



Universidade de São Paulo – São Carlos  
Instituto de Ciências Matemáticas e de Computação  
Redes Complexas para a Ciência da Computação



# Comparação de Algoritmos de Detecção de Comunidades

Glenda Michele Botelho  
Fabiano Berardo de Sousa



# Roteiro

- Introdução
- Proposta do Projeto
- Experimentos
  - Medidas avaliadas
  - Algoritmos comparados
  - Bases utilizadas
  - Resultados
- Conclusões



# Introdução

- **Detecção de Comunidades**
  - **Aprendizado de Máquina**
    - **Aprendizado Não-Supervisionado.**
    - **Técnicas de *Clustering*.**
- **Relevância**
  - **Redes Complexas modelam sistemas reais**
    - **Extração de características específicas.**
    - **Estudo da organização e evolução dinâmica da rede.**



# Introdução

- **Vários Algoritmos:**
  - Divisivos (*Betweenness*)
  - Aglomerativos (Modularidade)
  - Espectrais (Autovetores)
  - Locais (Modularidade Local)
  - Otimizações (Extrema, Monte Carlo)



# Introdução

- Como medir a “eficiência” do algoritmo?
  - Aprendizado não supervisionado
    - Não há atributo meta
  - Definição de comunidade não é clara
    - Depende do contexto do sistema

# Introdução

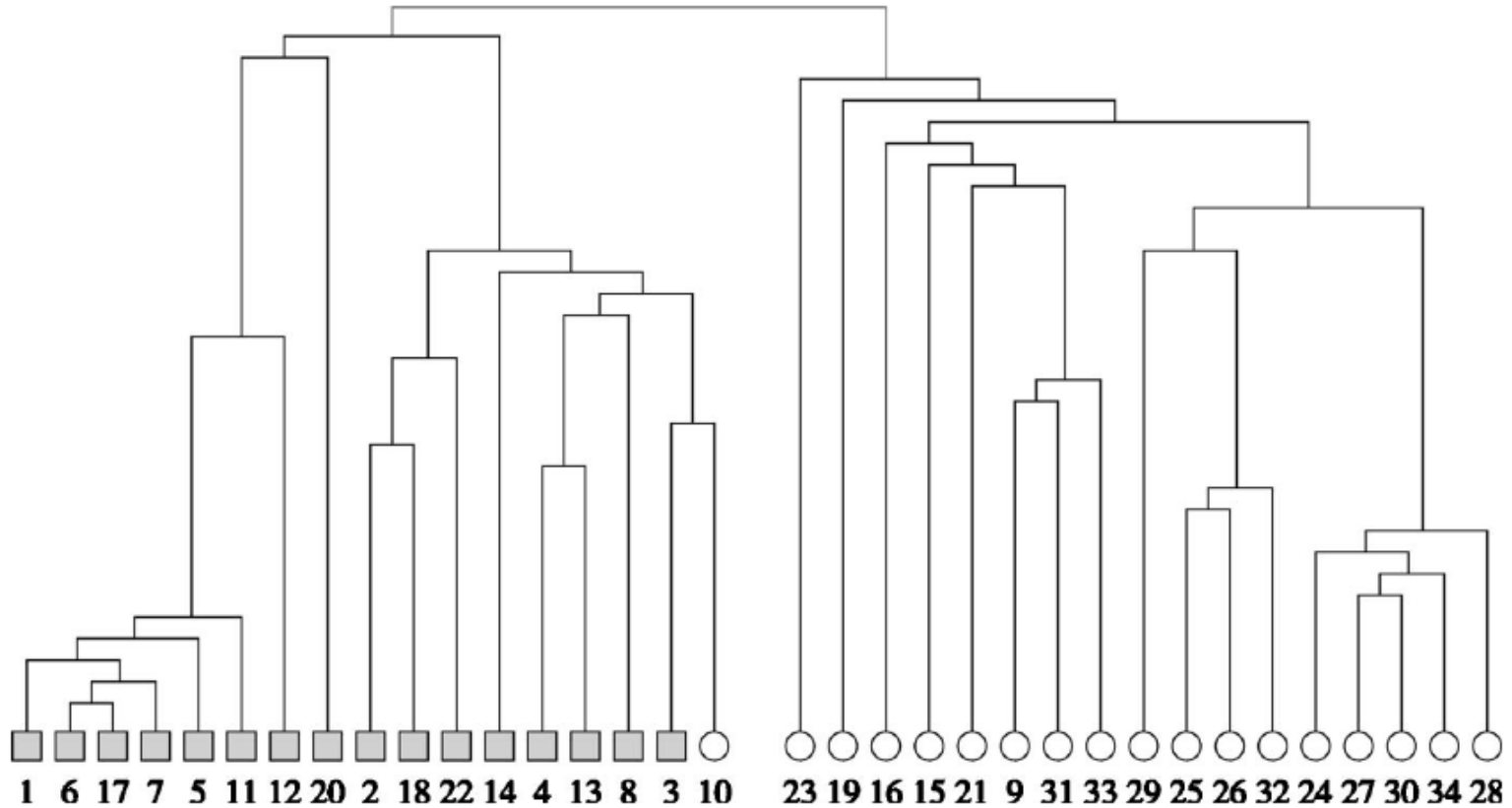
- Medida de Modularidade [Newman 2004]

$$Q = \sum_i (e_{ii} - a_i^2)$$

- $e_{ii}$  = fração das arestas inseridas no grupo  $i$ .
- $a_i^2$  = fração das arestas inseridas aleatoriamente no grupo  $i$ .

# Introdução

- Medida de Modularidade [Newman 2004]





# Proposta do Projeto

- Comparar diferentes algoritmos de detecção de comunidades
- Medidas utilizadas:
  - Modularidade
  - Tempo de execução
  - Número de comunidades detectadas





# Experimentos

- **Ambiente:**
  - Biblioteca IGRAPH para R
  - Core Duo 1.73GHz, 2GB RAM
  - Slackware 32bits.

# Experimentos

- Bases utilizadas
  - Zarachry's Karate Club (34, 78)
  - American College Football (115, 615)
  - Neural Network (297, 2359, D, W)
  - Coauthorship in Network Science -  
Netscience (1589, 2742, W, Desconexo)
  - ~~Internet (22962, ...)~~

# Experimentos

- **Algoritmos comparados:**
  - *Betweenness* [Newman and Girvan 2004]
    - Caminhos mínimos entre vértices.
  - *Caminhada Aleatória* [Pons and Latapy 2005]
    - Caminhos aleatórios curtos tendem a estar na mesma comunidade.
  - *FastGreedy* [Clauset et al. 2004]:
    - *Otimização da modularidade por meio de um algoritmo guloso.*
  - *Autovetores* [Newman 2006]
    - Matriz de modularidade modelada em função de autovetores e autovalores.
  - *Spinglass* [Reichardt and Bornholdt 2006]
    - *Otimização da modularidade usando Simulated Annealing.*

# Experimentos

- Medida de Modularidade da IGRAPH

$$Q = \frac{1}{2m} \sum_{ij} [A_{ij} - \frac{k_i k_j}{2m} \delta(c_i c_j)]$$

- Implementação para maximizar a medida, percorrendo o dendograma, segundo proposta de [Newman 2004]
- Não calculamos  $\Delta Q$ .

