POLITECNICO
MILANO 1863

European road network analysis

Project for the course of "Complessità nei sistemi e nelle reti"

Andrea Bertogalli
218342

Accademic year 2022/2023

The dataset

The graph represents the European road network :

- A node represents a city.
- An edge represents a road directly connecting two cities.

There are 1174 nodes (cities), connected by 1417 edges (direct roads).

The network is undirected and unweighted.

Source: http://konect.cc/networks/subelj_euroroad/
Author: Lovro Šubelj (*University of Ljubljana*)

Related paper:
Robust network community detection using balanced propagation, L. Šubelj et. Al, 2011 (arxiv.org)

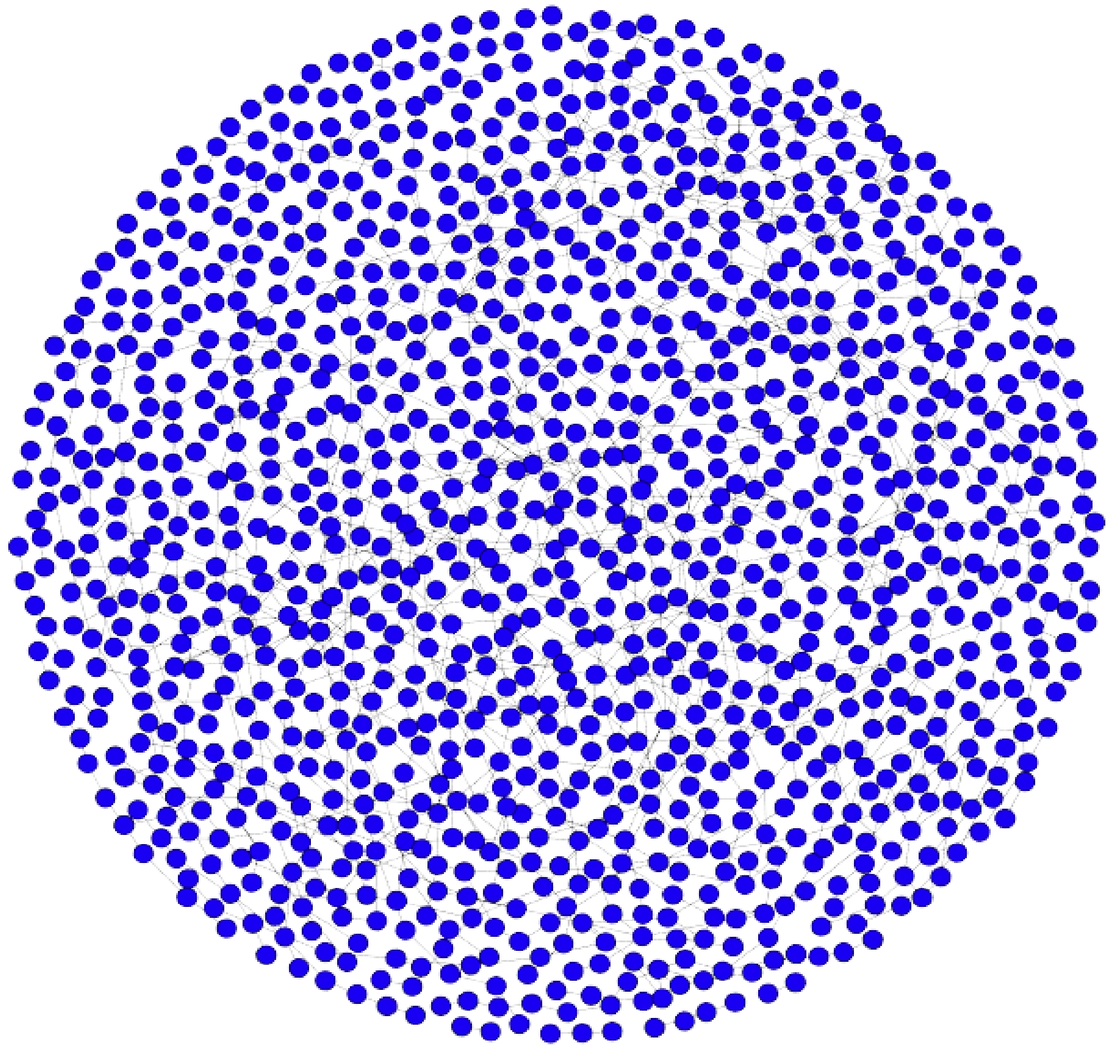


Fig.1: Fruchterman-Reingold layout plotted with Gephy.

Pre-processing

Before starting to analyze the network, some pre-processing on the data is necessary in order to improve the visualization and, consequently, to facilitate further analysis.

We can automatically retrieve the latitude and the longitude of each city exploiting the nodes' labels which correspond to the name.

Much better than the last slide…

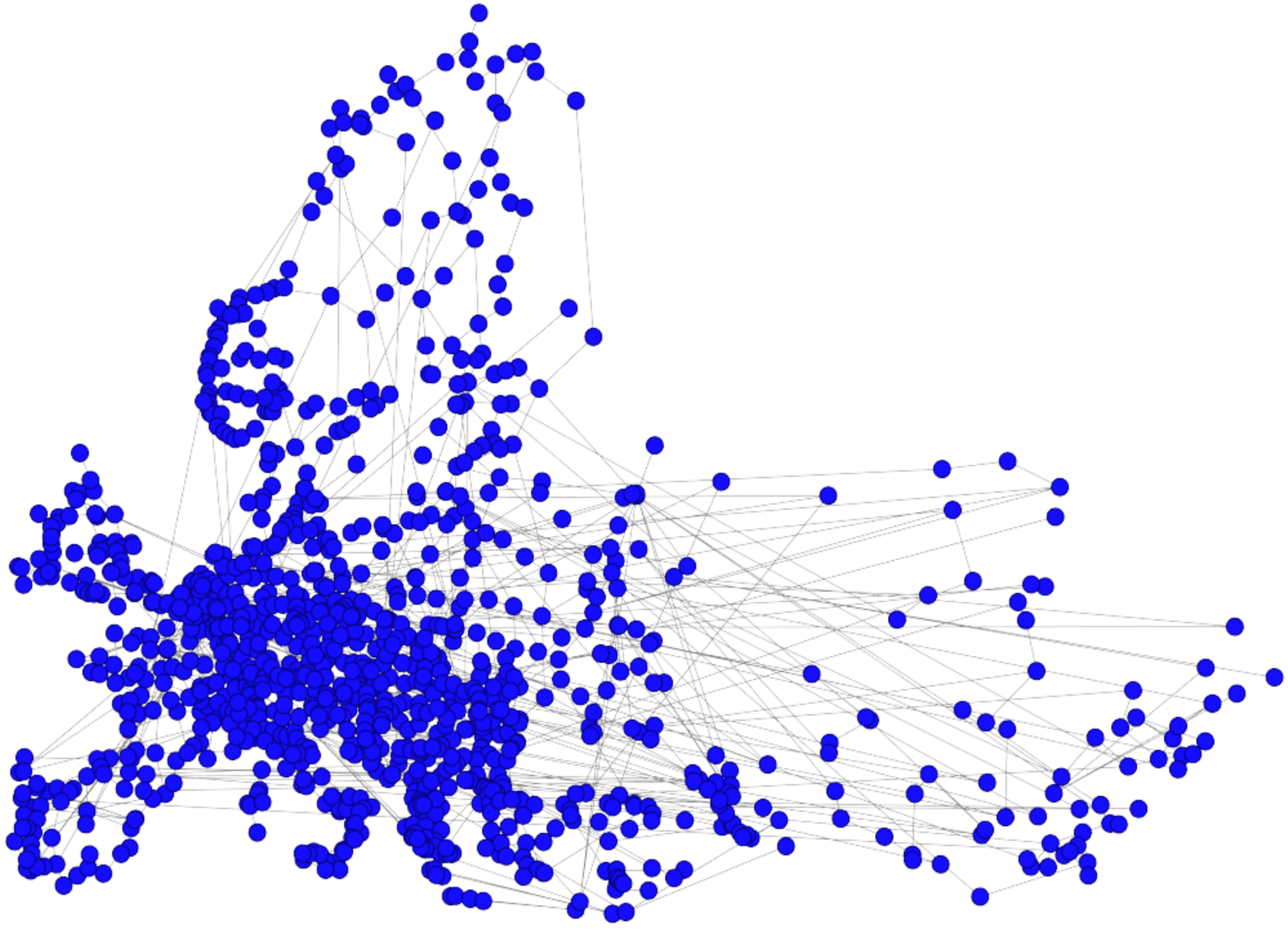


Fig.2: GeoLayout plotted with Gephy.

Pre-processing

We can also use a basemap to mark the Europe boundaries and see where the cities are located precisely.

Note that, due to the fact that coordinates are automatically retrieved, there may be some nodes drawn in the wrong position, but it's just a graphical matter.

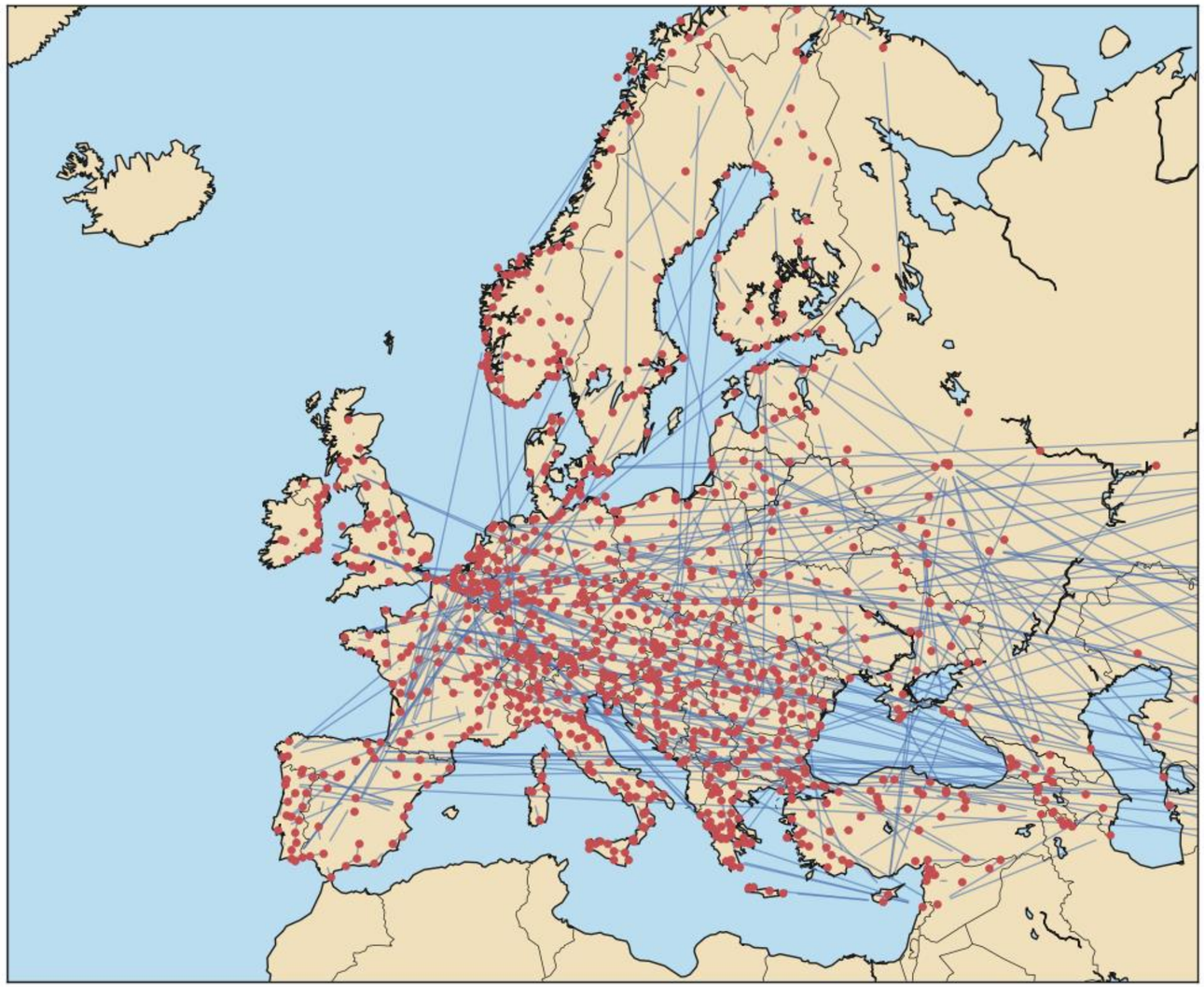


Fig.3: NetworkX graph plotted on a basemap using matplotlib.

