-clique label_propagation  6_4_4_13_18_p algorithm greedy_modularity louvain k-clique label_propagation  125 algorithm greedy_modularity louvain k-clique label_propagation  125 algorithm greedy_modularity louvain k-clique label_propagation	0.24596 0.37911 0.55579 0.20336 Jaccar 0.05867 0.71306 0.07978 0.04461 Jaccar 1.00006 0.24705 0.41333 Jaccar 0.28114 0.28114	88 0.0109 60 0.5162 10 0.0000 63 0.0411 MMI 10 0.2387 MMI 10 1.0000 10 1.0000 10 0.0000 10 0.0000 1	11 0.628715 34 0.364424 0 0.219805 41 0.717362 MI NVI 75 1.086858 96 0.155381 00 0.746947 1.113253  NVI 0.000000 0.000000 0.360674 0.263123  MI NVI 43 0.287653 00 0.967777
0 1 2 3 fo 0 1 2 3 25 0 1 2 3 rb	algorithm greedy_modularity louvain k-clique label_propagation  otball algorithm greedy_modularity louvain k-clique label_propagation  6_4_4_13_18_p algorithm greedy_modularity louvain k-clique label_propagation  125 algorithm greedy_modularity louvain k-clique label_propagation  125 algorithm greedy_modularity louvain k-clique louvain k-clique	0.24506 0.37911 0.55579 0.20336 Jaccar 0.05867 0.71300 0.04461 Jaccar 1.00008 0.24705 0.41333 Jaccar 0.28114 0.28114 0.3174 0.15913	88 0.0109 60 0.5162 61 0.0000 63 0.0411 6d N 75 0.1479 64 0.8922 66 0.0000 60 0.2387 6d NMI 60 1.0000 61 0.0000 63 0.6993 64 N 63 0.8254 63 0.8254 64 0.8254 65 0.7540	11 0.628715 34 0.364424 0 0.219805 41 0.717362 MI NVI 75 1.086858 96 0.155381 00 0.746947 1.113253  NVI 0.000000 0.000000 0.360674 0.263123  MI NVI 43 0.287653 00 0.967777
0 1 2 3 fo 0 1 2 3 25 0 1 2 3 rb	algorithm greedy_modularity louvain k-clique label_propagation  otball algorithm greedy_modularity louvain k-clique label_propagation  6_4_4_4_13_18_p algorithm greedy_modularity louvain k-clique label_propagation  125 algorithm greedy_modularity louvain k-clique label_propagation  125 algorithm greedy_modularity louvain k-clique	0.24506 0.37911 0.55579 0.20336 Jaccar 0.05867 0.71306 0.07978 0.04461 Jaccar 1.00006 0.24705 0.41333 Jaccar 0.28114 0.3174 0.15913	88 0.0109 66 0.5162 10.0000 63 0.0411 64 N 75 0.1479 94 0.8922 96 0.0000 90 0.2387 64 NMI 1.0000 90 0.0000 93 0.6993 64 N 93 0.8254 93 0.8254 94 0.8254 95 0.0000 96 0.0000	11 0.628715 34 0.36424 00 0.219805 41 0.717362 MI NVI 75 1.086858 96 0.155381 00 0.746947 07 1.113253  NVI 0.000000 0.360674 0.263123  MI NVI 43 0.287653 00 0.967777 33 0.420127  MI NVI 52 0.196748