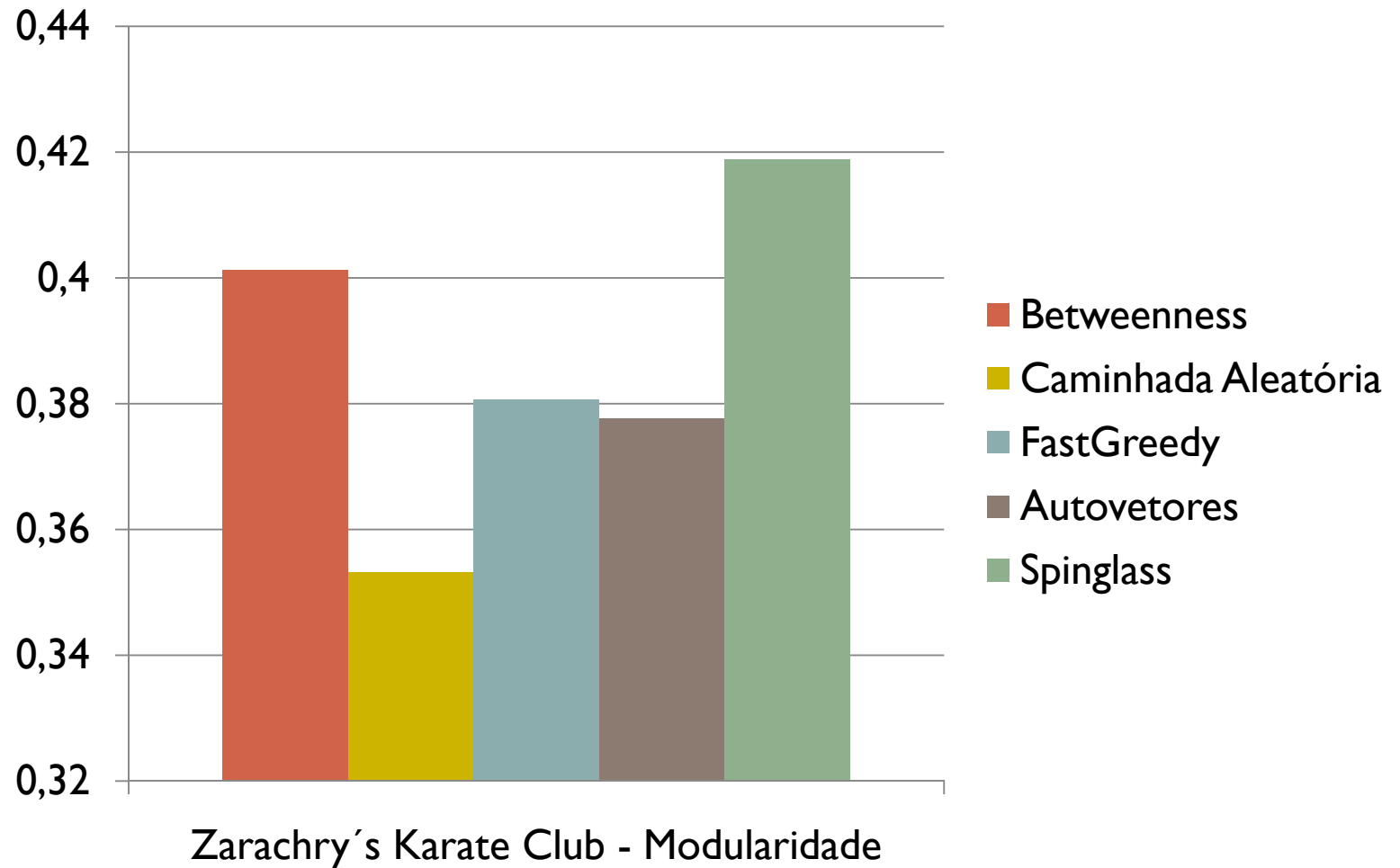
# Resultados

- Zarachry's Karate Club (grafo com 34 vértices, não direcionado e sem pesos ).

	Modularidade	Tempo de Execução	Qtde de Comunidades
<i>Betweenness</i>	0.4012985	0.019s	5
Caminhada Aleatória	0.3532216	0.001s	5
<i>FastGreedy</i>	0.3806706	0.001s	3
Autovetores	0.3776298	0.031s	5
<i>Spinglass</i>	0.4188034	2.118s	4

- Distribuição dos vértices nas comunidades:
  - [ 10 12 6 5 1 ] BT
  - [ 9 7 9 4 5 ] CA
  - [ 8 17 9 ] FG
  - [ 7 12 4 5 6 ] AV
  - [ 11 6 5 12 ] SG

