UNIVERSIDADE FEDERAL DO PIAUÍ
UFPI - CCN - DC
CIÊNCIA DA COMPUTAÇÃO
TÓPICOS ESPECIAL EM COMPUTAÇÃO:
COMPUTAÇÃO SOCIAL E URBANA
PROF. DR. GLAUBER DIAS GONÇALVES

Atividade 04

Agrupamento em grafos

Abril, 2025

Equipe

Enzo Borges Rocha Lima

Membro

Luis Felipe do Nascimento Moura

Membro

Objetivo

- Aplicar os algoritmos de detecção de comunidades:
  - Edge Betweenness
  - Louvain
  - Leiden
- Avaliar os resultados com modularidade.
- Comparar e discutir o melhor agrupamento.

Algoritmos

Edge Betweenness

- Remove arestas com maior centralidade de intermediação.
- Comunidades são formadas quando o grafo se desconecta.

Louvain (multilevel)

- Baseado em maximização de modularidade
- Rápido e eficaz para grandes redes
- Não garante comunidades conectadas

Leiden

- Melhoria sobre Louvain
- Garante comunidades conectadas

Algoritmos

Artigo de referência

- Uso do modelo LFR
  - É uma abordagem para gerar redes complexas que simula a estrutura de comunidades em grafos.
  - Gera redes não direcionadas e não ponderadas com comunidades não sobrepostas.
  - Amplamente utilizado como um benchmark para testar e comparar algoritmos de detecção de comunidades.
- A precisão é medida pela similaridade entre a estrutura modular gerada pelo modelo LFR e a partição identificada pelos algoritmos.

- Padrão de Mistura
  - O padrão de mistura refere-se à forma como os nós em uma rede estão conectados entre si, especialmente em relação à formação de comunidades.
- Parâmetro de Mistura (μ):
  - O parâmetro de mistura (μ) quantifica a proporção de conexões dentro de comunidades em comparação com conexões entre diferentes comunidades.
  - Valores baixos de μ indicam que a maioria das conexões está dentro das comunidades, enquanto valores altos indicam uma mistura maior entre as comunidades.

- A maioria dos algoritmos apresenta boa precisão quando o parâmetro de mistura (μ) é baixo ($\mu \leq 0.2$), mas a precisão diminui à medida que μ aumenta.
- O tempo de computação varia entre os algoritmos, com o **Multilevel** e o Label Propagation sendo os mais rápidos, enquanto Spinglass e Edge Betweenness são os mais lentos.

- O estudo avalia **oito algoritmos** de detecção de comunidades usando o benchmark LFR.
  - Infomap; Label Propagation; **Multilevel**; Walktrap; Spinglass; Edge Betweenness; Leading Eigenvector; Fastgreedy

Algoritmos

- **NetworkX**
  - Funciona super bem com pandas, matplotlib, numpy, etc.
  - Sintaxe intuitiva, mais próxima de um dicionário de listas.

  - Muito mais lento que iGraph em operações como centralidade, shortest path, clustering.
  - Estrutura interna usa dicionários aninhados, que consomem muita RAM.

- **iGraph**
  - Muito mais rápido e leve para grafos com milhares/milhões de nós.
  - Baixo consumo de memória

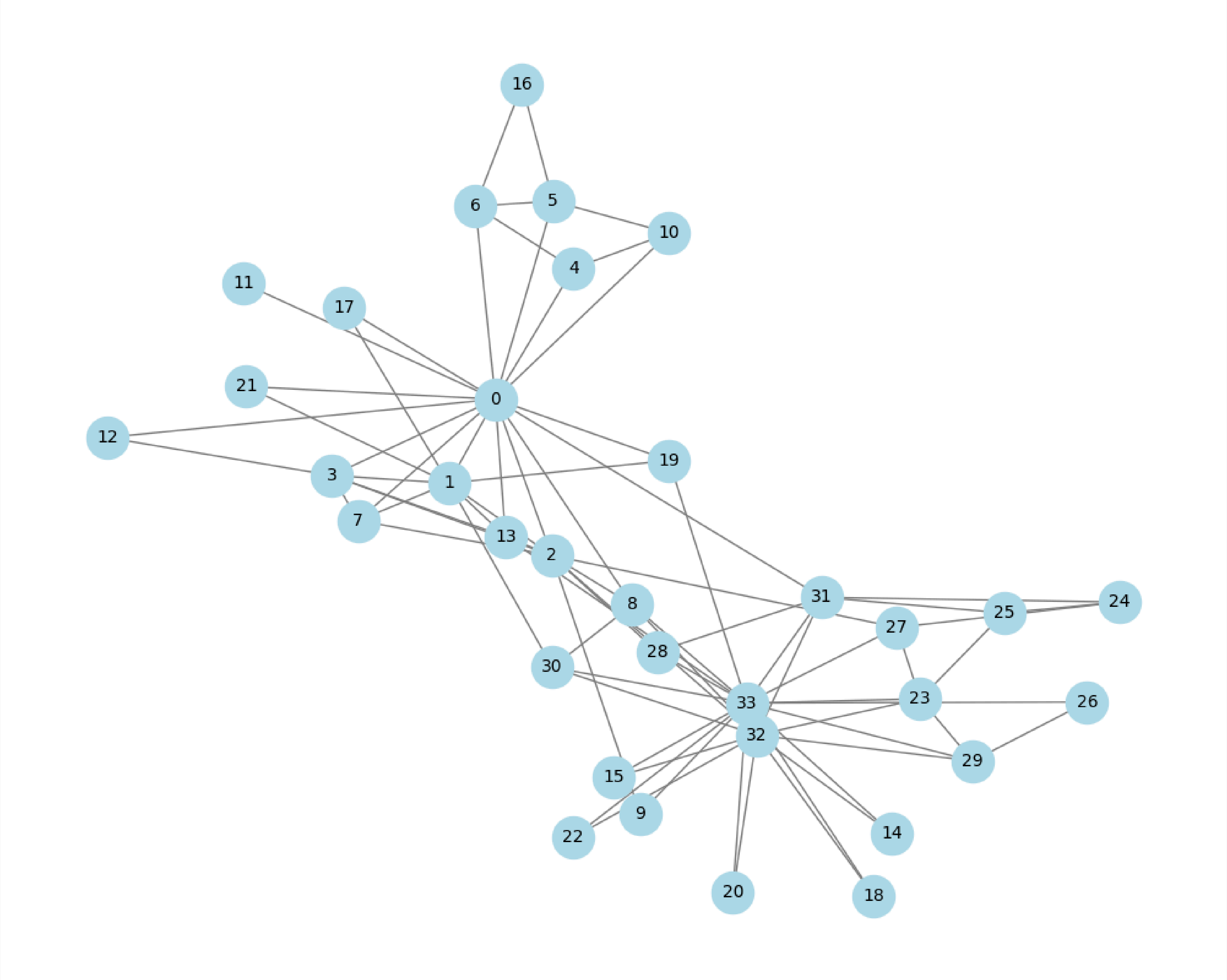
  - Integração com pandas/numpy fraca
  - Precisa adaptar os dados, pois não usa dicionários nativos.

