



Licenciatura em Ciência de Dados - 3º ano

Trabalho 2

Análise de Redes

6 de janeiro de 2025

Discentes: João Dias nº 110305 / Felipe Pereira nº 110861 /
David Franco nº 110733 / Diogo Aqueu nº 110705

2024/2025

ÍNDICE

| | |
|--|-----------|
| Introdução | 2 |
| Método de Remoção de Pontes | 2 |
| Método de Otimização de Modularidade (Fast Greedy) | 8 |
| Método de Otimização de Louvain | 13 |
| Método de Propagação de Etiquetas | 17 |
| Comparação entre os Métodos | 20 |
| Apêndice | 22 |
| Apêndice I - Importação das bibliotecas necessárias | 22 |
| Apêndice II - Importação da rede através do ficheiro disponibilizado no Moodle | 22 |
| Apêndice III - Plot visual da rede | 22 |
| Apêndice IV - Aplicação do método de remoção de pontes | 23 |
| Apêndice V - Dimensão das comunidades com o método de remoção de pontes | 23 |
| Apêndice VI - Ligações internas, externas e densidade interna do método de remoção de pontes | 24 |
| Apêndice VII - Modularidade do método de remoção de pontes | 25 |
| Apêndice VIII - Aplicação do método de otimização de modularidade (fast greedy) | 25 |
| Apêndice IX - Dimensão das comunidades com o método de otimização de modularidade (fast greedy) | 26 |
| Apêndice X - Ligações internas, externas e densidade interna do método de otimização de modularidade (fast greedy) | 27 |
| Apêndice XI - Modularidade do método de otimização da modularidade (fast greedy) | 28 |
| Apêndice XII - Aplicação do método de otimização de Louvain | 28 |
| Apêndice XIII - Dimensão das comunidades com o método de otimização de Louvain | 29 |
| Apêndice XIV - Ligações internas, externas e densidade interna do método de otimização de Louvain | 30 |
| Apêndice XV - Modularidade do método de otimização de Louvain | 31 |
| Apêndice XVI - Aplicação do método de propagação de etiquetas | 31 |
| Apêndice XVII - Dimensão das comunidades com o método de propagação de etiquetas | 32 |
| Apêndice XVIII - Ligações internas, externas e densidade interna do método de propagação de etiquetas | 32 |
| Apêndice XIX - Modularidade do método de propagação de etiquetas | 33 |

Introdução

Este projeto analisa métodos de detecção de comunidades em redes, onde os nodos representam pessoas e as arestas representam os contactos presenciais entre elas. São explorados os métodos de Remoção de Pontes, Otimização de Modularidade (Fast Greedy e Louvain) e Propagação de Etiquetas, com o objetivo de identificar subredes, avaliar o tamanho das comunidades, a coesão interna de cada grupo e a modularidade. A comparação entre os métodos visa determinar qual a abordagem que segmenta a rede de forma mais eficaz, considerando a densidade e o número de ligações internas e externas, bem como a qualidade da separação entre as comunidades. O estudo procura, assim, compreender melhor a estrutura das interações e relações entre as pessoas na rede fornecida.

Método de Remoção de Pontes

O método de remoção de pontes é uma abordagem eficaz para a detecção de comunidades em redes. Este método baseia-se na identificação de arestas que atuam como pontes entre diferentes partes da rede. Inicialmente, calcula-se a importância de cada aresta com base em métricas como a centralidade de intermediação. Em seguida, as arestas com maior pontuação são progressivamente removidas, o que resulta na fragmentação da rede em componentes menores. O processo continua até que as comunidades desejadas sejam formadas. No contexto desta rede, os nodos representam pessoas e as ligações representam contactos presenciais entre elas.

Na nossa rede, o método de remoção de pontes conseguiu identificar 11 comunidades distintas, o que demonstra a existência de subgrupos bem definidos. Isso significa que os 51 nodos da rede não estão interligados de forma homogénea, mas sim organizados em agrupamentos menores com padrões próprios de interação.

As comunidades variam em tamanho, desde a maior, composta por 17 nodos, até outras muito pequenas, com apenas um ou dois nodos. Esses resultados refletem a presença de interações mais concentradas dentro de cada grupo e conexões menos frequentes entre os diferentes grupos, destacando a estrutura modular da rede, conforme pode ser visto na Figura 1.

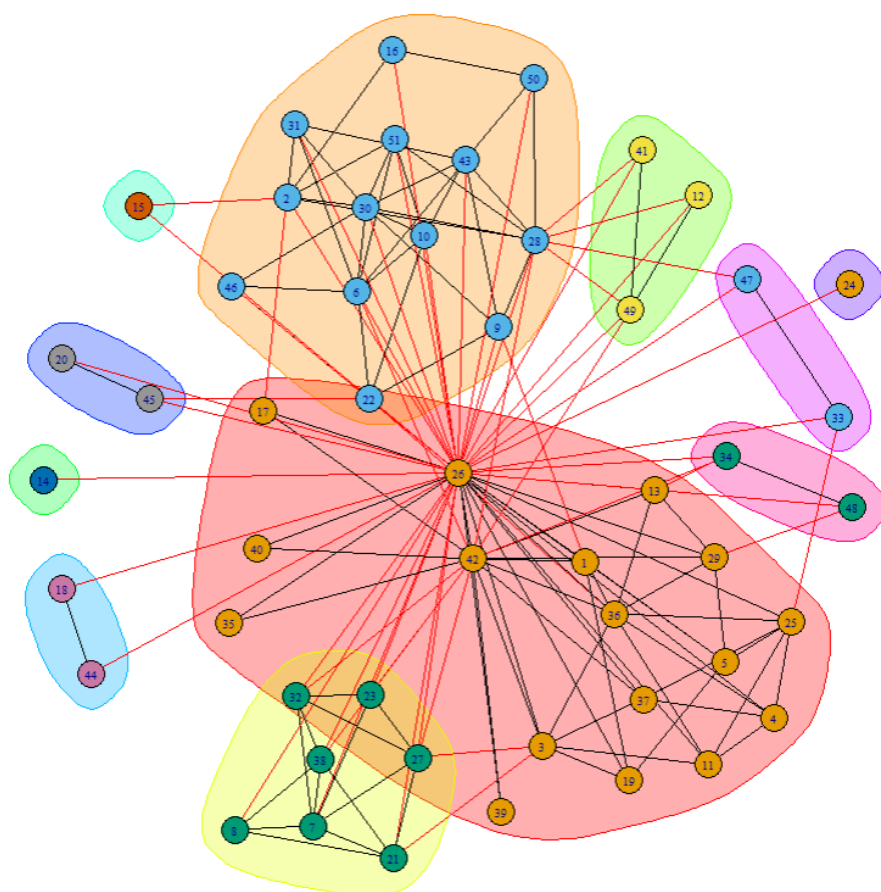


Figura 1 - Representação obtida pelo método de remoção de pontes

i. Número de comunidades obtidas:

O método de remoção de pontes identificou 11 comunidades na rede. Este número reflete a fragmentação da rede em subgrupos baseados nas arestas com maior centralidade de intermediação, que foram removidas para separar os nodos em grupos mais coesos internamente e com menos conexões externas.

ii. Dimensão de cada comunidade:

As comunidades identificadas possuem tamanhos variados, sendo a maior composta por 17 nodos e as menores por apenas um nodo cada. A distribuição dos nodos pelas comunidades pode ser vista na Tabela 1.

| Comunidade | Nº de nodos |
|------------|-------------|
| 1 | 17 |
| 2 | 13 |
| 3 | 7 |
| 4 | 3 |
| 5 | 1 |
| 6 | 1 |
| 7 | 2 |
| 8 | 2 |
| 9 | 1 |
| 10 | 2 |
| 11 | 2 |

Tabela 1 - Dimensões das comunidades pelo método de remoção de pontes

Essa distribuição evidencia uma forte hierarquia na rede, com alguns grupos mais densos e outros mais isolados.

iii. Número de ligações internas de cada comunidade e respetiva densidade interna:

Cada comunidade foi avaliada quanto ao número de ligações internas (arestas dentro da própria comunidade) e sua densidade interna (proporção de ligações presentes em relação ao máximo possível). Os resultados encontram-se na Tabela 2.

| Comunidade | Nº de ligações internas | Densidade interna |
|------------|-------------------------|-------------------|
| 1 | 47 | 0,346 |
| 2 | 32 | 0.410 |
| 3 | 14 | 0.667 |
| 4 | 2 | 0.667 |

| | | |
|----|---|-------|
| 5 | 0 | NaN |
| 6 | 0 | NaN |
| 7 | 1 | 1.000 |
| 8 | 1 | 1.000 |
| 9 | 0 | NaN |
| 10 | 1 | 1.000 |
| 11 | 1 | 1.000 |

Tabela 2 - Ligações internas e densidade interna pelo método de remoção de pontes

A Comunidade 3, com a maior densidade interna entre os grupos com mais de dois nodos (0.667), indica que as pessoas dentro dessa comunidade têm um número significativo de interações presenciais entre si, o que sugere um grupo de indivíduos bem conectado. Isso pode implicar uma rede forte de contactos, onde as interações são frequentes e as relações dentro do grupo são bem estabelecidas.