Network properties

By analyzing the network, it can be seen that there are multiple connected components (26) but only one of them is relevant, so only the latter has been analyzed.

The giant component has 1305 edges and 1039 nodes.

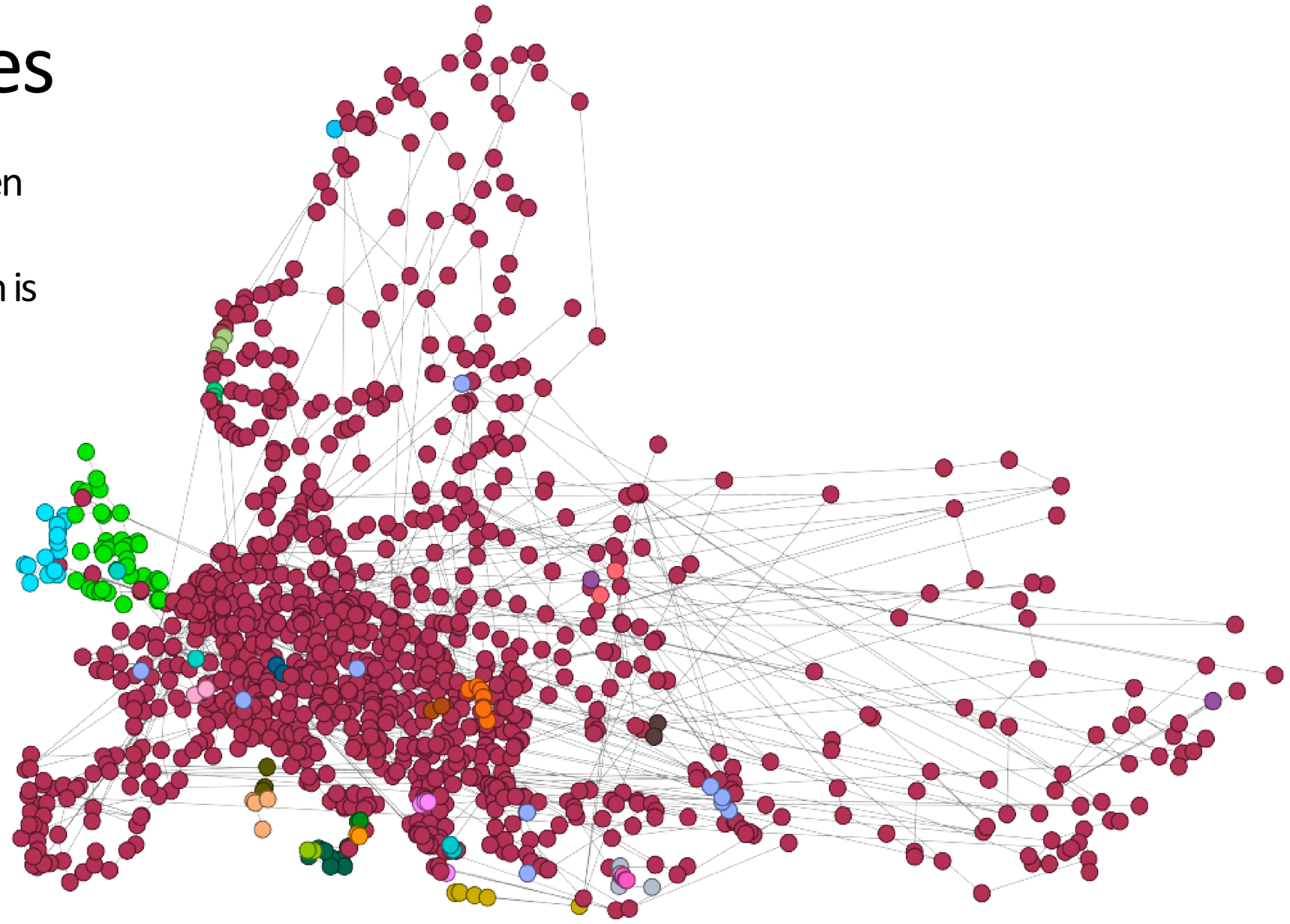


Fig.4: The network's components plotted with Gephy (dark red = giant component).

Network properties

Considering the average distance (≈18) and the degree distribution we can state that the network is neither a small word nor scale-free.

Number of nodes	1039
Number of edges	1305
Average degree	2.512
Average distance	18.395
Diameter	62
Global clustering coefficient	0.019
Density	0.002
Efficiency	0.077

Fig.5: Some properties of the network.

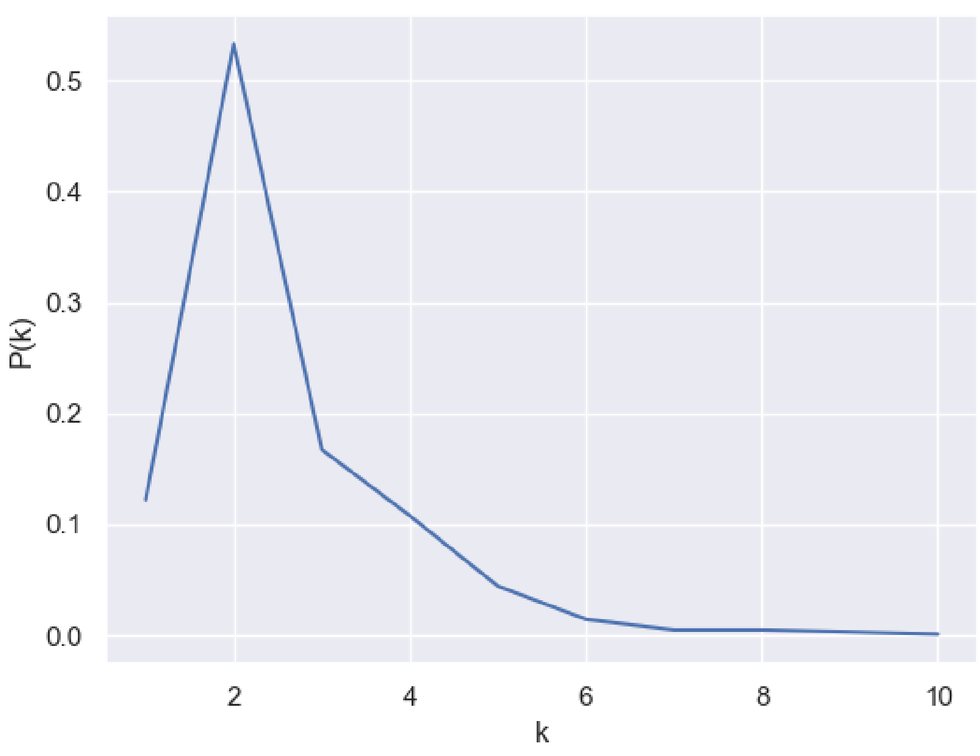


Fig.6: The degree distribution of the network.

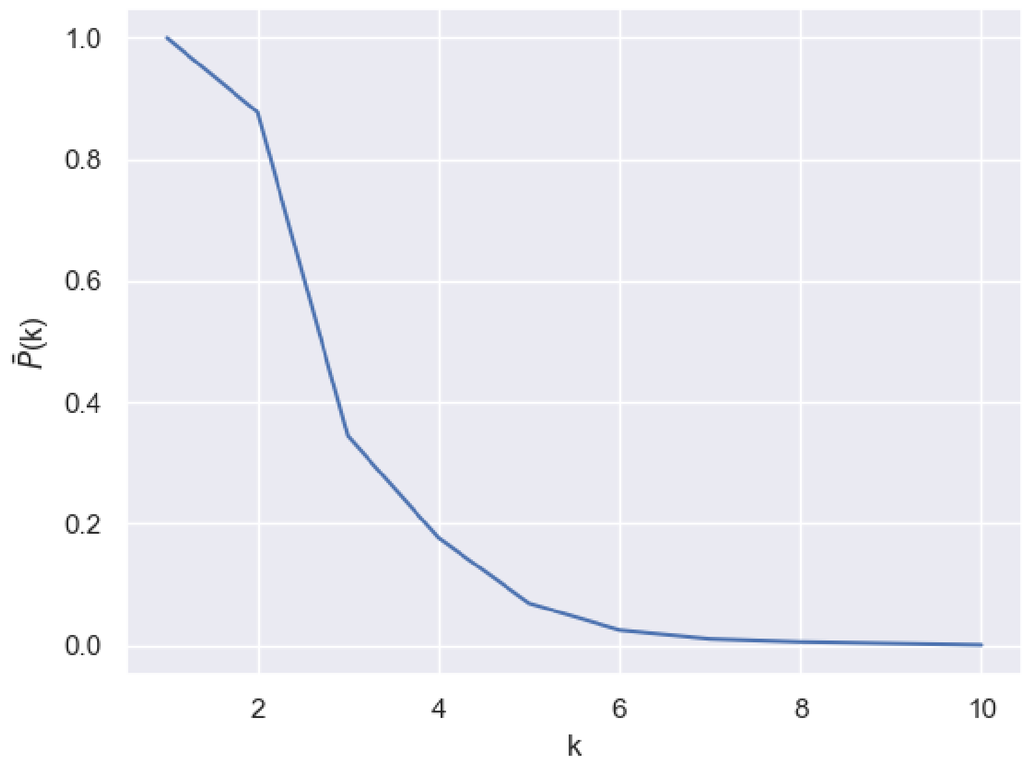


Fig.7: The cumulated degree distribution of the network.

Centralities - Degree

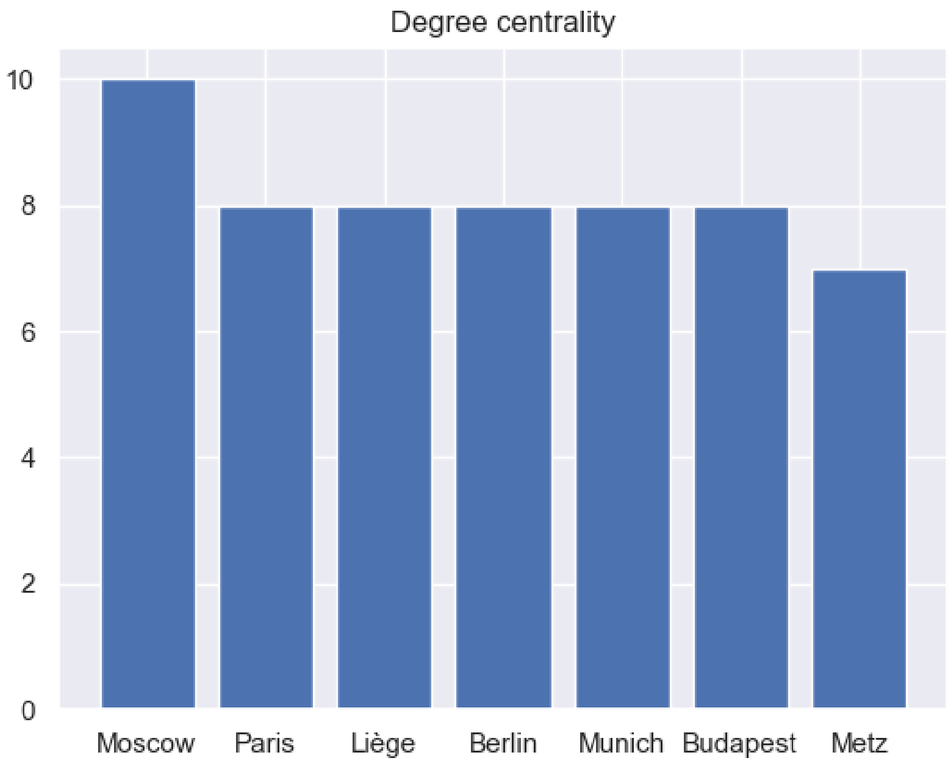


Fig.8: Top 7 for degree.

