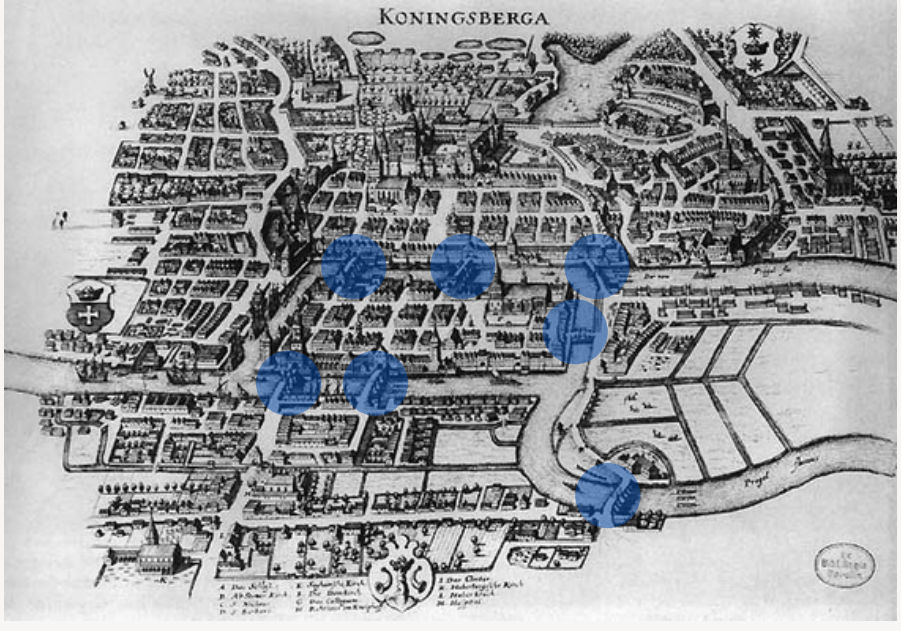
Análise de Redes Biológicas com a Linguagem Python

Prof. Dr. Gilderlanio Santana de Araújo
Arthur Ribeiro dos Santos

PPGBM0102 – Tópicos Avançados III – T01

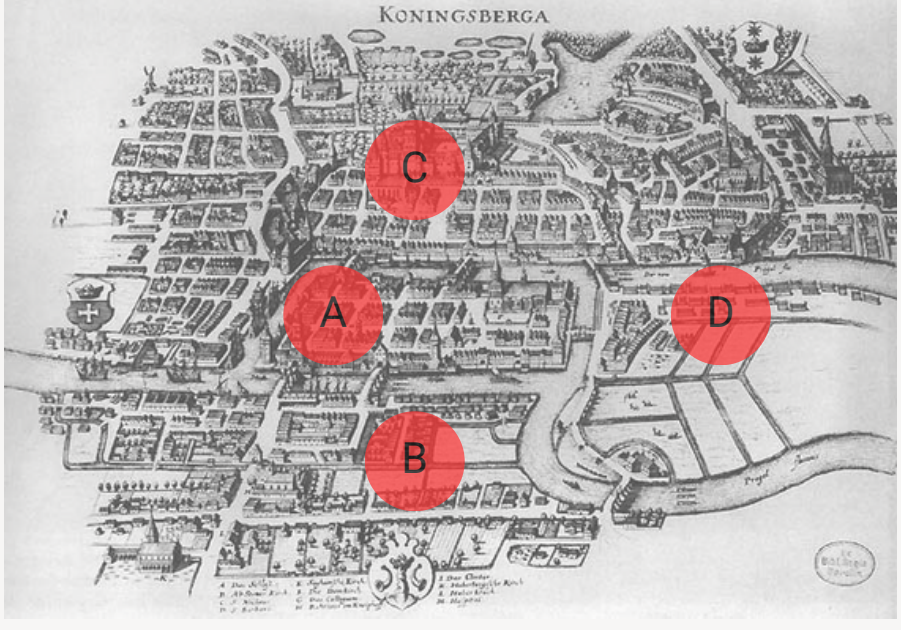
Sete Pontes de Königsberg

- Desafio contemporâneo
- É possível atravessar todas pontes da cidade sem repetições?



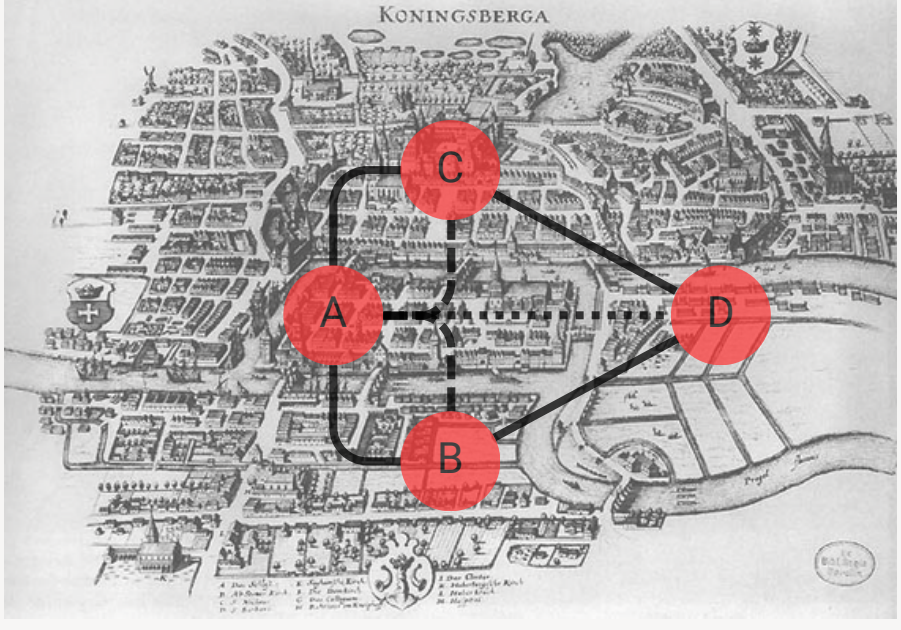
Sete Pontes de Königsberg

- Resolução do problema por Leonhard Euler (1736):
  - Comprovou matematicamente a solução negativa do problema
  - Simplificação do problema em pontos e linhas



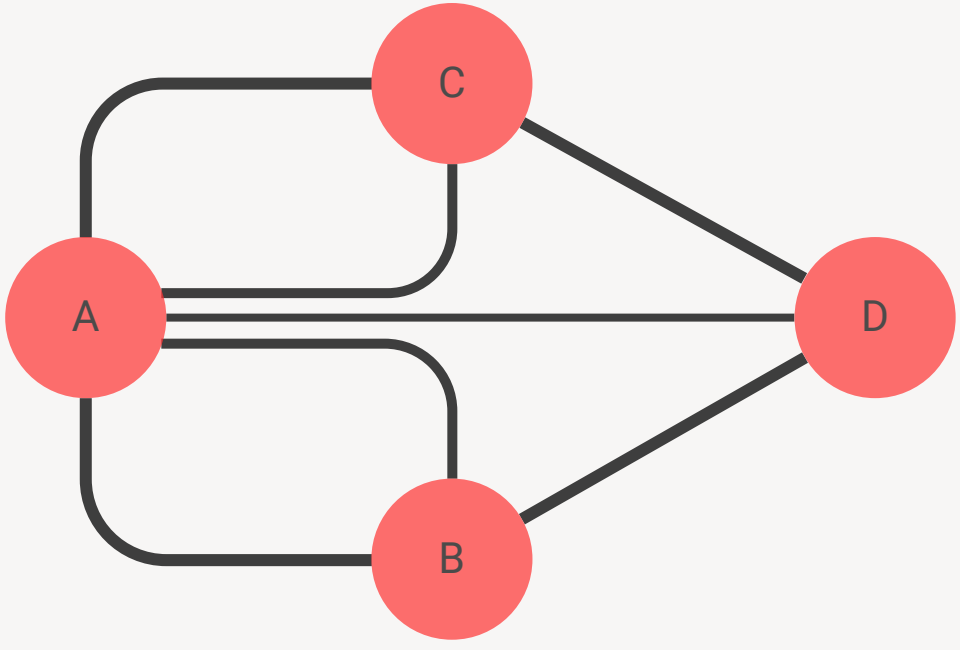
Sete Pontes de Königsberg

- Resolução do problema por Leonhard Euler (1736):
  - Comprovou matematicamente a solução negativa do problema
  - Simplificação do problema em pontos e linhas



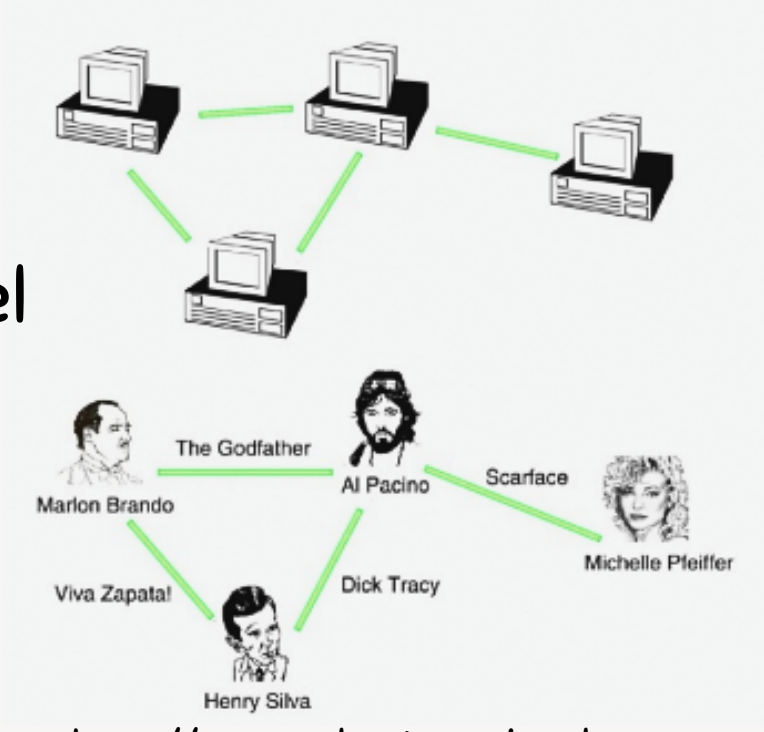
Sete Pontes de Königsberg

- Resolução do problema por Leonhard Euler (1736):
  - Comprovou matematicamente a solução negativa do problema
  - Simplificação do problema em pontos e linhas



Motivações

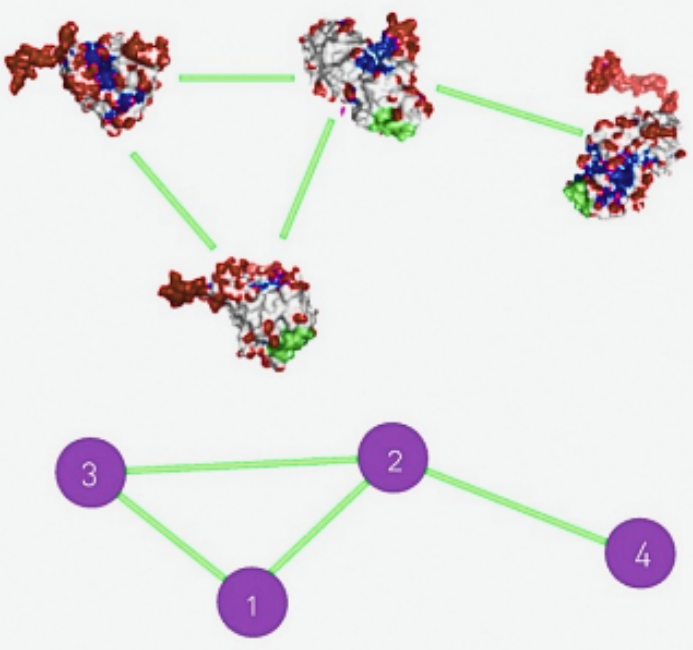
- Maior integração de dados
- Possibilitam análise a nível sistêmico



http://networksciencebook.com

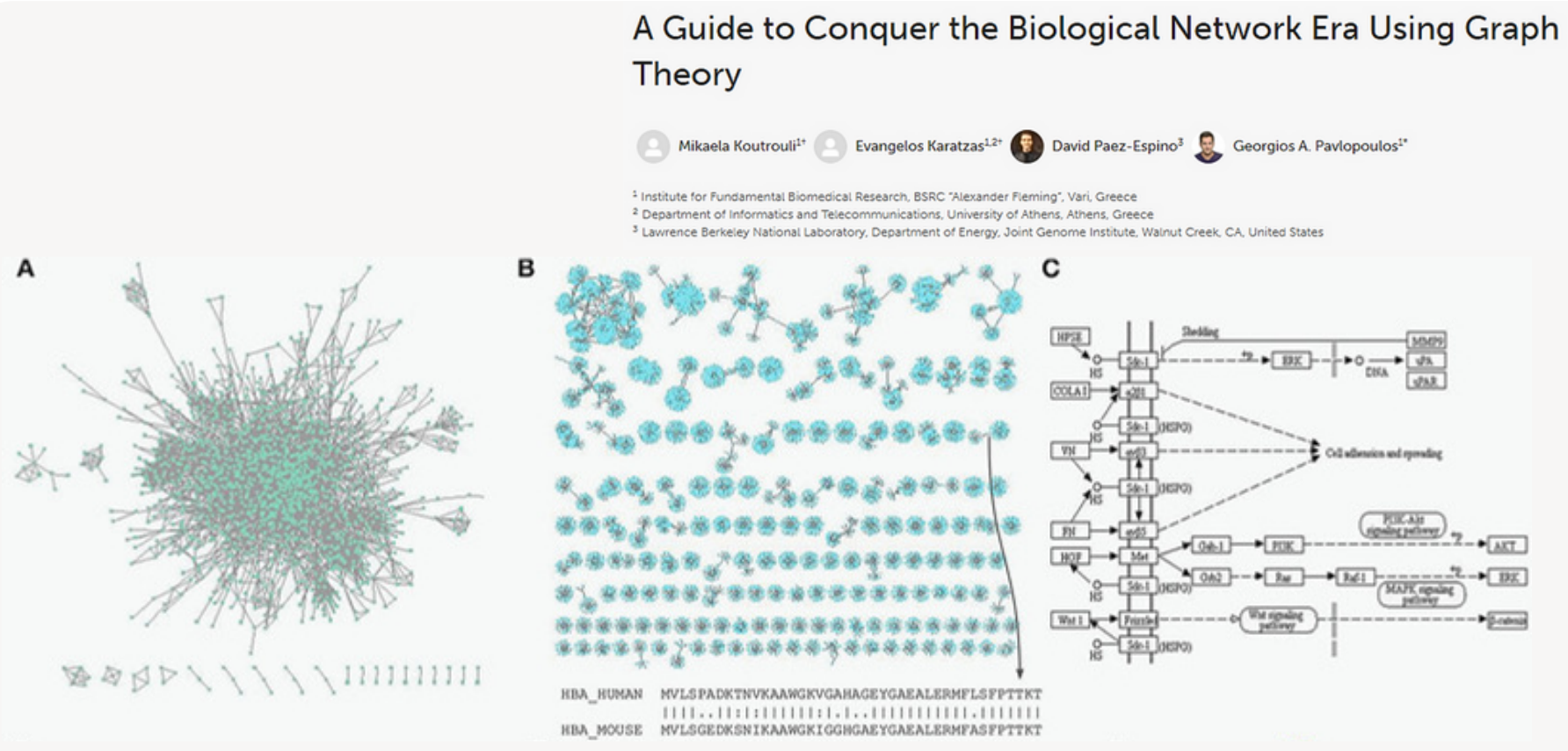
Motivações

- Grande volume de dados biológicos públicos
- Transformam problemas biológicos como questões de redes