Clube do karatê

Original

Nª de vértices: 33

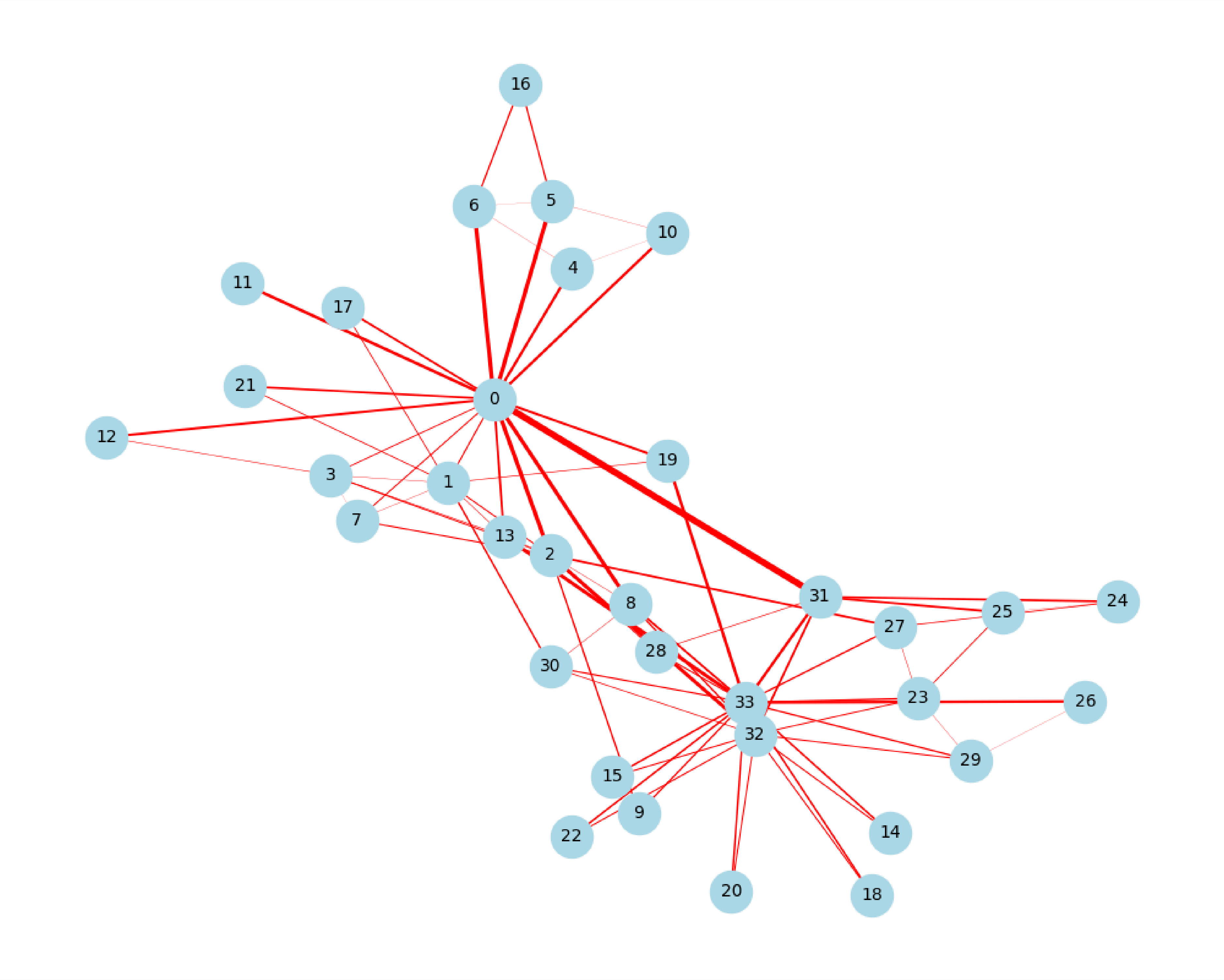
Nª de arestas: 78



Clube do karatê

Edge Betweenness

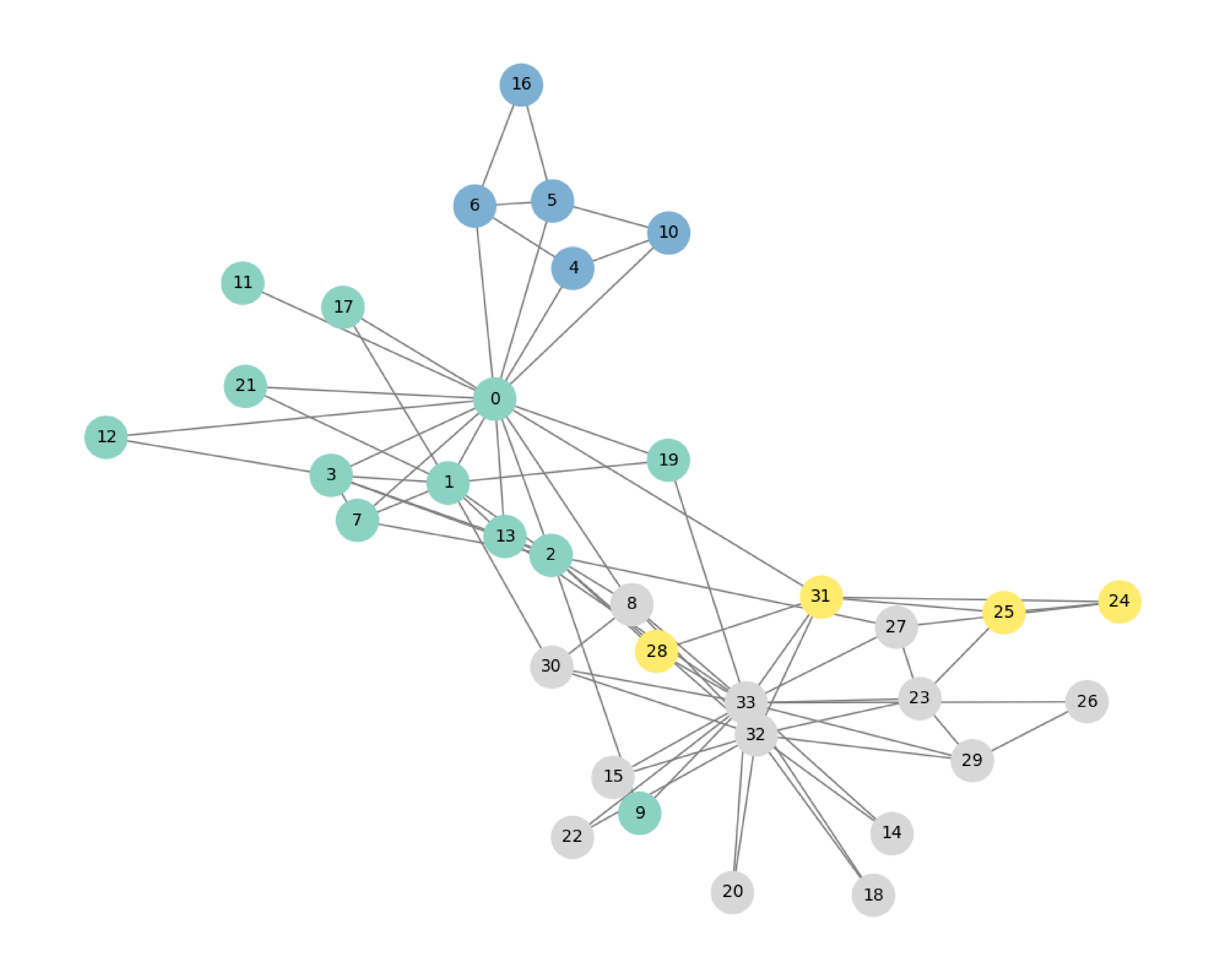
- Arestas centrais:
  - (0, 31): 0,1273
  - (0, 6) : 0,0781
  - (0, 5) : 0,0781
  - (0, 2) : 0,0778
  - (0, 8) : 0,0742