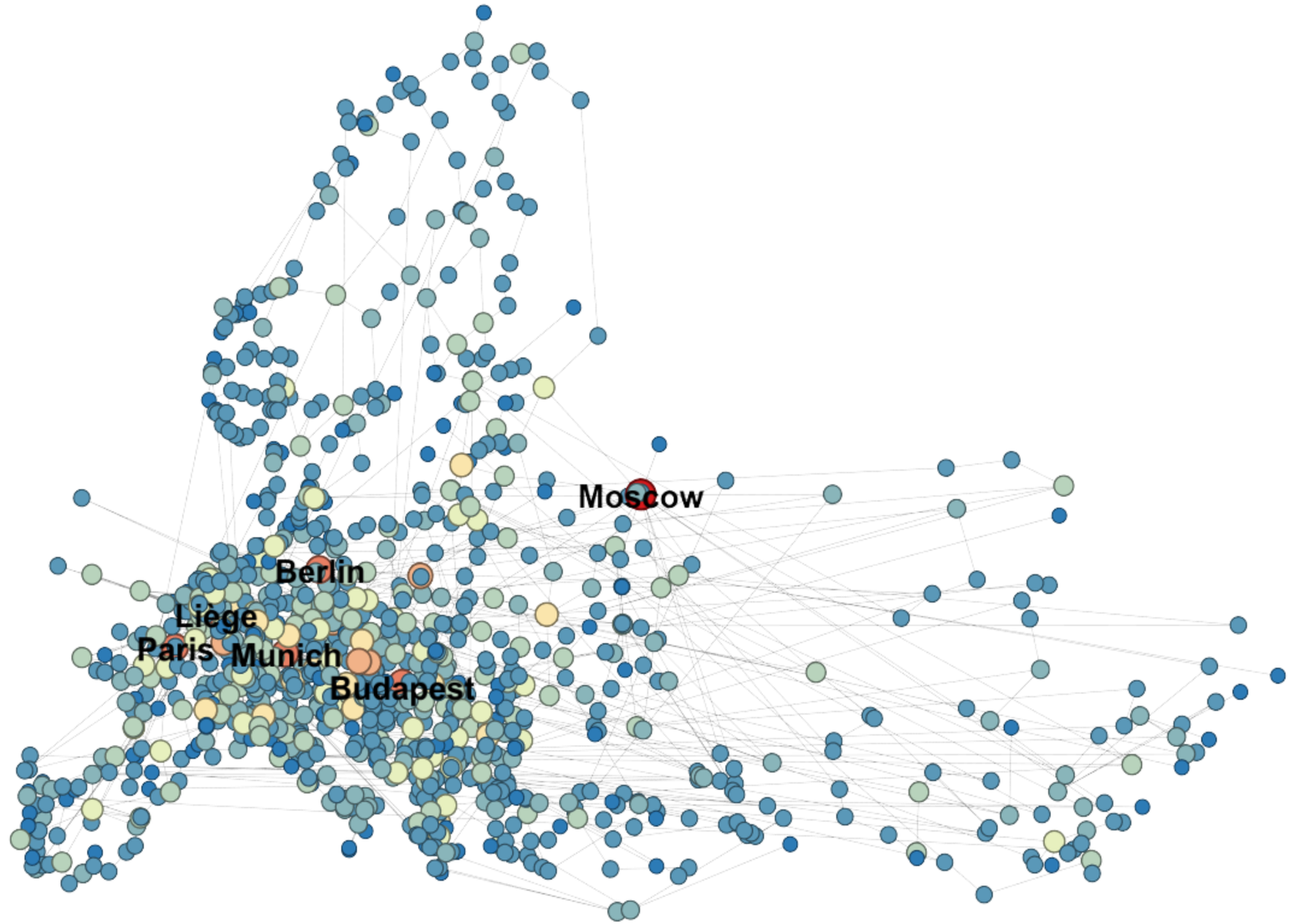


Fig.9: Degree centrality plotted with Gephy, red means high degree

Centralities - Closeness

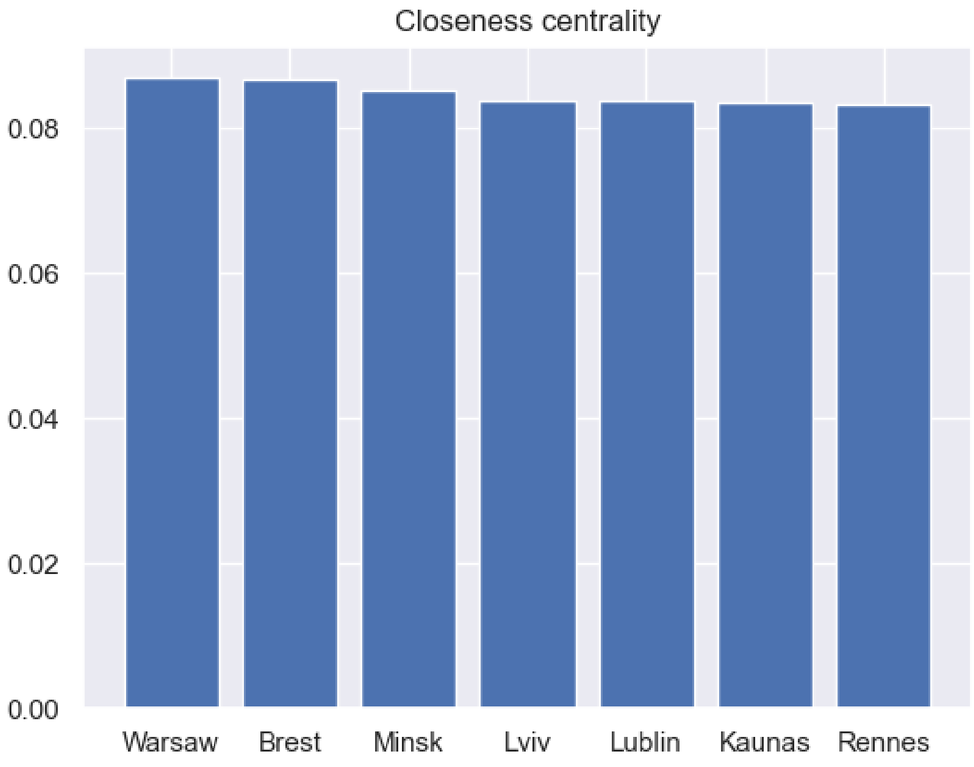


Fig.10: Top 7 for closeness centrality score.

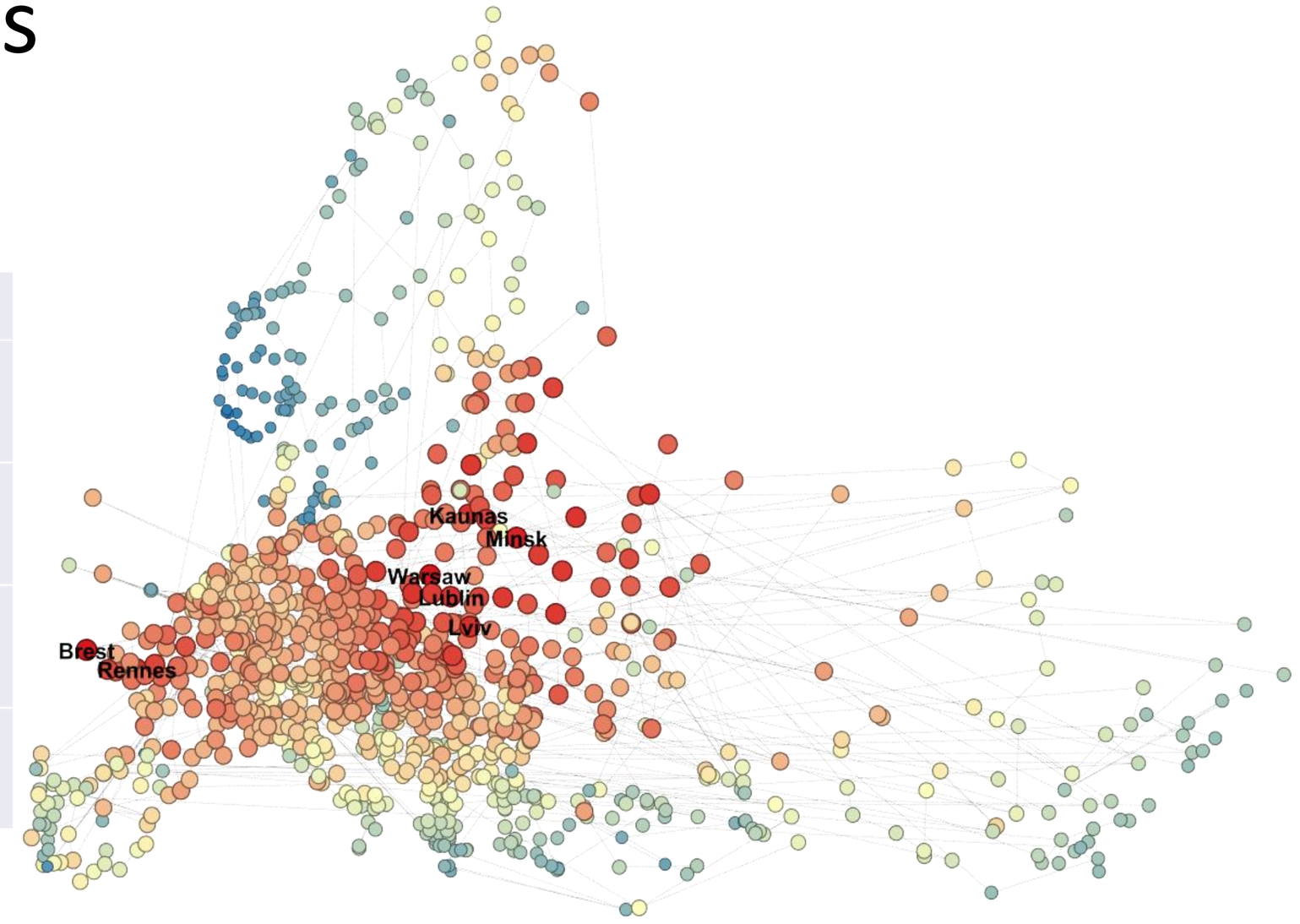


Fig.11: Closeness centrality score plotted with Gephy, rec means high closeness score.

Centralities - Betweenness

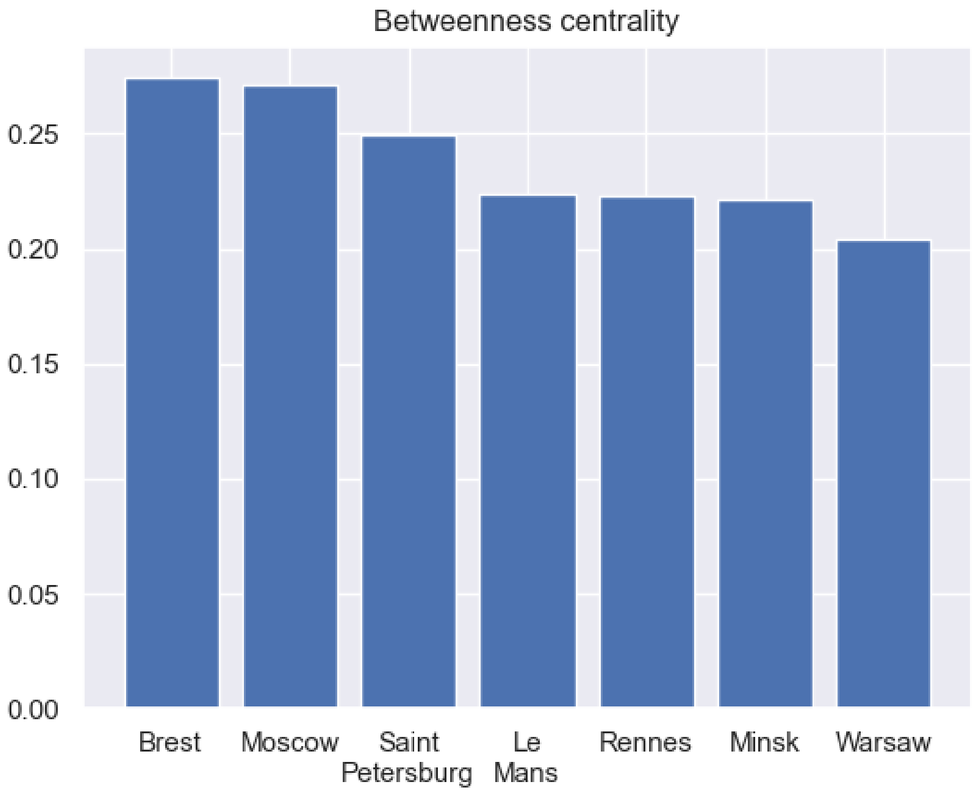


Fig.12: Top 7 for betweenness centrality score.