http://networksciencebook.com

Exemplos de redes biológicas



Rede proteína-proteína

Vias metabólicas do KEGG

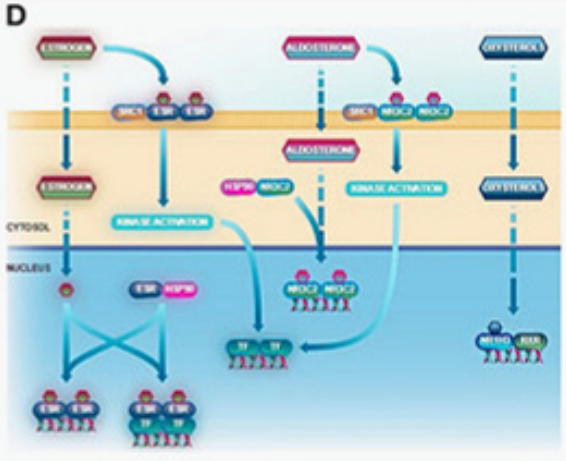
Rede de similaridade de sequência

Exemplos de redes biológicas

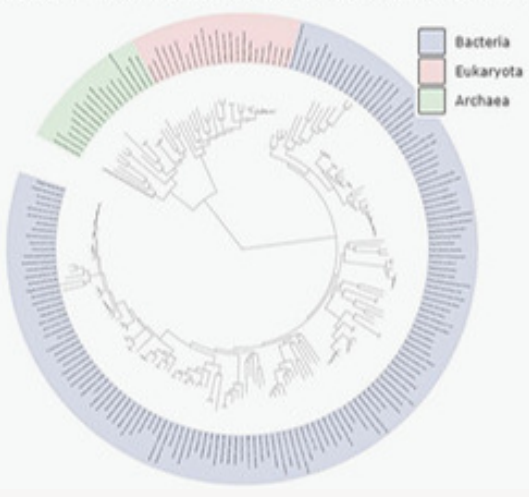
A Guide to Conquer the Biological Network Era Using Graph Theory

Mikaela Koutrouli[1*] Evangelos Karatzas[1,2*] David Paez-Espino[3] Georgios A. Pavlopoulos[1*]

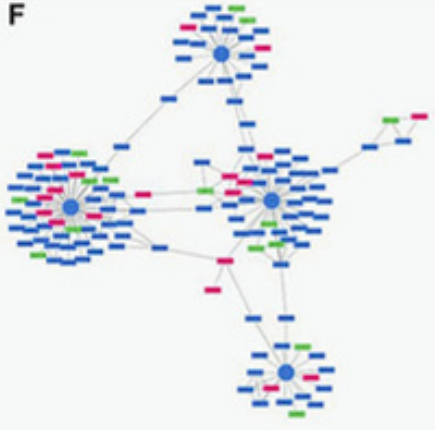
[1] Institute for Fundamental Biomedical Research, BSRC "Alexander Fleming", Vari, Greece
[2] Department of Informatics and Telecommunications, University of Athens, Athens, Greece
[3] Lawrence Berkeley National Laboratory, Department of Energy, Joint Genome Institute, Walnut Creek, CA, United States



Rede transdutora de sinais (REACTOME)



Árvore filogenétoca



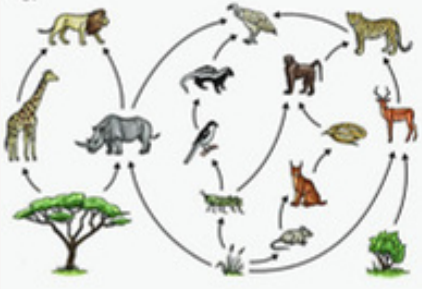
Rede de expressão gênica

Exemplos de redes biológicas

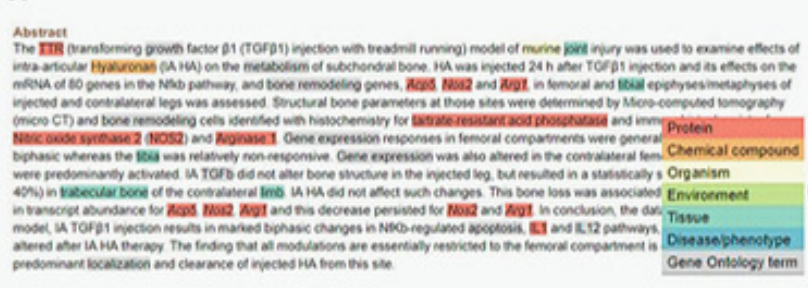
A Guide to Conquer the Biological Network Era Using Graph Theory

Mikaela Koutrouli[1*] Evangelos Karatzas[1,2*] David Paez-Espino[3] Georgios A. Pavlopoulos[1*]

[1] Institute for Fundamental Biomedical Research, BSRC "Alexander Fleming", Vari, Greece
[2] Department of Informatics and Telecommunications, University of Athens, Athens, Greece
[3] Lawrence Berkeley National Laboratory, Department of Energy, Joint Genome Institute, Walnut Creek, CA, United States