As comunidades menores (7, 8, 10 e 11), com densidade interna máxima de 1.000, significam que todas as pessoas dessas comunidades estão em contacto umas com as outras. Este tipo de configuração pode indicar pequenos grupos de indivíduos que têm interações pessoais frequentes.

Por outro lado, as comunidades isoladas (5, 6 e 9), que não possuem ligações internas (com densidade interna nula), indicam que as pessoas nesses grupos não têm interações presenciais entre si.

iv. Número de ligações externas de cada comunidade:

As ligações externas de cada comunidade, ou seja, as conexões com nodos fora da comunidade, estão descritas na Tabela 3.

| Comunidade | Nº de ligações externas |
|------------|-------------------------|
| 1 | 50 |
| 2 | 24 |

| | |
|----|----|
| 3 | 13 |
| 4 | 7 |
| 5 | 1 |
| 6 | 2 |
| 7 | 2 |
| 8 | 3 |
| 9 | 1 |
| 10 | 4 |
| 11 | 5 |

Tabela 3 - Ligações externas pelo método de remoção de pontes

A Comunidade 1, com o maior número de nodos (17), também tem o maior número de ligações externas (50), o que a torna a mais conectada com o resto da rede. Isto sugere que esta comunidade desempenha um papel central, funcionando como um ponto de transição entre diferentes grupos, com indivíduos que mantêm uma grande interação com pessoas de outras comunidades.

Por outro lado, as comunidades 5 e 9, com apenas 1 nodo cada, possuem apenas uma ligação externa, o que as coloca em posições periféricas. A falta de conexões externas indica que essas comunidades constituídas por apenas um indivíduo estão isoladas dentro da rede, com poucos contactos noutras partes da rede.

v. Análise da partição obtida quanto à coesão e à separação:

A coesão de uma comunidade está diretamente relacionada com a sua densidade interna, ou seja, com a quantidade de ligações entre os nodos dentro do próprio grupo. No caso das comunidades obtidas, observamos que as menores comunidades, como as comunidades 7, 8, 10 e 11, apresentam uma coesão máxima, com densidade interna de 1, o que significa que os nodos dentro dessas comunidades estão fortemente interconectados. É importante reforçar que estas comunidades são constituídas por apenas 2 nodos/pessoas, o que facilita a obtenção de uma densidade interna máxima, já que o número de ligações possíveis entre dois nodos é limitado a apenas uma. A comunidade 3, com a maior densidade interna de 0,667 entre as comunidades que apresentam mais de 2 nodos, também se destaca como um grupo coeso. Já a comunidade 1, que é composta pelo maior número de nodos, é

também a menos coesa, apresentando uma densidade interna de 0,346, o que reflete a complexidade de grupos maiores, onde é mais difícil manter uma interligação completa entre todos os membros.

Por outro lado, a separação entre as comunidades pode ser observada pelo número de ligações externas. A comunidade 1, com 50 ligações externas, é a que mais se conecta com outras comunidades, o que indica uma interação mais forte com o resto da rede. As comunidades 5, 6 e 9, por sua vez, apresentam um número reduzido de ligações externas, indicando um isolamento maior. No geral, as comunidades menores possuem alta coesão interna, mas a separação entre as comunidades não é totalmente clara, especialmente nas que possuem mais ligações externas, como a comunidade 1. Note-se que as subredes que apresentam uma separação elevada estão ligadas a outras comunidades ou subredes, por poucas ligações.

vi. Avaliação das partições:

A modularidade, que é uma medida da qualidade da divisão da rede, foi 0,322. Este valor não é muito alto, o que sugere que, embora a divisão em comunidades tenha sido eficaz em identificar subgrupos relativamente coesos, ainda há um número considerável de ligações entre as comunidades, o que indica que a separação não é perfeita. A modularidade de 0.322 indica que a divisão realizada é boa, mas não ideal. O valor não está perto de 1, o que indicaria uma separação perfeita, mas também está relativamente longe de 0, o que indicaria uma divisão sem estrutura clara.

Método de Otimização de Modularidade (Fast Greedy)

O método Fast Greedy é um algoritmo eficiente para identificar comunidades em redes, baseado na otimização da modularidade. No caso da rede que estamos a analisar, onde os nodos representam pessoas e as ligações indicam contactos presenciais entre elas, este método permite-nos identificar grupos de indivíduos que têm mais interações entre si do que com outras pessoas fora do grupo.

Este tipo de análise é especialmente útil para entender como as pessoas se organizam em grupos ou comunidades na nossa “rede social” representada na Figura 2. Ao maximizar a modularidade, o algoritmo tenta encontrar a melhor maneira de separar a rede em subconjuntos coesos, onde as conexões internas são densas e as externas são minimizadas. Assim, cada comunidade encontrada pode ser interpretada como um grupo de pessoas que interage frequentemente, o que reflete relações sociais ou padrões de comportamento na rede.

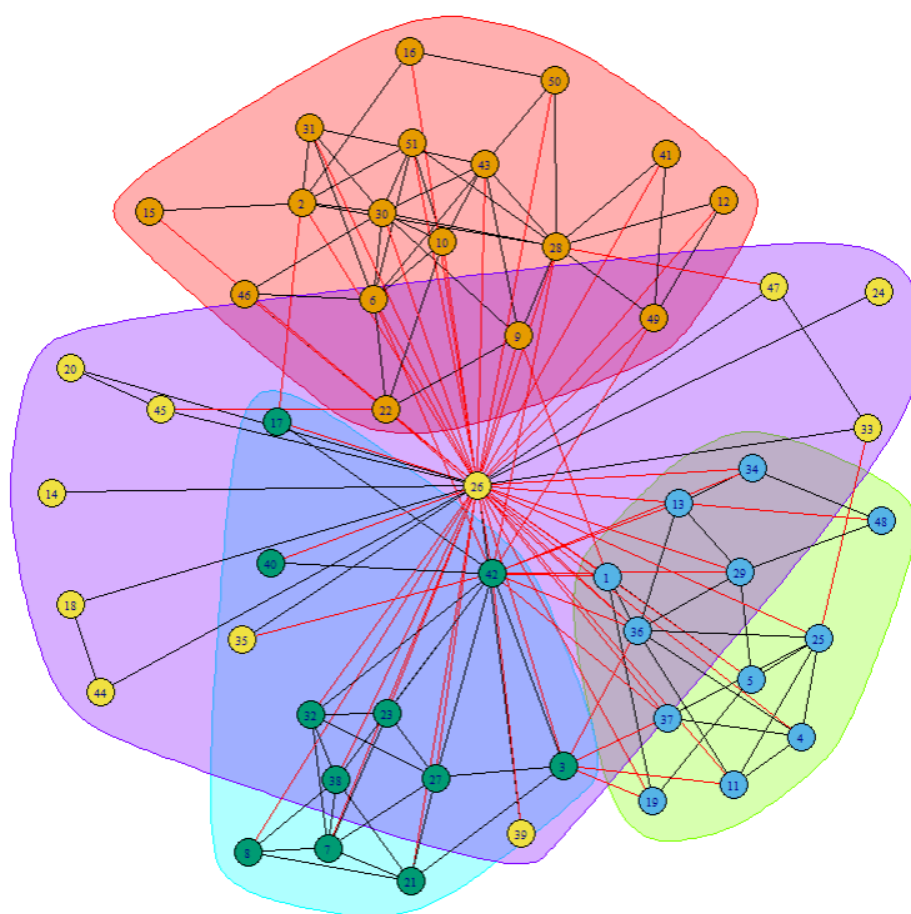


Figura 2 - Representação obtida pelo método de Otimização de Modularidade (Fast Greedy)

i. Número de comunidades obtidas:

O algoritmo Fast Greedy detectou 4 comunidades na rede, e este número de comunidades foi determinado com base na otimização da modularidade, que tenta dividir a rede da melhor forma possível para maximizar a separação entre os grupos, ou seja, ele procura criar comunidades que sejam bastante internas (fortemente conectadas) e, ao mesmo tempo, com o mínimo de ligações com outras comunidades.

ii. Dimensão de cada comunidade:

As dimensões das comunidades encontradas podem ver-se na Tabela 4.

| Comunidade | Nº de nodos |
|------------|-------------|
| 1 | 17 |
| 2 | 12 |
| 3 | 11 |
| 4 | 11 |

Tabela 4 - Dimensões das comunidades pelo método de otimização da modularidade (fast greedy)

As comunidades variam em tamanho, com a maior a ser composta por 17 nodos e as restantes a apresentarem entre 11 a 12 nodos. A distribuição relativamente equilibrada entre as comunidades sugere que o algoritmo fez uma segmentação eficaz, sem grandes disparidades no número de nodos.

iii. Número de ligações internas de cada comunidade e a respetiva densidade interna:

Para cada comunidade, calculamos o número de ligações internas (arestas que conectam nodos dentro da própria comunidade) e a densidade interna (proporção de ligações internas em relação ao número máximo de ligações possíveis dentro daquela comunidade). Os resultados apresentam-se na Tabela 5.

| Comunidade | Nº de ligações internas | Densidade interna |
|------------|-------------------------|-------------------|
| 1 | 38 | 0.279 |
| 2 | 20 | 0.303 |
| 3 | 23 | 0.418 |
| 4 | 13 | 0.236 |

Tabela 5 - Ligações internas e densidade interna pelo método de otimização da modularidade (fast greedy)

A densidade interna das comunidades varia de 0.236 a 0.418, o que significa que, no geral, as comunidades são bem conectadas internamente. A Comunidade 3 tem a maior densidade, o que mostra que os nodos dessa comunidade estão muito interconectados, por outro lado, a Comunidade 4 tem a densidade mais baixa, o que sugere que a sua coesão interna não é tão forte quanto nas outras.

iv. Número de ligações externas de cada comunidade:

Além das ligações internas, também se calculou o número de ligações externas (arestas que conectam cada nodo da comunidade aos nodos de outras comunidades). Os resultados foram agrupados na Tabela 6.

| Comunidade | Nº de ligações externas |
|------------|-------------------------|
| 1 | 25 |
| 2 | 25 |
| 3 | 27 |
| 4 | 45 |

Tabela 6 - Ligações externas pelo método de otimização da modularidade (fast greedy)

Podemos observar que o número de ligações externas varia entre 25 e 45. A Comunidade 4 tem o maior número de ligações externas, o que pode indicar que ela está mais conectada com as outras comunidades, já as Comunidades 1, 2 e 3 têm um número de ligações externas mais equilibrado.

v. Análise da partição obtida quanto à coesão e à separação:

A coesão de uma comunidade está relacionada com a sua densidade interna. Comunidades com maior densidade interna têm uma maior coesão, ou seja, os nodos dentro dessas comunidades estão mais fortemente conectados entre si. No caso das nossas comunidades:

- A Comunidade 3 tem a maior densidade interna (0.418), mostrando que é uma comunidade bem coesa.
- A Comunidade 4, por outro lado, tem a menor densidade interna (0.236), sugerindo que, embora seja uma comunidade separada, ela possui menos conexões internas, refletindo uma coesão mais fraca.

A separação entre as comunidades pode ser analisada pelo número de ligações externas, que representa a conexão de nodos de uma comunidade com nodos de outras comunidades:

- A Comunidade 4 tem o maior número de ligações externas (45), ao mesmo tempo que apresenta o menor número de nodos (11) entre todas as comunidades, o que sugere uma maior interação com outras comunidades e, portanto, uma separação mais fraca em relação às outras.
- As Comunidades 1, 2 e 3 têm um número mais equilibrado de ligações externas, o que indica uma separação relativamente mais forte entre elas, especialmente

No geral, o Fast Greedy conseguiu dividir a rede de uma forma razoável, criando comunidades que são relativamente coesas internamente, mas com algumas (como a Comunidade 4) que parecem ter mais interações com outras partes da rede e que por isso acabam por estar mais fracamente separadas.

vi. Avaliação das partições:

A modularidade foi utilizada para avaliar a qualidade da divisão realizada pelo algoritmo. O valor obtido foi de 0.3484, o que pode ser considerado razoável. A modularidade mede a qualidade da separação das comunidades: quanto mais alto o valor, melhor a divisão, ou seja, maior a conexão interna entre os nodos de cada comunidade e menor a conexão com outras comunidades. Embora o valor de modularidade não seja extremamente elevado, 0.3484 indica que a divisão feita pelo Fast Greedy é adequada, com espaço para melhorias.

Esse valor está distante de uma separação perfeita (valores próximos de 1), mas também não sugere uma divisão fraca.

Método de Otimização de Louvain

O método de otimização de Louvain é amplamente reconhecido pela sua eficiência em detectar comunidades em redes de grande escala. Funciona ao otimizar a modularidade, uma métrica que avalia a qualidade da divisão da rede em grupos, conforme já referido nos métodos anteriores. Por outras palavras, mede se dentro de cada grupo as conexões são mais fortes e, ao mesmo tempo, as ligações entre os grupos são mais fracas.

No caso da nossa rede, onde os nodos representam pessoas e as ligações indicam contactos presenciais entre elas, o método de Louvain ajuda a identificar grupos de indivíduos que interagem mais frequentemente entre si, e uma das coisas interessantes deste método é que ele trabalha de forma iterativa, primeiro encontra comunidades menores e depois "resume" a rede, agregando as comunidades encontradas em supernodos, para analisar a estrutura num nível mais alto, permitindo perceber diferentes camadas de agrupamento social dentro da rede. A Figura 3 ilustra as comunidades que se formaram na nossa rede com recurso ao método de otimização de Louvain.

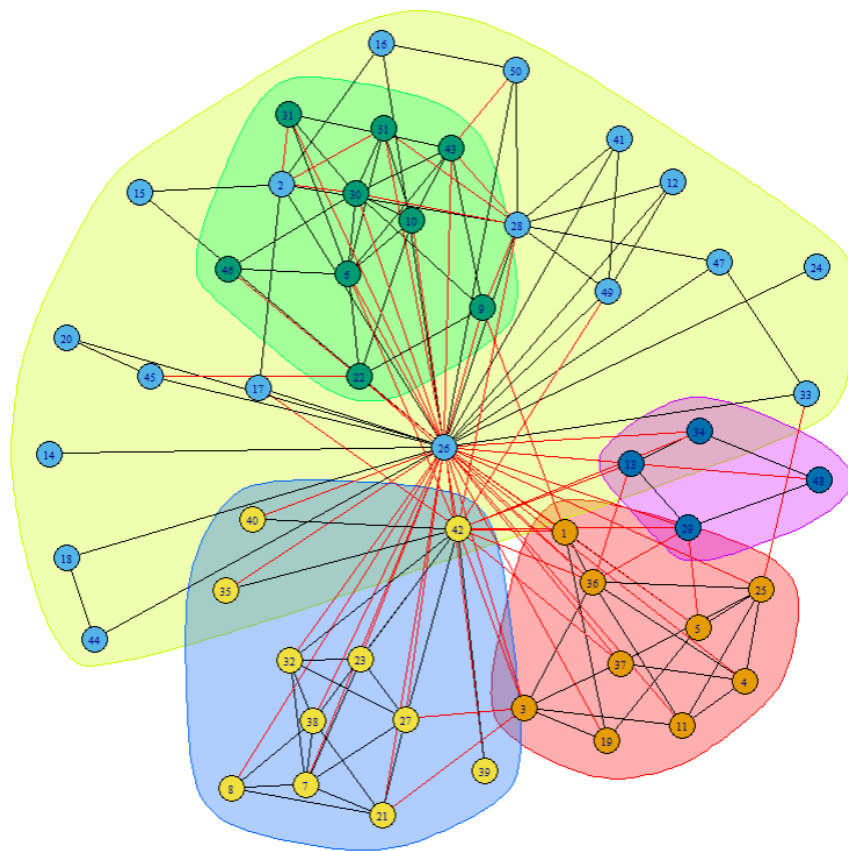


Figura 3 - representação obtida pelo método de Otimização de Louvain

i. Número de comunidades obtidas:

O algoritmo de Louvain identificou 5 comunidades na rede. Este número é definido com base na otimização da modularidade, uma métrica que, conforme já constatado, mede a qualidade da divisão da rede, favorecendo partições onde as conexões dentro das comunidades são densas e as entre comunidades são menos frequentes.

ii. Dimensão de cada comunidade:

As dimensões das comunidades estão descritas na Tabela 7.

| Comunidade | Nº de nodos |
|------------|-------------|
| 1 | 9 |
| 2 | 18 |
| 3 | 9 |
| 4 | 11 |
| 5 | 4 |

Tabela 7 - Dimensões das comunidades pelo método de otimização da modularidade de Louvain

A maior comunidade (2) possui 18 nodos, enquanto a menor (5) tem apenas 4 nodos. Apesar da divisão não ser perfeitamente equilibrada, a variação nos tamanhos reflete a estrutura subjacente da rede.

iii. Número de ligações internas de cada comunidade e a respetiva densidade interna:

Cada comunidade foi avaliada quanto ao número de ligações internas (arestas dentro da própria comunidade) e a sua densidade interna (proporção de ligações presentes em relação ao máximo possível). Os resultados obtidos encontram-se compilados na Tabela 8.

| Comunidade | Nº de ligações internas | Densidade interna |
|------------|-------------------------|-------------------|
| 1 | 17 | 0.472 |
| 2 | 32 | 0.209 |
| 3 | 20 | 0.556 |

| | | |
|---|----|-------|
| 4 | 21 | 0.382 |
| 5 | 4 | 0.667 |

Tabela 8 - Ligações internas e densidade interna pelo método de otimização da modularidade de Louvain

A densidade interna varia entre 0.209 e 0.667, indicando que algumas comunidades são mais coesas (como a Comunidade 5, que apresenta ao mesmo tempo o menor número de nodos), enquanto outras apresentam menor conectividade interna (como a Comunidade 2, que apresenta ao mesmo tempo o maior número de nodos).

iv. Número de ligações externas de cada comunidade:

O número de ligações externas (arestas conectando nodos da comunidade com outras comunidades) foi calculado tal como se pode ver na Tabela 9.

| Comunidade | Nº de ligações externas |
|------------|-------------------------|
| 1 | 21 |
| 2 | 46 |
| 3 | 21 |
| 4 | 24 |
| 5 | 10 |

Tabela 9 - Ligações externas pelo método de otimização da modularidade de Louvain

A Comunidade 2 tem significativamente mais conexões externas, sugerindo que está mais integrada com outras partes da rede, enquanto que a Comunidade 5 apresenta o menor número de ligações externas. Estas comunidades correspondem às que apresentam o maior e menor número de nodos, respetivamente, conforme já havia sido constatado no ponto imediatamente anterior a este.

v. Análise da partição obtida quanto à coesão e à separação:

A coesão interna é refletida pela densidade interna:

- A Comunidade 5 destaca-se como a mais coesa (densidade = 0.667).
- A Comunidade 2 tem a menor coesão (densidade = 0.209).

A separação entre as comunidades pode ser inferida pelo número de ligações externas:

- A Comunidade 2, com 46 ligações externas, está mais integrada com outras partes da rede, o que indica uma separação mais fraca e maior interação com as restantes comunidades.
- A Comunidade 5, com apenas 10 ligações externas, demonstra um maior isolamento, refletindo uma separação mais forte em relação às outras comunidades.

vi. Avaliação das partições:

A modularidade da divisão das comunidades gerada pelo método de Louvain foi de 0.362. Embora este valor não seja extremamente alto, indica uma divisão razoavelmente eficaz das comunidades dentro da rede. Em termos de modularidade, conforme já referido nas secções anteriores, valores próximos de 1 indicam uma separação clara entre as comunidades, com forte coesão interna e poucas ligações externas.

Com uma modularidade de 0.362, a partição revela uma divisão que, embora não seja ideal, apresenta grupos relativamente bem definidos, com comunidades mais conectadas internamente do que com as restantes. Esse valor sugere que o algoritmo foi capaz de identificar estruturas de agrupamento na rede, mas pode haver algum grau de sobreposição ou ligações inesperadas entre os grupos, deixando espaço para melhorias na segmentação. Em geral, o valor de modularidade indica que o método de Louvain gerou uma divisão relativamente eficiente, com uma separação razoável das comunidades, mas com oportunidades para aperfeiçoar a definição dos limites entre elas.

Método de Propagação de Etiquetas

O método de propagação de etiquetas é uma abordagem para detecção de comunidades em redes, onde inicialmente, cada nodo da rede é atribuído a uma comunidade própria e, de seguida, as etiquetas das comunidades são propagadas pelos seus vizinhos. O processo continua iterativamente até que as etiquetas se estabilizem, formando assim as comunidades finais. No contexto desta rede, tal como já referido no âmbito dos métodos anteriores, os nodos representam pessoas e as ligações representam contactos presenciais entre elas.

Na nossa rede, este método identificou apenas uma comunidade, o que significa que todos os 51 nodos da rede estão interligados num único grupo. Isso indica que, embora haja diferenças nos graus de conectividade entre os nodos, o algoritmo não foi capaz de identificar subgrupos de contactos suficientemente distintos que permitam a formação de comunidades separadas. Assim, a única comunidade identificada neste caso representa o conjunto completo das pessoas na rede, onde as interações ocorrem de forma global, tal como se pode ver na Figura 4.

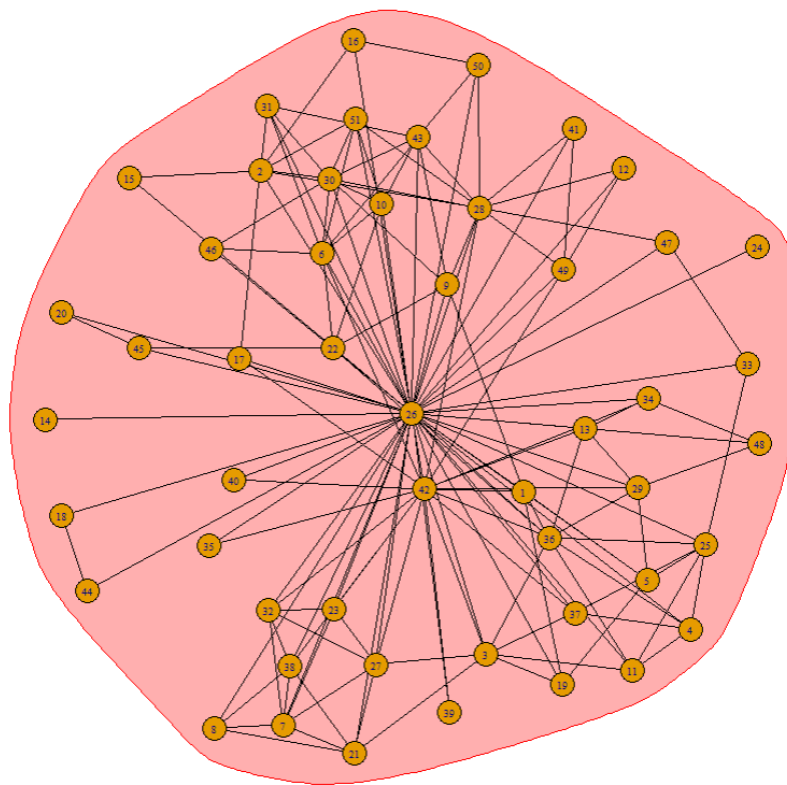


Figura 4 - Representação obtida pelo método de propagação de etiquetas

i. Número de comunidades obtidas:

Apenas uma comunidade foi obtida pelo método de propagação de etiquetas. Isso significa que todos os 51 nodos da rede foram agrupados numa única partição. Este resultado reflete a forma como o algoritmo funciona, pois, apesar das ligações internas entre os nodos, o método propaga etiquetas de forma simples e rápida, agrupando os nodos com base nas etiquetas dos seus vizinhos. No caso desta rede, o algoritmo não conseguiu identificar subgrupos distintos, resultando numa rede completamente interligada numa única comunidade. Além disso, o método depende da ordem de inspeção dos nodos e dos empates, o que pode influenciar o resultado final.

ii. Dimensão de cada comunidade:

Todos os nodos da rede (51) estão presentes em apenas uma comunidade, a única que foi identificada. Isso representa 100% dos nodos da rede, confirmando assim a ausência de divisões estruturais no grafo em estudo.

iii. Número de ligações internas de cada comunidade e respetiva densidade interna:

A comunidade tem 155 ligações internas, que corresponde também ao número de todas as ligações existentes no grafo. A densidade interna da comunidade é 0,122, o que mostra que a conectividade interna é relativamente baixa ao comparar com o número total de ligações possíveis num grafo completo com 51 nodos.

iv. Número de ligações externas de cada comunidade:

Não foram identificadas ligações externas para a comunidade. O resultado é expectável, uma vez que toda a rede foi agrupada exclusivamente numa só comunidade. Não há nodos e arestas fora dessa partição, logo não há possibilidade de existirem conexões externas.

v. Análise da partição obtida quanto à coesão e à separação:

A coesão da partição é máxima, pois todos os nodos pertencem à mesma comunidade e as 155 ligações ligam diretamente os membros dessa comunidade.

Não existe separação, pois a rede é composta por uma única comunidade, sem divisões claras. A falta de separação indica que a rede tem uma conectividade forte o suficiente para manter todos os nodos interligados num único grupo. A ausência de várias comunidades pode ser explicada pela forte coesão geral da rede.

vi. Avaliação das partições:

A modularidade calculada é 0, o que significa que a divisão feita pelo método de propagação de etiquetas não conseguiu destacar diferenças significativas entre as ligações internas das comunidades e as que seriam de esperar num grafo aleatório. Este valor sugere que a rede não tem uma estrutura modular clara, pois todos os nodos estão na mesma comunidade e não há subestruturas evidentes que favoreçam a criação de várias comunidades.

Em suma, o método de propagação de etiquetas identificou apenas uma comunidade na rede, composta por todos os 51 nodos. Este resultado destaca a forte coesão global da rede, sem divisões claras. A modularidade de 0 confirma a ausência de uma estrutura modular distinta.

Comparação entre os Métodos

A comparação entre os resultados obtidos pelos quatro métodos de detecção de comunidades na rede revela diferenças significativas nas abordagens de segmentação da rede e nas características das comunidades identificadas. Cada método apresenta as suas próprias peculiaridades, refletindo-se em aspetos como o número de comunidades, a coesão interna, a separação entre as comunidades e a modularidade. Alguns dos resultados gerais da aplicação dos quatro métodos encontram-se sumarizados na Tabela 10.

| Método | Nº comunidades | Dimensão mín. | Dimensão máx. | Modularidade |
|----------------------------|----------------|---------------|---------------|--------------|
| Remoção de Pontes | 11 | 1 | 17 | 0.3222268 |
| Otimização de Modularidade | 4 | 11 | 17 | 0.3484287 |
| Otimização de Louvain | 5 | 4 | 18 | 0.3616441 |
| Propagação de Etiquetas | 1 | 51 | 51 | 0 |

Tabela 10 - Principais indicadores da aplicação dos quatro métodos de detecção de comunidades

O método de propagação de etiquetas foi o único que resultou numa única comunidade abrangendo todos os 51 nodos. A modularidade zero e a densidade relativamente baixa indicam que, apesar das ligações entre os nodos, não há uma estrutura modular clara na rede, sugerindo que a rede não possui subgrupos bem definidos ou separados. Esse resultado destaca uma rede coesa globalmente, mas sem divisões internas evidentes. Em contraste, os outros métodos revelaram uma rede com várias comunidades, o que aponta para abordagens que procuram identificar subgrupos mais estruturados.

O método de remoção de pontes obteve o maior número de comunidades (11), refletindo uma segmentação mais fragmentada da rede. Algumas comunidades, com tamanhos que variam de 1 a 17 nodos, mostram que o método tende a separar a rede de forma mais agressiva. A densidade interna das menores comunidades é máxima, o que revela alta coesão entre os seus nodos. No entanto, a grande quantidade de ligações externas observada, especialmente na comunidade 1, sugere que as comunidades identificadas ainda estão interligadas de forma significativa, o que implica uma separação menos clara entre elas. A modularidade de 0.322 confirma essa fragmentação, com boa separação, mas ainda com interações externas consideráveis.

Por outro lado, os métodos Fast Greedy e Louvain geraram divisões mais equilibradas, com 4 e 5 comunidades, respetivamente. Ambas as abordagens identificaram subgrupos que são relativamente bem definidos, embora com características distintas. No caso do Fast Greedy, a maior comunidade é composta por 17 nodos, enquanto que as restantes variam entre 11 e 12. A densidade interna das suas comunidades também varia, com a comunidade 3 a apresentar a maior coesão (0.418), enquanto a comunidade 4, com a densidade mais baixa (0.236), sugere um grupo com menos capacidade de se conectar internamente. O número de ligações externas de cada comunidade varia, com a comunidade 4 a destacar-se por estar mais conectada com as outras comunidades. A modularidade de 0.348 do Fast Greedy é razoável, refletindo uma separação boa, mas não ideal.

Já o método de otimização de Louvain, por sua vez, registou o valor de modularidade mais elevado (0.362), o que indica uma divisão mais eficaz, com subgrupos mais bem separados entre si. Algumas das comunidades formadas pelo método de Louvain apresentam uma densidade interna relativamente elevada, o que reflete a melhor coesão dos grupos.

Em suma, a comparação dos quatro métodos revela que o Fast Greedy e o Louvain são mais eficazes na identificação de comunidades bem definidas, com boa coesão interna e separação entre as comunidades. O método de remoção de pontes apresentou uma divisão mais agressiva e mais fragmentada, com algumas comunidades com alta coesão, mas também com um número de nodos muito reduzido. Note-se que a 4ª maior comunidade obtida com o método de remoção de pontes é constituída por apenas 3 nodos. O método de propagação de etiquetas, por outro lado, não conseguiu identificar divisões dentro da rede, refletindo uma forte conectividade global, mas sem uma estrutura modular clara. Importa ainda realçar que cada abordagem tem as suas vantagens e limitações, e a escolha do método depende sempre dos objetivos da análise e do tipo de estrutura que se deseja identificar na rede.

Apêndice

Apêndice I - Importação das bibliotecas necessárias

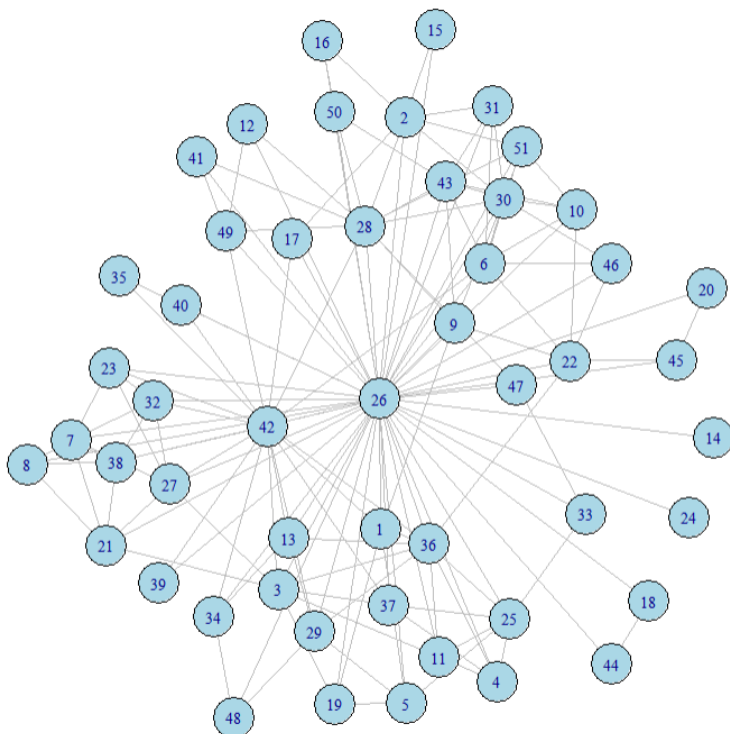
```
library(igraph)
```

Apêndice II - Importação da rede através do ficheiro disponibilizado no Moodle

```
rede <- read_graph("ligacoes_t2.txt", format = "edgelist", directed = F)
# Definir o layout fixo das redes
layout <- layout_with_fr(rede)
```