Clube do karatê

Louvain

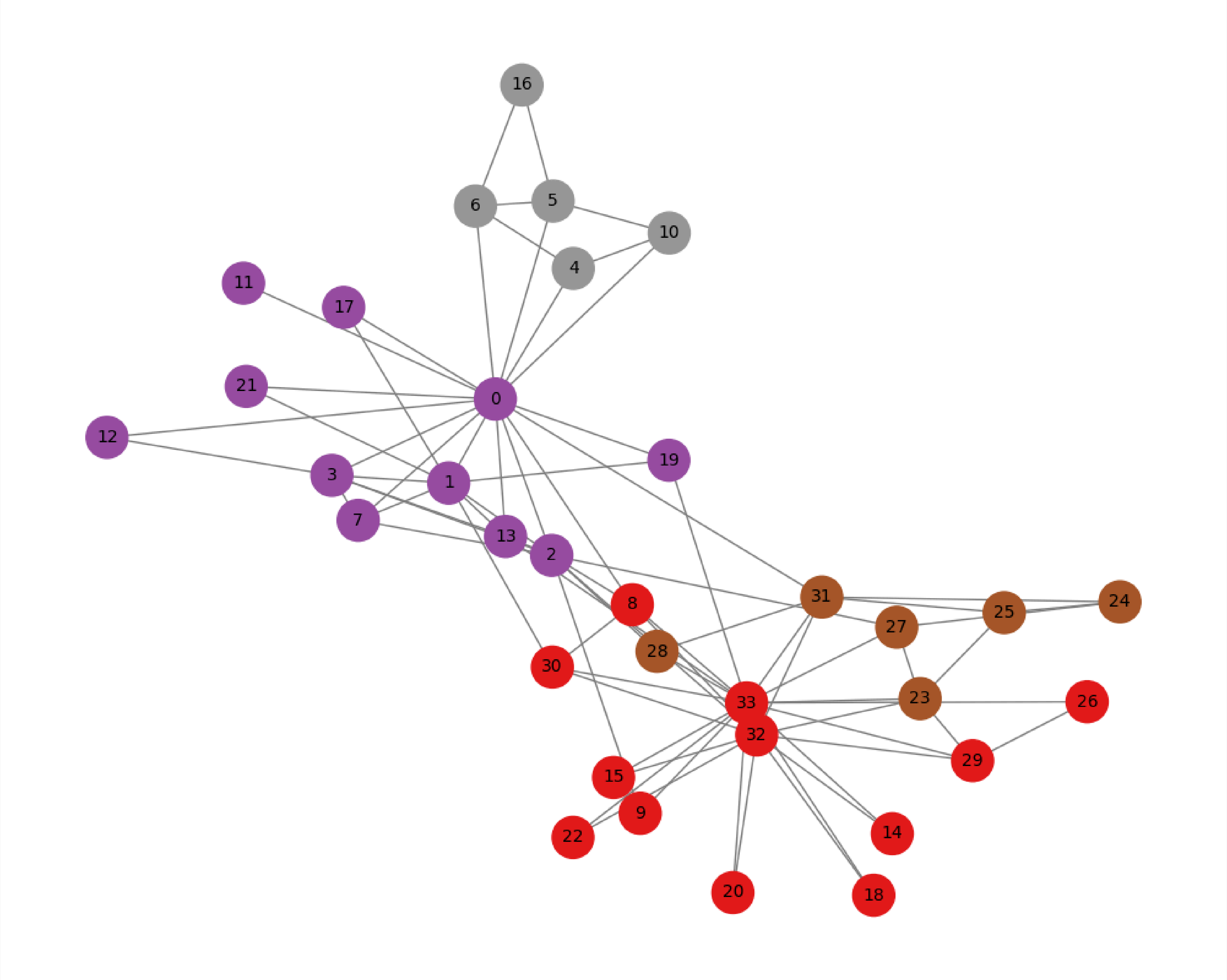
- Número de comunidades: 4



Clube do karatê

Leiden

- Número de comunidades: 4

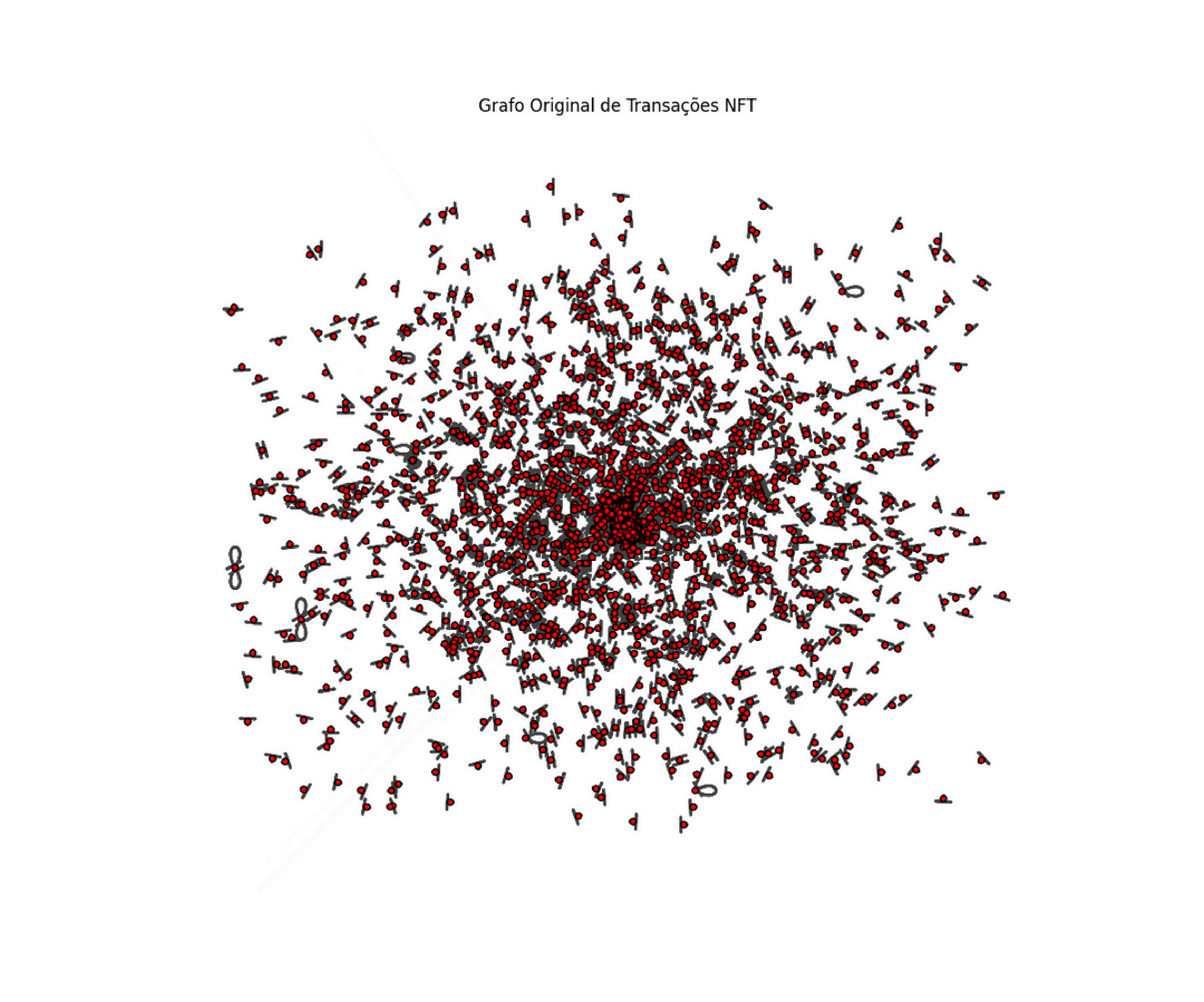


Transações de NFT

Original

Nª de vértices: 83.889

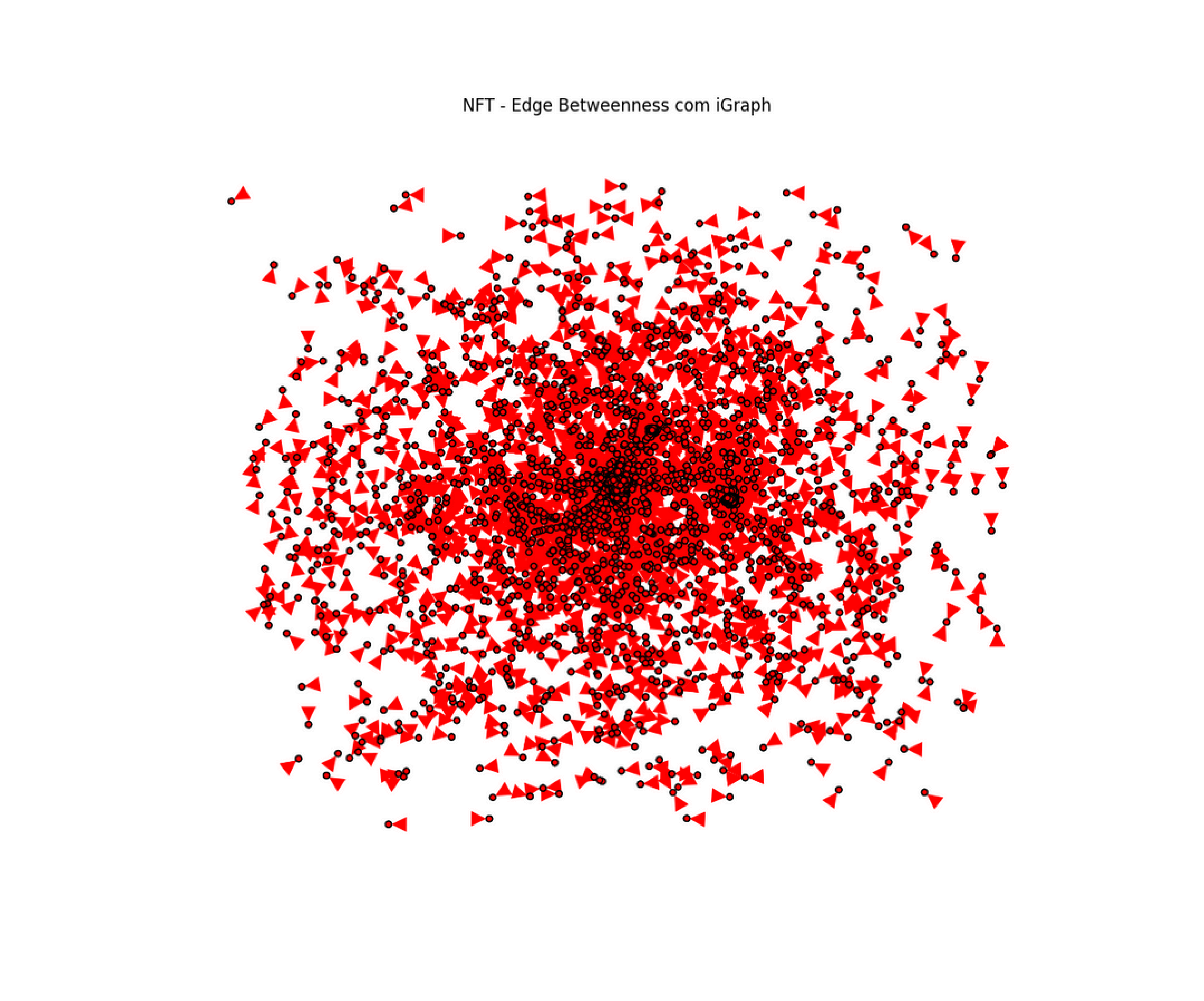