Cadeia alimentar

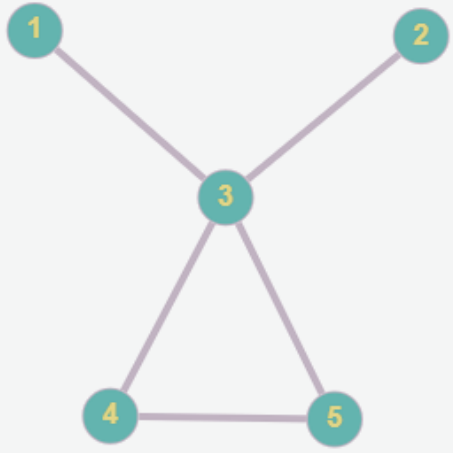


Co-ocorrências

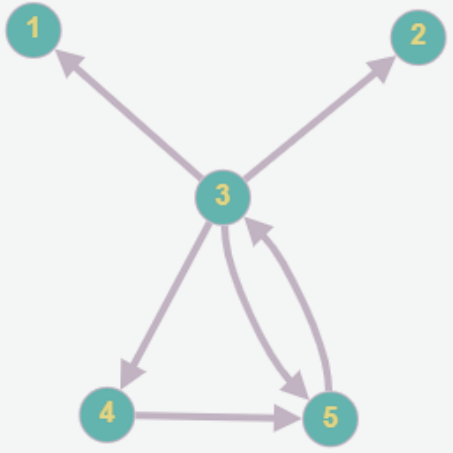
Rede PPI de múltiplas arestas

Tipos de grafos

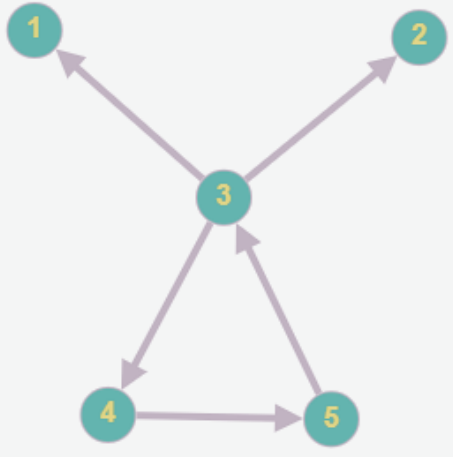
Não direcionado



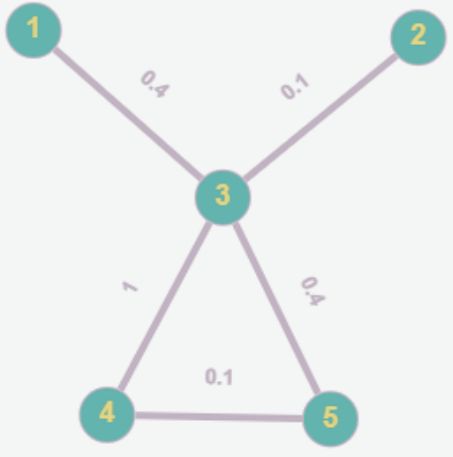
Misto



Direcionado



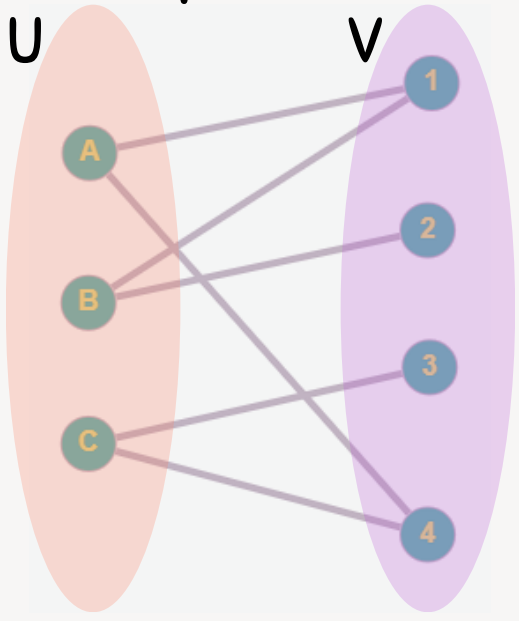
Com peso



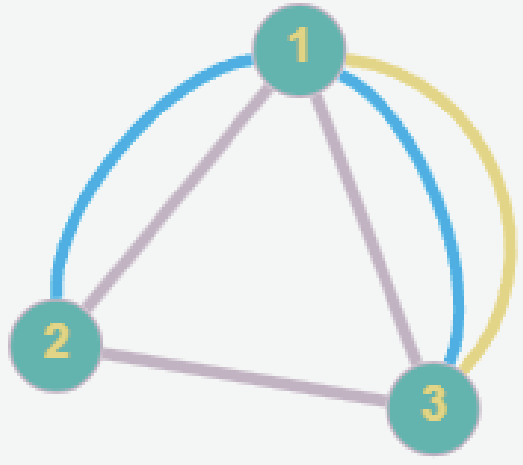
https://graphonline.ru/

Tipos de grafos

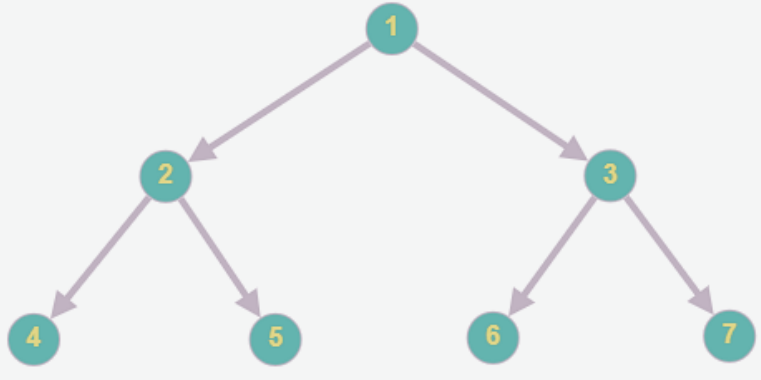
Bipartido



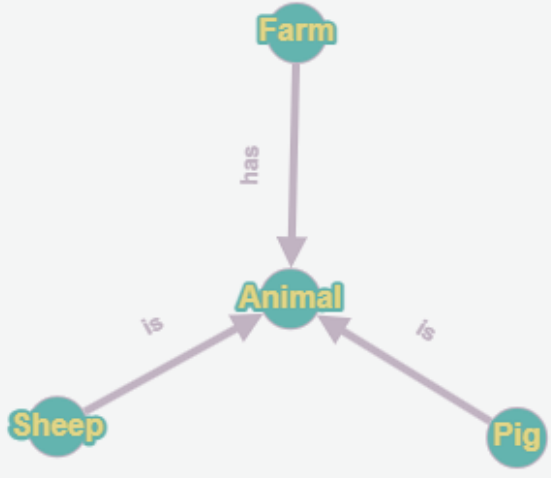
Multi-aresta



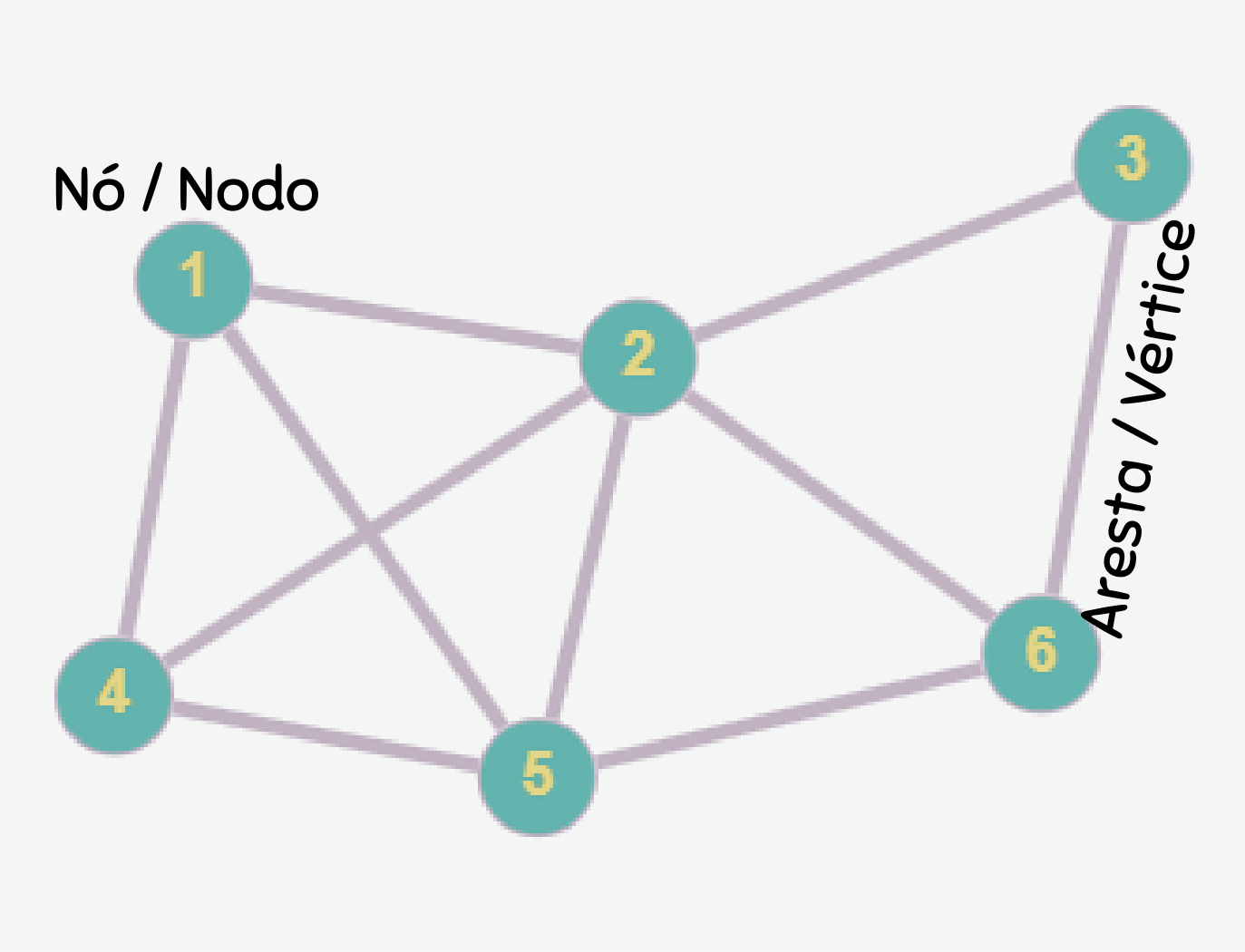
Árvore



Semântico

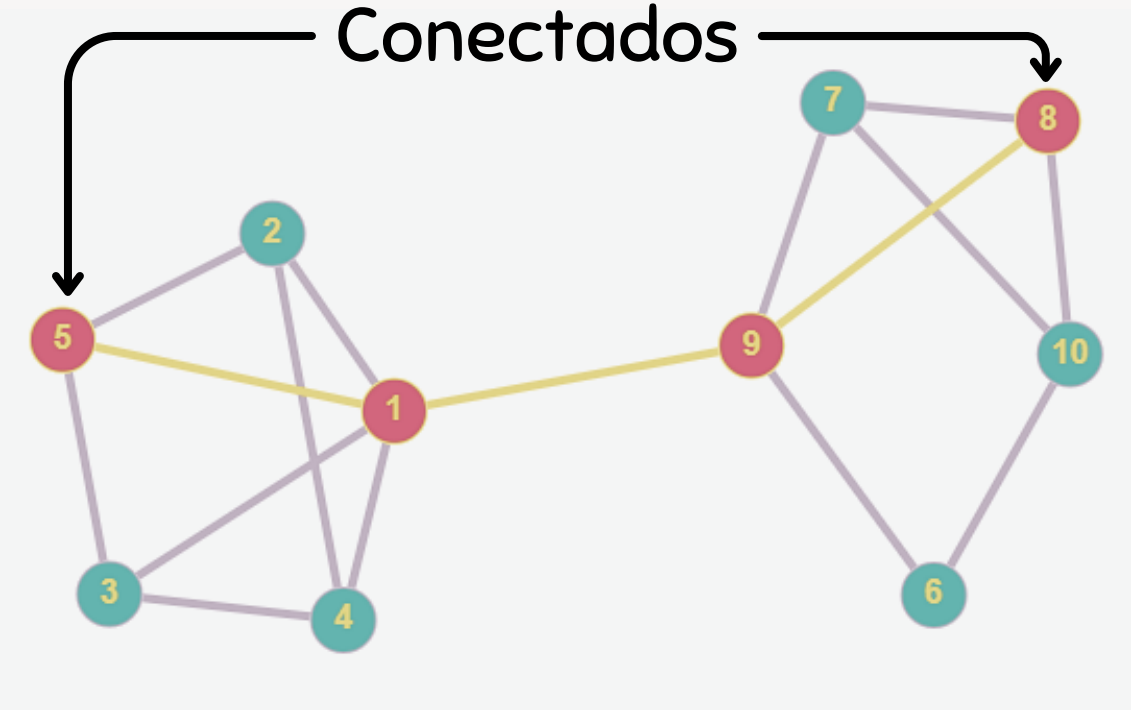


Estrutura básica de grafos



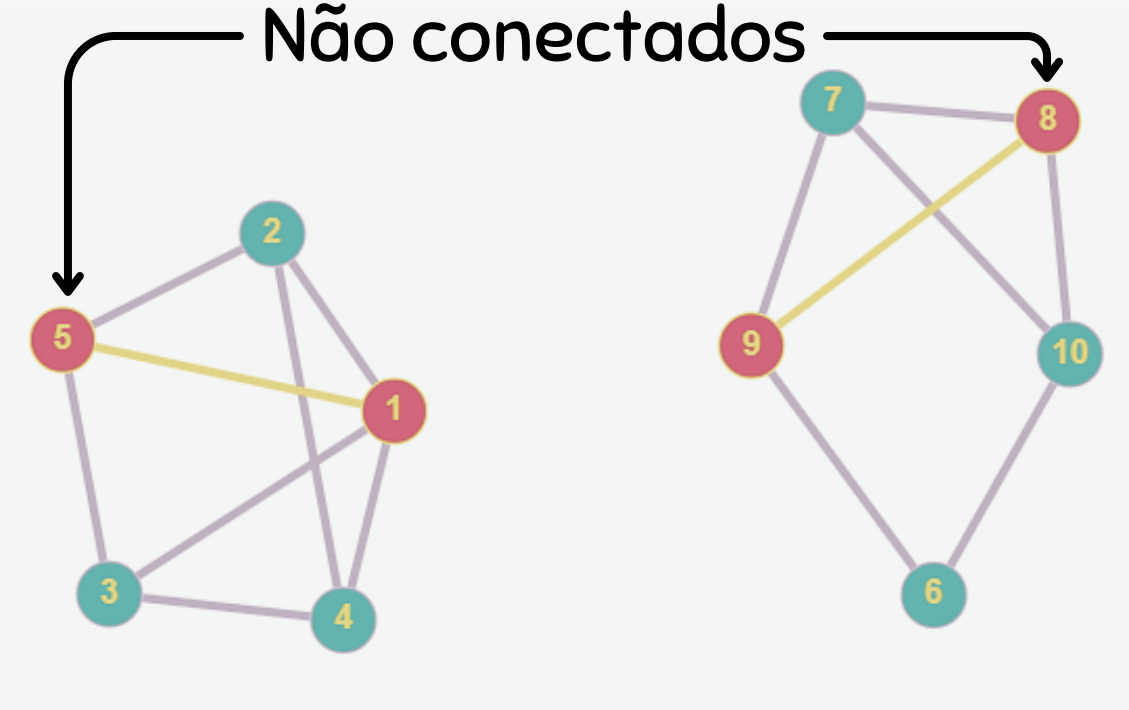
Conectividade

- Conceito relacionado a ligação entre elementos de uma rede
- Mede a robustez de um sistema



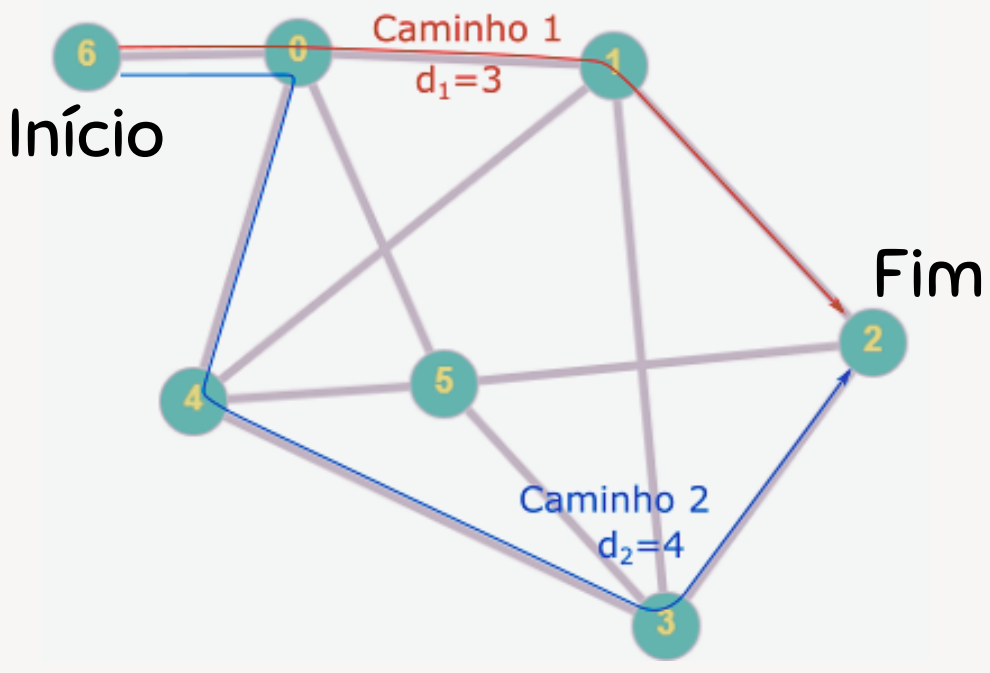
Conectividade

- Conceito relacionado a ligação entre elementos de uma rede
- Mede a robustez de um sistema



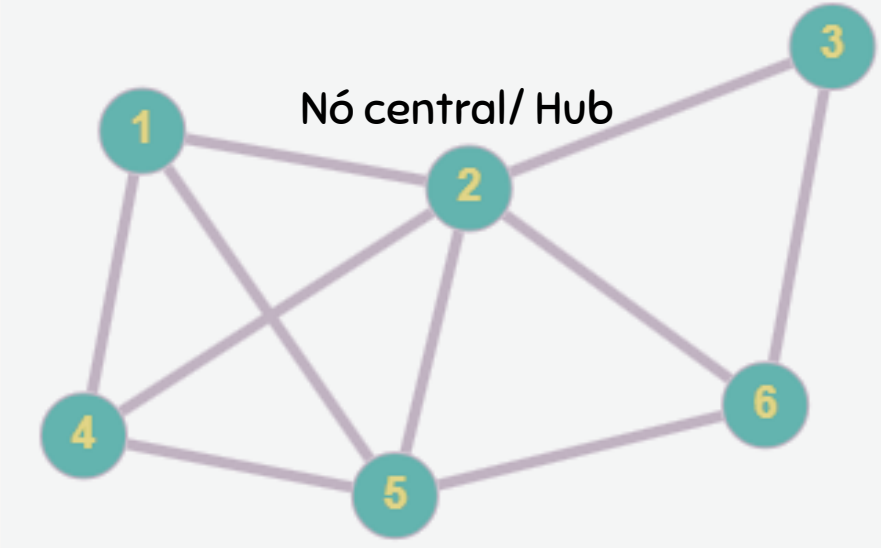
Conectividade

- Fortemente ligado a teoria dos problemas de fluxo
  - Medições de distância e caminho



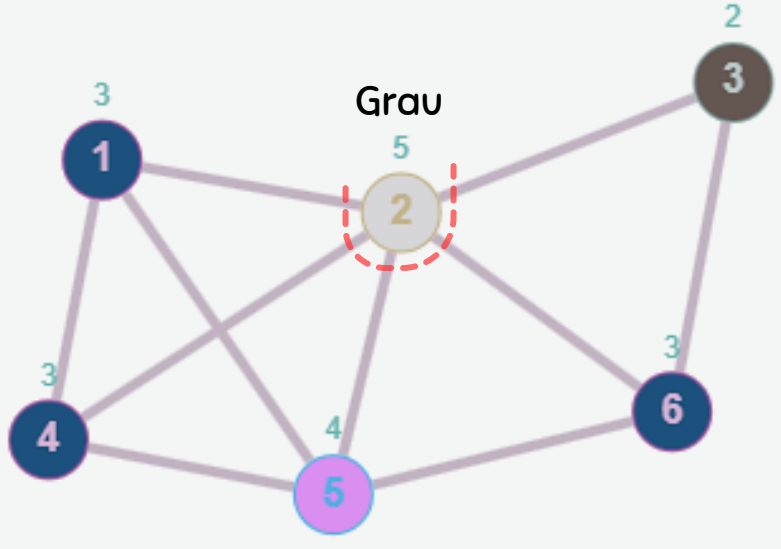
Centralidade

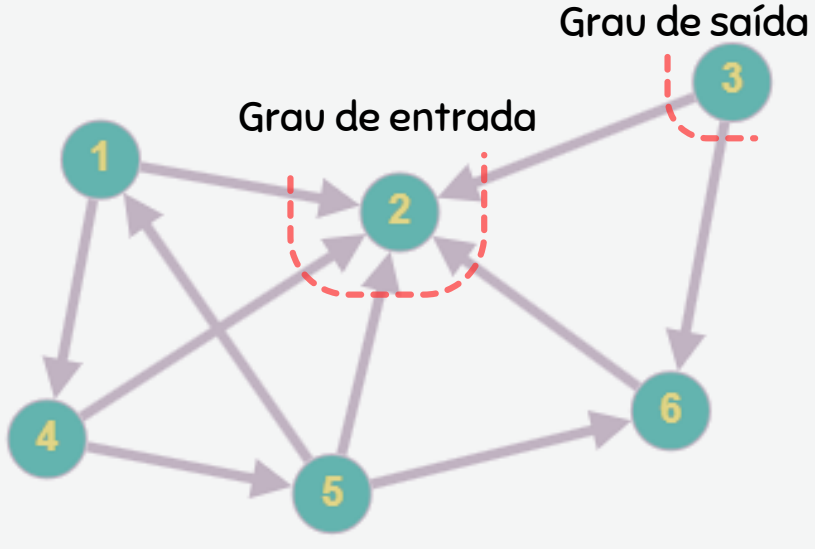
- Conceito que mede a influência das conexões de uma rede
- Permite a estimação de importância de um elemento dentro de um sistema



