UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

PEDRO HENRIQUE OLIVEIRA DA SILVA
Orientador: Jadson Castro Gertrudes

UM COMPARATIVO ENTRE A MODULARIDADE KNN E A MEDIDA DE ESTABILIDADE PARA EXTRAÇÃO ÓTIMA DE GRUPOS A PARTIR DE HIERARQUIA DE GRUPOS

Ouro Preto, MG
2024

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

PEDRO HENRIQUE OLIVEIRA DA SILVA

UM COMPARATIVO ENTRE A MODULARIDADE KNN E A MEDIDA DE ESTABILIDADE PARA EXTRAÇÃO ÓTIMA DE GRUPOS A PARTIR DE HIERARQUIA DE GRUPOS

Monografia II apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Jadson Castro Gertrudes

Ouro Preto, MG
2024

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E BIOLOGICAS
DEPARTAMENTO DE COMPUTACAO

FOLHA DE APROVAÇÃO

Pedro Henrique Oliveira da Silva

Um comparativo entre a Modularidade KNN e a medida de Estabilidade para extração ótima de grupos a partir de hierarquia de grupos

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal
de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 15 de Fevereiro de 2024.

Membros da banca:

Jadson Castro Gertrudes (Orientador) - Doutor - Universidade Federal de Ouro Preto
Anderson Almeida Ferreira (Examinador) - Doutor - Universidade Federal de Ouro Preto
Mardochee Ogécime (Examinador) - Doutor - PPGCCC UFOP

Jadson Castro Gertrudes, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 15/02/2024.

Documento assinado eletronicamente por **Jadson Castro Gertrudes**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 19/02/2024, às 13:59, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0667160** e o código CRC **D5988C22**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591692 - www.ufop.br

Agradecimentos

Agradeço ao Prof. Dr. Jadson Castro Gertrudes, orientador deste trabalho, por todo o apoio prestado e orientação durante a realização deste trabalho.

Também agradeço aos meus amigos de curso de Ciência da Computação pela companhia, risadas e bons momentos vividos ao longo do curso.

Por último, gostaria de agradecer a minha família por todo o apoio e consideração. Sem eles, eu não estaria aqui.

Resumo

Análises de agrupamento fazem parte do domínio do aprendizado não supervisionado. Elas envolvem a criação de grupos (*clusters*) a partir dos dados, reunindo-os com base em características compartilhadas, sem a necessidade de rótulos prévios. Este estudo tem como propósito avaliar a eficácia da métrica de qualidade *Modularidade Q*, fundamentada na estrutura de redes complexas, dentro do contexto do algoritmo *FOSC*, usado para extrair agrupamentos. O objetivo é investigar o desempenho dessa métrica ao ser aplicada em hierarquias produzidas por modelos tradicionais de agrupamento hierárquico. O trabalho visa a análise da métrica *Modularidade Q*, sua comparação com a medida original (*Stability*) empregada no *FOSC*, e sua avaliação tanto em cenários pré-definidos quanto em um cenário real, que abrange a análise de redes sociais. O estudo busca enriquecer o entendimento sobre a eficácia da métrica *Modularidade Q* no contexto do *FOSC*, bem como sua aplicabilidade prática em situações do mundo real. Os resultados obtidos nesse trabalho foram positivos, mostrando que *Modularidade Q* tem grande potencial para ser utilizada.

Palavras-chave: Aprendizado de máquina. Aprendizado não-supervisionado. Agrupamento de dados. Medidas de qualidade para extração de grupos.

Abstract

Cluster analysis is part of the unsupervised learning domain. It involves creating groups (clusters) from data by assembling them based on shared characteristics, without the need for pre-existing labels. This study aims to evaluate the effectiveness of the Modularity Q quality metric, grounded in the structure of complex networks, within the context of the FOSC algorithm used for extracting clusters. The objective is to investigate the performance of this metric when applied to hierarchies produced by traditional hierarchical clustering models. The work involves the analysis of the Modularity Q metric, its comparison with the original measure (Stability) employed in FOSC, and its evaluation in both predefined scenarios and a real-world scenario, encompassing the analysis of social networks. The study seeks to enhance the understanding of the effectiveness of the Modularity Q metric in the context of FOSC, as well as its practical applicability in real-world situations. The results obtained in this study were positive, demonstrating that Modularity Q has significant potential for use.

Keywords: Machine Learning, Unsupervised Learning, Clustering, Quality measures for optimal cluster extraction.

Lista de Ilustrações

Lista de Tabelas

Sumário

1 Introdução

No campo do aprendizado de máquina, as atividades de aprendizado descritivo, também conhecidas como não supervisionadas, estão relacionadas à detecção de informações significativas nos dados, sem depender de uma orientação externa, também chamado como rótulo, para direcionar o processo de aprendizado. Em outras palavras, trata-se de explorar os dados por conta própria, buscando padrões que melhor caracterizam certos conjuntos de dados (Faceli *et al.*, 2021).

Análises de agrupamento pertencem ao campo de aprendizado não supervisionado e consistem em obter uma partição, juntando dados em grupos (*clusters*) baseados nas características em comum que estes dados possuem (Faceli *et al.*, 2021). Um exemplo pode ser um conjunto de dados que possuem diversas espécies de animais vertebrados, dentre eles mamíferos, anfíbios, peixes, aves e anfíbios, e, dentre as características dos dados, possuir atributos como a capacidade de voar, tomar leite durante o desenvolvimento, qual o órgão responsável pela respiração, etc. Dependendo do valor destes atributos, algumas espécies, por terem mais características em comum, tendem a ficar mais próximas umas das outras, formando grupos. Técnicas de agrupamento tentam encontrar esses grupos utilizando diversos paradigmas distintos.

Algoritmos de agrupamento geralmente se enquadram nas seguintes categorias principais: algoritmos hierárquicos, algoritmos particionais, baseados em densidade e baseados em *grid*, como mencionado por Facelli em seu trabalho (Faceli *et al.*, 2021). De forma concisa, os algoritmos hierárquicos geram conjuntos de partições aninhadas, enquanto os algoritmos particionais produzem uma única partição como resultado com n *clusters* baseado em critérios de otimização. Já os baseados em densidade reconhecem *clusters* como uma área com alta concentração de objetos, e os baseados em *grid*, como o próprio nome diz, utiliza-se de um *grid* para fazer o agrupamento e detectar *outliers* (Faceli *et al.*, 2021). Esses conceitos serão explorados de maneira mais detalhada na seção de revisão bibliográfica (Capítulo 2), com foco especial nos algoritmos hierárquicos abordados na subseção 2.1.1.

Os algoritmos hierárquicos encontram aplicação em diversos campos, abrangendo

biologia, psicologia, medicina, marketing e ciência da computação, conforme indicado por Jain (Jain; Dubes, 1988). Devido à natureza hierárquica desses algoritmos, seus resultados frequentemente são representados por dendrogramas, uma estrutura amplamente empregada nas áreas de ciências biológicas e médicas, conforme discutido por Faceli *et al.* (2021). Exemplos notáveis de abordagens interdisciplinares que integram a computação e a biologia são vistos em diversos trabalhos da literatura (Li *et al.*, 2017; Li *et al.*, 2020; Teitz *et al.*, 2023).

Entretanto, partições de diferentes níveis hierárquicos não podem ser utilizadas em simultâneo, o que pode ocasionar em um problema na extração de grupos (Anjos, 2018). Dito isso, esforços são feitos para transpassar esse impecílio e, dentre eles, existe o *FOSC*, método que se utiliza de métricas sobre o conjunto de dados para selecionar partições em diferentes níveis hierárquicos proposto em Campello *et al.* (2013). Este trabalho irá abordar mais sobre este tema, discutindo sobre as métricas utilizadas no *FOSC* para extração ótima de partições em hierarquias.