Nª de arestas: 1.245.954



Transações de NFT

Edge Betweenness

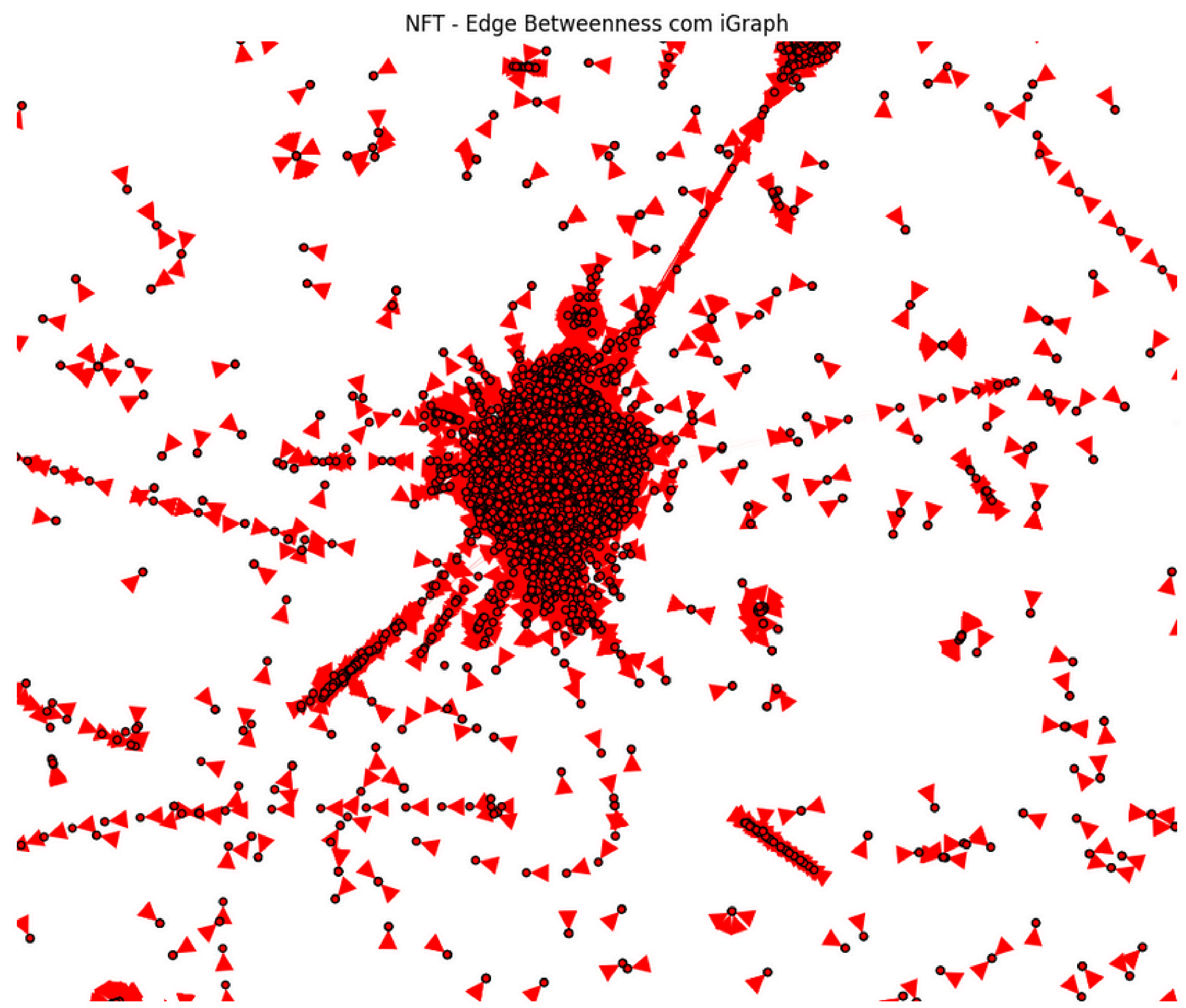
- Arestas centrais:
  - (60077, 27902) : 48.457.950,5692
  - (57151, 71067) : 29.201.312,4618
  - (45178, 60077) : 28.331.305,3150