Centralidade de grau

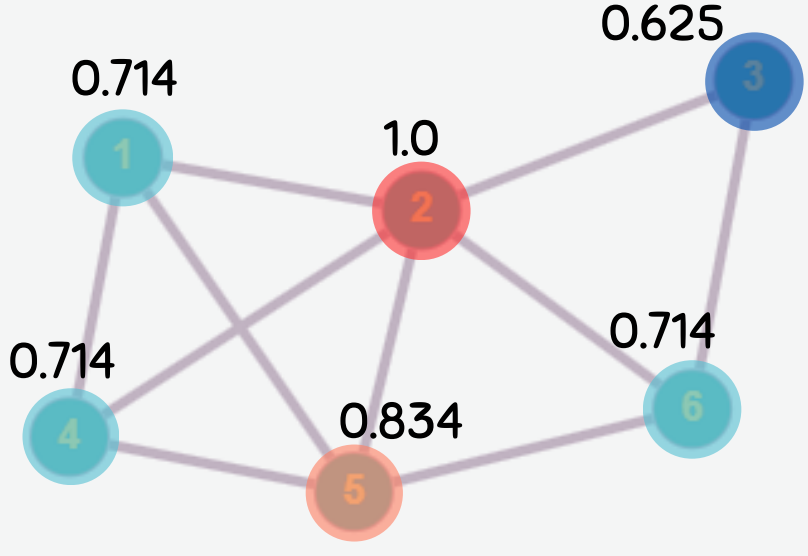
- Mede o número de ligações associados a cada elemento de uma rede
  - Aumento da chance de um elemento receber informações do sistema





Centralidade de proximidade

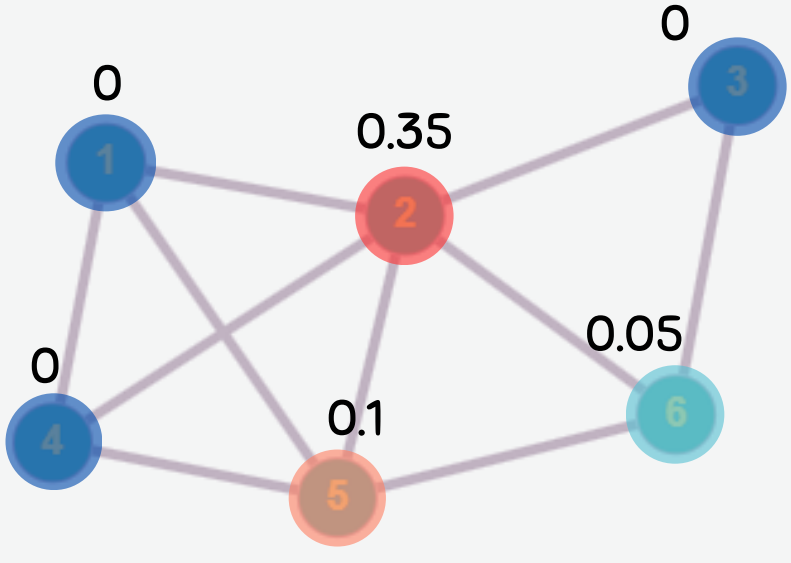
- Mede a menor distância entre todos pares de elementos de uma rede
  - Identifica o elemento com maior facilidade de transmitir informações



$$C(u) = \frac{n-1}{\sum_{v=1}^{n-1} d(v,u)}$$

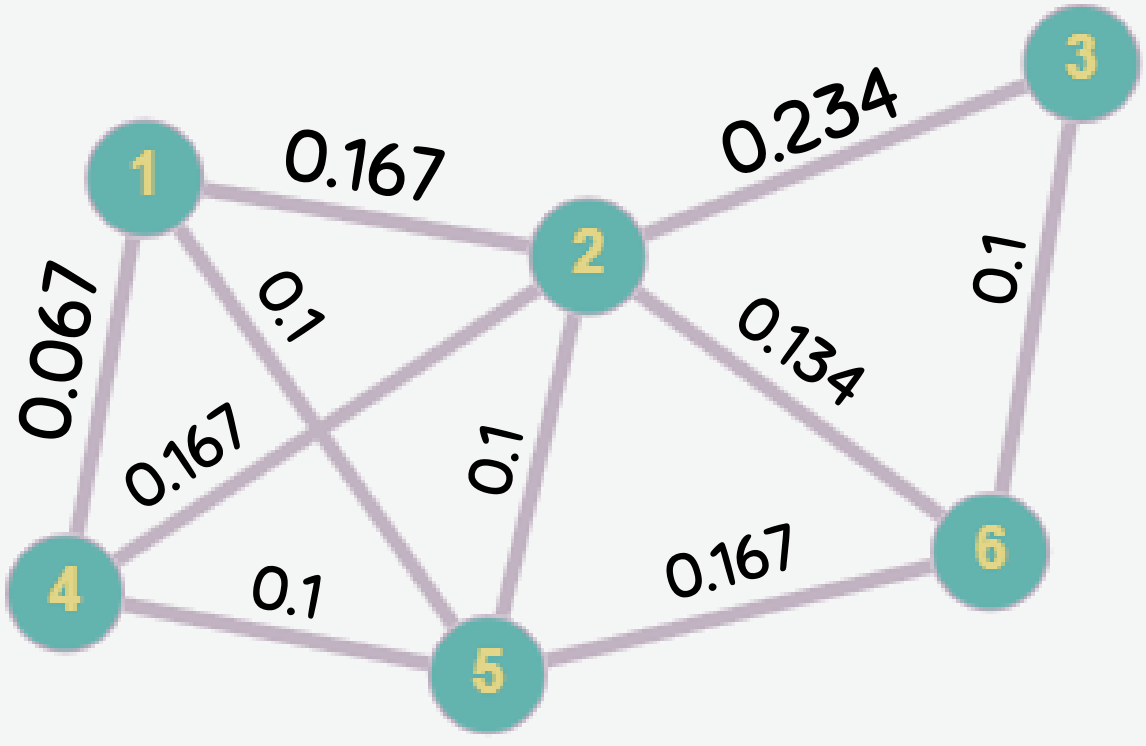
Centralidade de intermediação

- Mede o número de menores caminhos que atravessam um elemento/conexão
  - Identifica as estruturas principais envolvidas na transmissão de informação



$$c_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}$$

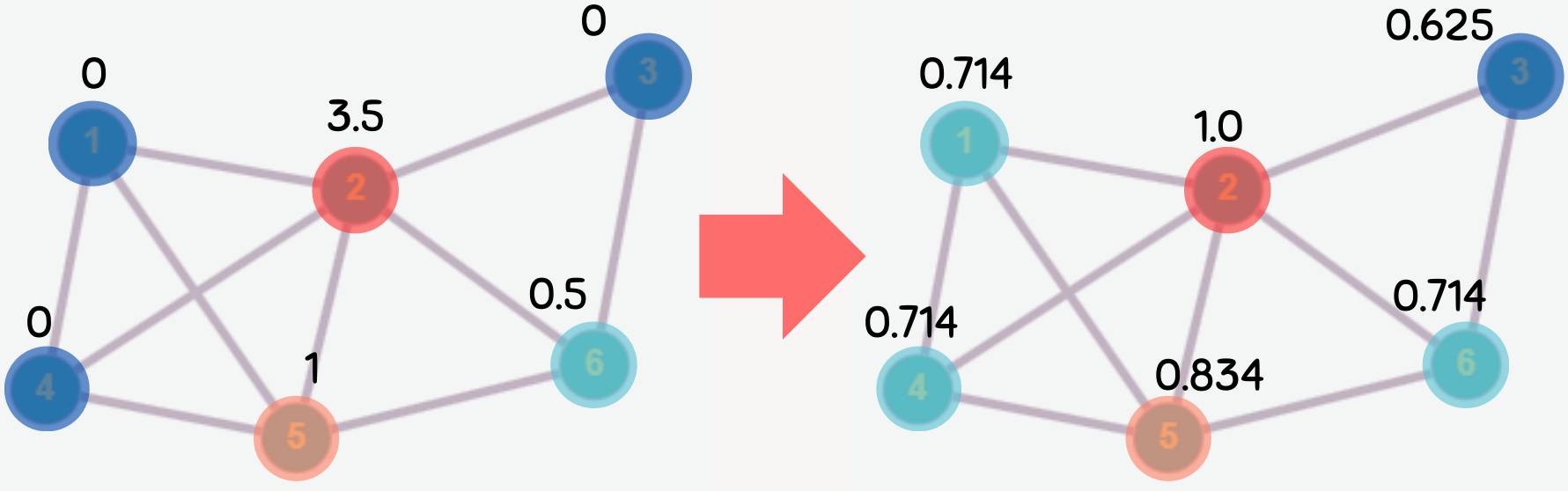
Centralidade de intermediação



$$c_B(e) = \sum_{s,t \in V} \frac{\sigma(s,t|e)}{\sigma(s,t)}$$

Centralidade de intermediação

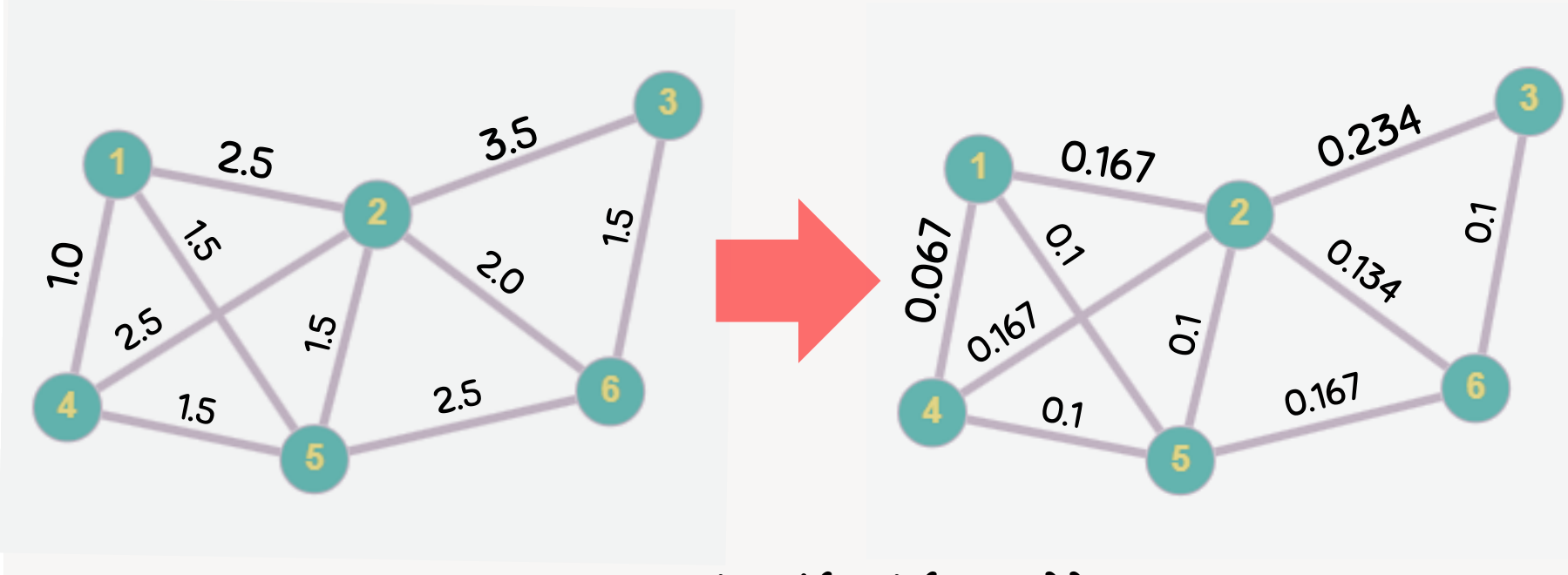
Coeficientes de intermediação não-normalizados