1.1 Justificativa

Técnicas de agrupamento são um ramo do aprendizado de máquina não supervisionado. Sua aplicação pode ser encontrada em muitos campos como biologia, psicologia, medicina, marketing e, logicamente, computação (Jain; Dubes, 1988). Em computação, métodos de agrupamento podem ser utilizados em diversas áreas como extração de características, reconhecimento de voz e processamento de imagens (Jain; Dubes, 1988).

Diante disto, diversos algoritmos foram criados para realizar a tarefa de agrupamento, para obtenção de partições de maneira precisa. Dentre eles, existe uma categoria chamada de algoritmos hierárquicos, os quais produzem uma hierarquia de partições dos dados como resultado de sua execução (Faceli *et al.*, 2021). Entretanto, um dos problemas de algoritmos hierárquicos é que partições formadas em diferentes níveis hierárquicos não podem ser utilizadas em simultâneo (Anjos, 2018). Para resolver esse problema, é necessário realizar cortes horizontais em diferentes níveis hierárquicos, de modo que o resultado desse corte produza os melhores grupos possíveis.

Uma iniciativa notável na busca por extrair grupos de uma hierarquia de partições resultou na formulação do método denominado *FOSC*. Este método emprega uma medida

de qualidade predefinida, denominada estabilidade (*Stability*), para avaliar e selecionar os *clusters* mais adequados dentro de um conjunto de dados (Campello *et al.*, 2013). Em outro estudo, Anjos (2018) introduziu uma medida de qualidade, denominada Modularidade Q, voltada para a extração de grupos, utilizando o método *FOSC*. No entanto, esse trabalho não abordou os modelos tradicionais de construção de hierarquias, como o *Single linkage* e o *Average linkage*, entre outros.

Portanto, dedicar esforços à exploração dessa área proporciona a oportunidade de realizar uma análise mais precisa, bem como uma seleção mais eficaz dos melhores *clusters*. Essa dedicação contribui para impulsionar avanços significativos no processamento e interpretação de conjuntos de dados, abrindo caminho para um entendimento mais profundo e abrangente.

1.2 Objetivos

O principal objetivo desse trabalho envolve analisar o desempenho de uma métrica baseada em modularidade de redes complexas, chamada de *Modularidade Q*, como medida a ser avaliada no algoritmo *FOSC* para extração ótima de *clusters*, a partir de hierarquias fornecidas por modelos tradicionais de agrupamento hierárquico.

Além disso, este trabalho também tem como objetivos específicos:

- Implementar e disponibilizar a medida de qualidade Modularidade Q.
- Comparar a medida Modularidade Q e a medida originalmente utilizada no método FOSC (*Stability*).

1.3 Organização do Trabalho

O restante do trabalho está organizado da seguinte forma: o Capítulo 2 mostra uma contextualização sobre o tema, dando definições sobre os modelos e apresentação dos trabalhos relacionados. O Capítulo 3 apresenta a metodologia proposta para os experimentos do trabalho, enquanto o Capítulo 4 os resultados obtidos. Por fim, o Capítulo 5 apresenta as conclusões e perspectivas de trabalhos futuros.

2 Revisão Bibliográfica

2.1 Fundamentação Teórica

Os algoritmos de agrupamento podem ser classificados de acordo com os paradigmas utilizados para formação de *clusters* (Faceli *et al.*, 2021). Uma dessas classes de algoritmos são os algoritmos hierárquicos. Nesta seção apresentamos os conceitos principais sobre agrupamento hierárquico, além dos métodos de extração de grupos a partir de hierarquia de grupos.

2.1.1 Algoritmos de agrupamento Hierárquico

Os algoritmos hierárquicos pertencem a uma categoria de métodos que utilizam métricas de distância (ou similaridade) entre os dados para realizar agrupamentos. Esses métodos podem ser subdivididos em dois tipos: agrupamento hierárquico divisivo e agrupamento hierárquico aglomerativo. No caso divisivo, é utilizada uma estratégia *top-down*, onde todos os objetos em um conjunto de dados começam juntos em um único *cluster* e, através de divisões sucessivas, cada objeto acaba em seu próprio *cluster* (*singleton*). No caso aglomerativo, o processo é o oposto, ou *bottom-up*, começando com cada objeto em um *cluster* individual (*singleton*) e terminando quando todos os objetos fazem parte do mesmo *cluster* (Faceli *et al.*, 2021).

Os algoritmos de agrupamento hierárquico geram uma sequência de partições aninhadas, que também são chamadas de hierarquia de agrupamentos (Faceli *et al.*, 2021). Essas hierarquias frequentemente são representadas visualmente por meio de dendrogramas, que são estruturas em forma de árvore que ilustram a progressão passo a passo na formação da hierarquia de agrupamentos.

Em relação aos algoritmos hierárquicos aglomerativos, várias abordagens são utilizadas para calcular a distância entre dois *clusters*, como o ***Single-Linkage***, o ***Complete-Linkage*** e o ***Average-Linkage***. Segundo Jain e Dubes (1988), na abordagem *Single-Linkage*, a distância entre dois *clusters* é determinada pela menor distância entre dois ob-

jetos pertencentes aos *clusters* comparados (Jain; Dubes, 1988). Na abordagem *Complete-Linkage*, a distância entre dois *clusters* é definida como a maior distância entre dois objetos pertencentes aos *clusters* comparados. Na abordagem *Average-Linkage*, a distância entre dois *clusters* é calculada como a média das distâncias entre todos os pares de objetos pertencentes aos *clusters* (Jain; Dubes, 1988).

Uma das vantagens do agrupamento hierárquico é a sua flexibilidade em relação ao nível de detalhamento dos agrupamentos, permitindo a utilização de diferentes formas de similaridade ou distâncias. Por outro lado, existem algumas desvantagens, como a falta de um critério de término bem definido e a limitação de que a maioria dos algoritmos não melhora os agrupamentos depois de construídos. Isso significa que um objeto só muda de cluster se o cluster que ele foi atribuido for fundido com outro (perspectiva *bottom-up*) ou o cluster ao qual ele faz parte for dividido em outros clusters (perspectiva *top-down*) (Faceli *et al.*, 2021).

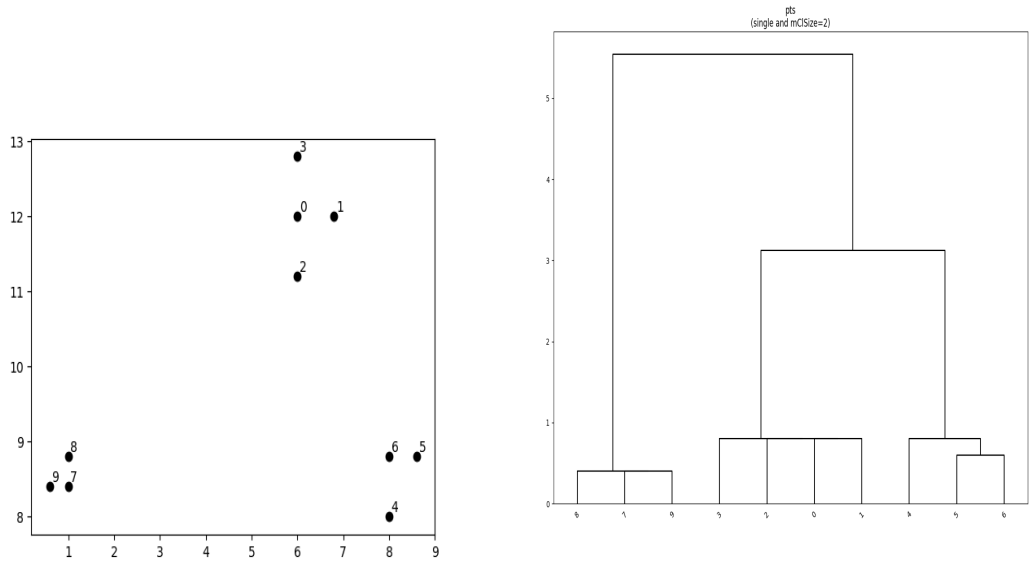
2.1.2 *Framework* para Extração Ótima de Clusters em Hierarquias (FOSC)

O Framework para Extração Ótima de Clusters em Hierarquias (FOSC) foi proposto por Campello *et al.* (2013) e utiliza o paradigma de programação dinâmica para definir uma partição ótima de *clusters*. Este método toma por base uma hierarquia simplificada de *clusters* e uma medida de qualidade especificada para cada grupo e, assim, realizar a extração ótima.