  - (46368, 4124) : 13.194.288,3803
  - (46368, 4124) : 13.194.288,3803



Transações de NFT

Edge Betweenness (zoom)

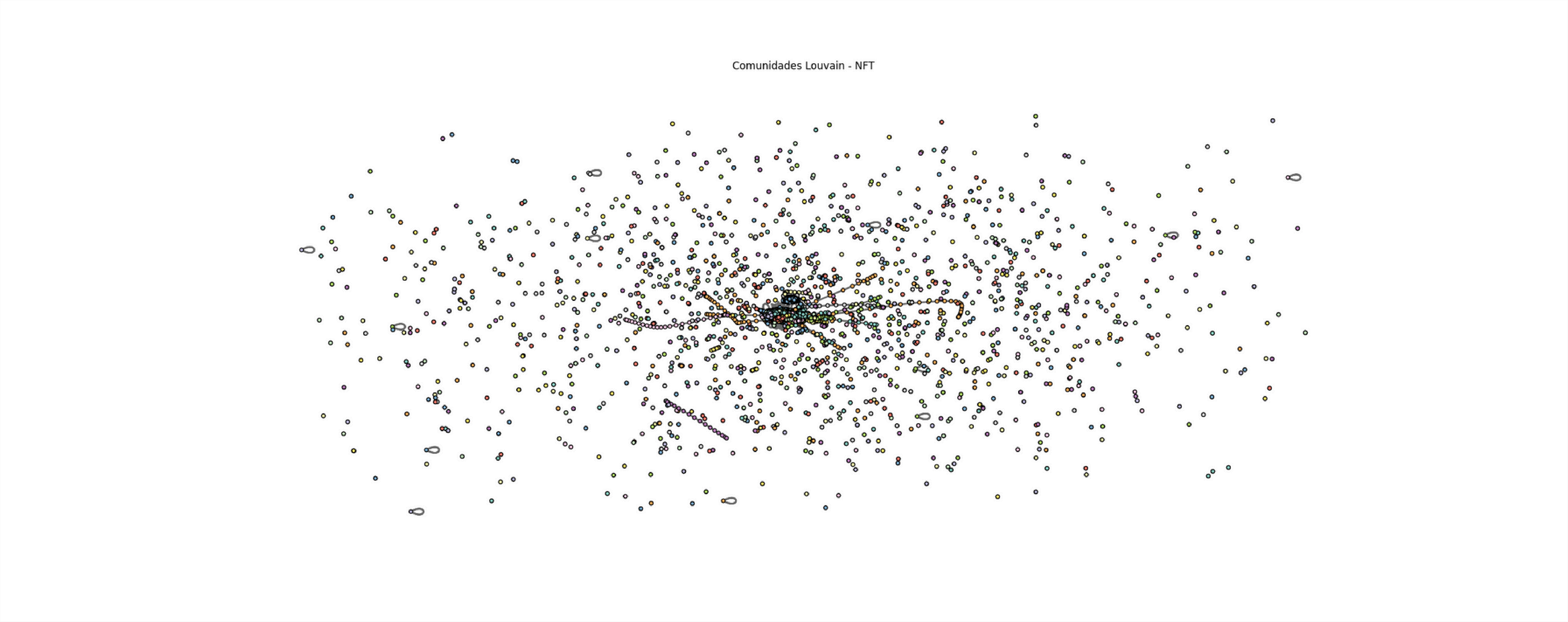
- Arestas centrais:
  - (0, 31): 0,1273
  - (0, 6) : 0,0781
  - (0, 5) : 0,0781
  - (0, 2) : 0,0778
  - (0, 8) : 0,0742