$$Gn = G*2/((N-1)(N-2))$$

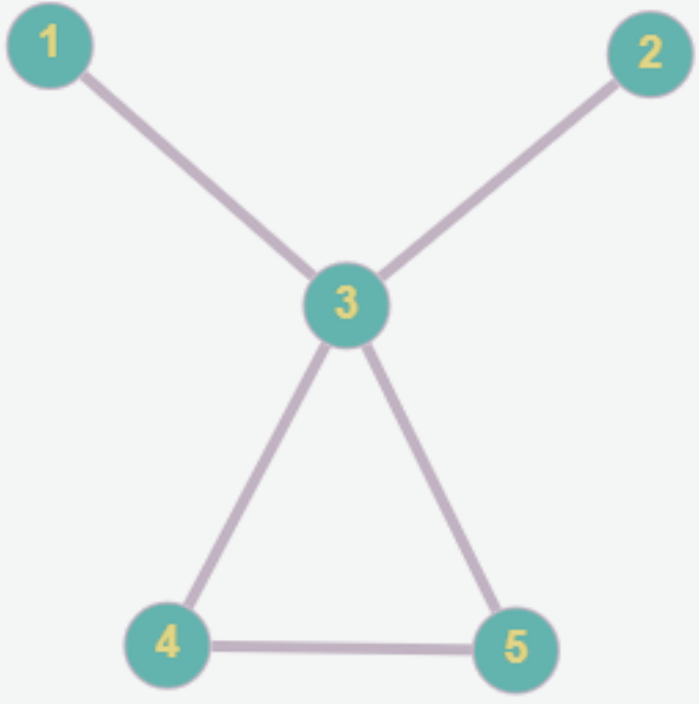
Centralidade de intermediação

Coeficientes de intermediação não-normalizados



$$Gn = G*2/(N*(N-1))$$

Matriz e lista de adjacência



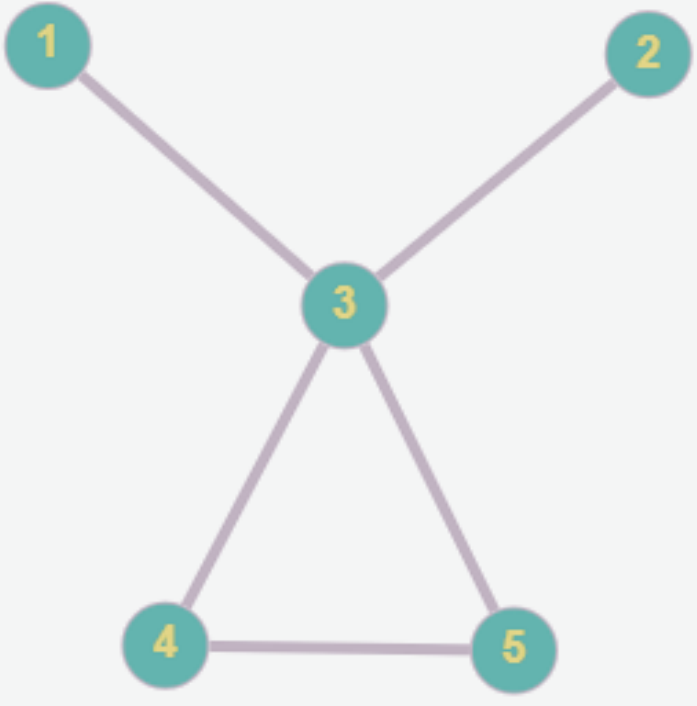
i\j	1	2	3	4	5
1	0	1	0	1	1
2	1	0	1	1	1
3	0	1	0	0	0
4	1	1	0	0	1
5	1	1	0	1	0

Matriz e lista de adjacência

i \ j	1	2	3	4	5
1	0	1	0	1	1
2	1	0	1	1	1
3	0	1	0	0	0
4	1	1	0	0	1
5	1	1	0	1	0

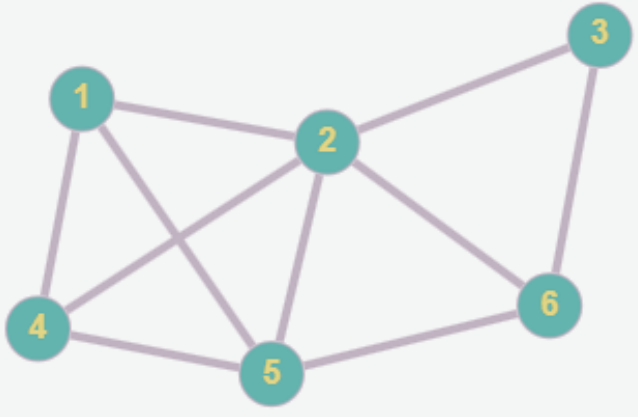
Origem	Destino	Valor
1	2	1
1	3	0
1	4	1
1	5	1
2	3	1
2	4	1
2	5	1
3	4	0
3	5	0
4	5	1

Matriz e lista de adjacência

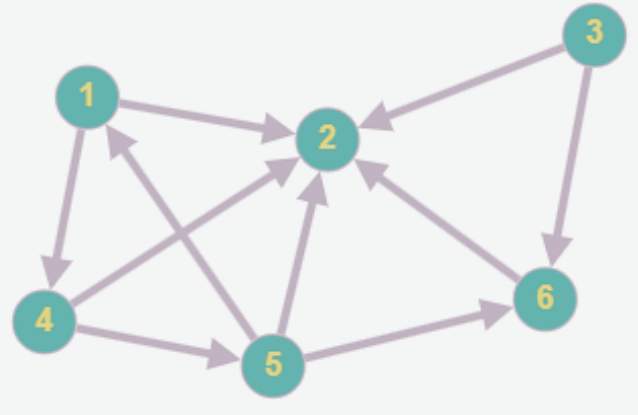


Origem	Destino	Valor
1	2	1
1	3	0
1	4	1
1	5	1
2	3	1
2	4	1
2	5	1
3	4	0
3	5	0
4	5	1

Matriz e lista de adjacência



i\j	1	2	3	4	5	6
1	0	1	0	1	1	0
2	1	0	1	1	1	1
3	0	1	0	0	0	1
4	1	1	0	0	1	0
5	1	1	0	1	0	1
6	0	1	1	0	1	0



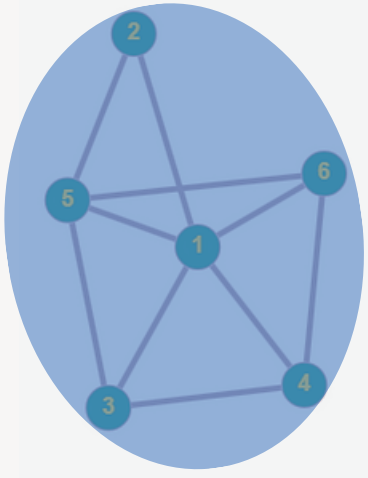
i\j	1	2	3	4	5	6
1	0	1	0	1	0	0
2	0	0	0	0	0	0
3	0	1	0	0	0	1
4	0	1	0	0	1	0
5	1	1	0	1	0	1
6	0	1	0	0	0	0

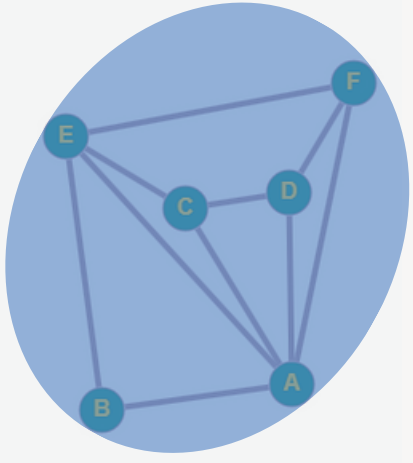
Matriz e lista de adjacência

i\j	1	2	3	4	5	6
1	0	1	0	1	0	0
2	0	0	0	0	0	0
3	0	1	0	0	0	1
4	0	1	0	0	1	0
5	1	1	0	1	0	1
6	0	1	0	0	0	0

Origem	Destino	Valor
1	2	1
1	3	0
...		
2	1	0
2	3	0
...		
3	1	0
3	2	1
...		

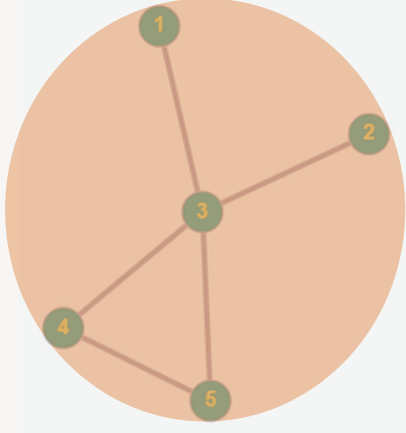
Isomorfismo entre grafos

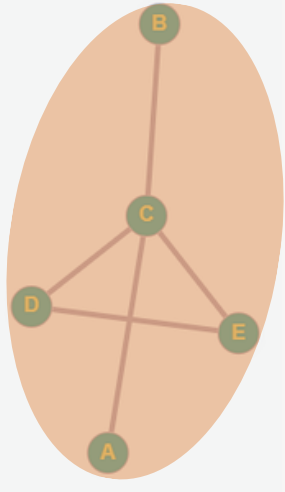




0	1	1	1	1	1
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	0	1
1	1	1	0	0	1
1	0	0	1	1	0

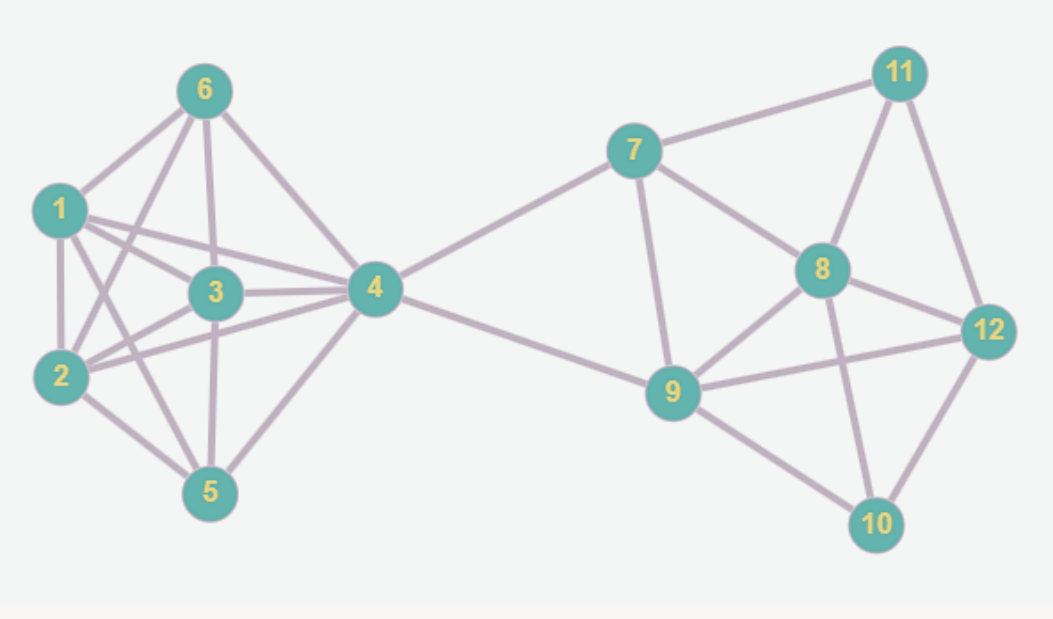
0	0	1	0	0
0	0	1	0	0
1	1	0	1	1
0	0	1	0	1
0	0	1	1	0





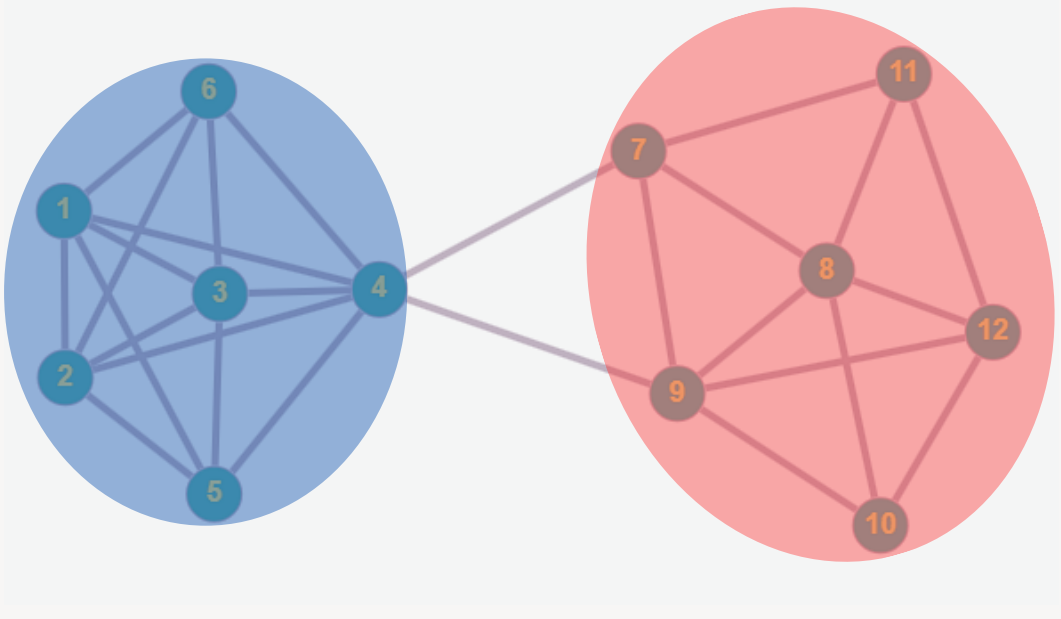
Partição de grafos

- Problema de teoria de grafos
- Capacidade de isolar um grafo em componentes menores
    - Compartilhamento de características

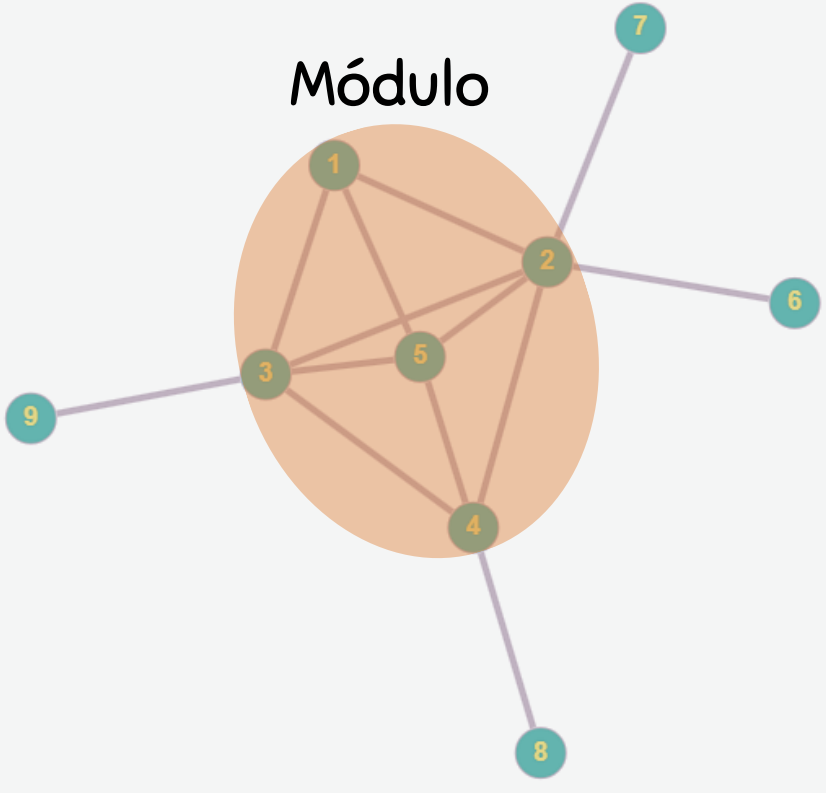


Partição de grafos

- Problema de teoria de grafos
- Capacidade de isolar um grafo em componentes menores
    - Compartilhamento de características