# Modularidade - Zarachry's Karate Club



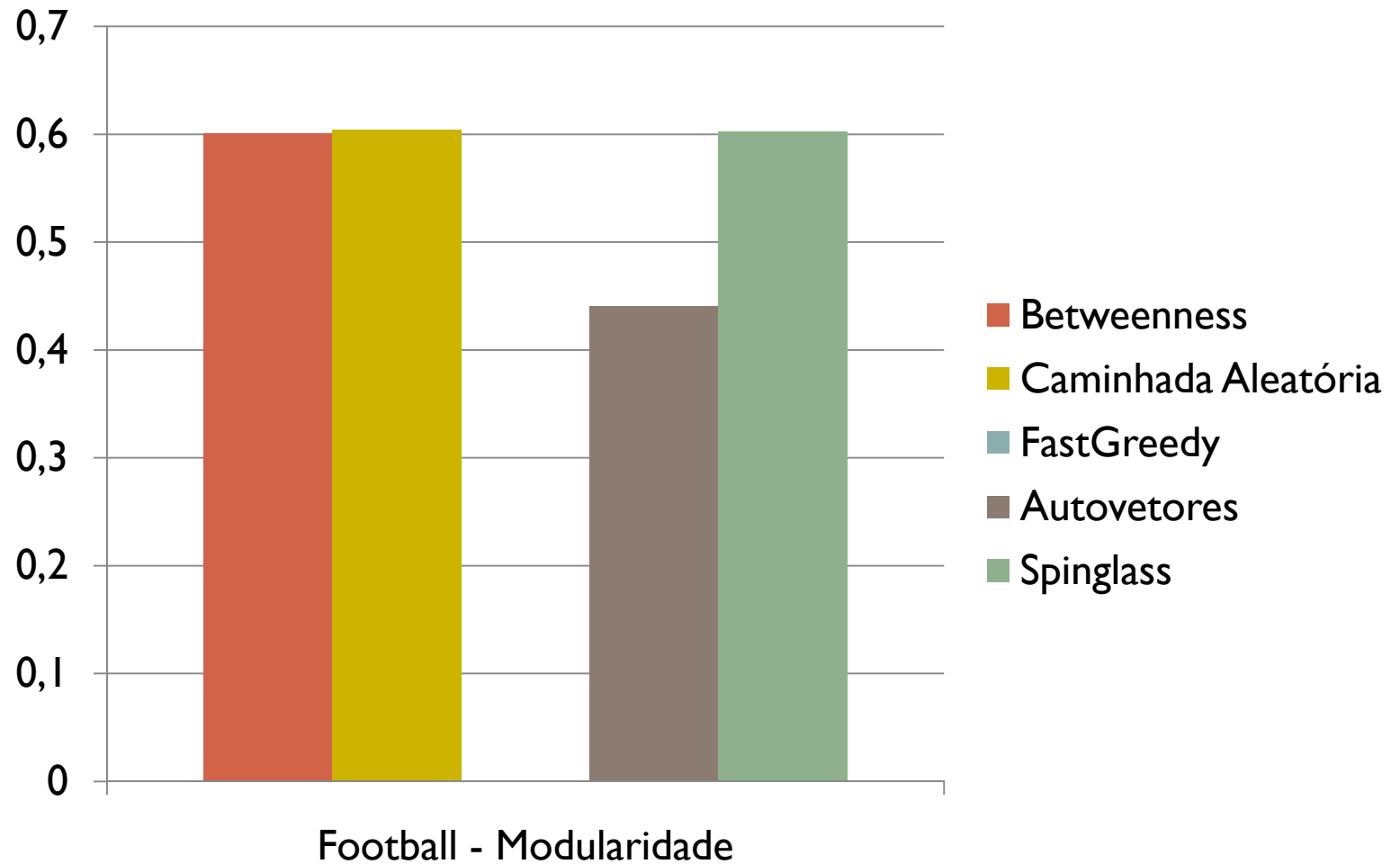
# Resultados

- Football (grafo com 115 vértices, não direcionado e sem peso).

	Modularidade	Tempo de Execução	Qtde de Comunidades
<i>Betweenness</i>	0.6005129	2.394s	10
Caminhada Aleatória	0.6038112	0.007s	10
<i>FastGreedy</i>	*	*	*
Autovetores	0.4402313	0.120s	13
<i>Spinglass</i>	0.6027933	3.117s	10

- \* Não funcionou.

# Modularidade - Football





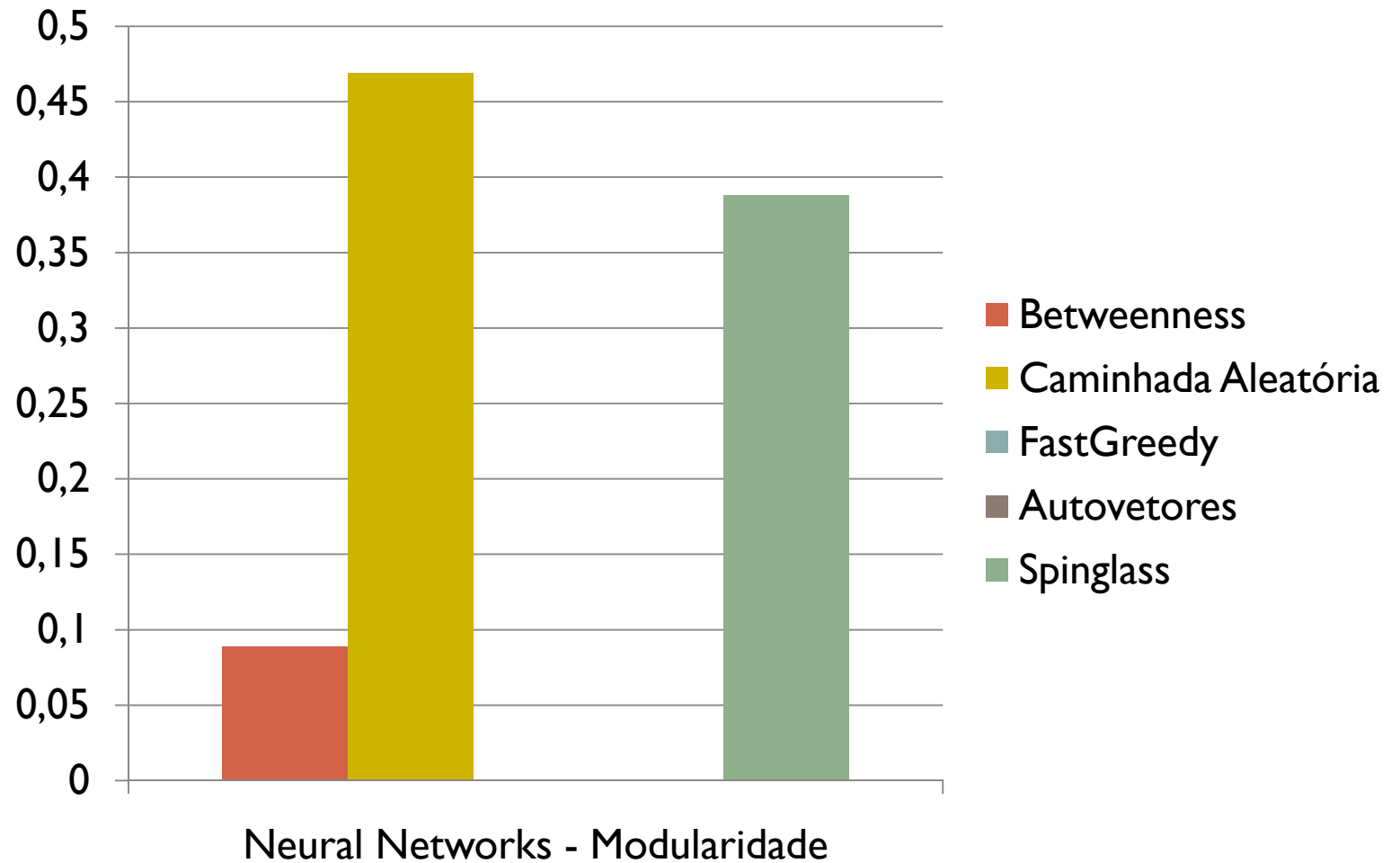
# Resultados

- Neural Network (grafo com 297 nós, direcionado e com pesos).

	Modularidade	Tempo de Execução	Qtde de Comunidades
<i>Betweenness</i>	0.0887715	22.493s	200
Caminhada Aleatória	0.469383	0.045s	24
<i>FastGreedy</i>	*	*	*
Autovetores	*	*	*
<i>Spinglass</i>	0.3883311	30.552s	5

- \* Somente para grafos não-direcionados.