Transações de NFT

Louvain

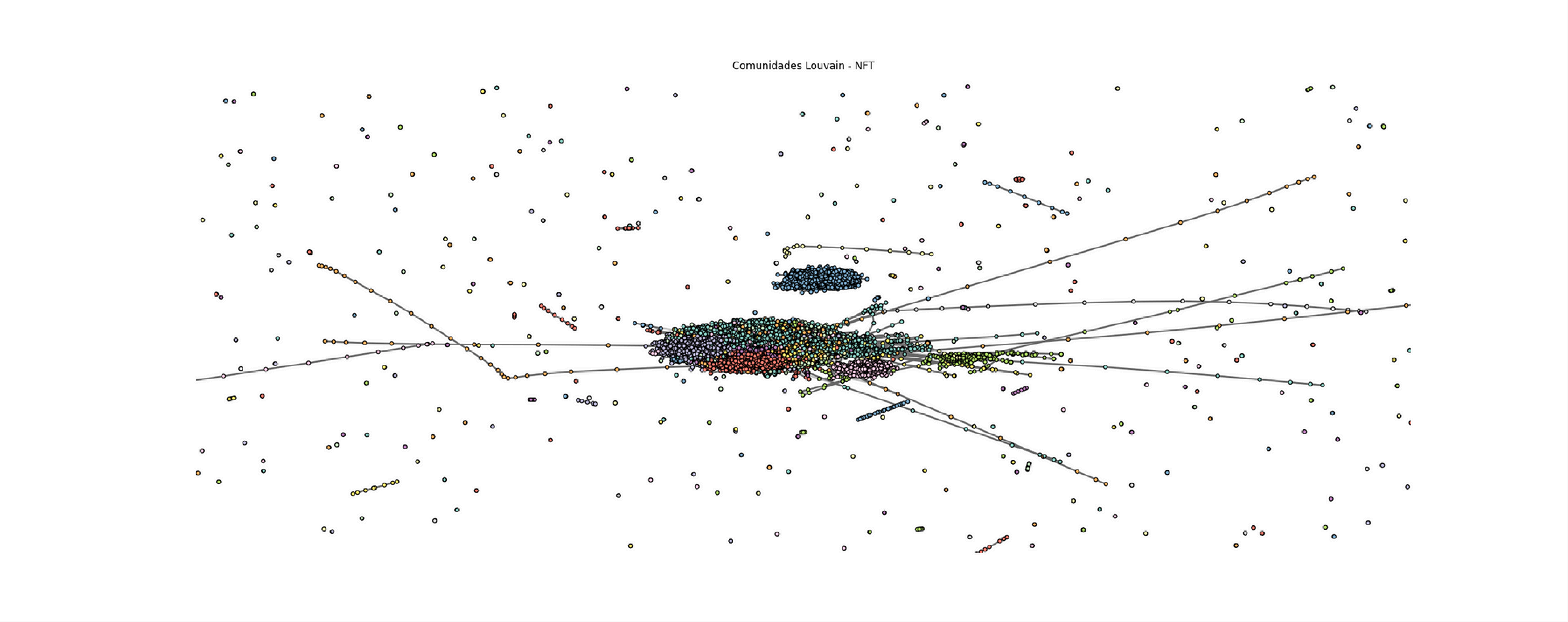
- **Número de comunidades: 1601**



Transações de NFT

Louvain (zoom)