0 1 2 3 fo 0 1 2 3 rb 0 1 2 3 25 0 1	algorithm greedy_modularity louvain k-clique label_propagation  otball algorithm greedy_modularity louvain k-clique label_propagation  6_4_4_13_18_p algorithm greedy_modularity louvain k-clique label_propagation  125 algorithm greedy_modularity louvain k-clique label_propagation  125 algorithm greedy_modularity louvain k-clique label_propagation 6_4_4_2_15_18_p algorithm greedy_modularity louvain greedy_modularity	0.24596 0.37911 0.55579 0.20336  Jaccar 0.05867 0.71396 0.07978 0.04461  Jaccar 1.00006 0.24705 0.41333  Jaccar 0.28114 0.28114 0.3174 0.15913	88 0.0109 60 0.5162 10.0000 63 0.0411 64 N 75 0.1479 64 0.8922 66 0.0000 60 0.2387 64 NMI 60 1.0000 60 1.0000 61 0.0000 63 0.6993 64 N 63 0.8254 63 0.8254 64 0.0000 62 0.7540 64 N 63 0.8254 65 0.8254 66 0.8858	11 0.628715 0.364424 00 0.219805 41 0.717362  MI NVI 75 1.086858 96 0.155381 00 0.746947 07 1.113253  NVI 0.000000 0.360674 0.263123  MI NVI 43 0.287653 43 0.287653 43 0.287653 43 0.287653 43 0.287653 43 0.287653 43 0.287653 43 0.287653
0 1 2 3 fo 0 1 2 3 rb 0 1 2 3 25 0	algorithm greedy_modularity louvain k-clique label_propagation otball algorithm greedy_modularity louvain k-clique label_propagation 6_4_4_13_18_p algorithm greedy_modularity louvain k-clique label_propagation 125 algorithm greedy_modularity louvain k-clique label_propagation 125 algorithm greedy_modularity louvain k-clique label_propagation 6_4_4_2_15_18_p algorithm greedy_modularity	0.24596 0.37911 0.55579 0.20336  Jaccar 0.05867 0.71396 0.07978 0.04461  Jaccar 1.00006 0.24705 0.41333  Jaccar 0.28114 0.28114 0.3174 0.15913	88 0.0109 66 0.5162 10.0000 63 0.0411 64 N 75 0.1479 944 0.8922 66 0.0000 60 0.2387 64 NMI 1.0000 63 0.6993 64 NM3 0.8254 62 0.0000 62 0.7540 64 N 63 0.8458 64 0.0000 64 0.0000	11 0.628715 34 0.364424 00 0.219805 41 0.717362  MI NVI 75 1.086858 96 0.155381 00 0.746947 07 1.113253  NVI 0.000000 0.360674 0.263123  MI NVI 43 0.287653 43 0.287653 43 0.287653 30 0.420127  MI NVI 52 0.967777 33 0.420127