# Modularidade – Neural Networks



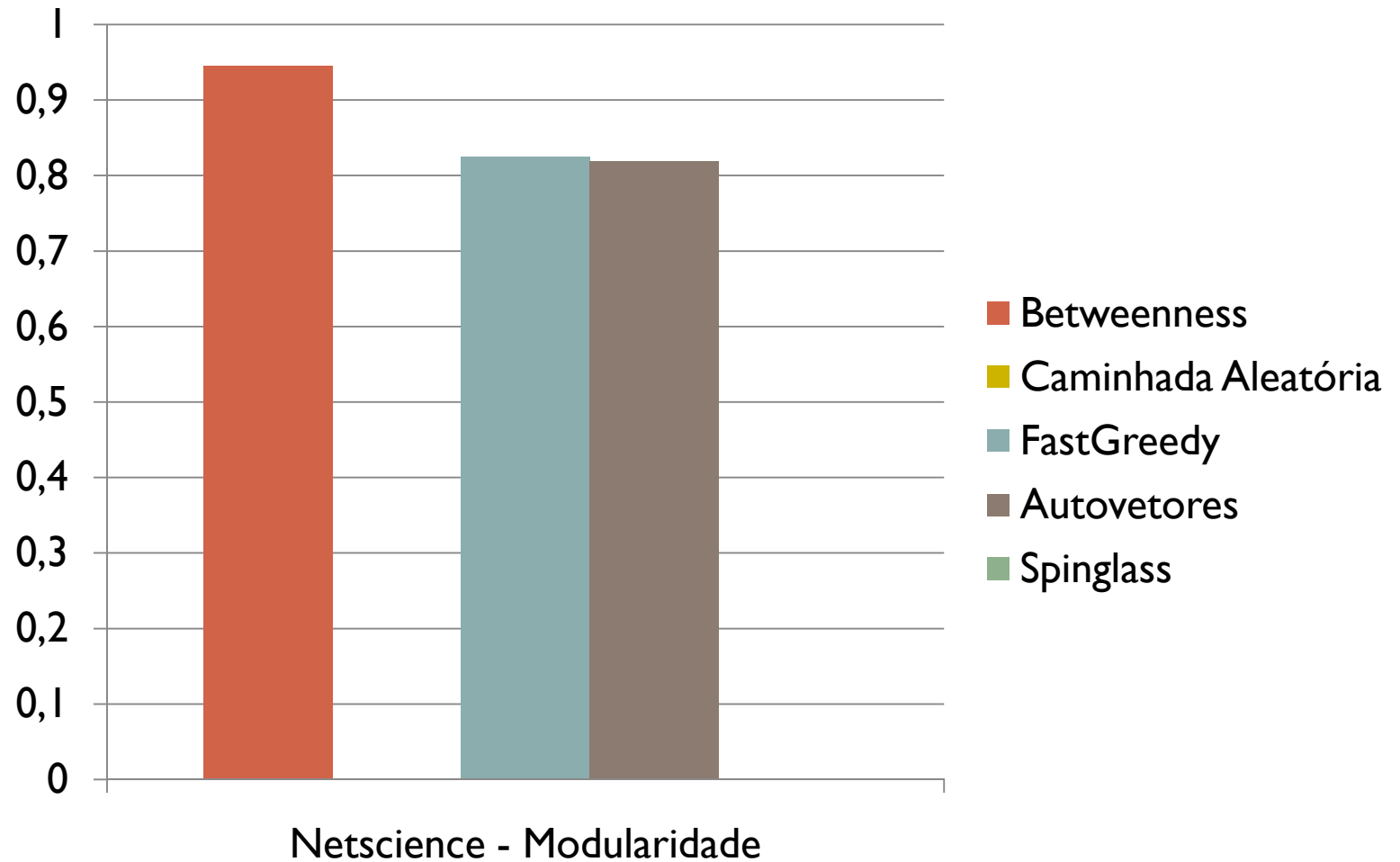
# Resultados

- Netscience (grafo com 1589 vértices, ponderado e desconexo).

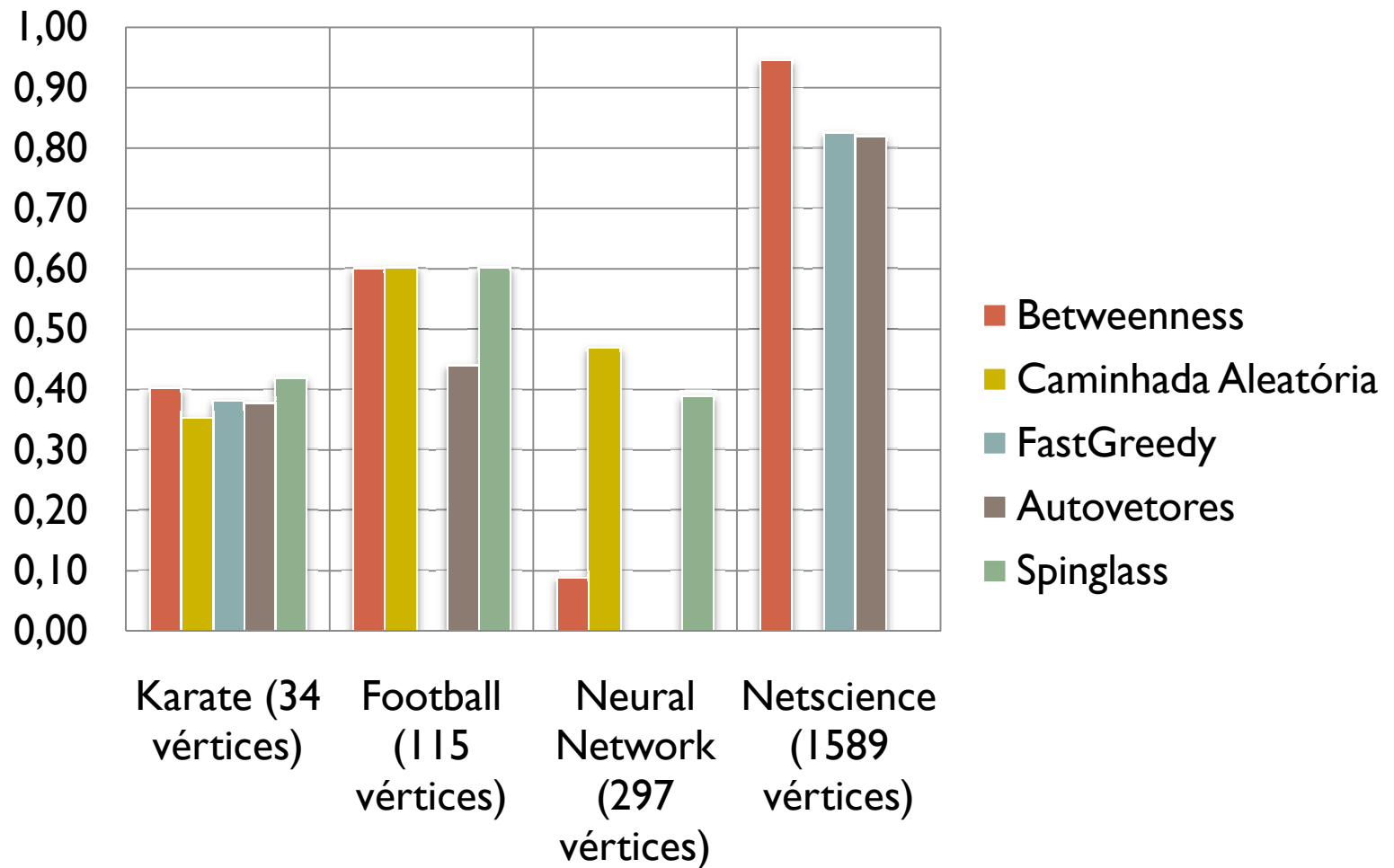
	Modularidade	Tempo de Execução	Qtde de Comunidades
<i>Betweenness</i>	0.9453312	24.440s	406
<i>Caminhada Aleatória</i>	*	*	*
<i>FastGreedy</i>	0.8252987	0.014s	396
<i>Autovetores</i>	0.81933	48.564s	234
<i>Spinglass</i>	*	*	*