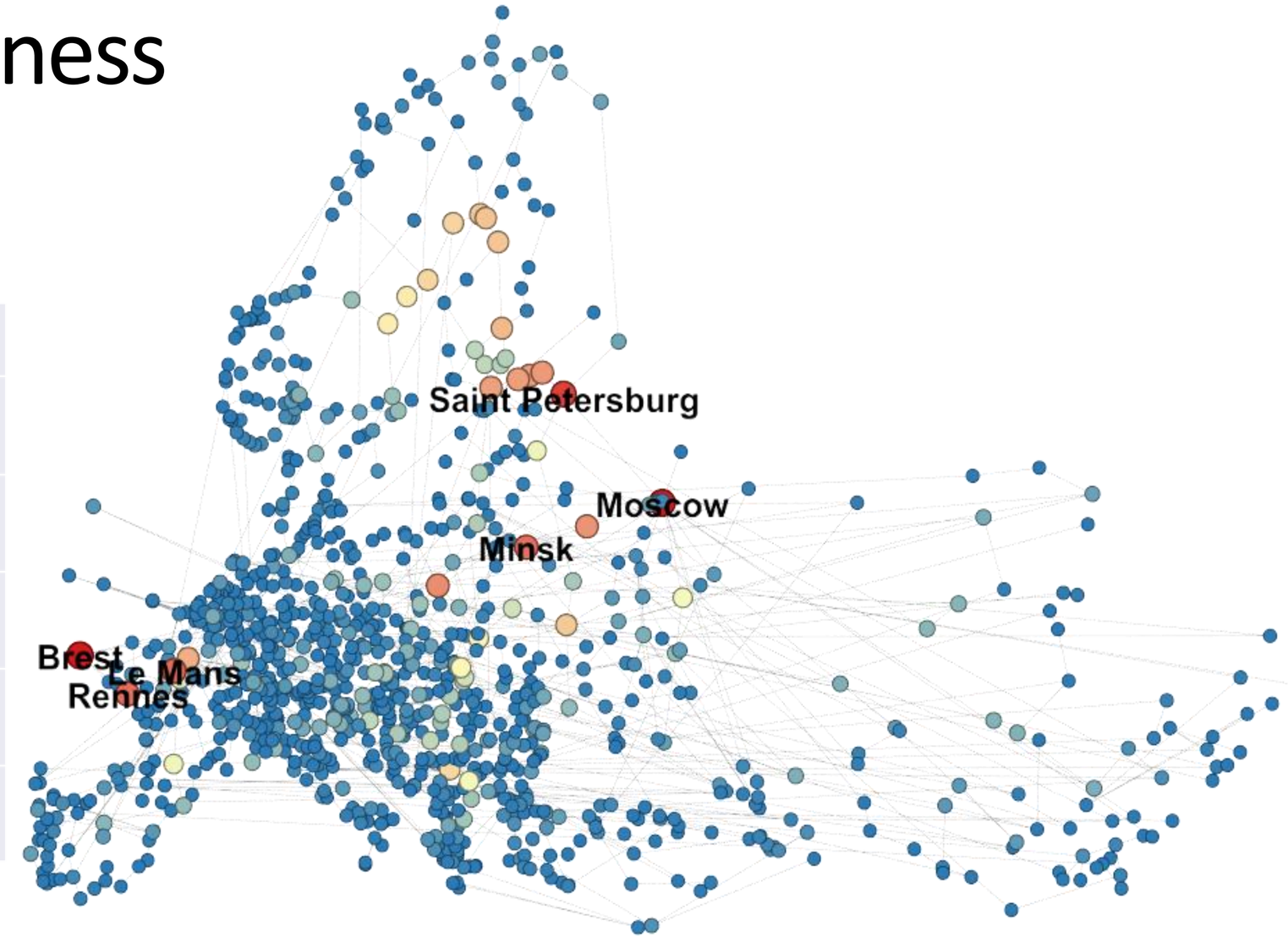


Fig.13: Betweenness centrality score plotted with Gephy, red means high betweenness score

Centralities - Eigenvector

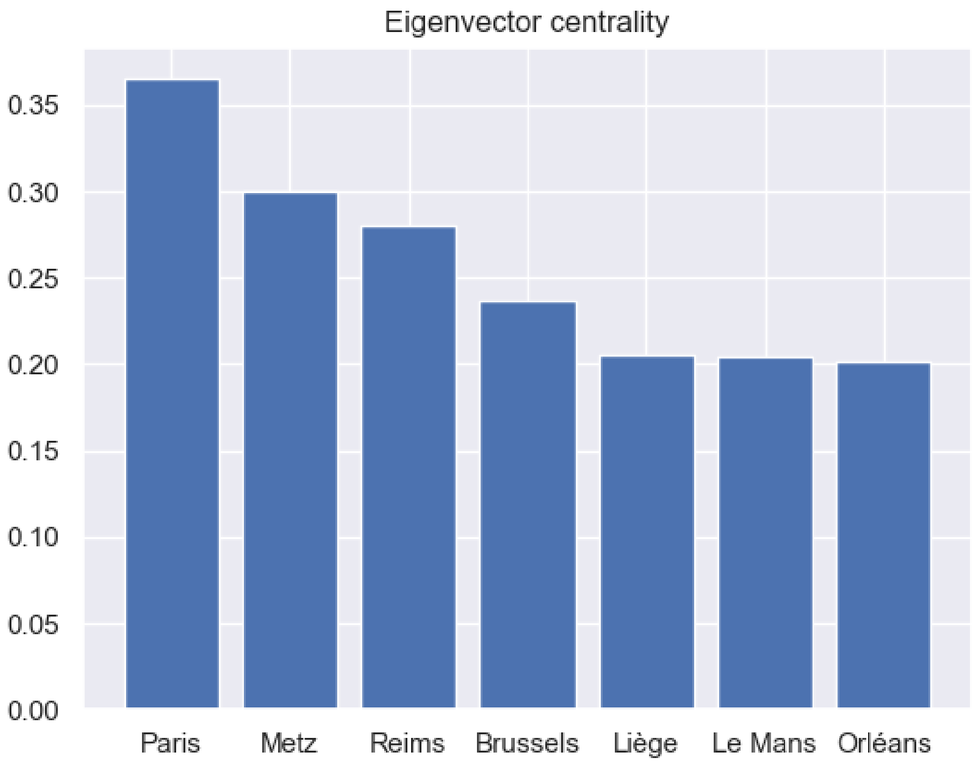


Fig.14: Top 7 for eigenvector centrality score.

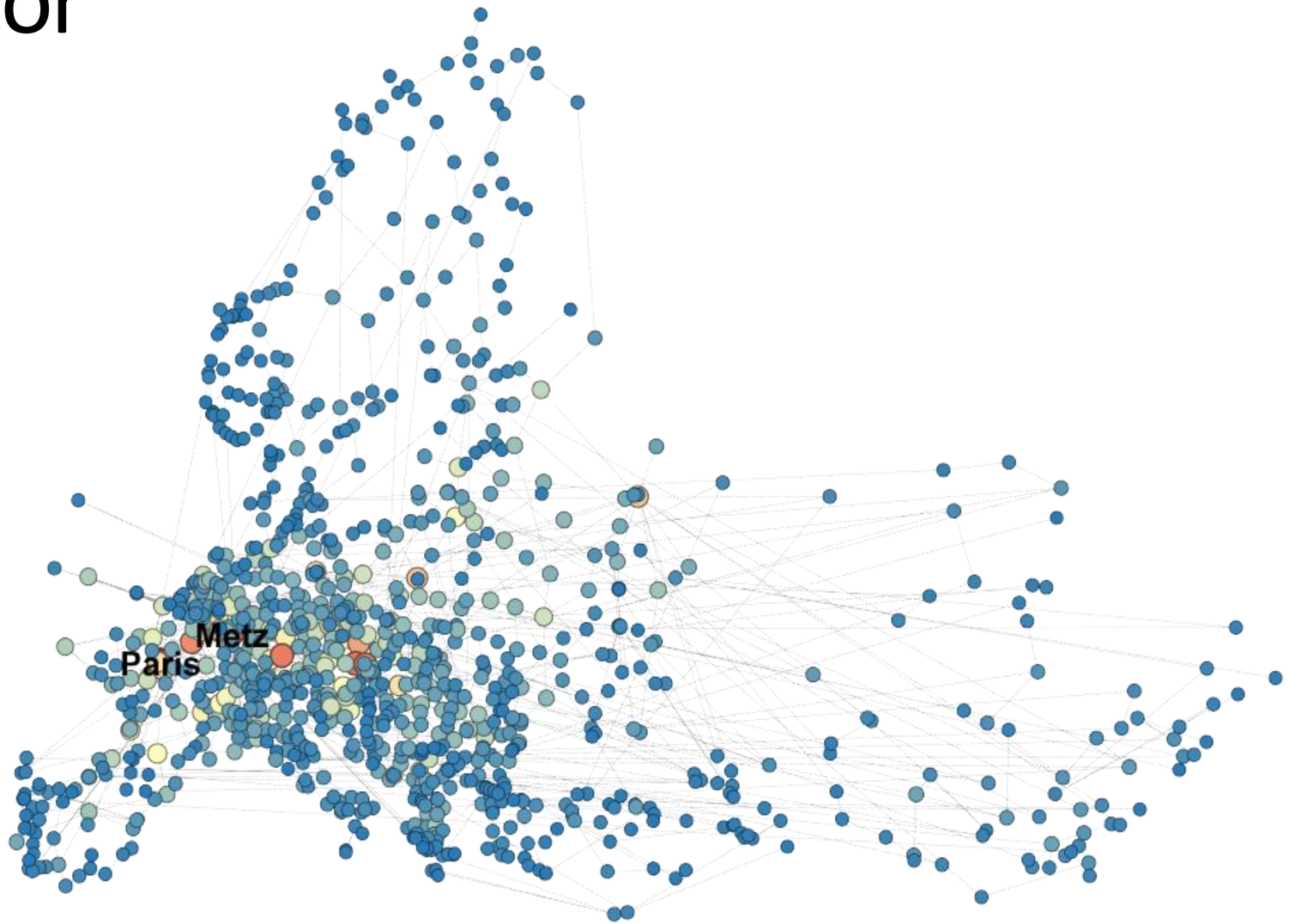


Fig.15: Eigenvector centrality score plotted with Gephy, red means high eigenvector score

Core-Periphery analysis

By applying the decomposition algorithm we notice that the network consists of only two shells. This binary division allows us to distinguish a periphery (1-Shell), composed by 303 nodes, and a core (2-Shell), composed by 736 nodes.

In the figure the **blue** nodes represent the 1-Shell while the **green** one the 2-Shell.