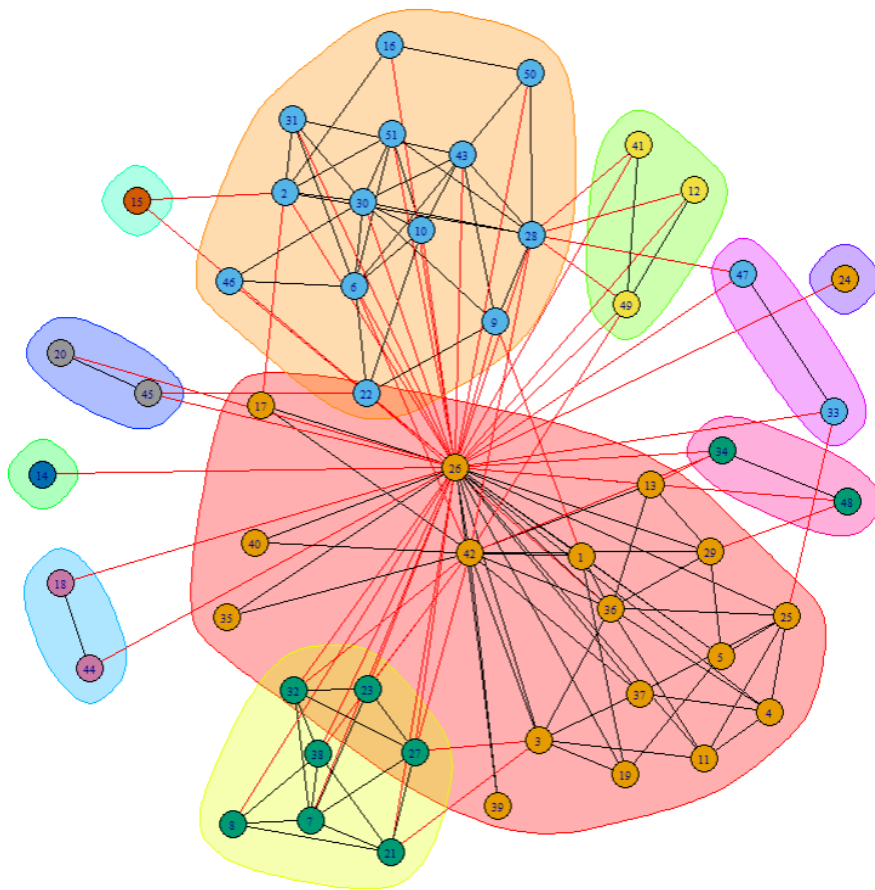
Apêndice III - Plot visual da rede

```
plot(
  rede,
  vertex.size = 12,           # Tamanho dos nodos
  edge.color = "gray",        # Cor das arestas
  vertex.color = "lightblue", # Cor dos nodos
  layout = layout
)
```



Apêndice IV - Aplicação do método de remoção de pontes

```
#####  
# MÉTODO 1: REMOÇÃO DE PONTES #  
#####  
  
comunidades_remocao_pontes <- cluster_edge_betweenness(rede)  
plot(comunidades_remocao_pontes, rede, vertex.size = 7, vertex.label.cex =  
.6, layout = layout)
```



Apêndice V - Dimensão das comunidades com o método de remoção de pontes

```
> # Número de comunidades  
> length(comunidades_remocao_pontes)  
[1] 11  
  
> # Dimensão das comunidades  
> sizes(comunidades_remocao_pontes)
```



```

Community sizes
 1  2  3  4  5  6  7  8  9 10 11
17 13  7  3  1  1  2  2  1  2  2
> print(paste("Tamanho mínimo da comunidade:",
min(sizes(comunidades_remocao_pontes))))
[1] "Tamanho mínimo da comunidade: 1"
> print(paste("Tamanho máximo da comunidade:",
max(sizes(comunidades_remocao_pontes))))
[1] "Tamanho máximo da comunidade: 17"

> # Frequência de tamanhos
> table(sizes(comunidades_remocao_pontes))

 1  2  3  7 13 17
 3  4  1  1  1  1

```

Apêndice VI - Ligações internas, externas e densidade interna do método de remoção de pontes

```

> # Número de ligações internas e densidade interna + ligações externas
> for (i in 1:length(comunidades_remocao_pontes)) {
+   # Subgrafo da comunidade
+   comunidade_nodes <- which(membership(comunidades_remocao_pontes) == i)
+   subgrafo <- induced_subgraph(rede, comunidade_nodes)
+
+   # Número de arestas internas (dentro da comunidade)
+   num_arestas_internas <- ecount(subgrafo)
+   densidade_interna <- edge_density(subgrafo)
+
+   # Número de ligações externas (arestas para fora da comunidade)
+   arestas_externas <- E(rede)[which(
+     (ends(rede, E(rede))[, 1] %in% comunidade_nodes & !(ends(rede, E(rede))[, 2]
+ %in% comunidade_nodes)) |
+     (ends(rede, E(rede))[, 2] %in% comunidade_nodes & !(ends(rede, E(rede))[, 1]
+ %in% comunidade_nodes))
+   )]
+
+   num_ligacoes_externas <- length(arestas_externas)
+
+   # Exibição dos resultados
+   print(paste("Comunidade", i, ": Ligações internas =", num_arestas_internas,
+     ", Densidade interna =", densidade_interna,
+     ", Ligações externas =", num_ligacoes_externas))
+   cat("Nodos :", paste(comunidade_nodes, collapse = " "), "\n")
+ }

```

```

[1] "Comunidade 1 : Ligações internas = 47 , Densidade interna = 0.345588235294118
, Ligações externas = 50"
Nodos : 1 3 4 5 11 13 17 19 25 26 29 35 36 37 39 40 42
[1] "Comunidade 2 : Ligações internas = 32 , Densidade interna = 0.41025641025641 ,
Ligações externas = 24"
Nodos : 2 6 9 10 16 22 28 30 31 43 46 50 51
[1] "Comunidade 3 : Ligações internas = 14 , Densidade interna = 0.666666666666667
, Ligações externas = 13"
Nodos : 7 8 21 23 27 32 38
[1] "Comunidade 4 : Ligações internas = 2 , Densidade interna = 0.666666666666667 ,
Ligações externas = 7"
Nodos : 12 41 49
[1] "Comunidade 5 : Ligações internas = 0 , Densidade interna = NaN , Ligações
externas = 1"
Nodos : 14
[1] "Comunidade 6 : Ligações internas = 0 , Densidade interna = NaN , Ligações
externas = 2"
Nodos : 15
[1] "Comunidade 7 : Ligações internas = 1 , Densidade interna = 1 , Ligações
externas = 2"
Nodos : 18 44
[1] "Comunidade 8 : Ligações internas = 1 , Densidade interna = 1 , Ligações
externas = 3"
Nodos : 20 45
[1] "Comunidade 9 : Ligações internas = 0 , Densidade interna = NaN , Ligações
externas = 1"
Nodos : 24
[1] "Comunidade 10 : Ligações internas = 1 , Densidade interna = 1 , Ligações
externas = 4"
Nodos : 33 47
[1] "Comunidade 11 : Ligações internas = 1 , Densidade interna = 1 , Ligações
externas = 5"
Nodos : 34 48