Em geral, não é fornecida diretamente esta hierarquia simplificada. Então, para obter tal hierarquia, Campello *et al.* (2013) propõem a utilização de algoritmos hierárquicos aglomerativos para produzirem os dendrogramas e utilização desses dendrogramas para geração da hierarquia simplificada. Durante este processo de simplificação é definido um parâmetro m_{clSize} que representa o número mínimo de observações que um *cluster* deve possuir.

A Figura 2.1 mostra um exemplo de conjunto de dados e seu respectivo dendrograma, obtido por meio da execução do algoritmo hierárquico usando *Single Linkage*. No dendograma, o eixo x representa os objetos do conjunto de dados, enquanto o eixo y representa a escala onde cada agrupamento foi se formando. A partir dele, é possível

observar que até a escala 5,55, todos os objetos pertencem ao mesmo *cluster*, neste caso o *cluster* 1. Após o nível 5,55 o *cluster* 1 é subdividido nos clusters 2 e 3, onde ambos possuem um número mínimo de objetos maior ou igual a 2 ($M_{ClSize} \geq 2$). A partir do nível 3,12, o *cluster* 2 se subdivide nos *clusters* 4 e 5 que duram até o nível 0,60 e 0,80, respectivamente. Vale a pena ressaltar que ao atingir o nível 0,80, o objeto x_4 deixa de pertencer ao *cluster* 4. O *cluster* 3 dura até o nível 0,40 e é a partição que dura mais tempo (com exceção do *cluster* 1, que é o raiz). O resultado desta simplificação é sumarizado na Tabela 2.1



(a) Conjunto de dados

(b) Dendrograma obtido por meio do método de ligação *Single linkage*

Figura 2.1 – Exemplo de um conjunto de dados e sua respectiva estrutura hierárquica.

Fonte: Elaborado pelo autor.

Dada a hierarquia simplificada e uma medida de qualidade, $\Gamma(\mathbf{C}_i)$ para cada i-ésimo *cluster* da hierarquia, o algoritmo FOSC é executada da seguinte forma:

1. Inicializar um vetor que corresponde ao conjunto solução (conjunto δ) com **n** posições correspondentes ao número de elementos da hierarquia de grupos. O vetor deve ser preenchido em todas as posições com o valor binário **1**;

Tabela 2.1 – Hierarquia de grupos produzida a partir do dendrograma gerado pelo método de *Single Linkage*.

Escala	$\mathbf{x}_0$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	$\mathbf{x}_9$
5.55	1	1	1	1	1	1	1	1	1	1
3.12	2	2	2	2	2	2	2	3	3	3
0.80	5	5	5	5	4	4	4	3	3	3
0.60	0	0	0	0	0	4	4	3	3	3
0.40	0	0	0	0	0	0	0	3	3	3
0.00	0	0	0	0	0	0	0	0	0	0

Fonte: Elaborado pelo autor

2. Retirar o *cluster* raiz da solução, colocando o valor **0** na posição correspondente;

3. Inicializar um vetor $\hat{\Gamma}$, onde somente as posições referentes aos nós folhas serão preenchidas com os valores das métricas calculadas, $\Gamma(C_i)$;

4. Para cada nó interno C_i:

   a) Compara-se o valor da métrica do nó analisado com os valores de $\hat{\Gamma}$ de seus nós filhos;

      i. Se $\Gamma(C_i) > \hat{\Gamma}(C_{il}) + \hat{\Gamma}(C_{ir})$, retirar todos os nós pertencentes a subárvores de C_i da solução;

      ii. Caso contrário, retirar o nó C_i da solução;

   b) Atribuir $\hat{\Gamma}(C_i) = \max\{\Gamma(C_i), \hat{\Gamma}(C_{il}) + \hat{\Gamma}(C_{ir})\}$;

5. Repetir o passo 4 até atingir o nó raiz;

Ao final desse processo, é obtido os clusters correspondentes a seleção ótima de partições, dado pelos clusters que possuem valor 1 no conjunto δ. Exemplos de execução do *FOSC* serão descritos posteriormente nas subseção 2.1.3.1 e subseção 2.1.3.2

2.1.3 Medidas de qualidade

Como mencionado na seção anterior, para realizar a extração ótima dos grupos no FOSC, se faz necessária a utilização de uma medida de qualidade, que deve possuir

as seguintes características: (i) aditividade: uma medida é descrita como aditiva se e somente se o valor dela puder ser decomposto em uma soma de dois ou mais fatores; (ii) localidade: uma medida é considerada como local se e somente se o valor da medida puder ser calculado baseando-se em informações obtidas somente pelo *cluster* analisado, sem precisar de informações de outros *clusters* para obter o valor da medida.

2.1.3.1 Estabilidade

No trabalho de Campello *et al.* (2013), é proposta uma medida de qualidade denominada estabilidade, que é dada pelo tempo em que cada objeto permaneceu em um determinado *cluster* $\mathbf{C}_i$. Matematicamente, pode-se definí-la na seguinte forma:

$$S(C_i) = \sum_{x_j \in C_i} \textit{lifetime}(x_j), \tag{2.1}$$

Sendo o *lifetime* do objeto x_j no cluster C_i corresponde à diferença do nível que ele começa a fazer parte de $\mathbf{C}_i$ com o nível que ele deixa de fazer parte do mesmo *cluster*. Por exemplo, no caso da medida de estabilidade, levando em consideração o exemplo da Figura 2.1 e de sua respectiva estrutura hierárquica Tabela 2.1, pode-se concluir que $S(C_4)$ = *lifetime*(x_4) + *lifetime*(x_5) + *lifetime*$(x_6) = (3,12 - 0,80) + (3,12 - 0,60) + (3,12 - 0,60) \approx 7,36$. A Figura 2.2 mostra com maior clareza o valor de estabilidade computado para cada um dos *clusters* da hierarquia de grupos apresentadas na Tabela 2.1.

Ao ser executado, o *FOSC* desconsideraria da solução a partição raiz, dado pelo *cluster* $\mathbf{C}_1$, considerando somente seus filhos. Após isso, o algoritmo atualizaria o $\hat{\Gamma}(\mathbf{C}_i)$ para todos os *clusters* que são folhas, que no caso são $\mathbf{C}_3$, $\mathbf{C}_4$ e $\mathbf{C}_5$. Seguindo o passo a passo descrito na subseção 2.1.2, é necessário iterar em cada *cluster* interno da hierarquia, verificando se a medida dele é maior que a soma da medida de seus filhos. No caso do exemplo, seria comparado se a medida de $\mathbf{C}_2$, que é $16,95$, é maior que a soma das medidas de $\mathbf{C}_4$ e $\mathbf{C}_5$, que são, respectivamente, $7,37$ e $9,29$. Como $7.37 + 9.29 = 16.66$, tal afirmação é verdadeira e os *clusters* $\mathbf{C}_4$ e $\mathbf{C}_5$ são retirados da solução, além de ter o valor de medida atualizado $\hat{\Gamma}$ como $16,95$. Após isso, o próximo nó a ser analisado é o $\mathbf{C}_1$. Como ele é o nó raiz, para-se a execução do *FOSC*, retornando como extração ótima os clusters $\mathbf{C}_2$ e $\mathbf{C}_3$.