- \* Grafo desconexo

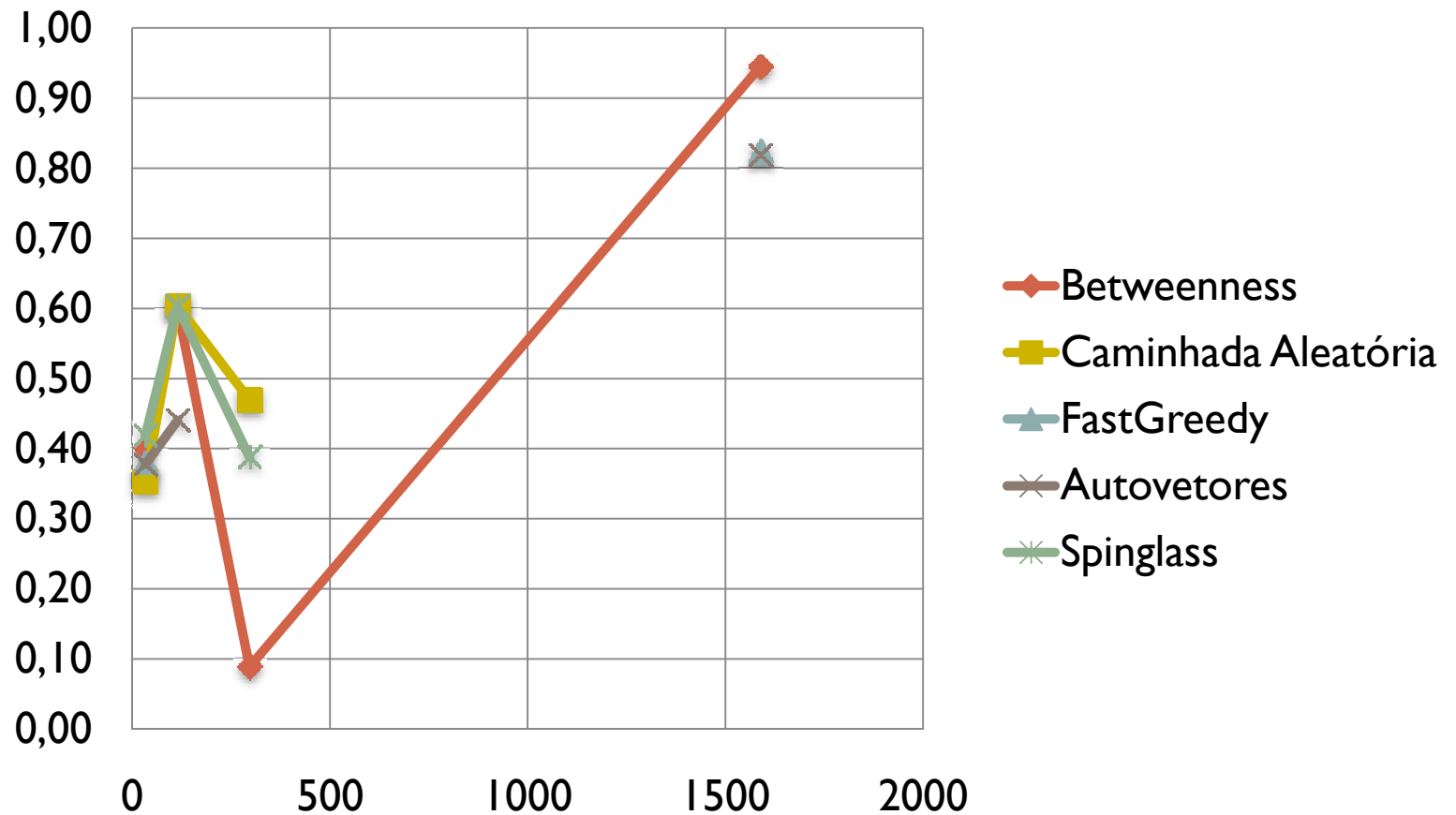
# Modularidade - Netscience



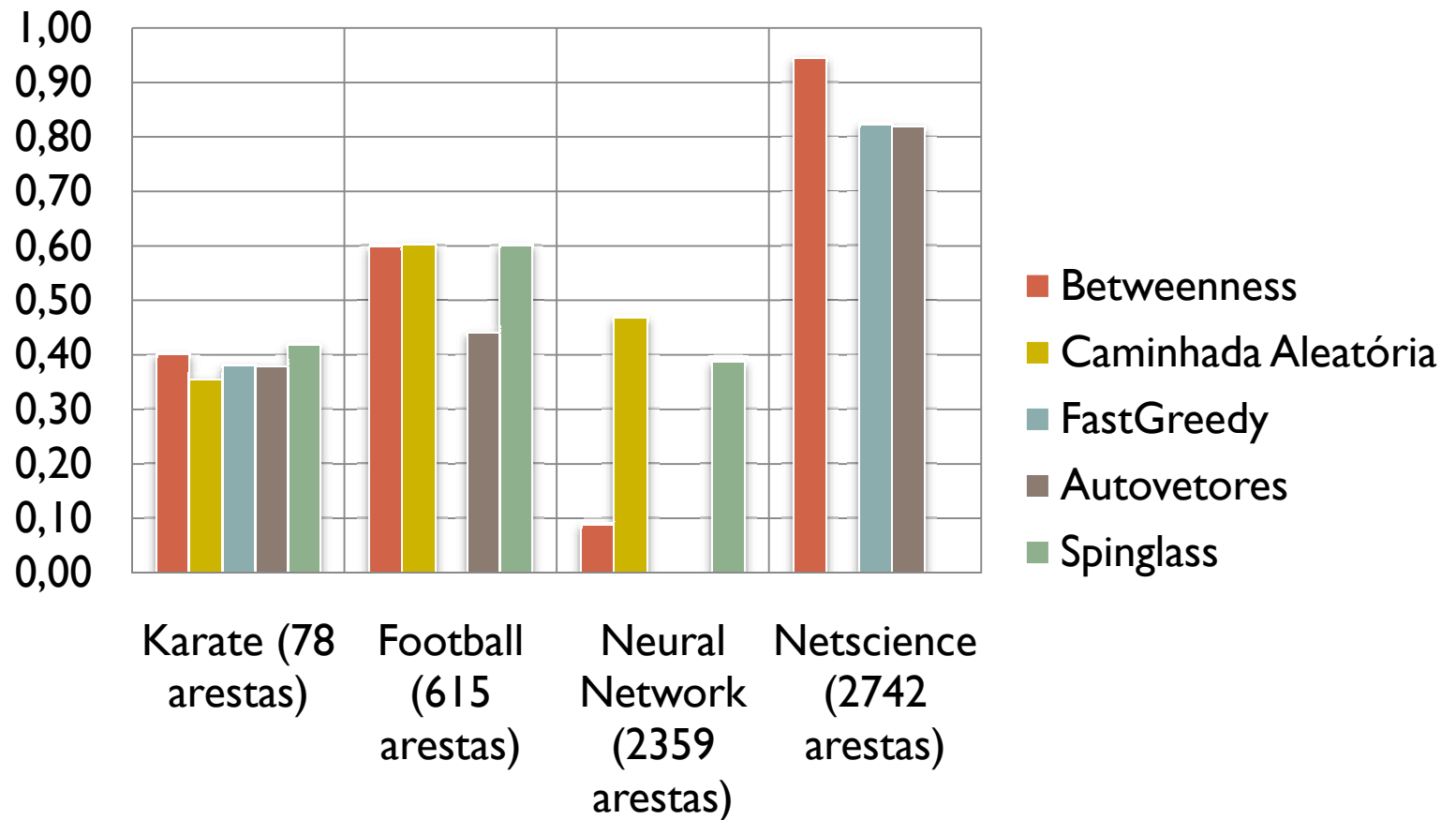
# Número de Vértices vs Modularidade