## 2.4 Modularity measures

This section's goal is to be able to provide a comparison of the modularity values between all the partitions obtained from the different algorithms, including the reference partitions. Modularity is a measure of the quality of a partition, which represents the degree to which a network can be divided into non-overlapping and densely connected groups.

By grouping the modularity values by network, we can compare the performance of each algorithm in identifying meaningful communities within each network. The table allows for a quick and easy comparison between the modularity values of the different partitions, allowing researchers to identify which algorithm performed better in identifying community structures in a particular network.

Table representing the modularity measures using igraph algorithms

	fastgreedy	louvain	label_prop
rb125	0.608733	0.608733	0.525830
256_4_4_4_13_18_p	0.696773	0.696773	0.663378
256_4_4_2_15_18_p	0.765660	0.781804	0.781804
zachary_unwh	0.380671	0.398176	0.132807
football	0.549741	0.604407	0.589858
dolphins	0.495491	0.523338	0.426526
cat_cortex_sim	0.260436	0.266097	0.000000
airports_UW	0.662490	0.699136	0.627621
grid-p-6x6	0.401235	0.416667	0.333333
graph4+4	0.423077	0.423077	0.423077
star	0.000000	0.000000	0.000000
graph3+1+3	0.367188	0.367188	0.367188
20x2+5x2	0.542579	0.542579	0.541586
		modu1	arity
graph3+	· <b>1</b> +3		arity 51562
graph3+ 20x2+5		0.3	
	ix2	0.3	51562
20x2+5	5x2 I+4	0.3 0.5 0.4	51562 41586
20x2+5 graph4	5x2 l+4	0.3 0.5 0.4 0.0	51562 41586 23077
20x2+5 graph4 star	5x2 l+4 x_sim	0.3 0.5 0.4 0.0	51562 41586 23077 00000
20x2+5 graph4 star cat_corte	5x2 l+4 x_sim wh-real	0.3 0.5 0.4 0.0 0.2	51562 41586 23077 00000 45996
20x2+5 graph4 star cat_corte zachary_un	5x2 l+4 x_sim wh-real	0.3 0.5 0.4 0.0 0.2 0.3	51562 41586 23077 00000 45996 71466
20x2+5 graph4 star cat_corte zachary_un dolphins	5x2 l+4 x_sim wh-real s-real	0.3 0.5 0.4 0.0 0.2 0.3 0.3	51562 41586 23077 00000 45996 71466 73482
20x2+5 graph4 star cat_corte zachary_un dolphins football-con	5x2 4+4 x_sim wh-real s-real ferences 13_18_p	0.3 0.5 0.4 0.0 0.2 0.3 0.3 0.5	51562 41586 23077 00000 45996 71466 73482 53973
20x2+5 graph4 star cat_corte zachary_un dolphins football-con	ix2 l+4 x_sim wh-real i-real ferences 13_18_p	0.3 0.5 0.4 0.0 0.2 0.3 0.5 0.6	51562 41586 23077 00000 45996 71466 73482 53973 96773
20x2+5 graph4 star cat_corte zachary_un dolphins football-con 256_4_4_4_ rb125	5x2 4+4 x_sim wh-real 6-real ferences 13_18_p -1	0.3 0.5 0.4 0.0 0.2 0.3 0.3 0.5 0.6 0.6	51562 41586 23077 00000 45996 71466 73482 53973 96773

#### 2.5 Conclusions about the algorithms

First, in the comparison part, we can observe that all algorithms successfully get a perfect score for every metric for several datasets, including graph4+4, star, and dolphins. This shows that clustering these datasets might be simple. Other datasets i te other hand, like catcortexsim, football, and zacharyunwh, have scores that differ between algorithms, indicating that clustering may be more difficult for these datasets.

When it comes to the algorithms presented in the igraph algorithms comparison table, we can see that the "labelpropagation" algorithm performs well when the data contains distinct clusters or communities, as shown by the high Jaccard and NMI scores it receives on numerous datasets. The "fast-greedy" and "leadingeigenvector" algorithms, on the other hand, typically perform well across many graph types, scoring highly in the Jaccard and NMI metrics on the majority of the datasets we used. Last but not least, the "louvain" algorithm performs occasionally well but less regularly than the others.

If we compare the results we got after using networkx algorithms, we can conclude that in terms

of Jaccard, NMI, and NVI scores, the algorithms "greedymodularity" and "louvain" generally outperform "kclique" and "labelpropagation." On many datasets, these two algorithms frequently gave us excellent results, occasionally even perfect results and scores. However, 'k-clique' and 'labelpropagation' frequently received lower scores, particularly in terms of NMI and NVI.

What can we conclude from these tables is that the algorithm "label propagation" is the one that most often gives more similar results to the reference partitions. Therefore it is not always the case since the algorithm is not deterministic, so running it again with the same graph may give us different outcomes.

In the modularity scores part, we can see that both Louvain and Label Propagation methods have greater modularity scores than Fast Greedy for the majority of network graphs. For instance, the Louvain and Label Propagation algorithms have modularity scores of 0.604407 and 0.589858, respectively, for the "football" network graph, while Fast Greedy got a score of 0.549741.

However, there are several exceptions to this observed pattern or conclusion. For instance, the Louvain and Label Propagation methods have higher modularity scores than Fast Greedy in the case of the "dolphins" network graph and the difference is not as pronounced as in other examples.

To conclude, we can say that selecting a community detection algorithm might affect the modularity score of a network graph and that in some cases we are required to test out various algorithms in order to get the best outcomes.