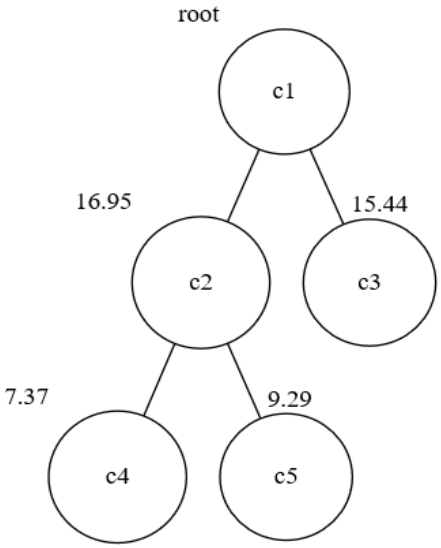


Figura 2.2 – Representação gráfica da árvores de grupos apresentada na Tabela 2.1.

Fonte: Elaborado pelo autor.

2.1.3.2 Mod-Knn

Baseado nas propriedades de aditividade e localidade, Anjos *et al.* (2019) propõem uma nova medida de qualidade e a aplicação dela para seleção ótima em hierarquia de partições. Essa métrica é denominada de *Mod-Knn* e baseia-se em métodos de seleção de comunidades de redes complexas.

Em redes complexas, a medida de qualidade para detecção de comunidades ou grupos mais utilizada é denominada *Modularity Q* (Modularidade Q em português) (Anjos, 2018). Essa métrica é utilizada em redes as quais as arestas não são ponderadas. Fazendo-se uma adaptação para a versão que aceita pesos nas arestas, a Modularidade Q, agora ponderada, pode ser descrita pela seguinte equação:

$$Q_s(P) = \sum_{i=1}^{k} \left(\frac{IS(\mathbf{C}_i)}{TS} - \left(\frac{DS(\mathbf{C}_i)}{TS} \right)^2 \right) \tag{2.2}$$

Sendo $\mathbf{P} = \{\mathbf{C}_1, \mathbf{C}_2, ..., \mathbf{C}_k\}$ é uma partição contendo comunidades $\mathbf{C}_i$ que são disjuntas, k é o total de comunidades, IS($\mathbf{C}_i$) é a soma dos pesos das arestas internas à comunidade $\mathbf{C}_i$, ou seja, os dois vértices que compõem a aresta pertencem à comunidade $\mathbf{C}_i$, DS($\mathbf{C}_i$) é a soma dos pesos das arestas que possuem ao menos um vértice em $\mathbf{C}_i$ e TS é a soma

dos pesos de todas as arestas da rede.

Anjos (2018) mostram que a Modularidade Q ponderada pode ser utilizada como medida de estabilidade do algoritmo *FOSC*. A propriedade aditividade é cumprida observando a estrutura da Equação 2.2, a qual é um somatório do valor de cada comunidade. Em relação a propriedade de localidade, ela é cumprida, uma vez que os valores dos dados necessários para o calculo de cada comunidade são obtidos sem precisar de dados de outros *clusters*. O valor de IS($\mathbf{C}_i$) é obtido baseado em dados da própria comunidade, já que ambos os vértices pertencem a aquele grupo, e o valor de DS($\mathbf{C}_i$), apesar de envolverem outras comunidades, não necessita delas para realizar o calculo, já que somente é necessário saber de um dos vértices daquela aresta faz parte da comunidade (Anjos, 2018).

Tendo isso em vista, é necessário criar um grafo de similaridade contendo as informações da base de dados para o calculo da métrica modularidade Q ponderada, uma vez que ela só funciona em redes complexas. Em redes complexas, similaridade em grafos ponderados é dado pela seguinte equação:

$$\sigma(u,v) = \frac{\sum_{x \in \Phi_u \cap \Phi_v} w(u,x)w(v,x)}{\sqrt{\sum x \in \Phi_u w^2(u,x)} + \sqrt{\sum x \in \Phi_v w^2(v,x)}} \tag{2.3}$$

Sendo $\sigma(u,v)$ é a similaridade entre os vértices *u* e *v*, Φ_u é o conjunto de vértices conectados ao vértice *u* e $w(u,x)$ é o peso da aresta que liga os vértices *u* e *x*.

Para criar tal grafo, começa-se obtendo o grafo de dissimilaridade G_d da base de dados. Tal grafo corresponde a todos os objetos do conjunto de dados e arestas ligando todos os vértices, cujo peso é igual a distância de dissimilaridade dos vértices (Anjos, 2018). Em seguida, transforma-se o grafo G_d em um grafo de similaridade G_s, uma vez que, na equação 2.3, os pesos das arestas tratam-se dos valores de similaridade dos vértices. A transformação é dada pela seguinte equação:

$$s(u,v) = 1 - \left(\frac{d(u,v)}{d_{max}}\right) \tag{2.4}$$

Sendo $s(u,v)$ é um valor entre 0 e 1 que corresponde a similaridade entre dois objetos *u* e *v*, $d(u,v)$ é a distância de dissimilaridade entre dois objetos *u* e *v* e d_{max} é a maior distância encontrada entre dois objetos (Anjos, 2018).

Nesse ponto, ainda é impossível aplicar a Equação 2.2 e calcular a modularidade Q ponderada, uma vez que todos os objetos do conjunto de dados estão conectados entre si, o que não faz sentido no calculo da modularidade, que é utilizado para detectar comunidades nas redes. Para resolver esse problema, Anjos (2018) propõem fazer um corte nas arestas do grafo, criando um subgrafo de G_s, denominado G_{KNN}. Nesse subgrafo, é mantido somente as arestas dos k objetos mais similares para cada ponto. Segundo Anjos (2018), o melhor valor para k é o número mínimo de objetos para se ter em um cluster (m_{clSize}).

Após a obtenção do grafo G_{KNN}, é possível obter o grafo de similaridade desejado, aplicando a Equação 2.3 para se obter a similaridade ponderada dos dados e o grafo resultante G_σ (Anjos, 2018). Por fim, basta executar o algoritmo hierárquico para gerar as partições e calcular o valor da métrica *Mod-Knn* de cada cluster utilizando-se da equação Equação 2.2.

Assim como apresentado para a medida de qualidade estabilidade, a Figura 2.3 apresenta a árvore de grupos para o exemplo da Figura 2.1 e sua respectiva estrutura hierárquica 2.1.

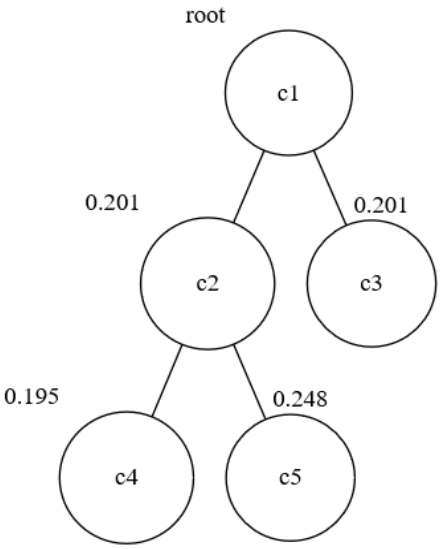


Figura 2.3 – Cluster Tree da hierarquia 2.1 utilizando-se da *Mod-Knn* como valor de métrica de estabilidade do algoritmo *FOSC*.

Fonte: Elaborado pelo autor.

Ao ser executado, o *FOSC* desconsideraria da solução a partição raiz, dado pelo

cluster $\mathbf{C}_1$, considerando somente seus filhos. Após isso, o algoritmo atualizaria o $\hat{\Gamma}(\mathbf{C}_i)$ para todos os *clusters* que são folhas, que no caso são $\mathbf{C}_3$, $\mathbf{C}_4$ e $\mathbf{C}_5$. Seguindo o passo a passo descrito na subseção 2.1.2, é necessário iterar em cada *cluster* interno da hierarquia, verificando se a medida dele é maior que a soma da medida de seus filhos. No caso do exemplo, seria comparado se a medida de $\mathbf{C}_2$, que é $0,201$, é maior que a soma das medidas de $\mathbf{C}_4$ e $\mathbf{C}_5$, que são, respectivamente, $0,195$ e $0,248$. Como $0,195 + 0,248 = 0,443$, tal afirmação é falsa e o *cluster* $\mathbf{C}_2$ é retirado da solução, além de ter o valor de métrica atualizado $\hat{\Gamma}$ como $0,443$. Após isso, o próximo nó a ser analisado é o $\mathbf{C}_1$. Como ele é o nó raiz, para-se a execução do *FOSC*, retornando como extração ótima os *clusters* $\mathbf{C}_3$, $\mathbf{C}_4$ e $\mathbf{C}_5$.