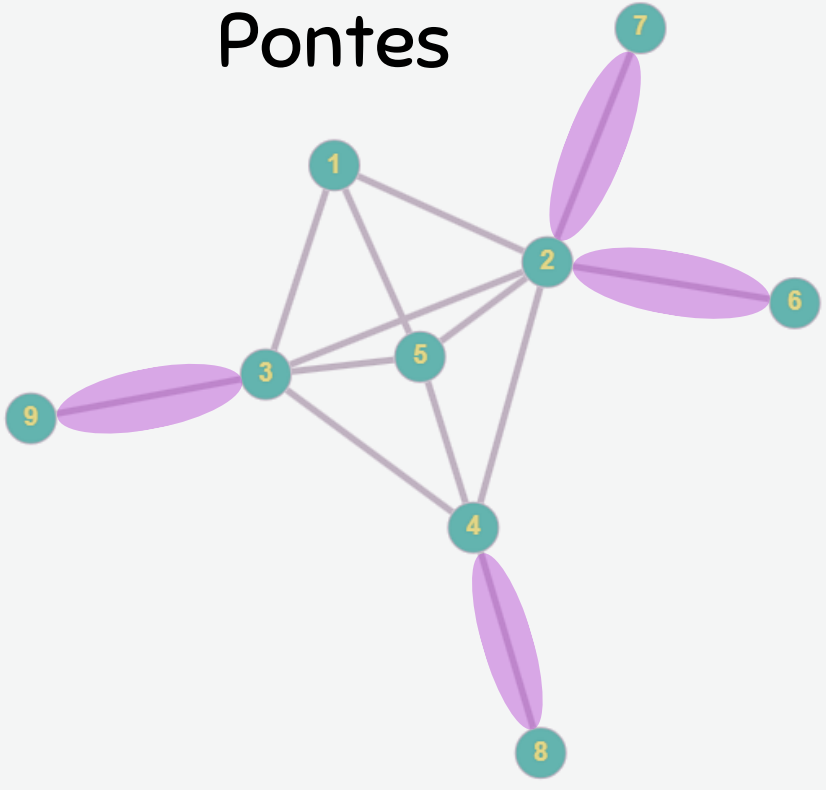


Partição de grafos



i\j	1	2	3	4	5	6	7	8	9
1	0	1	1	0	1	0	0	0	0
2	1	0	1	1	1	1	1	0	0
3	1	1	0	1	1	0	0	0	1
4	0	1	1	0	1	0	0	1	0
5	1	1	1	1	0	0	0	0	0
6	0	1	0	0	0	0	0	0	0
7	0	1	0	0	0	0	0	0	0
8	0	0	0	1	0	0	0	0	0
9	0	0	1	0	0	0	0	0	0

Partição de grafos



i\j	1	2	3	4	5	6	7	8	9
1	0	1	1	0	1	0	0	0	0
2	1	0	1	1	1	1	1	0	0
3	1	1	0	1	1	0	0	0	1
4	0	1	1	0	1	0	0	1	0
5	1	1	1	1	0	0	0	0	0
6	0	1	0	0	0	0	0	0	0
7	0	1	0	0	0	0	0	0	0
8	0	0	0	1	0	0	0	0	0
9	0	0	1	0	0	0	0	0	0

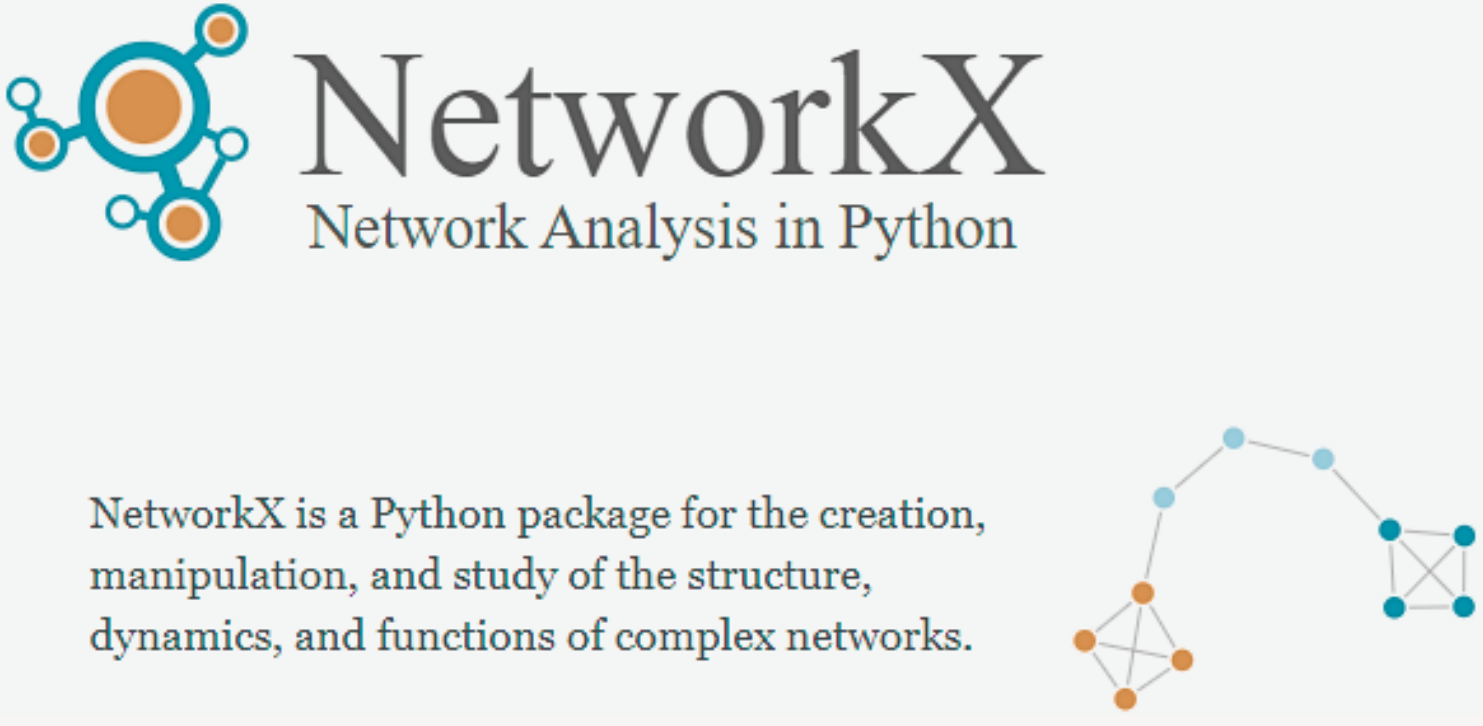
NetworkX

NetworkX
Network Analysis in Python

NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.

- Algoritmos para análise de métricas de grafos
- Métodos de visualização e layouts para redes

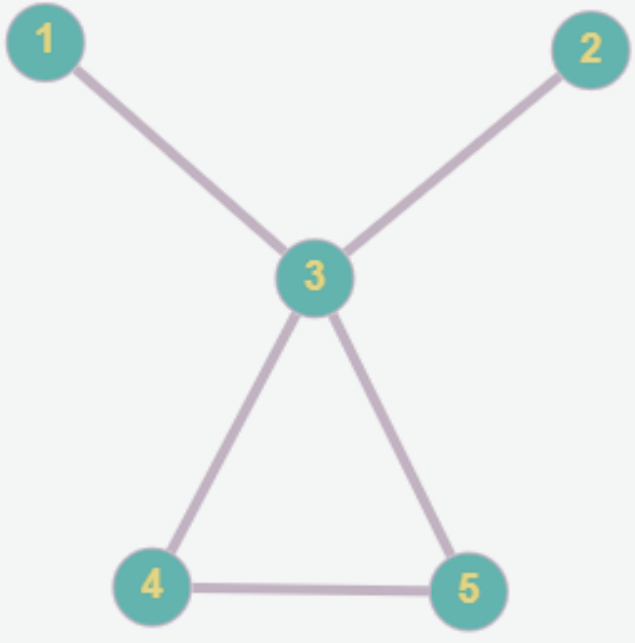
NetworkX




```
$ pip install networkx[default]
```

Manipulação de grafos no NetworkX

G




```
>>> import networkx as nx
>>> G = nx.Graph()


>>> G.add_edge(1, 3)
>>> G.add_edge(2, 3)

...
```

Manipulação de grafos no NetworkX

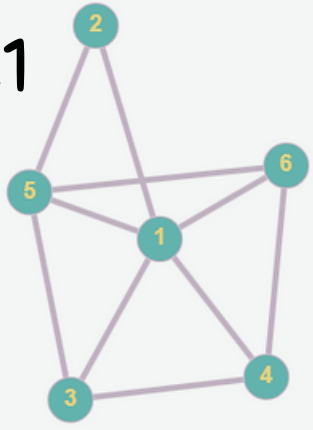
```
>>> G.edges()
EdgeView([(1, 3), (3, 2), (3, 4), (3, 5), (4, 5)])

>>> G.nodes()
NodeView((1, 3, 2, 4, 5))

>>> G.degree()
DegreeView({1: 1, 3: 4, 2: 1, 4: 2, 5: 2})
```

```
>>> nx.degree_centrality(G)
EdgeView([(1, 3), (3, 2), (3, 4), (3, 5), (4, 5)])


>>> nx.closeness_centrality(G)
{1: 0.5714285714285714,
 3: 1.0,
 2: 0.5714285714285714,
 4: 0.6666666666666666,
 5: 0.6666666666666666}
```

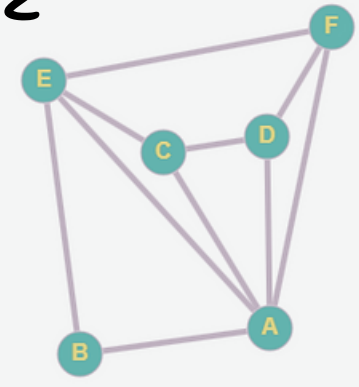
```
>>> nx.betweenness_centrality(G)
{1: 0.0, 3: 0.8333333333333, 2: 0.0, 4: 0.0, 5: 0.0}

>>> nx.edge_betweenness_centrality(G)
{(1, 3): 0.4,
 (3, 2): 0.4,
 (3, 4): 0.30000000000000004,
 (3, 5): 0.30000000000000004,
 (4, 5): 0.1}
```

Manipulação de grafos no NetworkX

C1

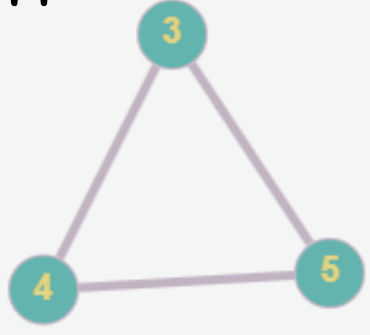


C2




```
>>> nx.is_isomorphic(C1, C2)
True
```

H

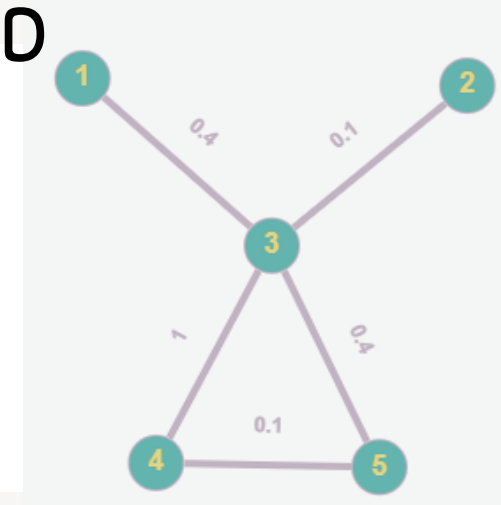



```
>>> H = G.subgraph([3, 4, 5])
```

Manipulação de grafos no NetworkX

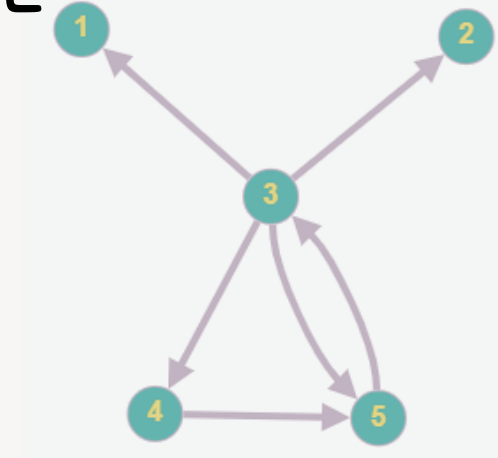
```
>>> D = nx.Graph()
>>> D.add_edge(3, 5, weight = 0.4)
>>> D.add_edge(4, 5, weight = 0.1)
...
```