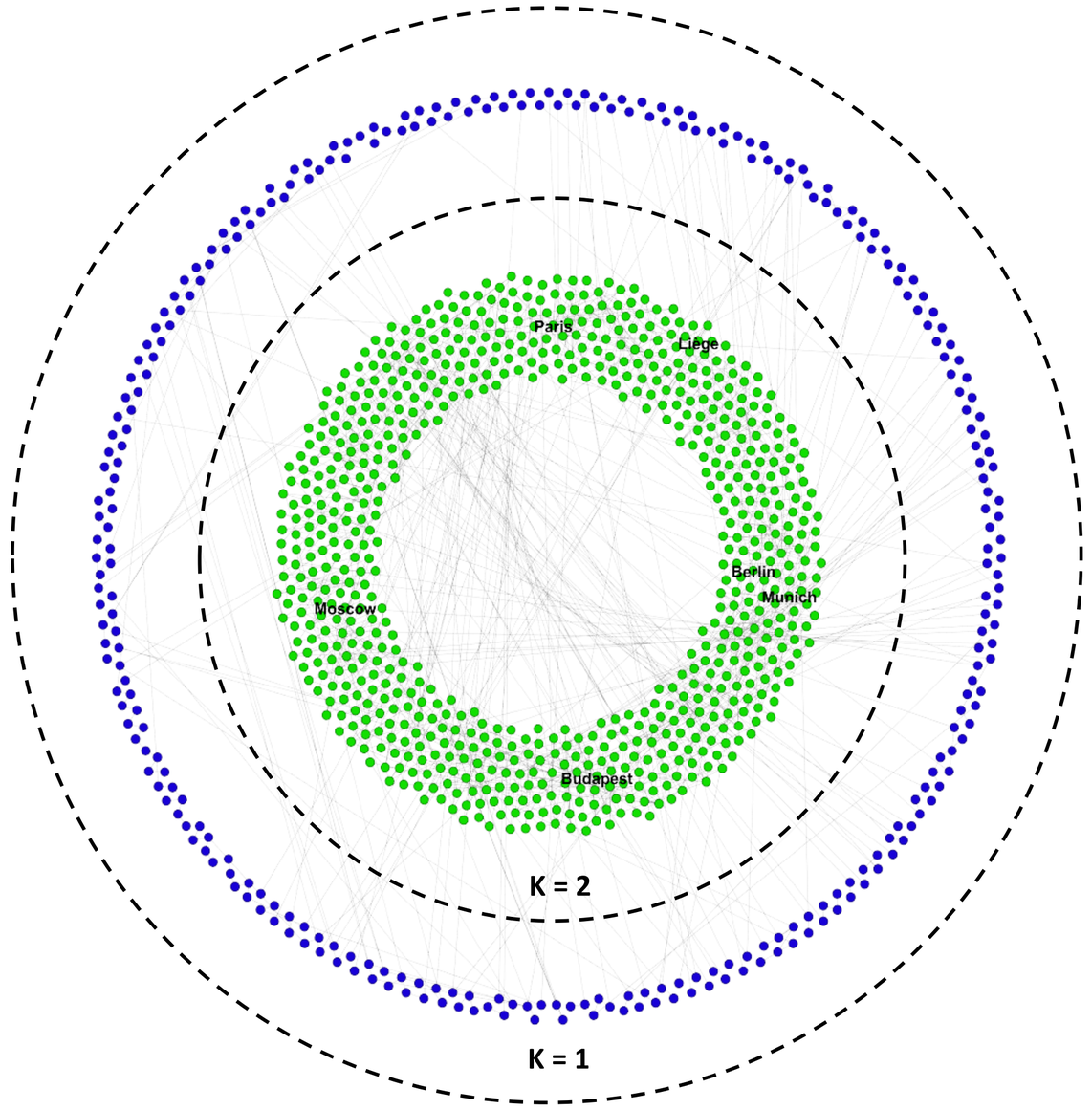


Fig.16: K-Core decomposition computed with networkX and plotted with Gephy (Dual Circle Layout).

Core-Periphery analysis

It's interesting to note that the periphery coincides fairly well with the geographic periphery of Europe, this highlights that the European road network is more sparse in the periphery while more dense in the core.

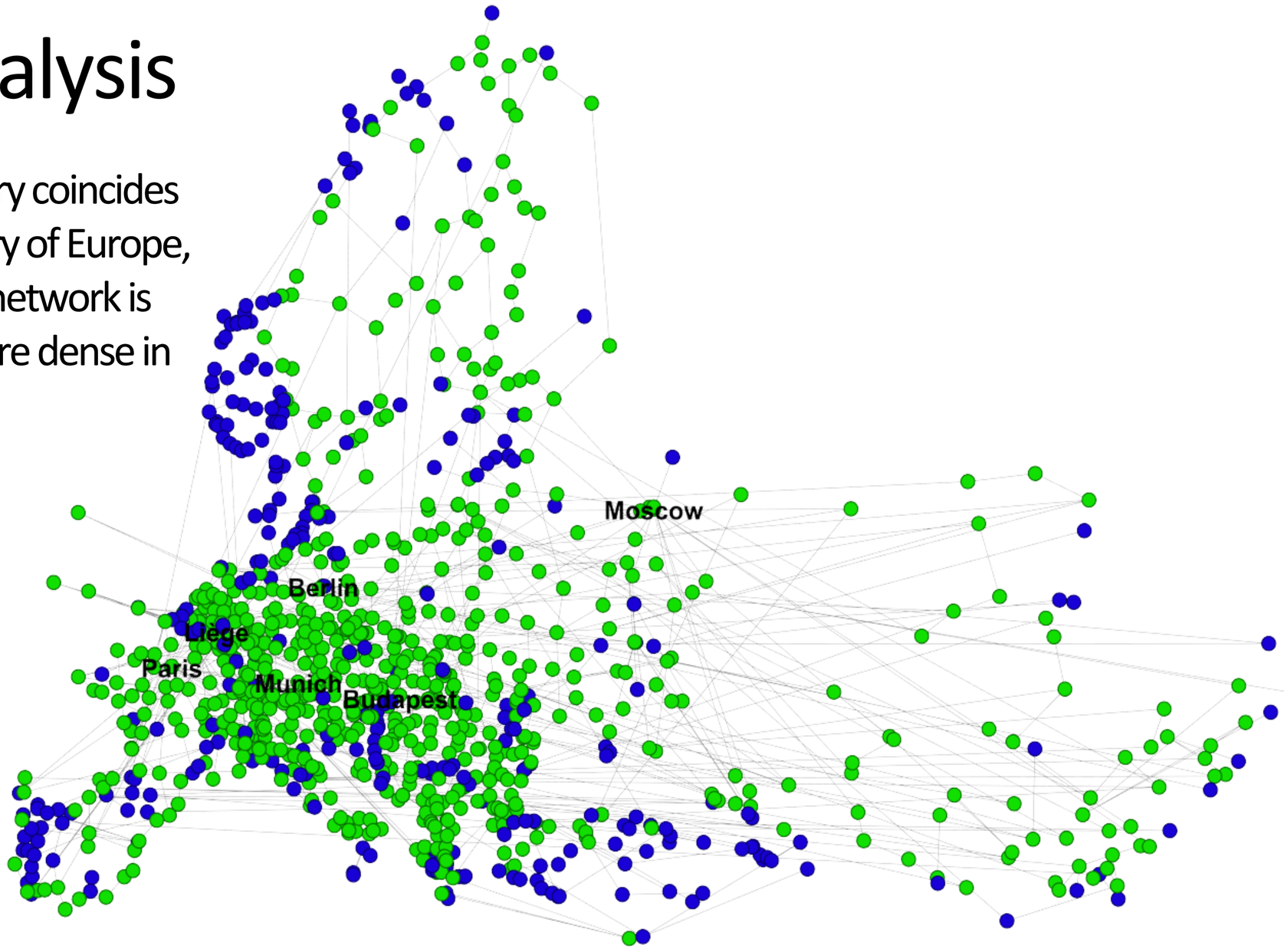


Fig.17: K-Core decomposition computed with networkX and plotted with Gephy.

Core-Periphery analysis

We can also consider the core-periphery profile of the network to highlight an ordering of nodes from the periphery to the core.

We can see that the top 7 degree nodes are in the core.

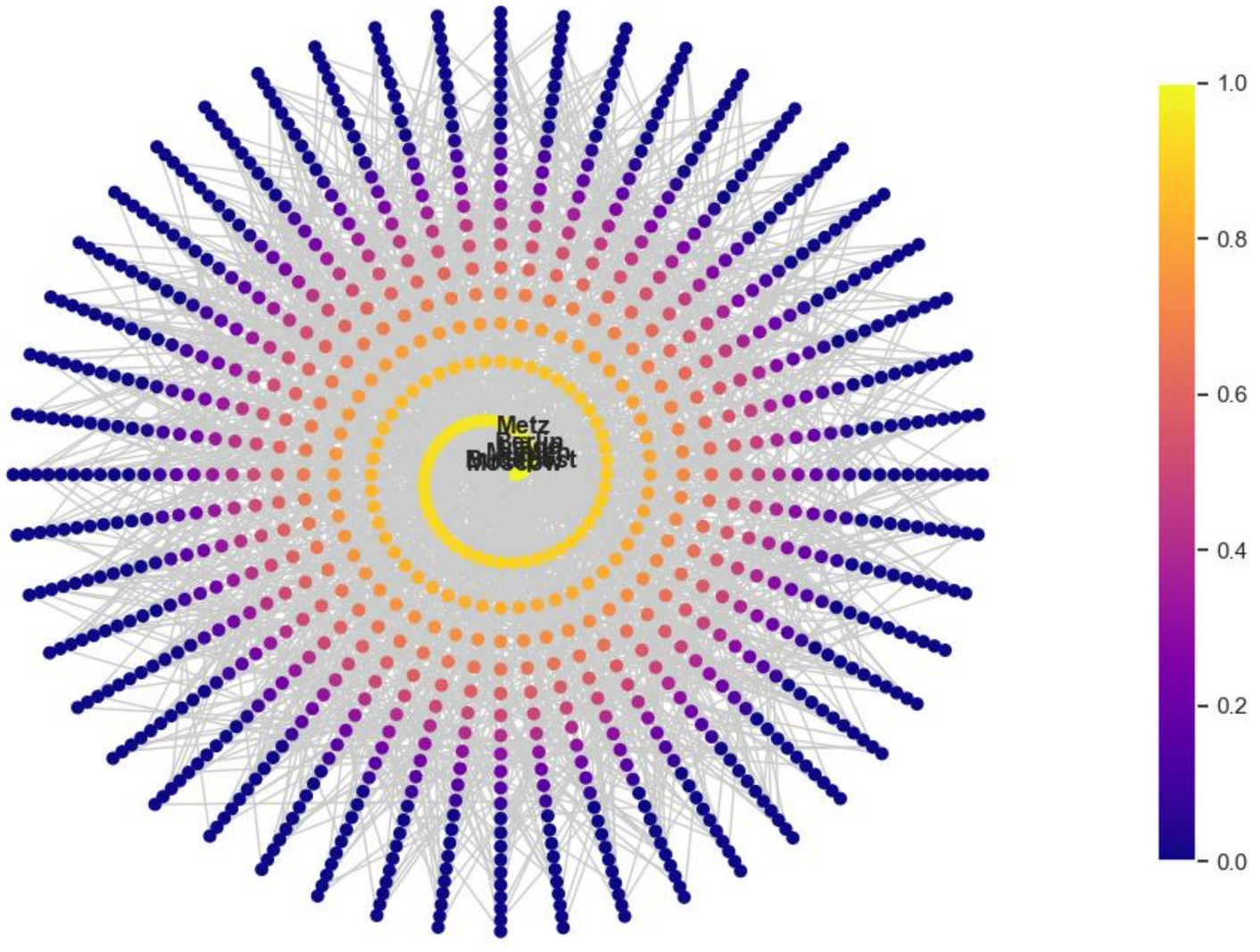
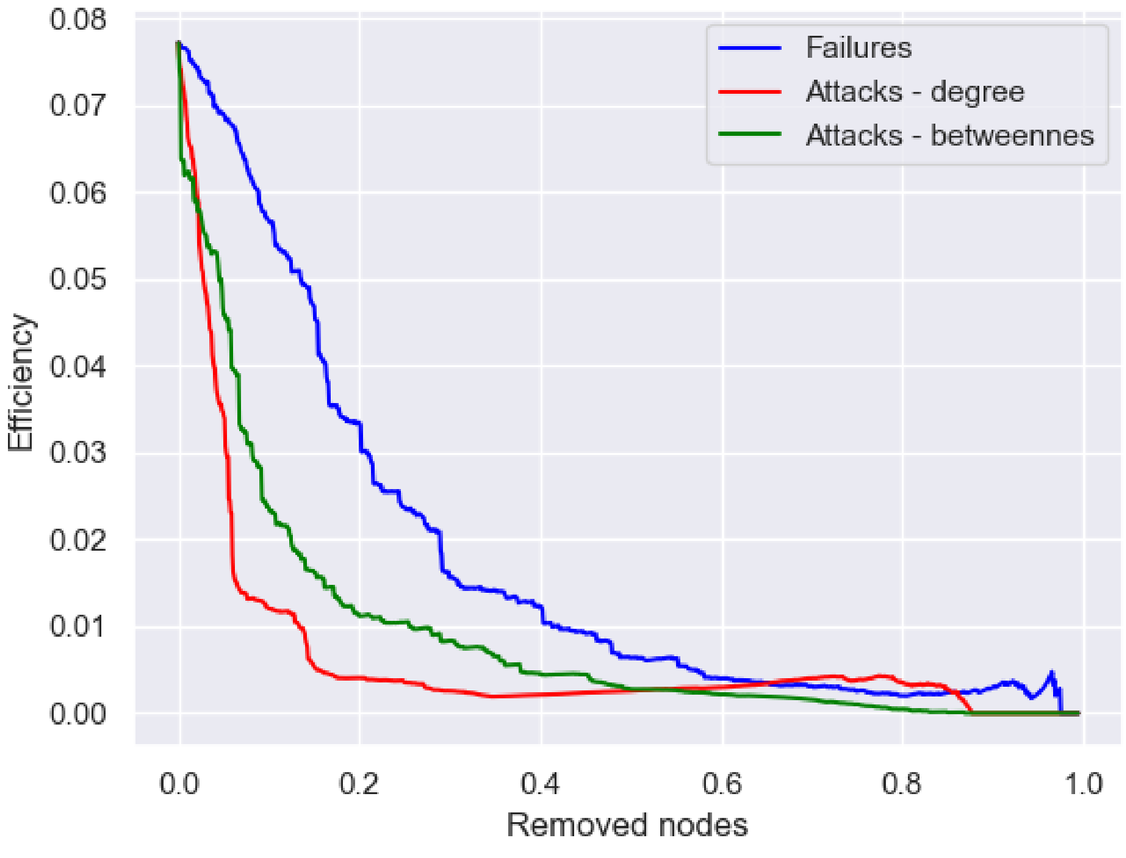