2.2 Trabalhos Relacionados

No trabalho de Campello *et al.* (2013), é apresentado um algoritmo para extração de grupos de clusters ótimos utilizando de hierarquias pré-calculadas, apelidado de *FOSC (Framework for Optimal Selection of Clusters)*. No *FOSC*, é utilizado um algoritmo de agrupamento hierárquico para extrair as hierarquias de cada cluster. Após isso, é feito uma análise baseado numa métrica, chamada de estabilidade, para descobrir quais são os clusters mais promissores para representar o conjunto de dados. No trabalho de Campello *et al.* (2013), a métrica utilizada é o tempo de vida do cluster, ou seja, qual é a duração do cluster na escala de hierarquia dos dados. O trabalho apresenta que o *FOSC* possue resultados promissores, principalmente comparando-o com algoritmos especificos em seus respectivos cenários (Campello *et al.*, 2013).

No trabalho de Anjos *et al.* (2019), é apresentado uma nova métrica utilizada a ser utilizada no *FOSC* para extração ótima de clusters. A métrica escolhida foi a *Mod-Knn*, baseada na *Modularity Q* (modularidade Q em português). A modularidade Q é amplamente utilizada no contexto de detecção de comunidades em redes complexas e pode ser aplicada como critério de otimização no problema de extração de clusters ótimos utilizando-se de hierarquias (Anjos *et al.*, 2019). Segundo os autores, a *Mod-Knn* é uma alternativa viável para a Estabilidade proposta em Campello *et al.* (2013), a qual a primeira mostra comportamentos mais robustos que a segunda.

Zhao e Zhang (2011) propuseram em seu trabalho um novo método de agrupa-

mento baseado em detecção de cliques em uma rede complexa. Um clique é um subgrafo de uma rede complexa em que todos os vértices pertencentes a esse subgrafo estão conectados entre si, ou seja, é um grafo completo (Goldbarg, 2012). Como a detecção de cliques máximos (de maior tamanho possível) em um grafo é um problema NP-Completo, os autores utilizam uma heurística para fazer essa detecção. Em seu método de agrupamento, os autores também utilizam de conceitos como densidade de um subgrafo, que corresponde a proporcionalidade de arestas e vértices de um subgrafo e a contribuição de um vértice a uma comunidade para a geração das partições (Zhao; Zhang, 2011). Por fim, Zhao e Zhang (2011) avaliam se um vértice contribui o suficiente para fazer parte de uma comunidade já criada, adicionando-o se ele contribuir o necessário. Tal calculo é baseado na densidade da comunidade e de parâmetros passados pelo usuário (Zhao; Zhang, 2011).

O estudo de Gertrudes *et al.* (2019) introduz um novo *framework* que combina algoritmos de clustering baseados em densidade com abordagens de aprendizado semi-supervisionado. Os autores investigam a interação entre esses métodos, propondo uma estratégia inovadora para a classificação eficaz de dados. Eles enfatizam o papel crucial do algoritmo central generalizado, HDBSCAN*, na facilitação do clustering semi-supervisionado dentro desse contexto. Esse trabalho apresenta uma visão integrada e promissora para a aplicação de técnicas baseadas em densidade em cenários de clustering e classificação semi-supervisionados.

Em suma, Zhao e Zhang (2011), assim como este trabalho, trabalham na análise de agrupamento em um conjunto de dados. Entretanto, em sua proposta, Zhao e Zhang (2011) não trabalha com algoritmos hierárquicos e, consequentemente, extração ótimas de grupos em hierarquias. Já Campello *et al.* (2013), Anjos *et al.* (2019) e Gertrudes *et al.* (2019) estão extremamente atrelados a este trabalho, uma vez que as propostas feitas pelos dois primeiros (*FOSC*, estabilidade como métrica do *FOSC* e *Mod-Knn*) serão utilizadas. A diferença é que, tanto Campello *et al.* (2013) e Anjos *et al.* (2019) não utilizaram algoritmos hierárquicos puros na construção de suas hierarquias. Ambos utilizam o *HDBSCAN* (Hierarchical Density-Based Spatial Clustering of Applications with Noise Star)*, o qual consiste em um algoritmo que aborda aspectos hierárquicos, mas que constroem as partições baseado em densidade. Neste trabalho, os algoritmos de agrupamento a serem utilizados serão os algoritmos de agrupamento puro. Já Gertrudes

et al. (2019), será utilizado de conjunto de dados que foram pré-processados nesse artigo.

3 Desenvolvimento

Como apresentado na seção 1.2, o objetivo principal desta proposta envolve a comparação entre as medidas de qualidade *stability* (estabilidade) e *Mod-Knn* na definição de partições em algoritmos de agrupamento hierárquicos aglomerativos. Para realização do trabalho, foram definidas as seguintes etapas, apresentadas na seção a seguir.

3.1 Conjunto de dados

Foram utilizados 29 conjuntos de dados distintos para realização dos experimentos, descritos na Tabela 3.1. Os conjuntos ACE ECFP4, ACE ECFP6, COX2 ECPF6, DHFR ECFP4, DHFR ECFP6, Fontaine ECFP4, Fontaine ECFP6,M1 ECFP4 e M1 ECFP6 representam moleculas e buscam encontrar relação entre estrutura molecular e atividade biológica (Gertrudes *et al.*, 2019). Analcatdata authorship corresponde um conjunto de dados com 841 objetos e 70 atributos, onde os objetos correspondem a livros que devem ser categorizados para qual autor escreveu o respectivo objeto. Armstrong-v1, Chowdary, Gordon e Yeast Galactose são conjuntos de dados que representam análises de características genéticas para detectar câncer. Wdbc consiste em analise de células para detectar câncer de mama. Iris contêm 3 classes com 5 atributos cada, onde cada classe se refere um tipo de planta da espécie Íris. Seeds corresponde a um conjunto de dados para classificar 3 espécies de trigos baseados nos atributos dos mesmos. Wine é proveniente de uma análise química de vinhos cultivados na mesma região na Itália, mas provenientes de três cultivares diferentes, sendo composto de 13 atributos e 3 classes. Todos os conjuntos foram pré-processados e, neste trabalho, os mesmos conjuntos utilizados em Gertrudes *et al.* (2019) serão utilizados na experimentação aqui. Os conjunto de dados estão disponíveis em <https://github.com/jadsoncastro/UnifiedView/tree/master/data>.

Além disso, a Tabela 3.1 mostra que, para cada conjunto de dados, existe uma medida de distância distinta utilizada. Tais escolhas de distâncias foram feitas de acordo com Gertrudes *et al.* (2019) e, segundo eles, tais distâncias são aquelas que apresentarão melhor resultado para cada base de dados.

3.2 Métodos de ligação

Durante os experimentos serão utilizados, a princípio, um algoritmo hierárquico aglomerativo puro, utilizando *Single linkage*, *Complete linkage* e *Average linkage* como métodos de ligação para geração dos dendrogramas que serão utilizados para geração da hierarquia de grupos e posterior extração pelo método FOSC.

3.3 Parâmetros do *FOSC*

Vale lembrar que na geração da hierarquia de grupos faz-se o uso do parâmetro m_{clSize} que determina a quantidade mínima de objetos que um *cluster* deve possuir. Neste caso, realizaremos os experimentos seguindo o valor padrão determinado em (Campello *et al.*, 2013), com $m_{clSize} = 4$. Além disso, os valores de m_{clSize} serão variados entre os valores de 4, 8, 12, 16 e 20.