D




```
>>> E = nx.DiGraph()
>>> E.add_edge(3, 5)
>>> E.add_edge(5, 3)
...
```

E

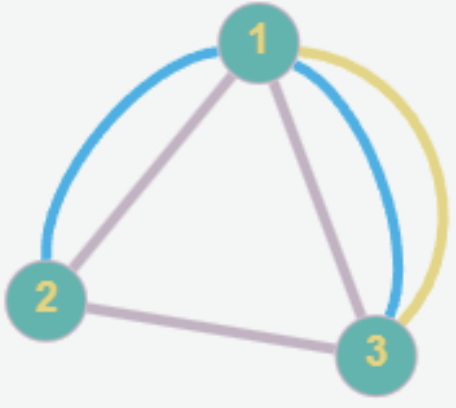


Manipulação de grafos no NetworkX

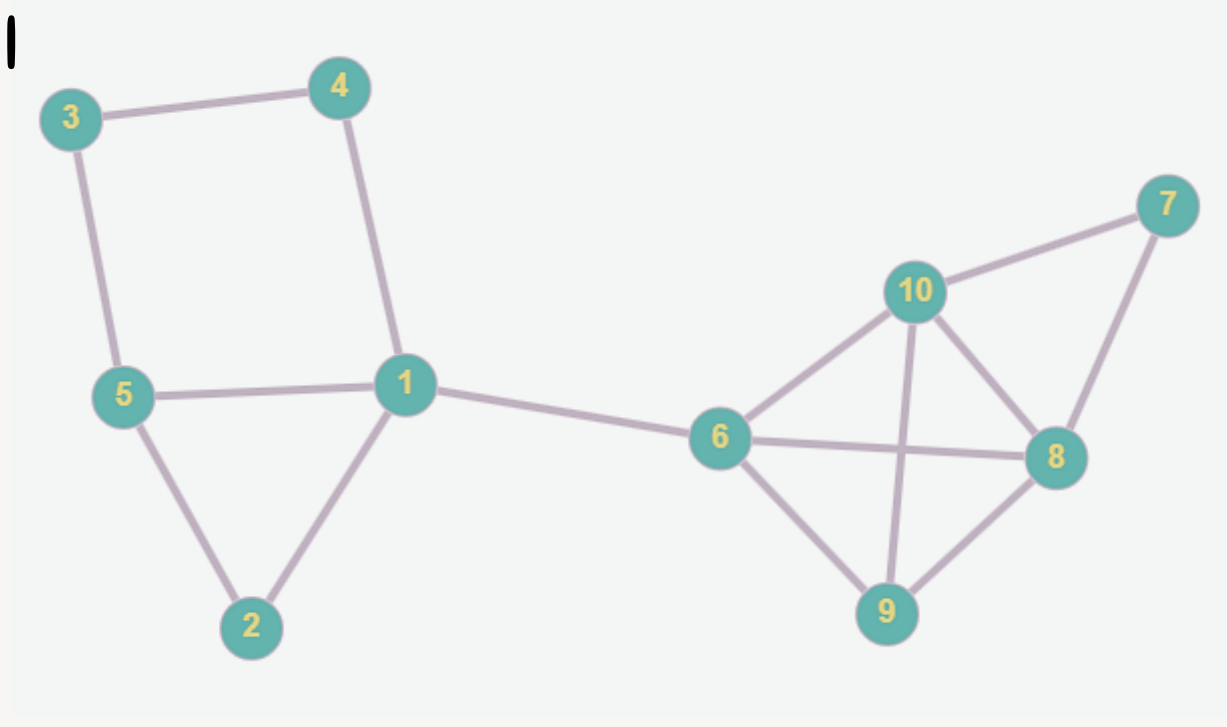
```
>>> F = nx.MultiGraph()
>>> F.add_edge(1, 2, color = "gray")
>>> F.add_edge(1, 2, color = "blue")
...
```

- Armazenamento de atributos:
  - Dicionário

F



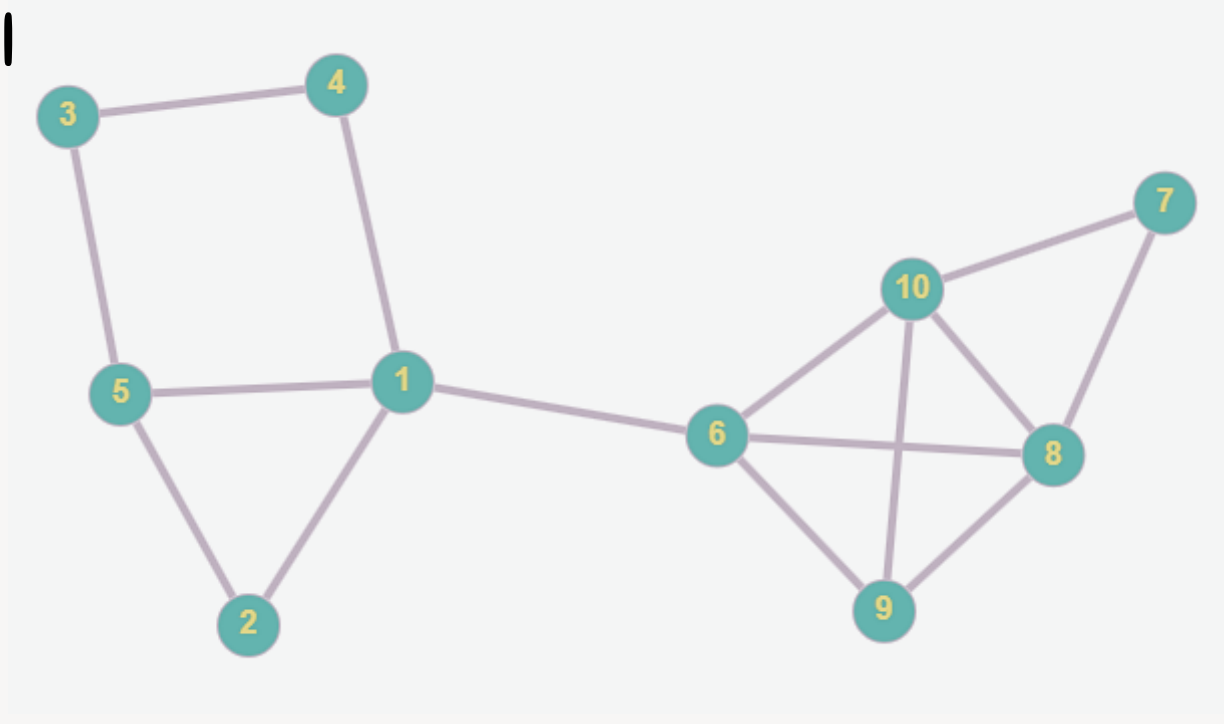
Manipulação de grafos no NetworkX




```
>>> nx.has_bridges(I)
True
```

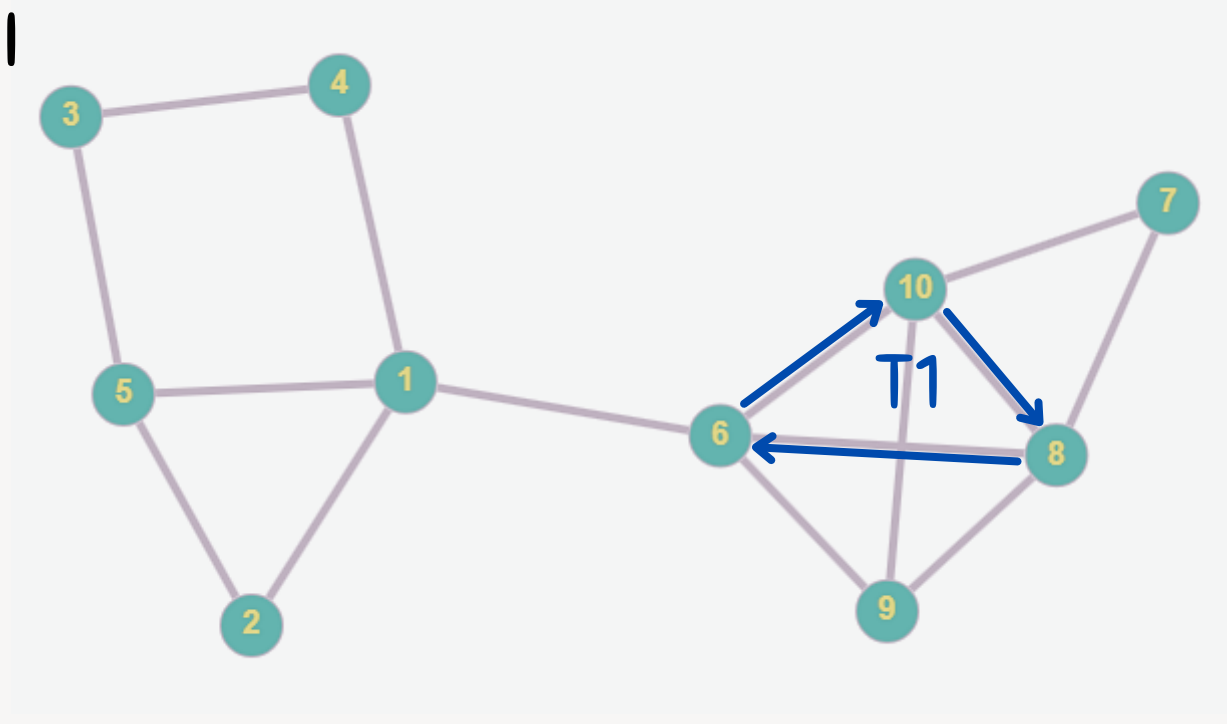
```
>>> list(nx.bridges(I))
[(1, 6)]
```

Coeficiente de agrupamento



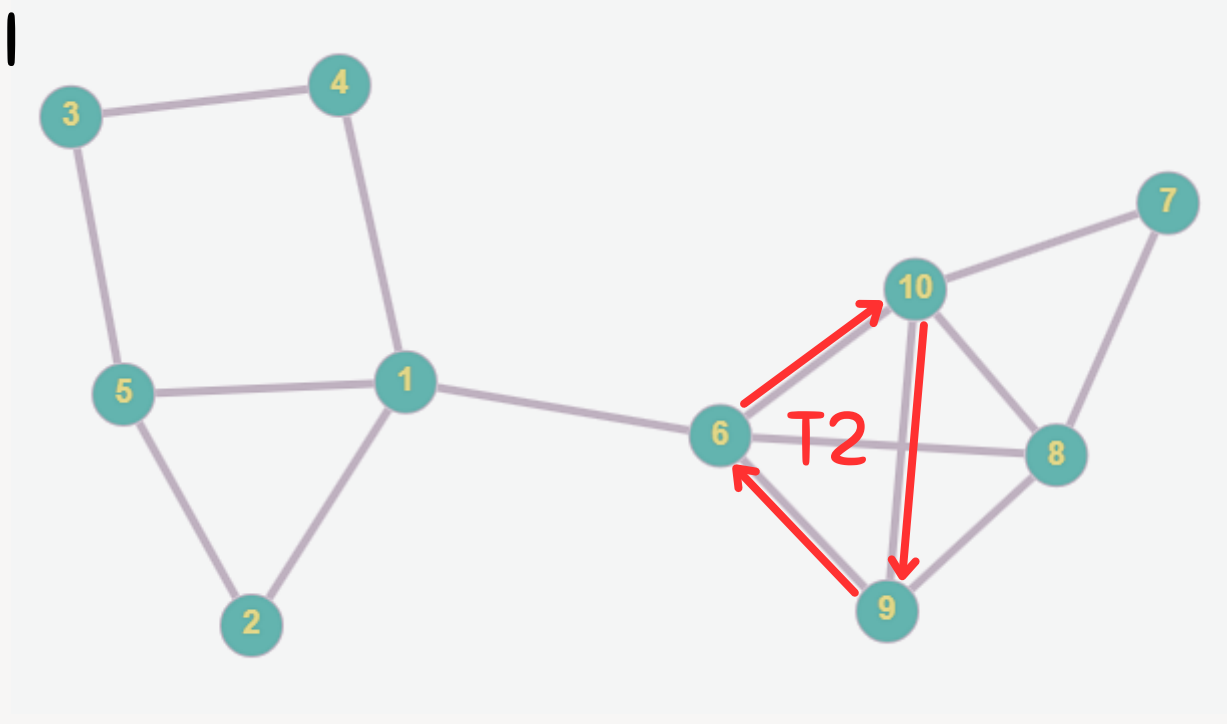
$$c_u = \frac{2T(u)}{deg(u)(deg(u)-1)}$$

Coeficiente de agrupamento



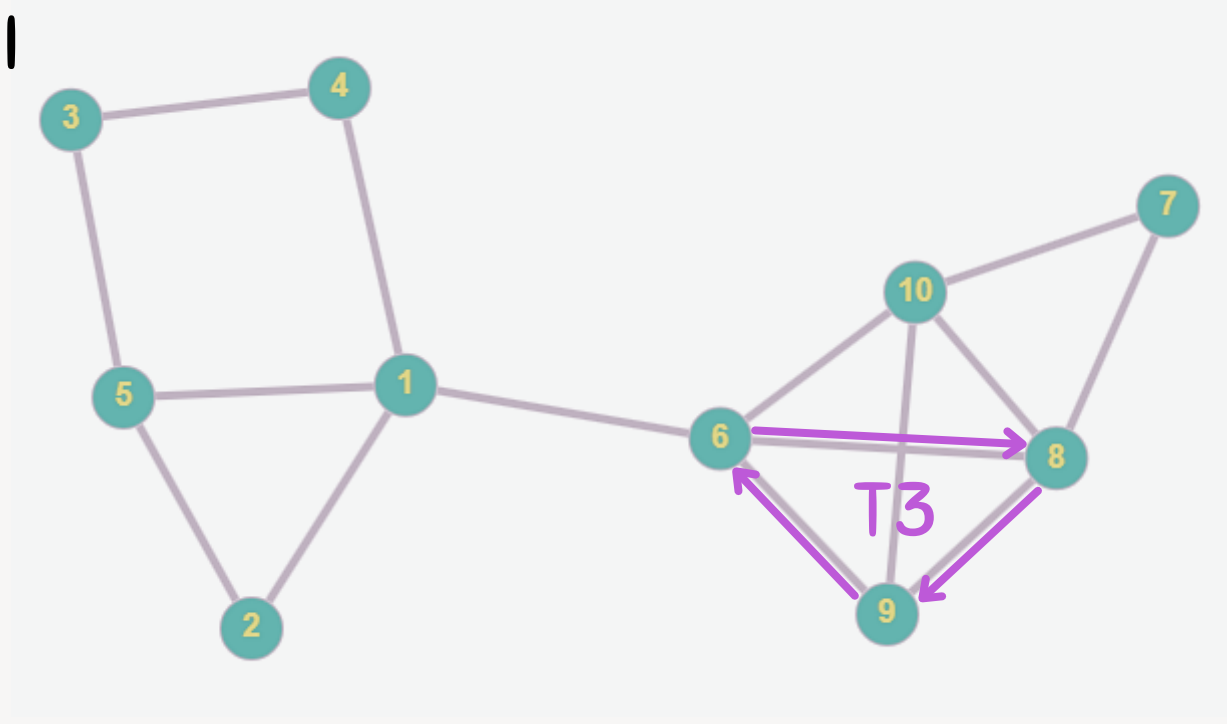
$$c_u = \frac{2T(u)}{deg(u)(deg(u)-1)}$$