```

Apêndice VII - Modularidade do método de remoção de pontes

```

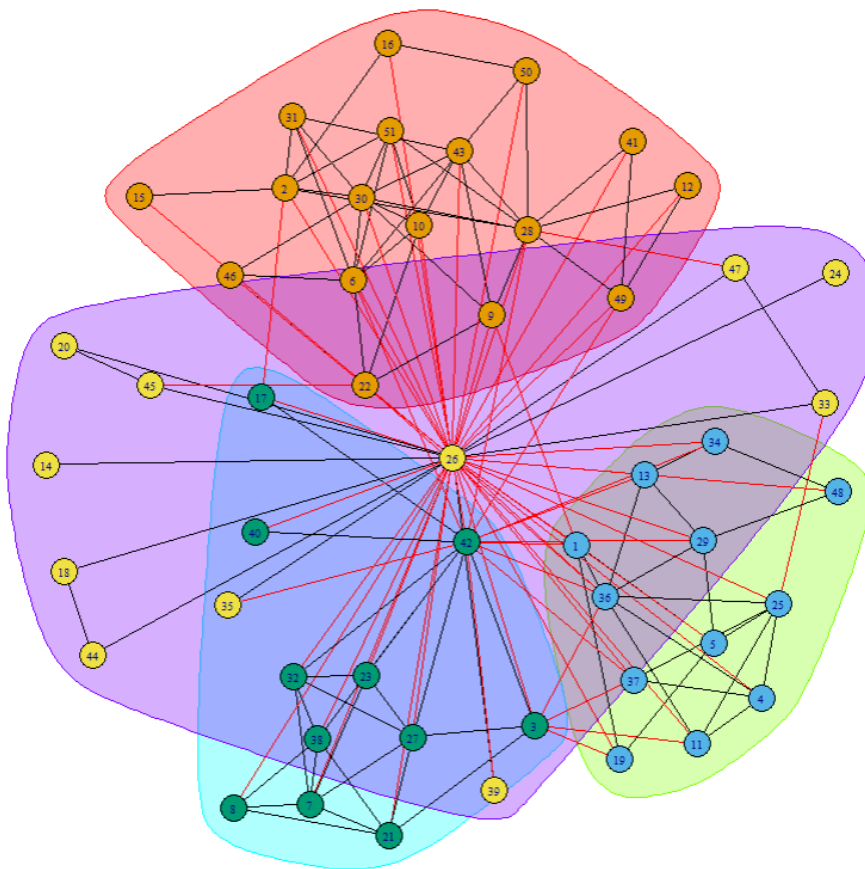
> # Modularidade
> modularity(comunidades_remocao_pontes)
[1] 0.3222268

```

Apêndice VIII - Aplicação do método de otimização de modularidade (fast greedy)

```
#####
# MÉTODO 2: OTIMIZAÇÃO DE MODULARIDADE (FAST GREEDY) #
#####

comunidades_fast_greedy <- cluster_fast_greedy(rede)
plot(comunidades_fast_greedy, rede, vertex.size = 7, vertex.label.cex = .6,
     layout = layout)
```



Apêndice IX - Dimensão das comunidades com o método de otimização de modularidade (fast greedy)

```
> # Número de comunidades
> length(comunidades_fast_greedy)
[1] 4

> # Dimensão das comunidades
> sizes(comunidades_fast_greedy)
Community sizes
 1  2  3  4
```

```

17 12 11 11
> print(paste("Tamanho mínimo da comunidade:",
min(sizes(comunidades_fast_greedy))))
[1] "Tamanho mínimo da comunidade: 11"
> print(paste("Tamanho máximo da comunidade:",
max(sizes(comunidades_fast_greedy))))
[1] "Tamanho máximo da comunidade: 17"

> # Frequência de tamanhos
> table(sizes(comunidades_fast_greedy))

11 12 17
 2  1  1

```

Apêndice X - Ligações internas, externas e densidade interna do método de otimização de modularidade (fast greedy)

```

> # Número de ligações internas e densidade interna + ligações externas
> for (i in 1:length(comunidades_fast_greedy)) {
+   # Subgrafo da comunidade
+   comunidade_nodes <- which(membership(comunidades_fast_greedy) == i)
+   subgrafo <- induced_subgraph(rede, comunidade_nodes)
+
+   # Número de arestas internas (dentro da comunidade)
+   num_arestas_internas <- ecount(subgrafo)
+   densidade_interna <- edge_density(subgrafo)
+
+   # Número de ligações externas (arestas para fora da comunidade)
+   arestas_externas <- E(rede)[which(
+     (ends(rede, E(rede))[, 1] %in% comunidade_nodes & !(ends(rede, E(rede))[, 2]
+ %in% comunidade_nodes)) |
+     (ends(rede, E(rede))[, 2] %in% comunidade_nodes & !(ends(rede, E(rede))[, 1]
+ %in% comunidade_nodes))
+   )]
+
+   num_ligacoes_externas <- length(arestas_externas)
+
+   # Exibição dos resultados
+   print(paste("Comunidade", i, ": Ligações internas =", num_arestas_internas,
+     ", Densidade interna =", densidade_interna,
+     ", Ligações externas =", num_ligacoes_externas))
+   cat("Nodos :", paste(comunidade_nodes, collapse = " "), "\n")
+ }

[1] "Comunidade 1 : Ligações internas = 38 , Densidade interna = 0.279411764705882

```

```
, Ligações externas = 25"
Nodos : 2 6 9 10 12 15 16 22 28 30 31 41 43 46 49 50 51
[1] "Comunidade 2 : Ligações internas = 20 , Densidade interna = 0.303030303030303
, Ligações externas = 25"
Nodos : 1 4 5 11 13 19 25 29 34 36 37 48
[1] "Comunidade 3 : Ligações internas = 23 , Densidade interna = 0.418181818181818
, Ligações externas = 27"
Nodos : 3 7 8 17 21 23 27 32 38 40 42
[1] "Comunidade 4 : Ligações internas = 13 , Densidade interna = 0.236363636363636
, Ligações externas = 45"
Nodos : 14 18 20 24 26 33 35 39 44 45 47
```

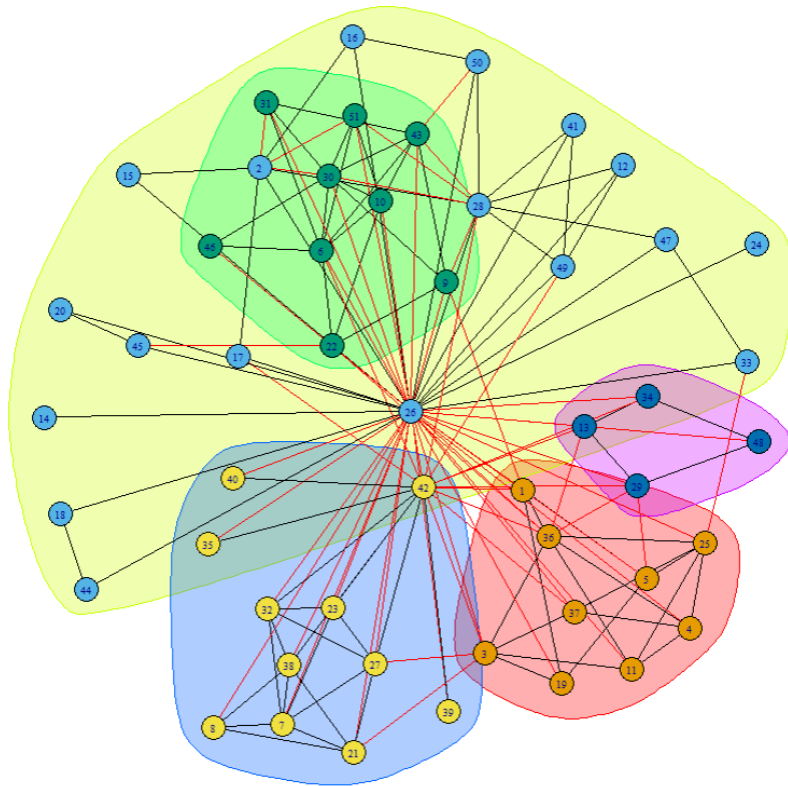
Apêndice XI - Modularidade do método de otimização da modularidade (fast greedy)

```
> # Modularidade
> modularity(comunidades_fast_greedy)
[1] 0.3484287
```

Apêndice XII - Aplicação do método de otimização de Louvain

```
#####
# MÉTODO 3: OTIMIZAÇÃO DE LOUVAIN #
#####

set.seed(123)
comunidades_louvain <- cluster_louvain(rede)
plot(comunidades_louvain, rede, vertex.size = 7, vertex.label.cex = .6,
layout = layout)
```