3.4 Métrica de Validação

A medida de validação a ser utilizada nos experimentos será o índice Rand ajustado (ARI), proposto em Hubert e Arabie (1985). Este índice compara um conjunto de *clusters* (partição) gerados por um algoritmo de agrupamento e uma partição verdadeira (*ground truth*) do conjunto de dados. Na literatura de agrupamento de dados, este indice é considerado um indice de validação externa, pois utiliza de resultados externos para avaliar a qualidade de um agrupamento. Para o cálculo deste índice, são consideradas as seguintes variáveis:

- a que representa o número de objetos no mesmo *cluster* tanto no gerado pelo algoritmo de agrupamento quanto no *cluster* do *ground truth*;
- b que representa o número de objetos que estão nos mesmos *clusters* no *ground truth* mas em *clusters* distintos no resultado gerado pelo algoritmo de agrupamento;
- c que representa o número de objetos que estão em diferentes *clusters* no *ground truth* mas nos mesmos *clusters* no resultado gerado pelo algoritmo de agrupamento;

- d que representa o número de objetos em *cluster* distintos tanto no gerado pelo algoritmo de agrupamento quanto no *cluster* do *ground truth*;

Matematicamente, o índice de Rand ajustado é dado por

$$ARI = \frac{a - \frac{(a+c)(a+b)}{M}}{\frac{(a+c)+(a+b)}{2} - \frac{(a+c)(a+b)}{M}}, \tag{3.1}$$

onde $M = a + b + c + d = \frac{N(N-1)}{2}$. O ARI uma medida que se estende na faixa de -1 a 1. Valores próximos de zero, ou menores, indicam que qualquer similaridade entre as divisões de conjuntos ocorre devido a acasos aleatórios, enquanto o valor 1 denota que as divisões são exatamente iguais, ou seja, objetos na mesma partição no *ground truth* estão na mesma partição no resultado do algoritmo de agrupamento a ser validado (Faceli *et al.*, 2021). Embora o índice possa teoricamente chegar ao mínimo de -1, na prática, esse ponto não é alcançado. Normalmente, quando as divisões são significativamente discrepantes, o índice tende a se aproximar de 0. Apesar de ser possível ter um limite inferior de 0 rigidamente estabelecido, a normalização necessária para tal refinamento não leva a nenhuma vantagem, uma vez que valores negativos não têm relevância nenhuma (Faceli *et al.*, 2021).

3.5 Método de análise dos resultados obtidos

Feito isso, será utilizado um método estatístico para análise dos resultados obtidos, descritos em Demšar (2006). Este método é o teste dos postos sinalizados de Wilcoxon, criado em 1945, o qual irá criar *ranks* para as diferenças obtidas pelos resultados obtidos por cada métrica (*modknn* e *stability*) para cada base de dados. Tal método é uma alternativa ao teste T-pareado, uma vez que, segundo Demšar (2006), o teste T-pareado possui três fraquezas:

- O teste estatistico T só faz sentido quando a diferença entre as bases de dados são mensuráveis
- Caso não haja bases de dados suficientes (segundo Demšar (2006) mais que 30), a diferença entre as bases de dados tem que possuir distribuição normal.

- O teste T-pareado sofre com *outliers*, reduzindo seu poder dedutivo

O experimento proposto não cumpre com os itens 2 e 3, uma vez que são utilizados menos de 30 datasets e possui *outliers* nos resultados, uma vez que houve resultados bons e ruins dependendo do *dataset*.

Portanto, o teste dos postos sinalizados de Wilcoxon surge como uma alternativa, uma vez que não precisa que os resultados estejam na distribuição Gaussiana e é menos suscetível a *outliers*.

Tabela 3.1 – Lista de conjuntos de dados coletados para realizar os experimentos de agrupamento não supervisionados.

Datasets	**#objetos**	**#atributos**	**#classes**	**Distância**
ACE ECFP4	114	1025	2	Tanimoto
ACE ECFP6	114	1025	2	Tanimoto
Analcatdata authorship	841	70	4	Cosseno
Armstrong-v1	72	1082	2	Cosseno
Auto Price	159	16	2	Euclidiana
Bank note–Authentication	1372	5	2	Euclidiana
Cardiotocography	2126	36	10	Euclidiana
Chowdary	104	183	2	Cosseno
Chcase Geyser1	222	2	2	Euclidiana
COX2 ECFP6	322	1025	2	Tanimoto
DHFR ECFP4	397	1025	2	Tanimoto
DHFR ECFP6	397	1025	2	Tanimoto
Diggle table	310	8	9	Euclidiana
Fontaine ECFP4	435	1024	2	Tanimoto
Fontaine ECFP6	435	1024	2	Tanimoto
Gordon	181	1627	2	Cosseno
Iris	150	5	3	Euclidiana
M1 ECFP4	769	1025	2	Tanimoto
M1 ECFP6	769	1025	2	Tanimoto
Mfeat-factors	2000	216	10	Euclidiana
Mfeat-Karhunen	2000	65	10	Euclidiana
Seeds	210	8	3	Euclidiana
Segmentation	2100	20	7	Euclidiana
Semeion	1593	256	10	Cosseno
Stock	950	10	2	Euclidiana
Transplant	131	4	2	Euclidiana
WDBC	569	32	2	Euclidiana
Wine	178	13	3	Euclidiana
Yeast galactose	205	81	4	Euclidiana

Fonte: Adaptado de Gertrudes *et al.* (2019).

4 Resultados

4.1 Análise de Resultados

Como descrito no Capítulo 3, foi realizado os agrupamentos para cada método de ligação, utilizando as medidas de qualidade ModKNN e *Stability* para extração ótima de gupos no framework *FOSC*. Os resultados "brutos" são apresentados nas Tabelas A.1, A.2, A.3, A.4, A.5 e A.6. Vale ressaltar que nos conjuntos de dados com classificação de moléculas, percebeu-se um desempenho não favorável de ambos os algoritmos de agrupamento.

Ademais, as Figuras 4.1, 4.2, 4.3 mostram o desempenho médio das métricas obtidas em cada método de ligação: *single*, *average* e *complete*, respectivamente. Observando e analisando os gráficos, percebe-se que o desempenho dos métodos vão melhorando a medida que o m_{Clsize} aumenta, além de que o desempenho médio do *FOSC* com a modknn foi ligeiramente melhor que a com *stability*. Verificando os resultados brutos, percebe-se que, em geral, a combinação do método de ligação *average linkage* e do $m_{Clsize} = 20$ produziu a melhor média de resultados para as medidas de qualidade estudadas.

4.2 Teste estatístico

Para uma verificação de uma possível diferença estatística entre as medidas de qualidade utilizadas para extração de grupos, selecionamos, para cada base de dados, o melhor desempenho do *FOSC* para cada medida, *stability* e mod-knn, visando uma comparação justa entre as métricas. Com isso, sumarizamos os resultados na Tabela 4.1, que demonstra qual foi a melhor combinação de m_{ClSize} e método de ligação que obteve o melhor desempenho em cada *dataset*.