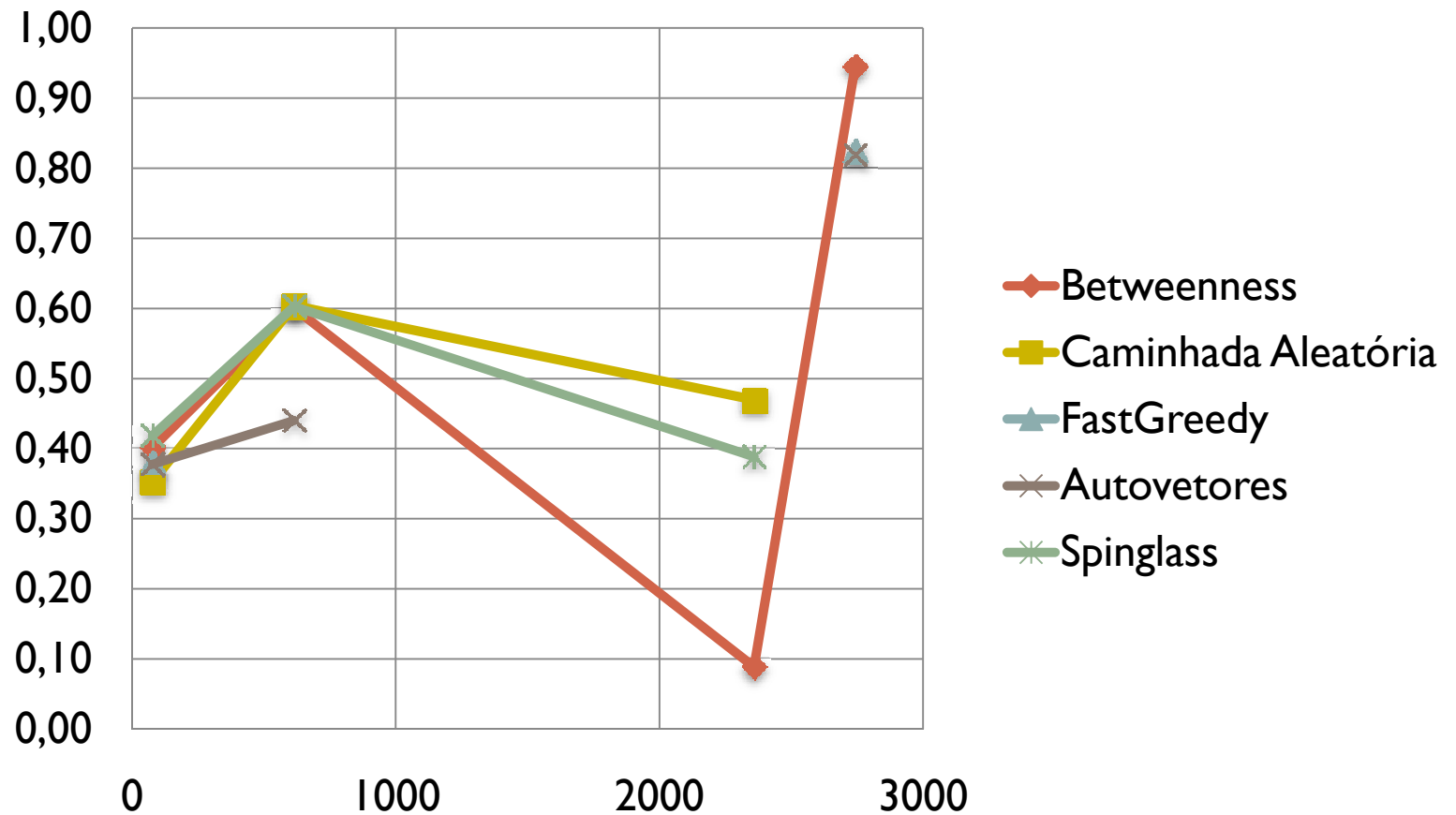
# Número de Vértices vs Modularidade



# Número de Arestas vs Modularidade

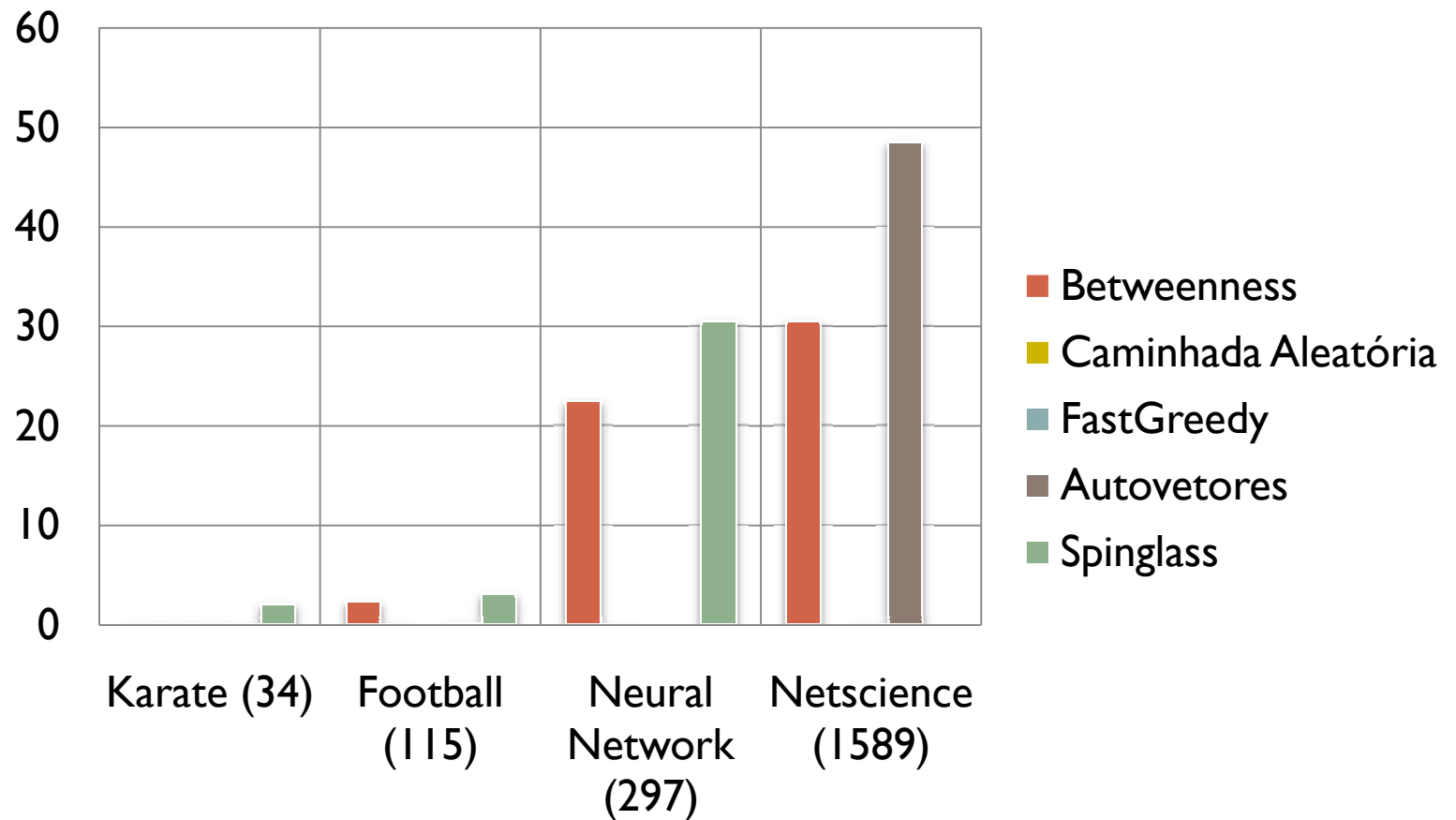