Apêndice XIII - Dimensão das comunidades com o método de otimização de Louvain

```
> # Número de comunidades
> length(comunidades_louvain)
[1] 5

> # Dimensão das comunidades
> sizes(comunidades_louvain)
Community sizes
 1  2  3  4  5
 9 18  9 11  4

> print(paste("Tamanho mínimo da comunidade:", min(sizes(comunidades_louvain))))
[1] "Tamanho mínimo da comunidade: 4"

> print(paste("Tamanho máximo da comunidade:", max(sizes(comunidades_louvain))))
[1] "Tamanho máximo da comunidade: 18"

> # Frequência de tamanhos
> table(sizes(comunidades_louvain))

 4  9 11 18
 1  2  1  1
```

Apêndice XIV - Ligações internas, externas e densidade interna do método de otimização de Louvain

```
> # Número de ligações internas e densidade interna + ligações externas
> for (i in 1:length(comunidades_louvain)) {
+   # Subgrafo da comunidade
+   comunidade_nodes <- which(membership(comunidades_louvain) == i)
+   subgrafo <- induced_subgraph(rede, comunidade_nodes)
+
+   # Número de arestas internas (dentro da comunidade)
+   num_arestas_internas <- ecount(subgrafo)
+   densidade_interna <- edge_density(subgrafo)
+
+   # Número de ligações externas (arestas para fora da comunidade)
+   arestas_externas <- E(rede)[which(
+     (ends(rede, E(rede))[, 1] %in% comunidade_nodes & !(ends(rede, E(rede))[, 2]
+ %in% comunidade_nodes)) |
+     (ends(rede, E(rede))[, 2] %in% comunidade_nodes & !(ends(rede, E(rede))[, 1]
+ %in% comunidade_nodes))
+   )]
+
+   num_ligacoes_externas <- length(arestas_externas)
+
+   # Exibição dos resultados
+   print(paste("Comunidade", i, ": Ligações internas =", num_arestas_internas,
+     ", Densidade interna =", densidade_interna,
+     ", Ligações externas =", num_ligacoes_externas))
+   cat("Nodos :", paste(comunidade_nodes, collapse = " "), "\n")
+ }

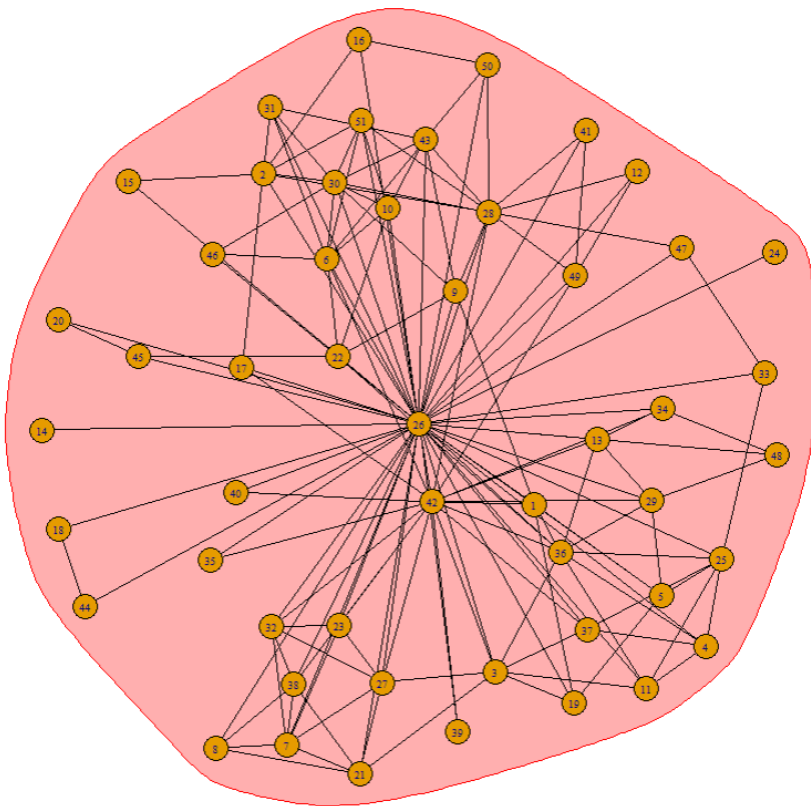
[1] "Comunidade 1 : Ligações internas = 17 , Densidade interna = 0.472222222222222
, Ligações externas = 21"
Nodos : 1 3 4 5 11 19 25 36 37
[1] "Comunidade 2 : Ligações internas = 32 , Densidade interna = 0.209150326797386
, Ligações externas = 46"
Nodos : 2 12 14 15 16 17 18 20 24 26 28 33 41 44 45 47 49 50
[1] "Comunidade 3 : Ligações internas = 20 , Densidade interna = 0.5555555555555556
, Ligações externas = 21"
Nodos : 6 9 10 22 30 31 43 46 51
[1] "Comunidade 4 : Ligações internas = 21 , Densidade interna = 0.381818181818182
, Ligações externas = 24"
Nodos : 7 8 21 23 27 32 35 38 39 40 42
[1] "Comunidade 5 : Ligações internas = 4 , Densidade interna = 0.666666666666667 ,
, Ligações externas = 10"
Nodos : 13 29 34 48
```

Apêndice XV - Modularidade do método de otimização de Louvain

```
> # Modularidade  
> modularity(comunidades_louvain)  
[1] 0.3616441
```

Apêndice XVI - Aplicação do método de propagação de etiquetas

```
#####  
# MÉTODO 4: PROPAGAÇÃO DE ETIQUETAS #  
#####  
  
set.seed(2024)  
comunidades_label_propagation <- cluster_label_prop(rede)  
plot(comunidades_label_propagation, rede, vertex.size = 7, vertex.label.cex  
= .6, layout = layout)
```



Apêndice XVI - Dimensão das comunidades com o método de propagação de etiquetas

```
> # Número de comunidades
> length(comunidades_label_propagation)
[1] 1

> # Dimensão das comunidades
> sizes(comunidades_label_propagation)
Community sizes
 1
51

> print(paste("Tamanho mínimo da comunidade:",
min(sizes(comunidades_label_propagation))))
[1] "Tamanho mínimo da comunidade: 51"
> print(paste("Tamanho máximo da comunidade:",
max(sizes(comunidades_label_propagation))))
[1] "Tamanho máximo da comunidade: 51"
>
> # Frequência de tamanhos
> table(sizes(comunidades_label_propagation))

51
 1
```

Apêndice XVII - Ligações internas, externas e densidade interna do método de propagação de etiquetas

```
> # Número de ligações internas e densidade interna + ligações externas
> for (i in 1:length(comunidades_label_propagation)) {
+   # Subgrafo da comunidade
+   comunidade_nodes <- which(membership(comunidades_label_propagation) == i)
+   subgrafo <- induced_subgraph(rede, comunidade_nodes)
+
+   # Número de arestas internas (dentro da comunidade)
+   num_arestas_internas <- ecount(subgrafo)
+   densidade_interna <- edge_density(subgrafo)
+
+   # Número de ligações externas (arestas para fora da comunidade)
+   arestas_externas <- E(rede)[which(
+     (ends(rede, E(rede))[, 1] %in% comunidade_nodes & !(ends(rede, E(rede))[, 2]
+ %in% comunidade_nodes)) |
+     (ends(rede, E(rede))[, 2] %in% comunidade_nodes & !(ends(rede, E(rede))[, 1]
```

```

%in% comunidade_nodes))
+   )]
+
+   num_ligacoes_externas <- length(arestas_externas)
+
+   # Exibição dos resultados
+   print(paste("Comunidade", i, ": Ligações internas =", num_arestas_internas,
+               ", Densidade interna =", densidade_interna,
+               ", Ligações externas =", num_ligacoes_externas))
+   cat("Nodos :", paste(comunidade_nodes, collapse = " "), "\n")
+ }

[1] "Comunidade 1 : Ligações internas = 155 , Densidade interna = 0.12156862745098
, Ligações externas = 0"
Nodos : 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51

```

Apêndice XVIII - Modularidade do método de propagação de etiquetas

```

> # Modularidade
> modularity(comunidades_label_propagation)
[1] 0

```