Com o intuito de sumarizar os resultados, foi confeccionada a Tabela 4.2. Nela, é possível observar que o mod-knn obteve 15 vitórias, 7 empates e 7 derrotas ao utilizar todos os datasets e, realizando o teste de postos de Wilcoxon como descrito em Capítulo 3,

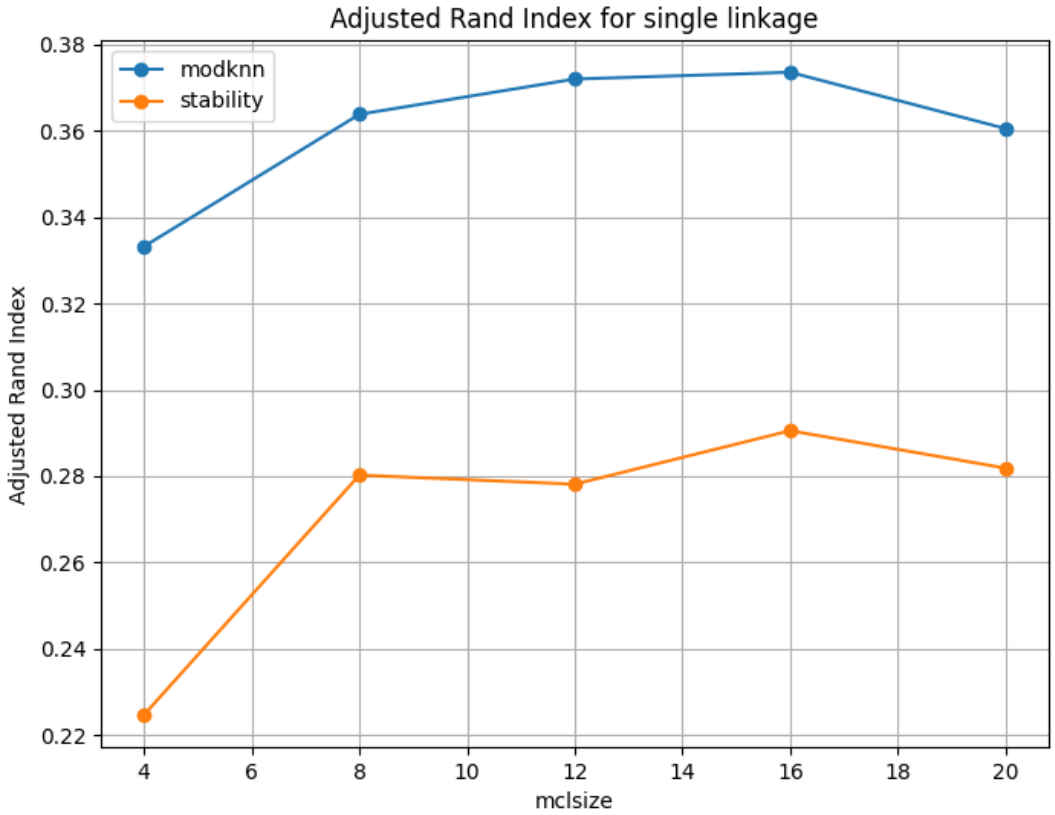


Figura 4.1 – Resultados agregados para o método de *single linkage*

Fonte: Elaborado pelo autor.

obteve-se que os resultados obtidos pelo *FOSC* utilizando mod-knn como medidade de qualidade possui diferença relevante o suficiente em relação a medida *stability* para afirmar que ela possui um desempenho superior nessas bases de dados, com 96% de confiança, indicado pelo valor de p-value encontrado (0,0327).

Entretanto, percebe-se que em muitos casos houve desempenho não ideal dos algoritmos utilizando tanto a *stability* quanto a modknn. Isso indica, como descrito na Capítulo 3, que os agrupamentos encontrados podem ser considerados aleatórios, sem nenhum relacionamento com o resultado ideal. Portanto, visando contornar esse problema, também foi analisado os casos onde o ARI obteve somente valores maiores ou iguais a 0,5 em qualquer uma das duas métricas. Feito isso, é possível analisar, também na

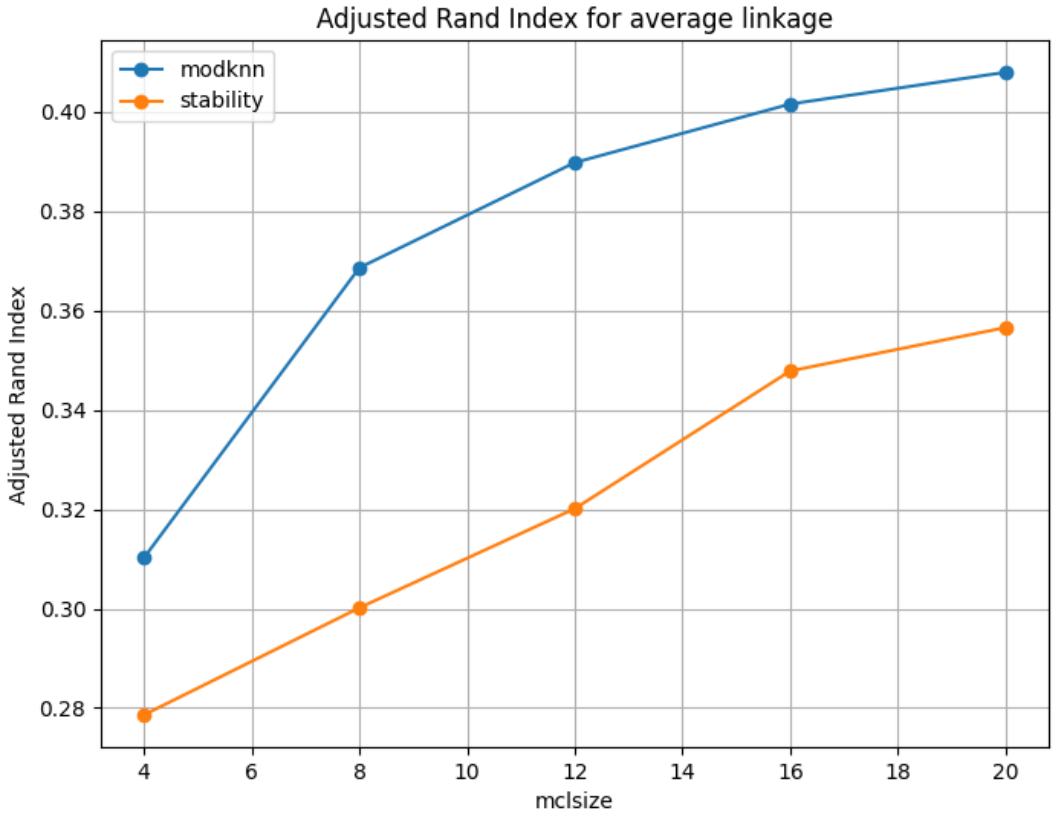


Figura 4.2 – Resultados agregados para o método de *average linkage*

Fonte: Elaborado pelo autor.

Tabela 4.2, que a mod-knn obteve 9 vitórias, 3 empates e 4 derrotas. Realizando o mesmo teste estatístico, obteve-se um valor de p-value de 0,0970, o que indica que, com 90% de confiança, o mod-knn possui desempenho superior nessas bases de dados em relação a *stability*.