Fig.18: Core-periphery profile ordered nodes, custom layout NetworkX.

Reference:
Profiling core-periphery network structure by random walkers*, F. Della Rossa et al, 2013.*
(nature.com)

Robustness analysis

To analyze the robustness of the network, both attacks and failures were simulated. The robustness evaluation is made by considering the efficiency (E) and the ratio of the largest connected component to the entire network (S/N).



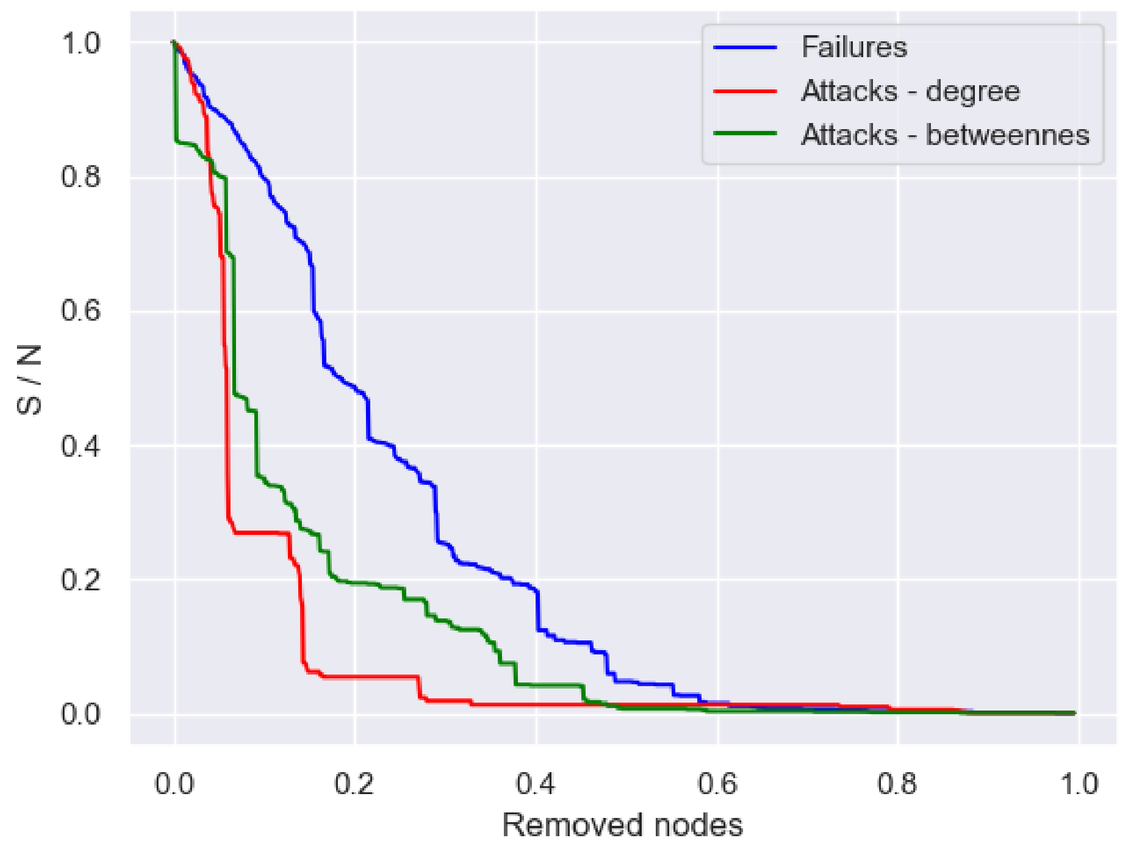


Fig.19: Attacks and failures simulation.

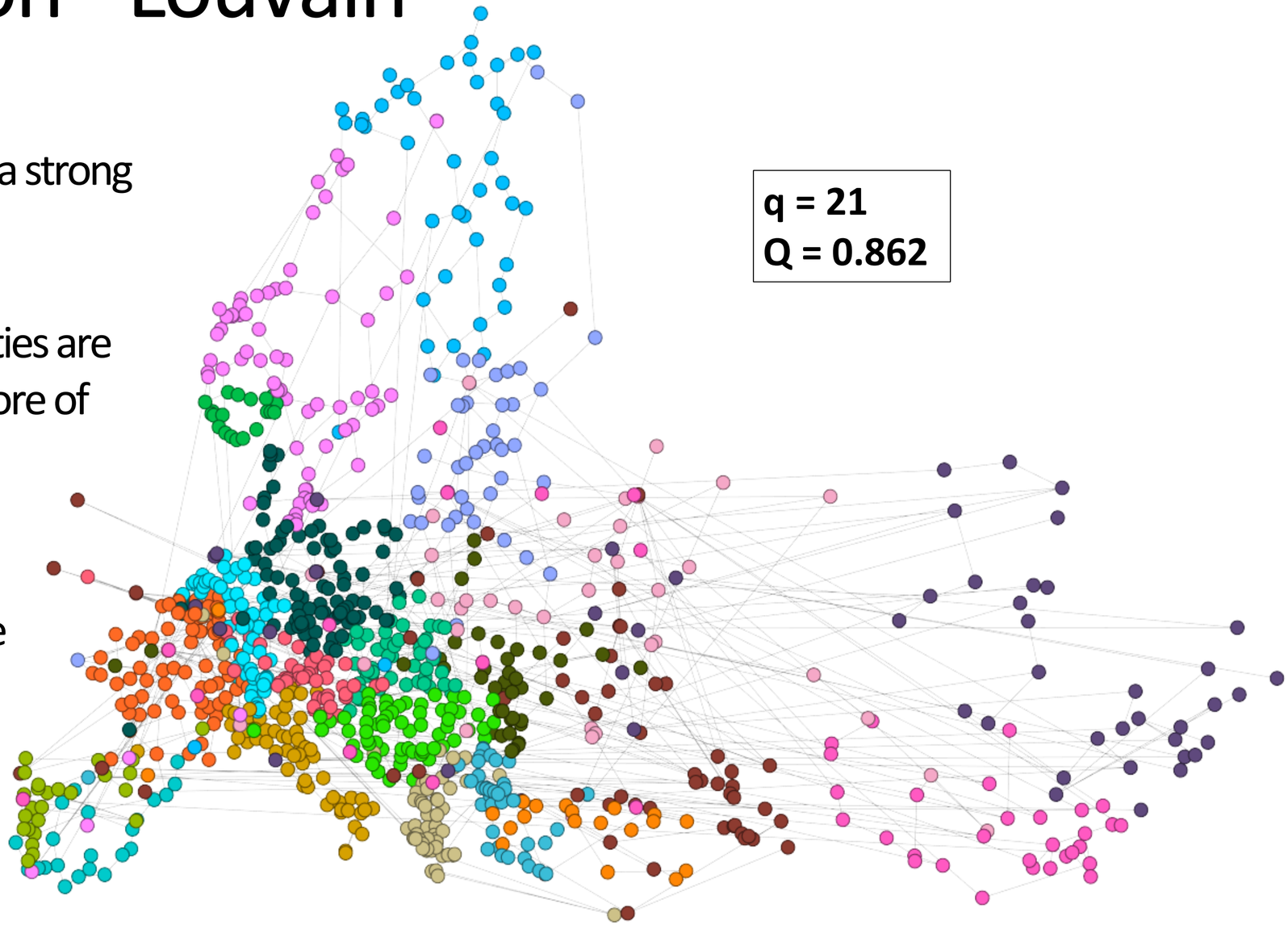
Community detection - Louvain

Analyzing the network from a community perspective revealed that the network has a strong community structure.

By using the Louvain method 21 communities are found, with a corresponding modularity score of 0.862.

We also note that (especially in the central Europe) the communities correspond quite well with countries.

Reference:
Fast unfolding of communities in large networks, Blondel et al. 2008. (arxiv.org).



q = 21
Q = 0.862

Fig.20: Community detection using the Louvain method, plotted eith gephy.

Community detection – Persistence probabilities

An alternative method to reveal the network's community structure is by using the persistence probabilities.

Increasingly granular partitions are generated using (divisive) hierarchical clustering (using Floyd-Warshall to generate the distance matrix), then the partitions are evaluated by means of a persistence probability threshold (α), finally, the more granular partition is taken.

Reference:
Finding and testing network communities by lumped Markov chains, C. Piccardi, 2011. (arxiv.org).