# Número de Arestas vs Modularidade

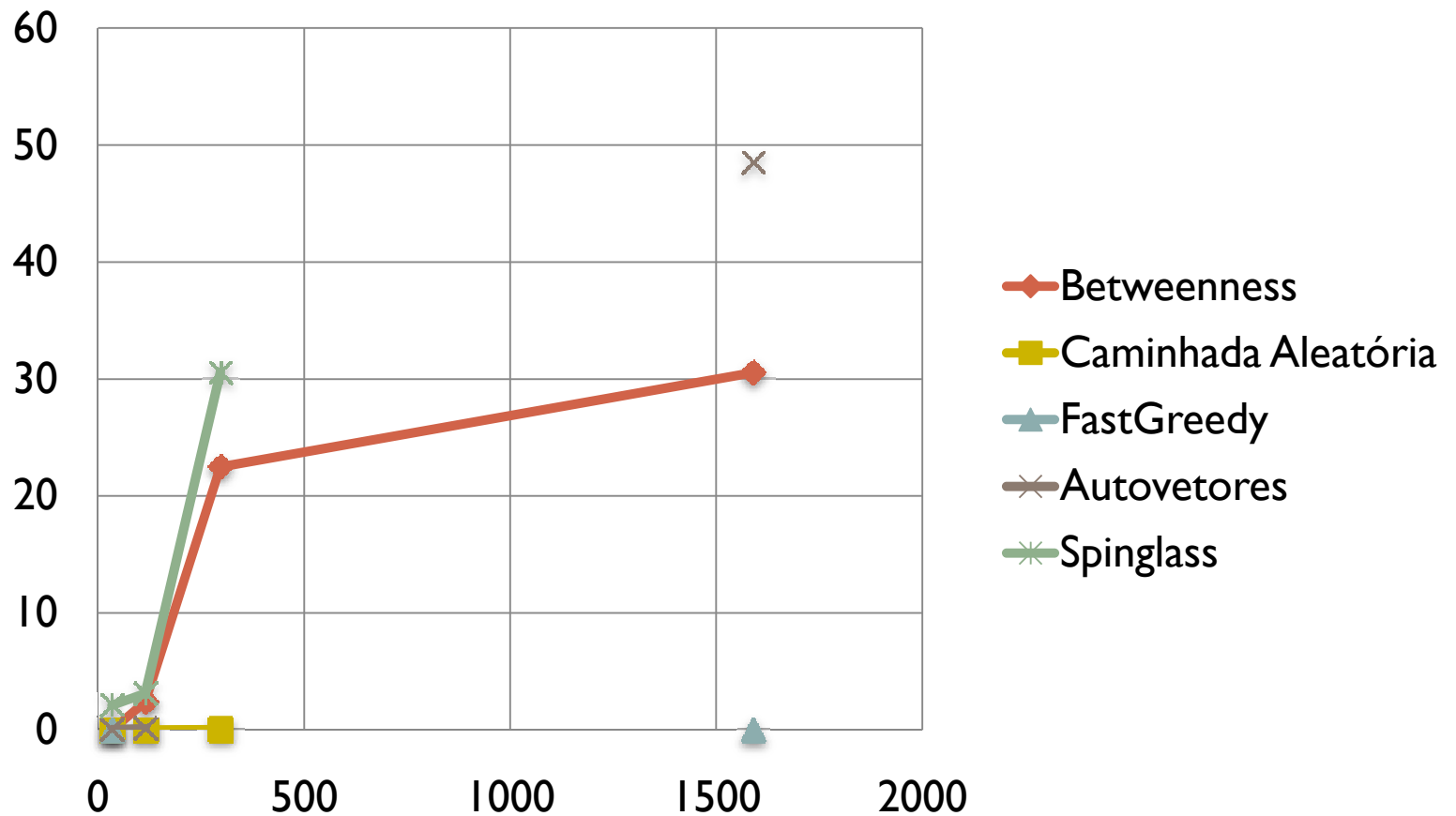




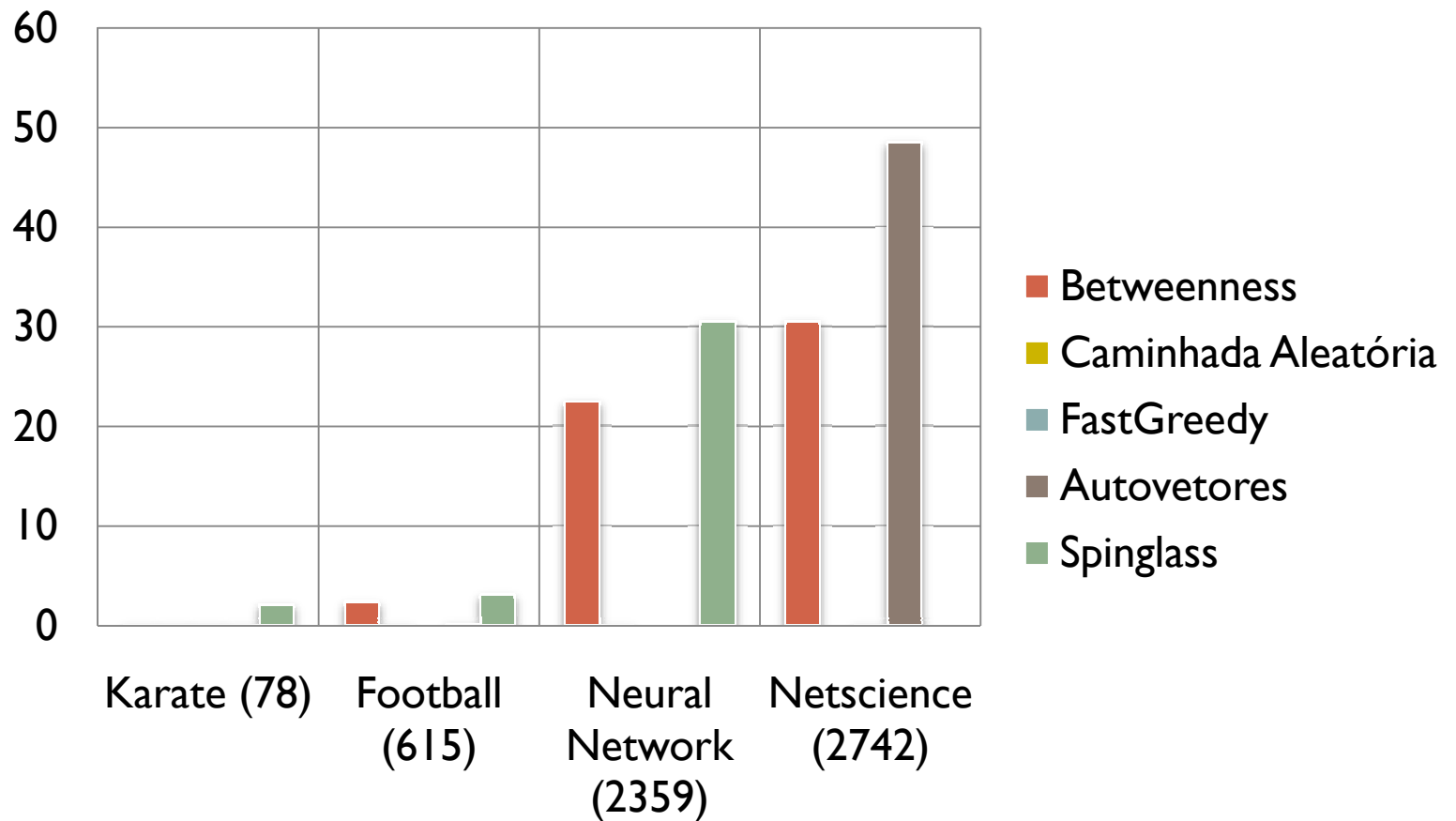
# Tempo de Execução (segundos) vs Número de Vértices