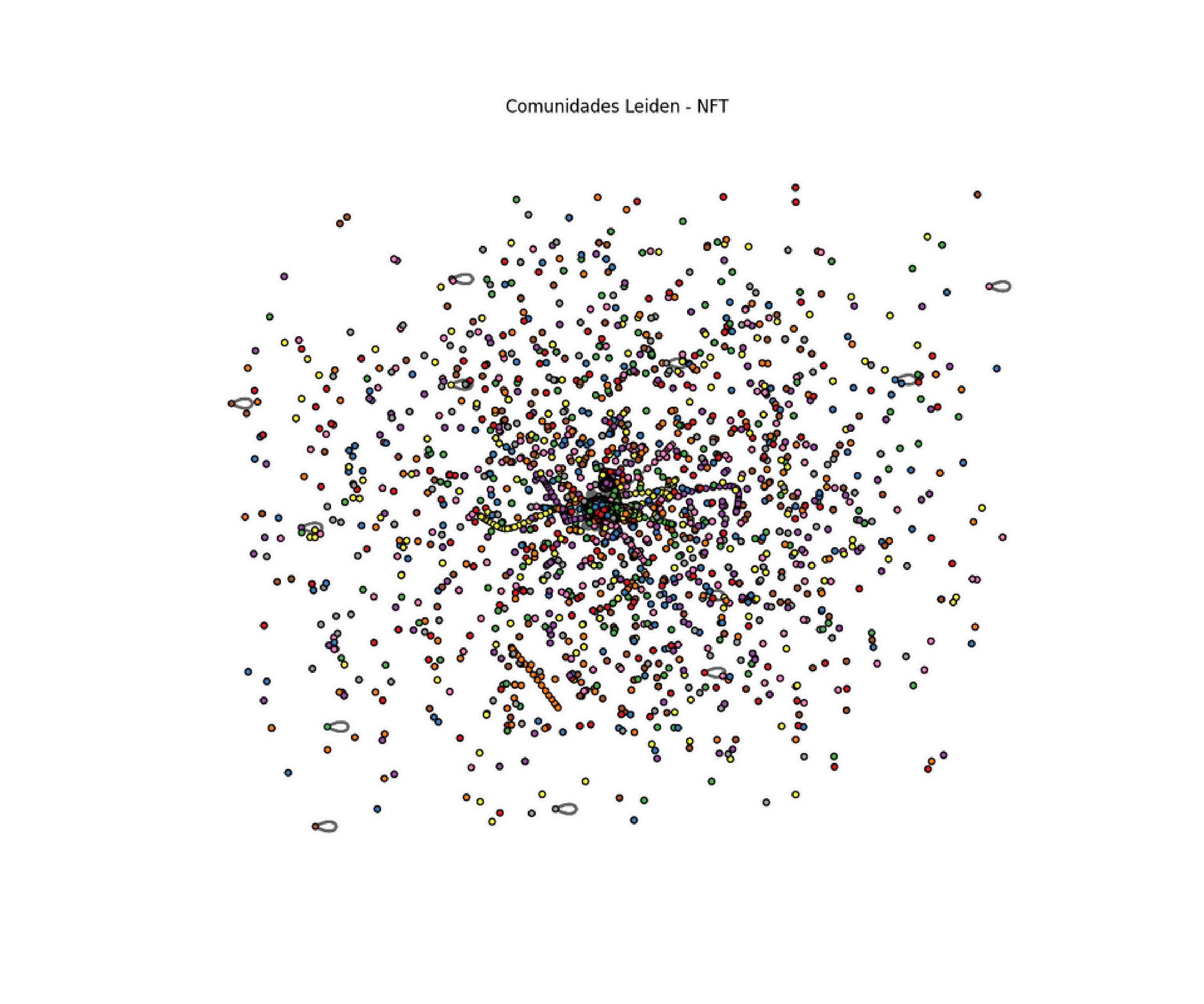
- Número de comunidades: 1601



Transações de NFT

Leiden

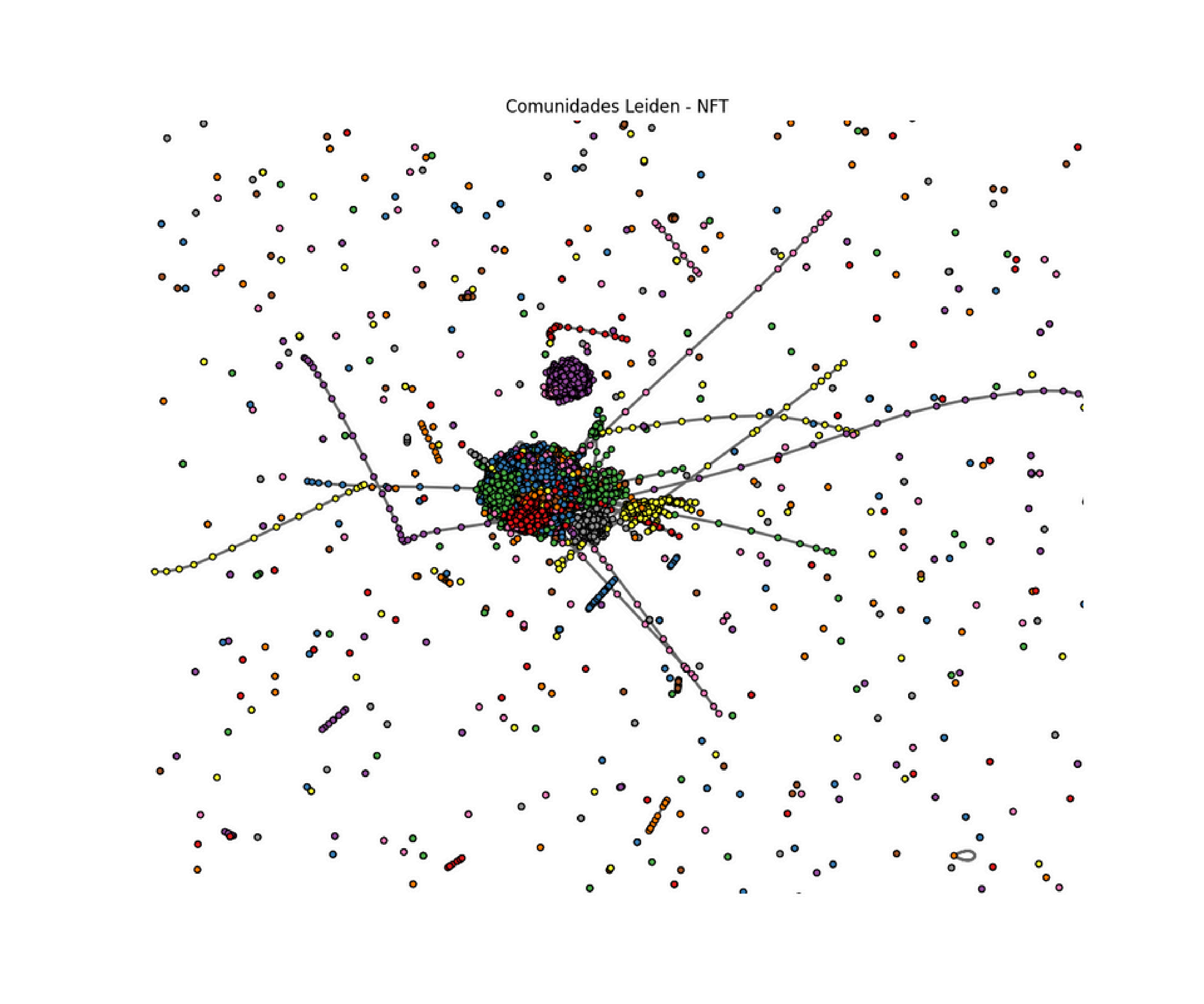
- Número de comunidades: 1602



Transações de NFT

Leiden (zoom)

- Número de comunidades: 1602



Conclusão

- Os três algoritmos oferecem boas abordagens para detectar comunidades.

- Leiden e Louvain são melhores para grandes redes e modularidade.

- Edge Betweenness é mais ‘interpretável’, mas computacionalmente custoso.