Coeficiente de agrupamento



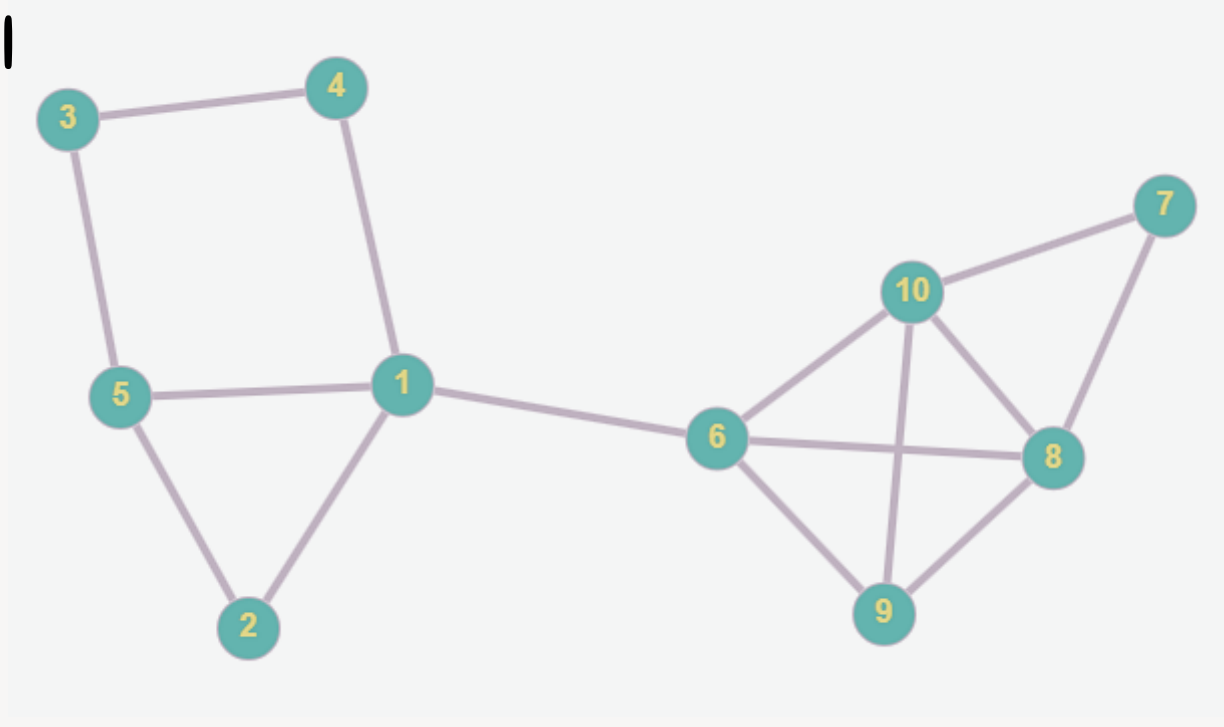
$$c_u = \frac{2T(u)}{deg(u)(deg(u)-1)}$$

Coeficiente de agrupamento



$$c_u = \frac{2T(u)}{deg(u)(deg(u)-1)}$$

Manipulação de grafos no NetworkX



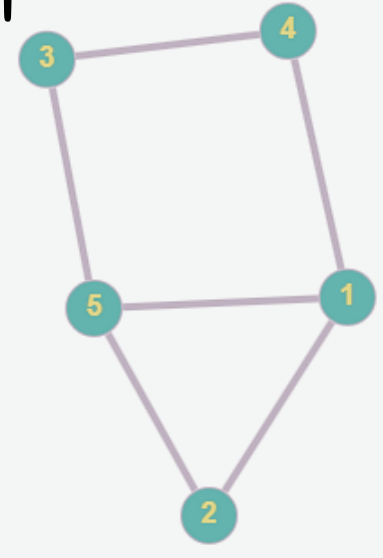
$$c_u = \frac{2T(u)}{deg(u)(deg(u)-1)}$$

```
>>> nx.clustering(I, 6)
0.5
```

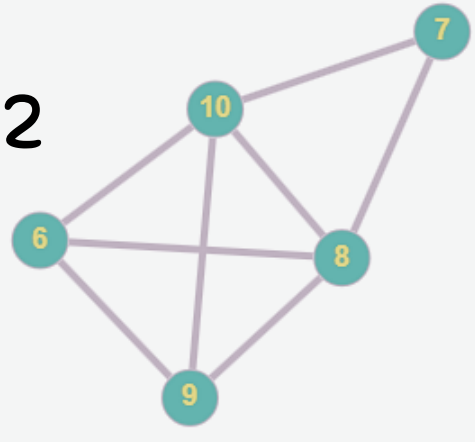
Manipulação de grafos no NetworkX

```
>>> from networkx.algorithms.community.centrality import girvan_newman
>>> comp = girvan_newman(I)
>>> tuple(sorted(c) for c in next(comp))
([1, 2, 3, 4, 5], [6, 7, 8, 9, 10])
```

I1

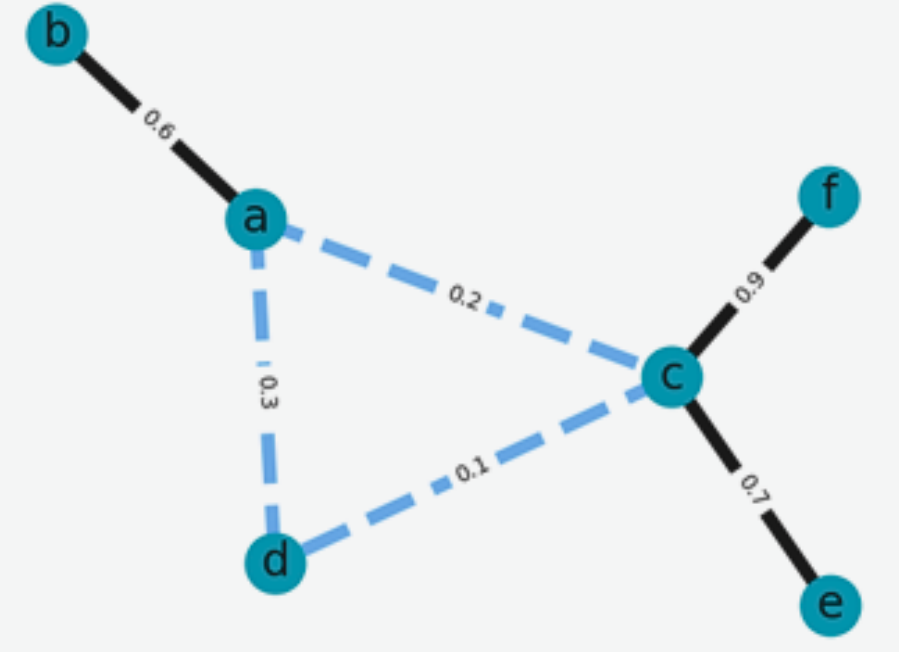


I2

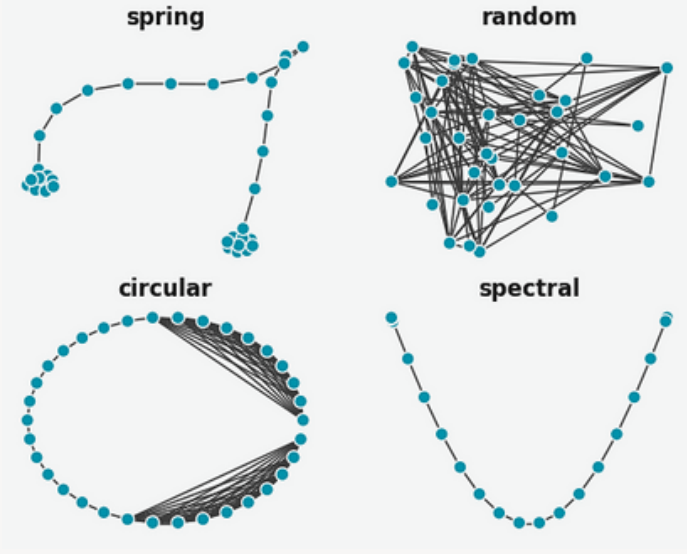


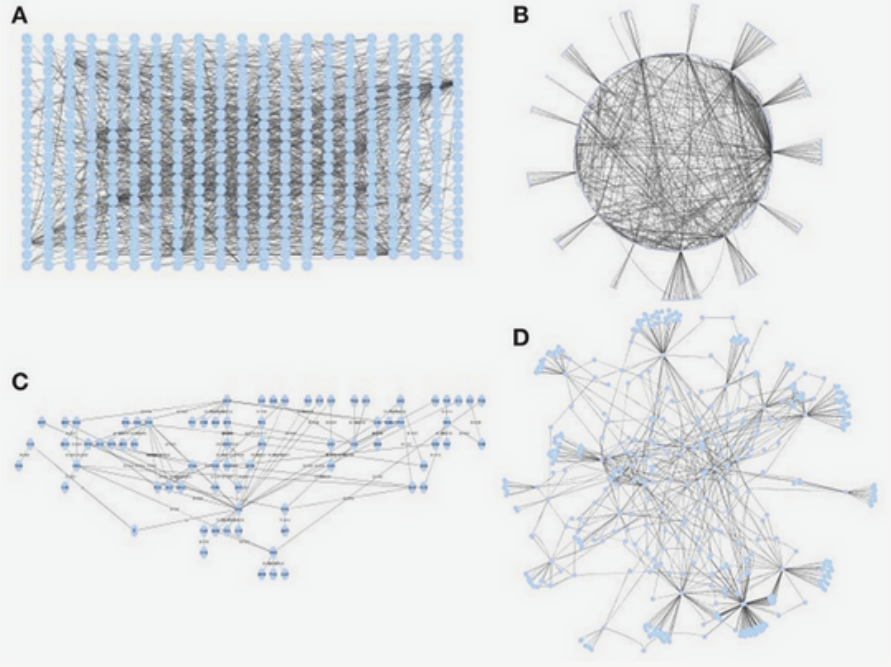
Modelagem de grafos no NetworkX

```
>>> nx.draw(G)
>>> nx.draw_networkx_nodes(G, pos)
>>>  x.draw_networkx_edges(G, pos)
>>> nx.draw_networkx_labels(G, pos)
>>> nx.draw_networkx_edge_labels(G, pos)
```



Layout da rede





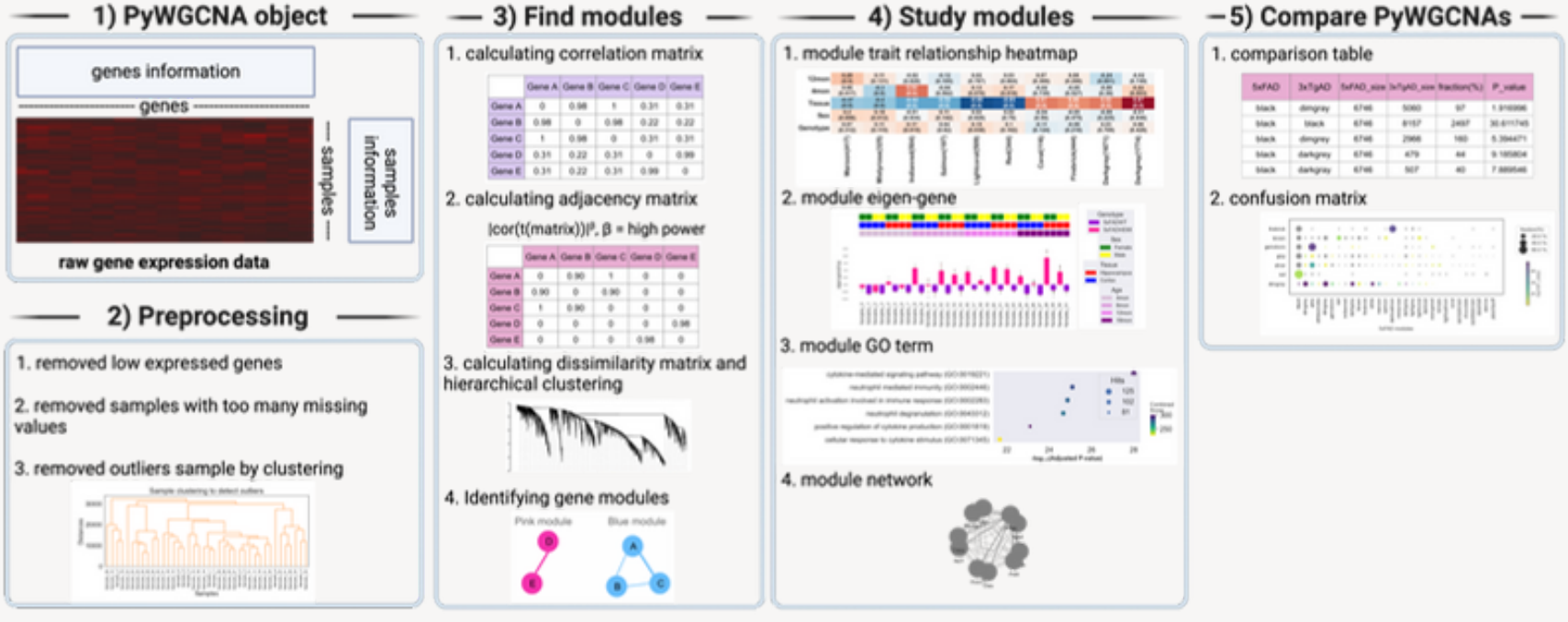
pyWGCNA

JOURNAL ARTICLE

PyWGCNA: a Python package for weighted gene co-expression network analysis

Narges Rezaie, Farilie Reese, Ali Mortazavi

Bioinformatics, Volume 39, Issue 7, July 2023, btad415,
https://doi.org/10.1093/bioinformatics/btad415
Published: 03 July 2023 **Article history** ▾




```
$ pip install pyWGCNA
```