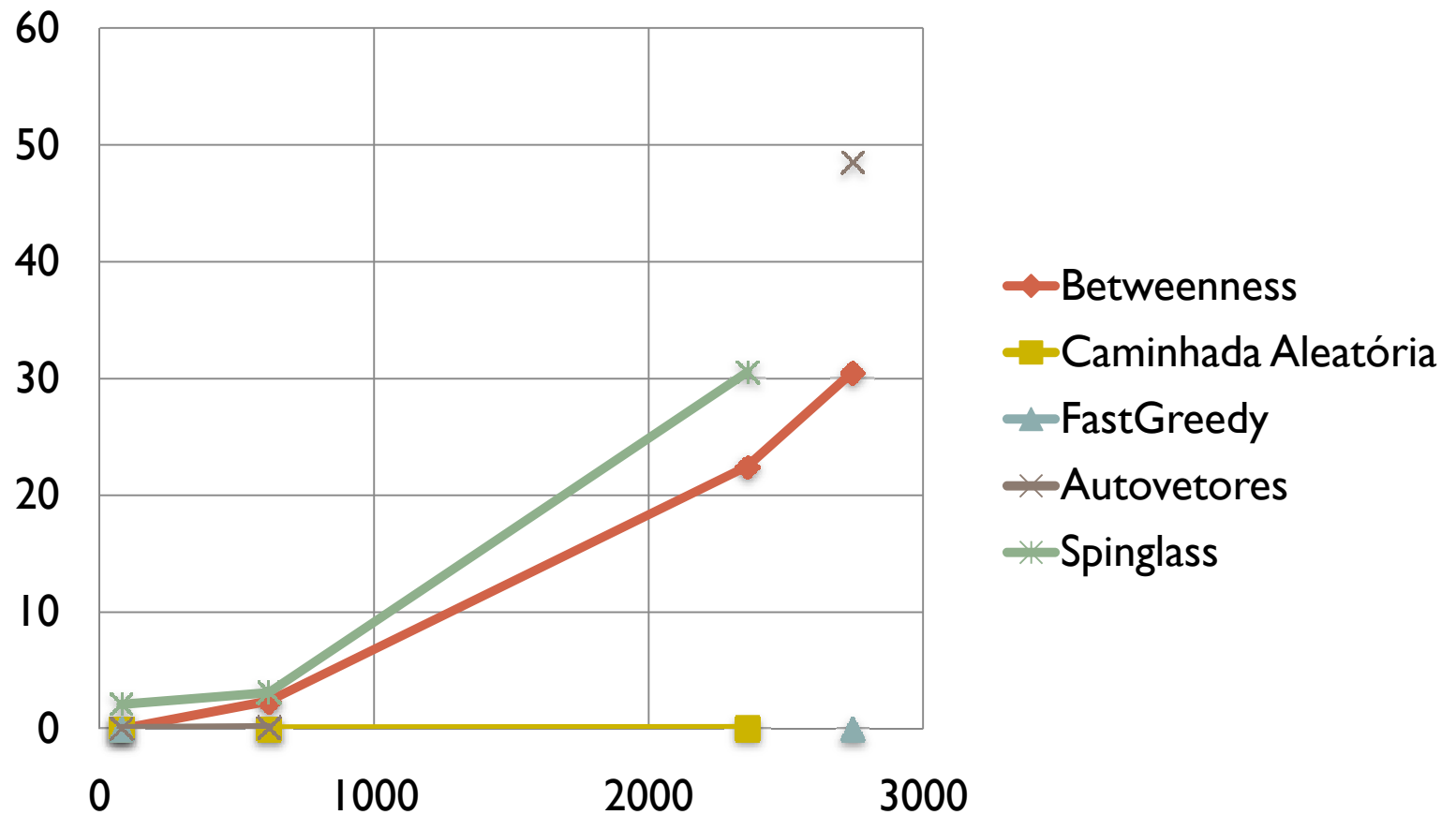
# Tempo de Execução (segundos) vs Número de Vértices



# Tempo de Execução (segundos) vs Número de Arestas



# Tempo de Execução (segundos) vs Número de Arestas



# Conclusões

- Altos valores de modularidade.
- Quantidade de comunidades obtida é variável.
- Alto custo computacional dos algoritmos.
- A modularidade varia em mesmo padrão tanto com o aumento do número de vértices quando com o aumento do número de arestas.
- Com poucos vértices os algoritmos mostram uma modularidade similar.
- Caminhada Aleatória e *FastGreedy* mantêm baixo tempo de execução.



# Conclusões

- Resultados ainda pouco significativos.
- Necessidade de testes com maior número de bases.
- Necessidade de melhor sistema computacional (base “Internet”).



# Bibliografia

- [Clauset et al. 2004] Clauset, A., Newman, M. E. J., and Moore, C. (2004). Finding Community Structure in Very Large Networks. *Physical Review E*, 79(066111).
- [Newman and Girvan 2004] Newman, M. E. J. and Girvan, M. (2004). Finding and Evaluating Community Structure in Networks. *Physical Review E*, 69(026113).
- [Newman 2004] Newman, M. E. J. (2004). Fast Algorithm for Detecting Community Structure in Networks. *Physical Review E*, 69(066133).
- [Newman 2006] Newman, M. E. J. (2006). Finding Community Structure in Networks using the Eigenvectors of Matrices. *Physics/0605087*.
- [Pons and Latapy 2005] Pons, P. and Latapy, M. (2005). Computing Communities in Large Networks using Random Walks. *Proceedings of the 20th International Symposium on Computer and Information Sciences, ISCIS'05, LNCS 3733, 284-293*.
- [Reichardt and Bornholdt 2006] Reichardt, J. and Bornholdt, S. (2006). Statistical Mechanics of Community Detection, *Physical Review E*, 74, 016110.



**Dúvidas?**