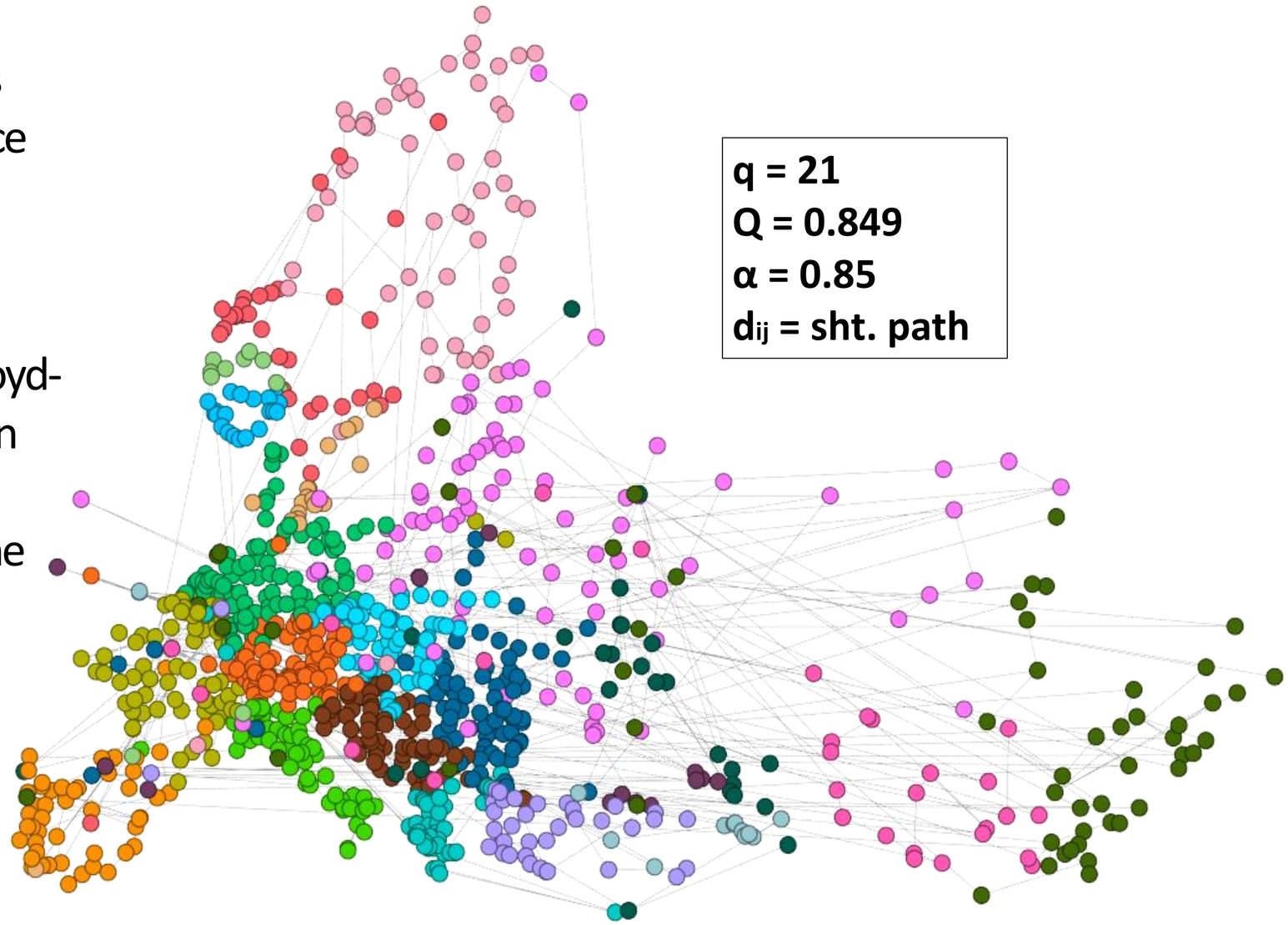


Fig.21: Community detection using the α-community & α-partition method.

Community detection – Persistence probabilities

From the plot we can notice that the "last" partition, which is composed by communities above the dashed line, is the one with 21 communities.

Other tests can be done by varying alpha and by changing the distance used for the clustering.

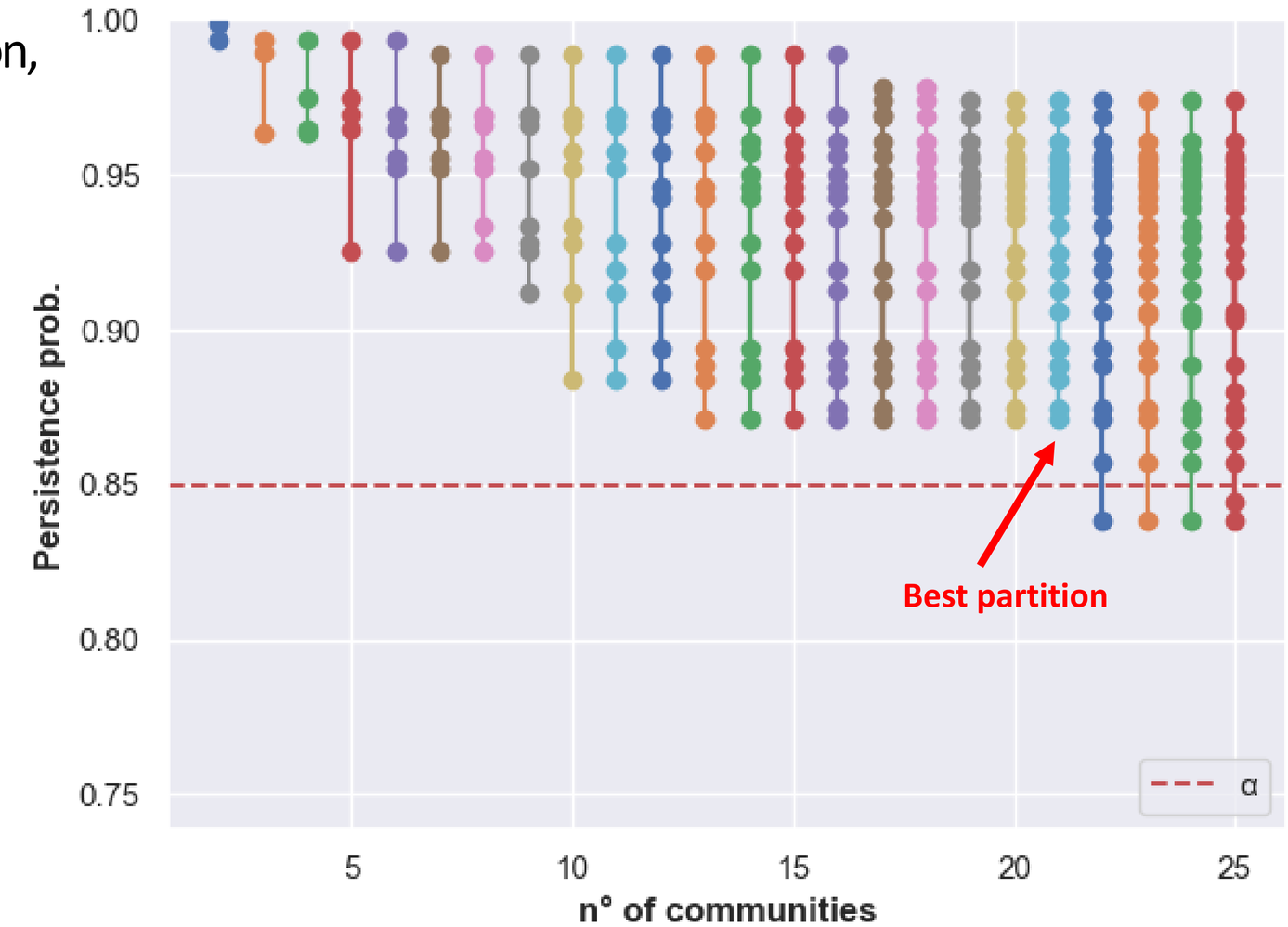
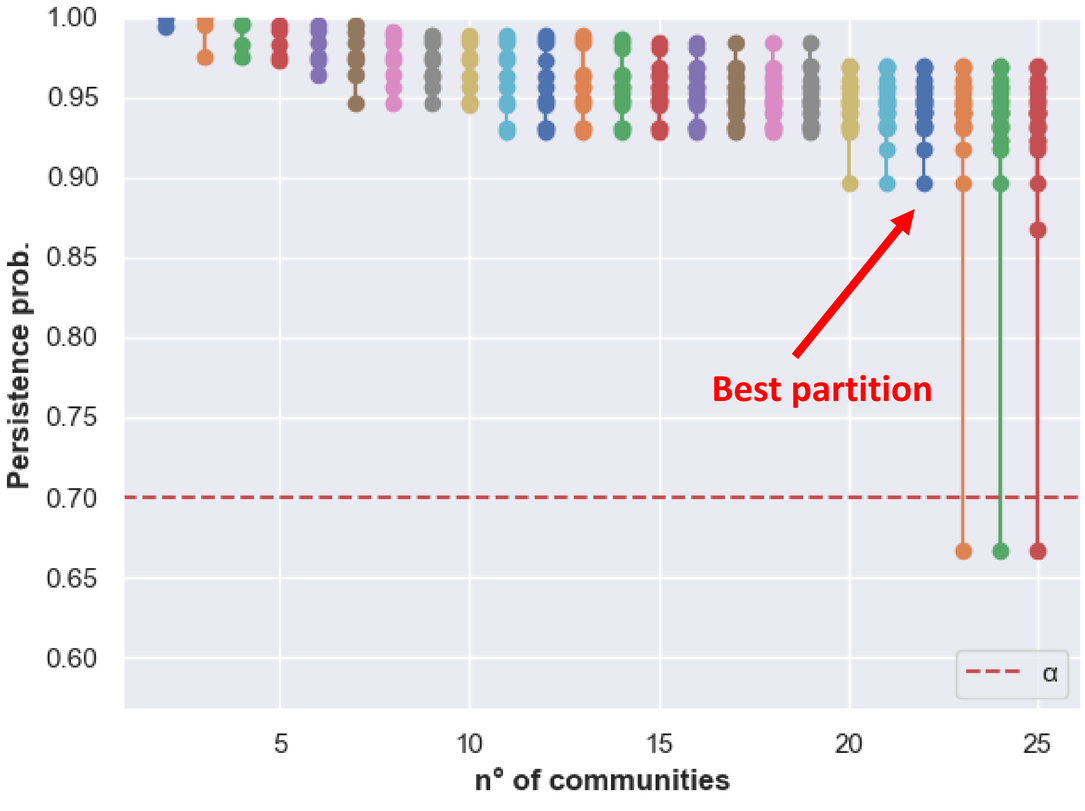
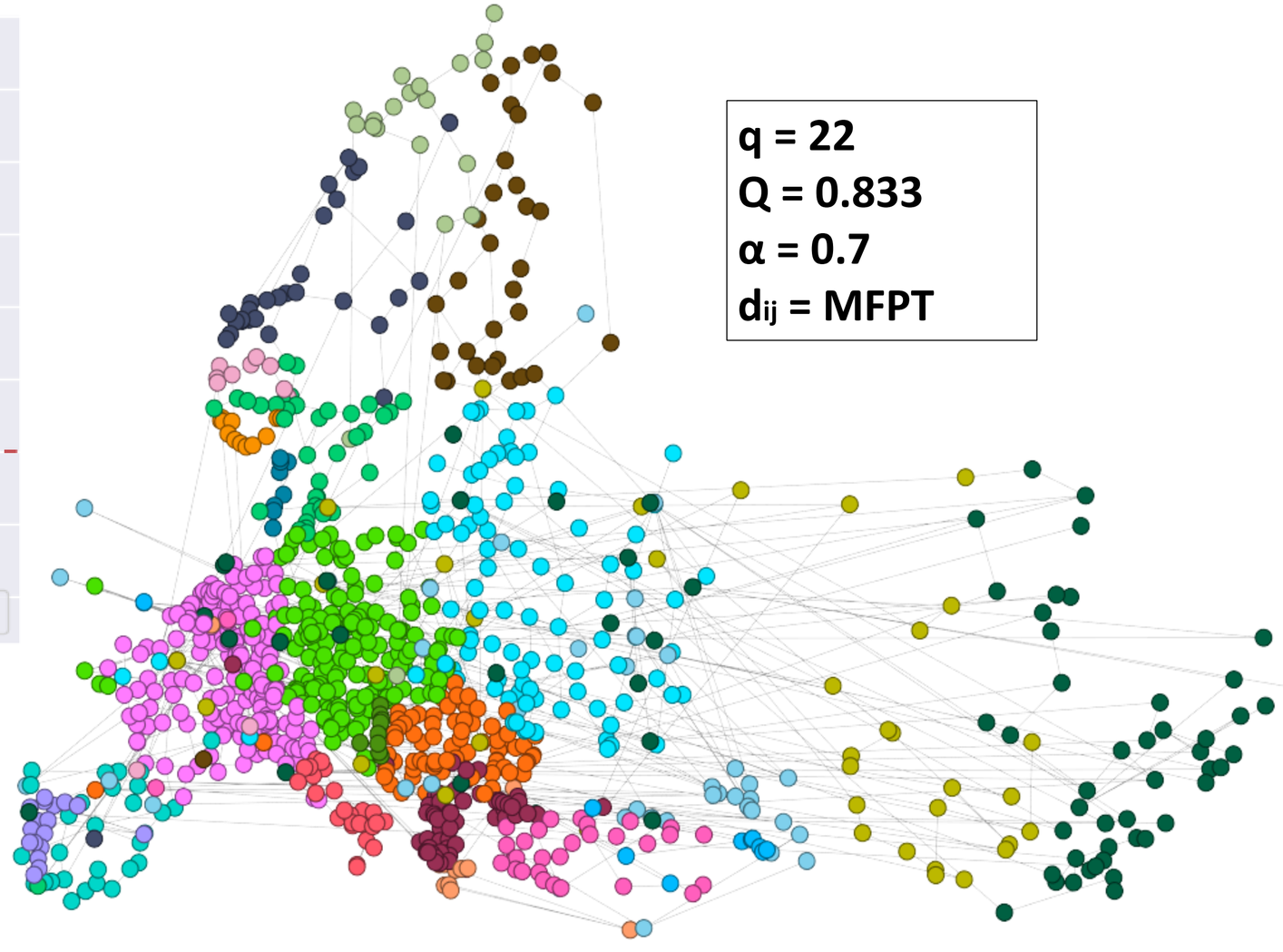


Fig.22: α-partitions method plot, NetworkX.