Referências

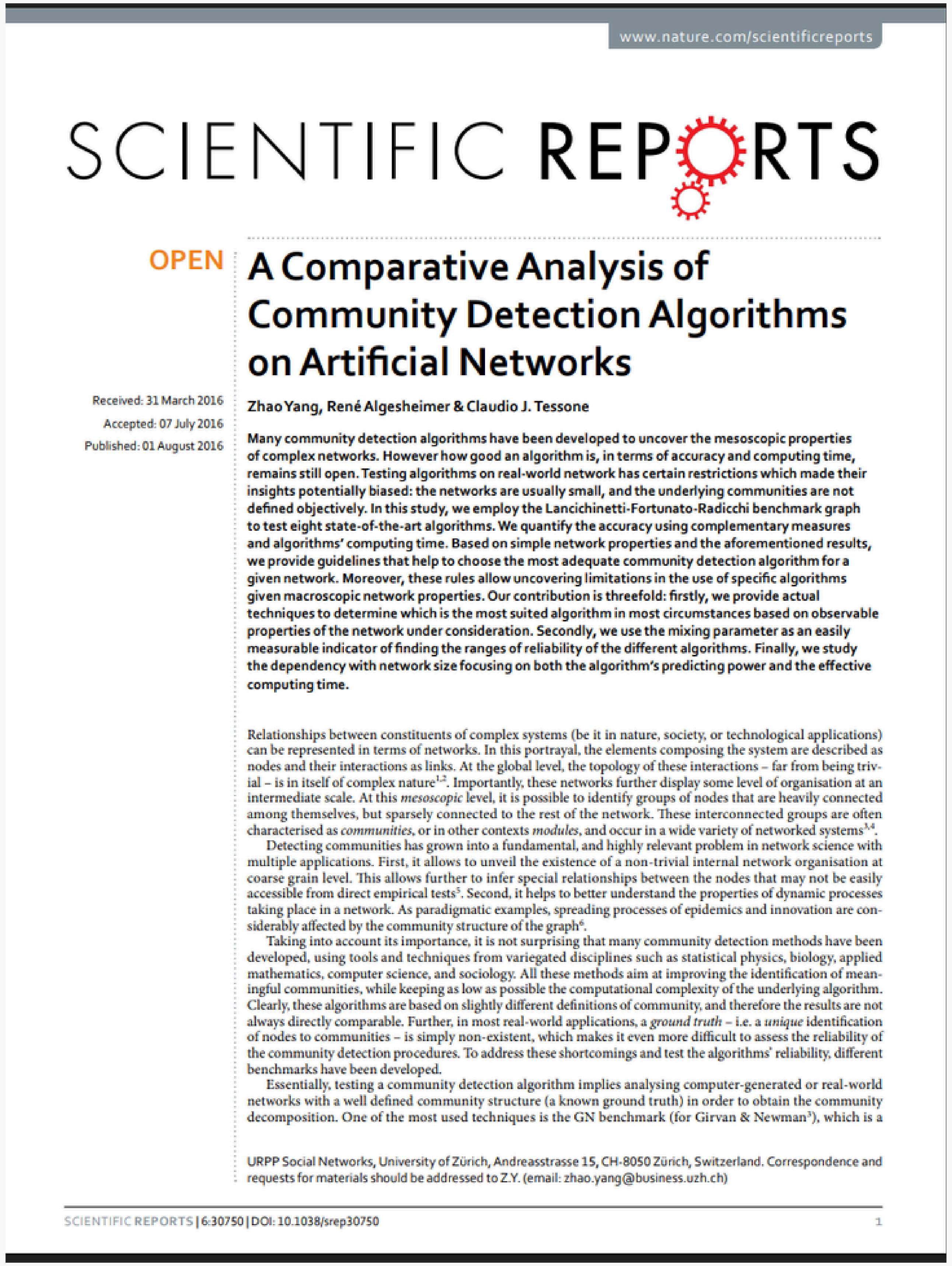
Nome do artigo:

A Comparative Analysis of Community Detection Algorithms on Artificial Networks

Fonte:

https://www.nature.com/srep

Link para o artigo



SCIENTIFIC REPORTS

OPEN

A Comparative Analysis of Community Detection Algorithms on Artificial Networks

Received: 31 March 2016
Accepted: 07 July 2016
Published: 01 August 2016

Zhao Yang, René Algesheimer & Claudio J. Tessone

Many community detection algorithms have been developed to uncover the mesoscopic properties of complex networks. However how good an algorithm is, in terms of accuracy and computing time, remains still open. Testing algorithms on real-world network has certain restrictions which made their insights potentially biased: the networks are usually small, and the underlying communities are not defined objectively. In this study, we employ the Lancichinetti-Fortunato-Radicchi benchmark graph to test eight state-of-the-art algorithms. We quantify the accuracy using complementary measures and algorithms' computing time. Based on simple network properties and the aforementioned results, we provide guidelines that help to choose the most adequate community detection algorithm for a given network. Moreover, these rules allow uncovering limitations in the use of specific algorithms given macroscopic network properties. Our contribution is threefold: firstly, we provide actual techniques to determine which is the most suited algorithm in most circumstances based on observable properties of the network under consideration. Secondly, we use the mixing parameter as an easily measurable indicator of finding the ranges of reliability of the different algorithms. Finally, we study the dependency with network size focusing on both the algorithm's predicting power and the effective computing time.

Relationships between constituents of complex systems (be it in nature, society, or technological applications) can be represented in terms of networks. In this portrayal, the elements composing the system are described as nodes and their interactions as links. At the global level, the topology of these interactions – far from being trivial – is in itself of complex nature[1,2]. Importantly, these networks further display some level of organisation at an intermediate scale. At this *mesoscopic* level, it is possible to identify groups of nodes that are heavily connected among themselves, but sparsely connected to the rest of the network. These interconnected groups are often characterised as *communities*, or in other contexts *modules*, and occur in a wide variety of networked systems[3,4].

Detecting communities has grown into a fundamental, and highly relevant problem in network science with multiple applications. First, it allows to unveil the existence of a non-trivial internal network organisation at coarse grain level. This allows further to infer special relationships between the nodes that may not be easily accessible from direct empirical tests[5]. Second, it helps to better understand the properties of dynamic processes taking place in a network. As paradigmatic examples, spreading processes of epidemics and innovation are considerably affected by the community structure of the graph[6].

Taking into account its importance, it is not surprising that many community detection methods have been developed, using tools and techniques from variegated disciplines such as statistical physics, biology, applied mathematics, computer science, and sociology. All these methods aim at improving the identification of meaningful communities, while keeping as low as possible the computational complexity of the underlying algorithm. Clearly, these algorithms are based on slightly different definitions of community, and therefore the results are not always directly comparable. Further, in most real-world applications, a *ground truth* – i.e. a *unique* identification of nodes to communities – is simply non-existent, which makes it even more difficult to assess the reliability of the community detection procedures. To address these shortcomings and test the algorithms' reliability, different benchmarks have been developed.

Essentially, testing a community detection algorithm implies analysing computer-generated or real-world networks with a well defined community structure (a known ground truth) in order to obtain the community decomposition. One of the most used techniques is the GN benchmark (for Girvan & Newman[3]), which is a

URPP Social Networks, University of Zürich, Andreasstrasse 15, CH-8050 Zürich, Switzerland. Correspondence and requests for materials should be addressed to Z.Y. (email: zhao.yang@business.uzh.ch)

Obrigado