Reference:
Finding and testing network communities by lumped Markov chains, C. Piccardi, 2011. (arxiv.org).

Community detection – Persistence probabilities





q = 22
Q = 0.833
α = 0.7
d_{ij} = MFPT

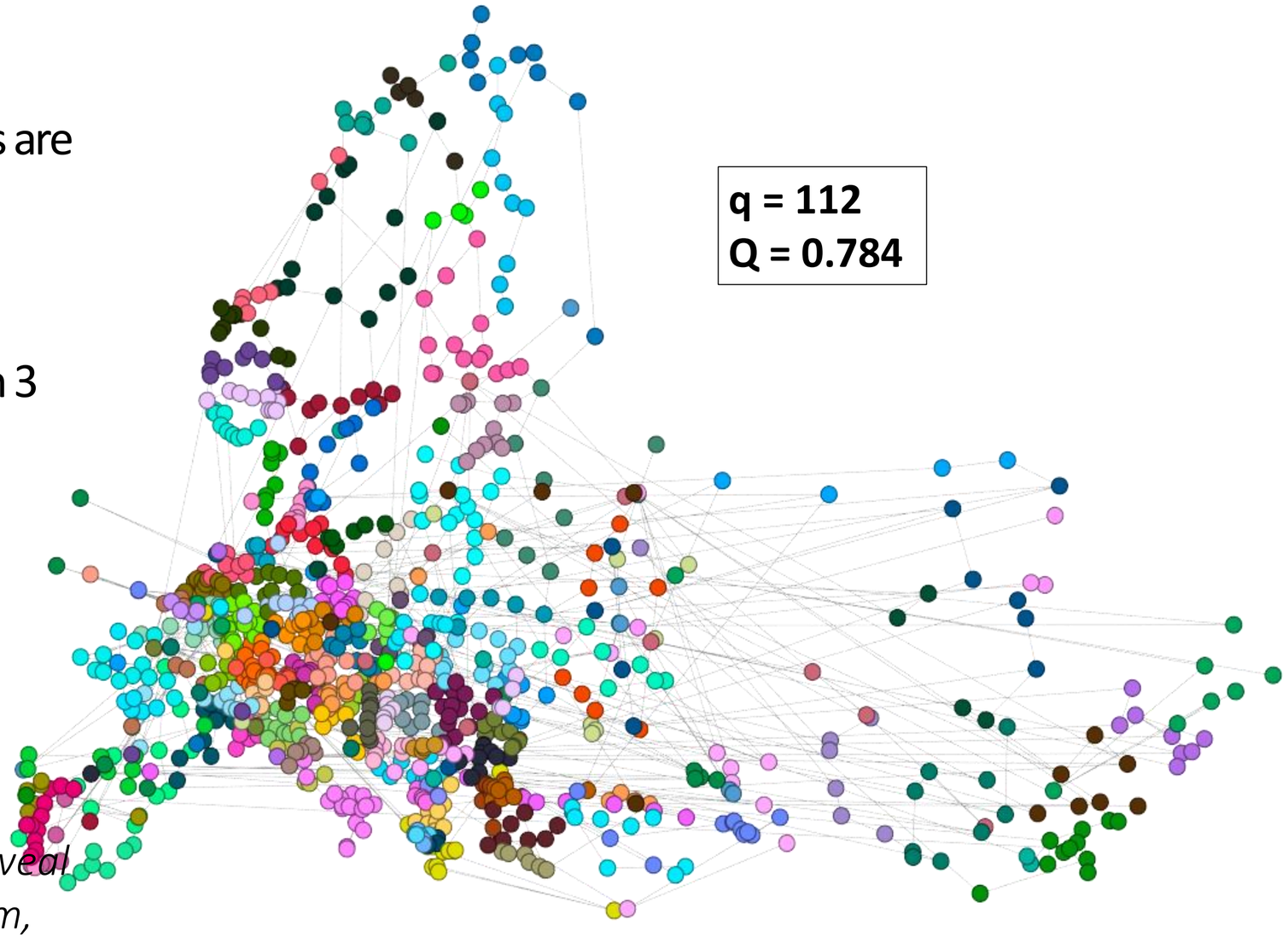
Fig.23: Community detection using the α-community & α-partition method.

Reference:
Finding and testing network communities by lumped Markov chains, C. Piccardi, 2011. (arxiv.org).

Community detection – Infomap

By applying Infomap we obtain a finer division between communities, in fact 112 communities are highlighted by the algorithm.

In this case we can distinguish some relevant regions, for example Italy seems to be divided in 3 main parts (north, center, south).



q = 112
Q = 0.784

Fig.24: Infomap partition plot, plotted with Gephy.

Reference:
Maps of random walks on complex networks reveal community structure, M. Rosvall, C. T. Bergstrom, 2008. (arxiv.org)

Community detection – LPA

1. Given an undirected network where N is the set of nodes, a unique label is assigned to each node ($\forall n \in N, c_n = l_n$).
2. Then, at each iteration, each node adopts the label shared by most of its neighbors (considering also edge weights).

$$c_n = argmax_{\,l} \sum_{m \in B^l(n)} w_{nm}$$

where $B^l(n)$ is the set of neighbors of $n \in N$ that share the label l and w_{nm} is the weight of the edge incident to nodes $n, m \in N$.

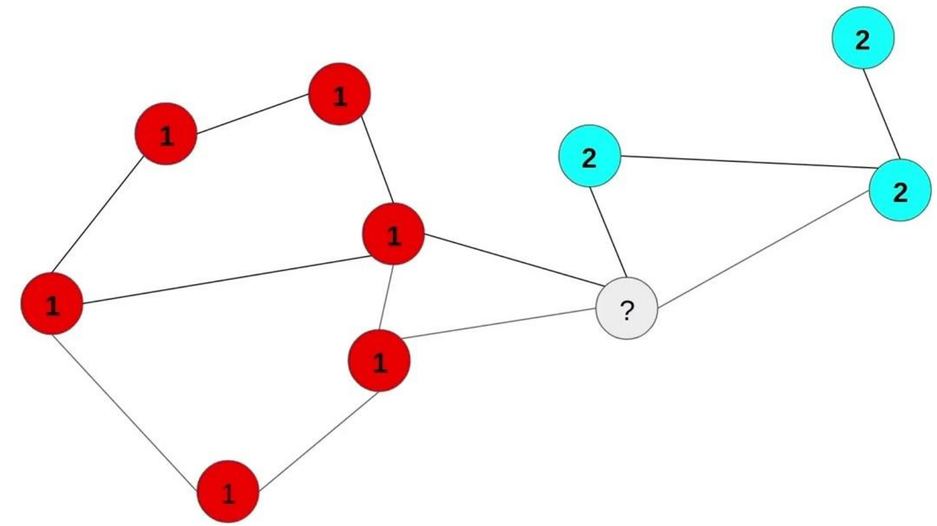


Fig.25: LPA for community detection.

The label update can be improved by introducing a preference term p_m:

$$c_n = argmax_{\,l} \sum_{m \in B^l(n)} p_m w_{nm}$$

Reference:
Near linear time algorithm to detect community structures in large-scale networks, U. N. Raghavan et al., 2007. (arxiv.org) [LPA]

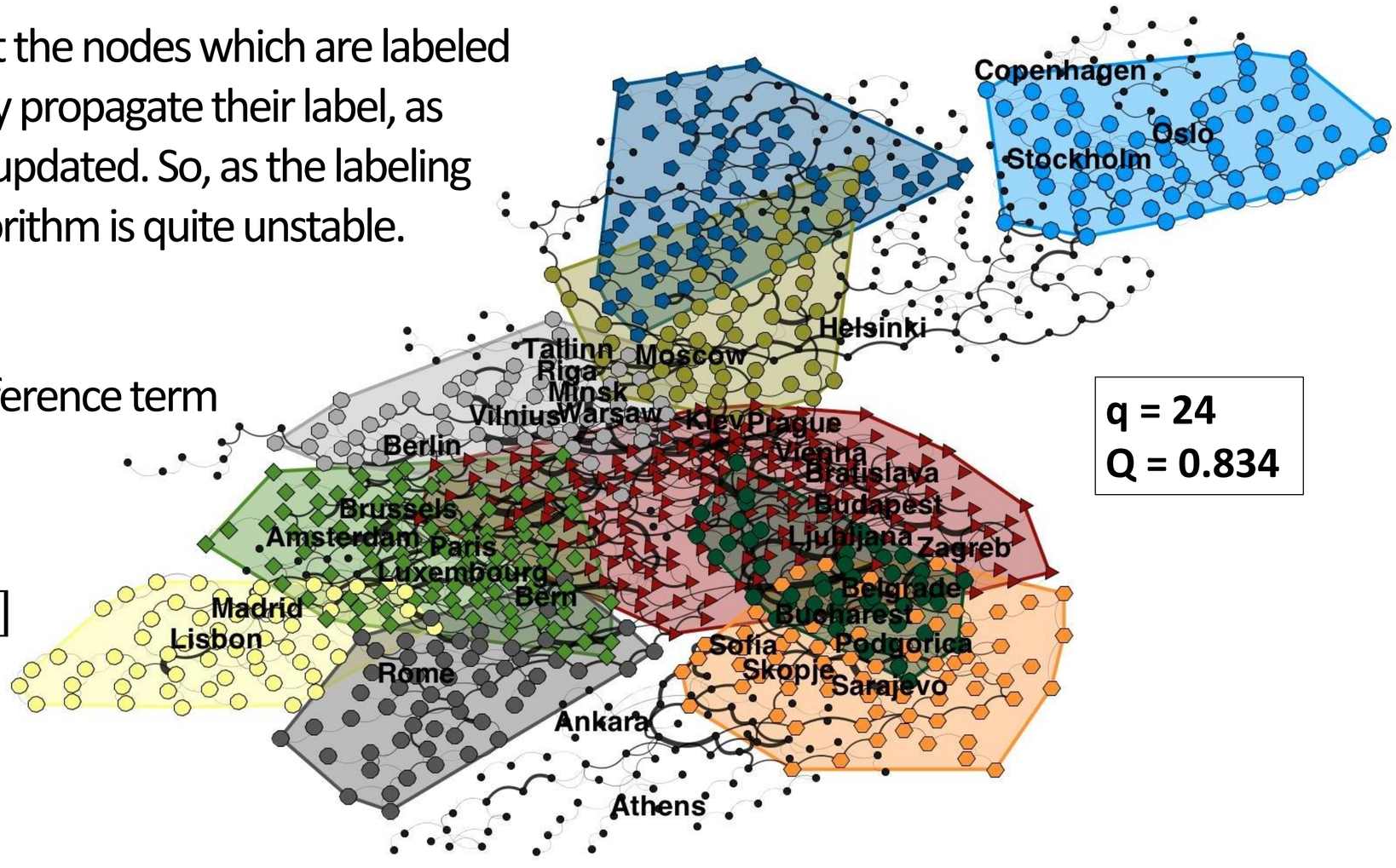
Reference:
Towards real-time community detection in large networks, I. X. Y Leung et al., 2009. (arxiv.org) [LPA with preferences]

Community detection – BPA

A problem related with LPA is the fact that the nodes which are labeled at the end of an iteration cannot efficiently propagate their label, as most of their neighbors are already been updated. So, as the labeling order at each iteration is random, the algorithm is quite unstable.

BPA fix this issue by adding an ad-hoc preference term computed as:

$$p_n = \frac{index\ of\ node\ n}{|N|} \quad , p_n \in (0,1]$$



q = 24
Q = 0.834

Fig.26: 10 top communities computed with BPA.

Reference:
Robust network community detection using balanced propagation, L. Šubelj et. Al, 2011 (arxiv.org) [BPA]

POLITECNICO
MILANO 1863

Thanks for your attention

European road network analysis

Andrea Bertogalli
218342

Accademic year 2022/2023