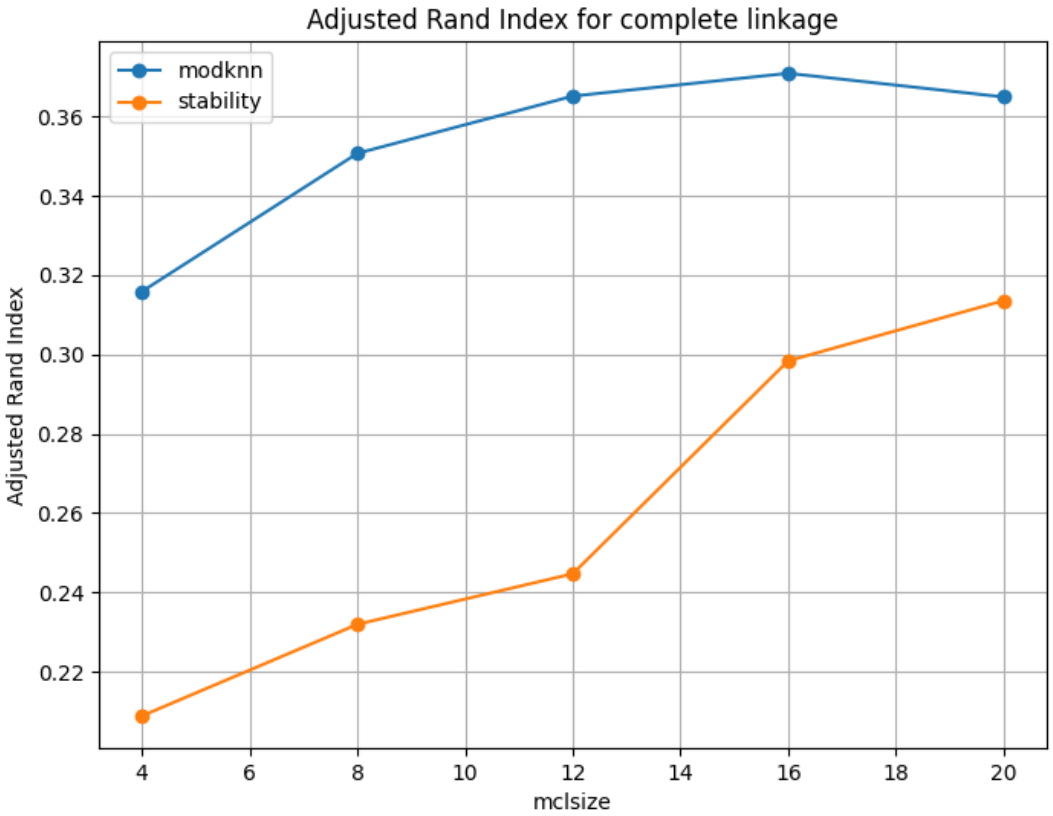


Figura 4.3 – Resultados agregados para o método de *complete linkage*

Fonte: Elaborado pelo autor.

Tabela 4.1 – Síntese dos resultados obtidos

Dataset	Modknn	Modknn Params	Stability	Stability Params
ace_ECFP_4	0.332	complete_linkage_mclsize_4	0.278	single_linkage_mclsize_12
ace_ECFP_6	0.456	single_linkage_mclsize_20	0.456	single_linkage_mclsize_20
analcatdata	0.847	complete_linkage_mclsize_4	0.973	average_linkage_mclsize_4
armstrong2002v1	0.407	complete_linkage_mclsize_4	0.233	complete_linkage_mclsize_8
autoPrice	0.692	complete_linkage_mclsize_4	0.502	complete_linkage_mclsize_4
banknote-authentication	0.143	single_linkage_mclsize_16	0.100	average_linkage_mclsize_4
cardiotocography	0.947	single_linkage_mclsize_12	0.527	average_linkage_mclsize_20
chowdary2006	0.924	single_linkage_mclsize_4	0.924	single_linkage_mclsize_16
chscase_geyser1	0.308	single_linkage_mclsize_8	0.310	average_linkage_mclsize_4
cox2_ECFP_6	0.077	single_linkage_mclsize_8	0.077	single_linkage_mclsize_8
dhfr_ECFP_4	0.088	average_linkage_mclsize_20	0.089	complete_linkage_mclsize_16
dhfr_ECFP_6	0.090	complete_linkage_mclsize_12	0.087	average_linkage_mclsize_20
diggle_table	0.884	single_linkage_mclsize_20	0.959	single_linkage_mclsize_8
fontaine_ECFP_4	0.189	average_linkage_mclsize_20	0.189	average_linkage_mclsize_20
fontaine_ECFP_6	0.201	average_linkage_mclsize_12	0.201	average_linkage_mclsize_20
gordon2002	0.805	average_linkage_mclsize_12	0.805	average_linkage_mclsize_8
iris	0.759	average_linkage_mclsize_20	0.568	single_linkage_mclsize_4
m1_ECFP_4	0.094	single_linkage_mclsize_4	0.104	average_linkage_mclsize_8
m1_ECFP_6	0.070	average_linkage_mclsize_8	0.056	single_linkage_mclsize_12
mfeat-factors	0.757	average_linkage_mclsize_20	0.603	average_linkage_mclsize_20
mfeat-karhunen	0.811	average_linkage_mclsize_20	0.581	average_linkage_mclsize_12
seeds	0.617	average_linkage_mclsize_16	0.515	average_linkage_mclsize_4
segmentation-normcols	0.542	single_linkage_mclsize_16	0.274	complete_linkage_mclsize_16
semeion	0.502	average_linkage_mclsize_8	0.404	average_linkage_mclsize_16
stock	0.143	complete_linkage_mclsize_20	0.140	single_linkage_mclsize_4
transplant	0.346	complete_linkage_mclsize_12	0.665	average_linkage_mclsize_4
wdbc	0.308	single_linkage_mclsize_16	0.538	complete_linkage_mclsize_12
wine-187	0.810	average_linkage_mclsize_8	0.810	average_linkage_mclsize_8
yeast_Galactose	0.978	average_linkage_mclsize_8	0.872	average_linkage_mclsize_16

Tabela 4.2 – Tabela contendo o número de vitórias, empates e derrotas da mod-knn x *stability*

	Mod-KNN x *Stability*			
datasets	**Vitórias**	**Empates**	**Derrotas**	**p-value (Wilcoxon)**
Datasets reais	15	7	7	0,0327
datasets reais onde ARI $\geq 0,50$	9	3	4	0,0970

5 Considerações Finais

5.1 Conclusão

Neste trabalho, foi descrito métodos de extração de agrupamentos em conjuntos de dados. No campo do aprendizado descritivo, tais métodos são de suma importância e permitem a viabilização e realização de diversas atividades em diferentes campos de pesquisa, seja ela no âmbito da computação ou não.

Nos métodos descritos, foi dado um enfoque aos algoritmos hierárquicos e a meios de extração de partições dessas hierarquias, de modo com que a extração dos *clusters* fosse ótima. Um dos algoritmos com esse viés é o *FOSC*.

O *FOSC* recebe uma hierárquia de partições e, baseado numa métrica de estabilidade, tenta achar um conjunto ótimo de partições, escolhendo os grupos que máximizem o valor máximo da métrica de estabilidade. Na implementação inicial do FOSC, a métrica escolhida foi a *lifetime* da partição, que corresponde ao intervalo na estrutura hierárquica que aquele agrupamento existiu antes de se dividir em novos agrupamentos.

Outra implementação do *FOSC* utiliza-se de uma métrica chamada *Mod-Knn* para realizar a seleção das partições ótimas. Tal métrica basea-se na modularidade Q de redes complexas, a qual representa o quão bem separado em grupos uma rede está. Na adaptação do uso da modularidade Q para o contexto de agrupamento de dados, é necessário criar um grafo de similaridade dos dados, para assim ser possível a aplicação do calculo da Modularidade Q e, consequentemente, utiliza-la como métrica de estabilidade do *FOSC*.

Tendo isso em vista, foi realizado experimentos, comparando o desempenho do *FOSC* utilizando as duas métricas descritas acima em bases de dados reais. Para fazer tal comparação e análise de resultados, foi utilizado o teste dos postos sinalizados de Wilcoxon e obteve-se um resultado positivo, mostrando que o *FOSC* utilizando a *Mod-Knn* como métrica obteve desempenhos superiores do que utilizando a métrica *stability* com, no mínimo, 90% de confiança.

5.2 Trabalhos Futuros

Uma possível continuidade desse trabalho é a avaliação de desempenho do *FOSC* junto a mod-knn em bases de dados que contenham uma características de ser derivados de redes complexas. A mod-knn é derivada do conceito de Modularidade Q, proveniente do campo de redes complexas e tal conexão pode vir a ser um ponto interessante a ser explorado em trabalhos futuros.

Além disso, outra possível trabalho futuro é analisar os agrupamentos obtidos por esses métodos usando métricas de validação intrínsecas, uma vez que nesse trabalho foi utilizado o ARI, que consiste em uma métrica extrínseca. Desta forma, seria possível analisar se os agrupamentos obtidos pelos métodos tiveram sentido (por exemplo agrupar elementos próximos no mesmo cluster enquanto elementos mais afastados estao em clusters distintos) mesmo em conjunto de dados onde o valor de ARI foi baixo.

Referências

ANJOS, F. d. A. R. d. **Seleção de grupos a partir de hierarquias: uma modelagem baseada em grafos**. Tese (Doutorado) — Universidade de São Paulo, 2018.

ANJOS, F. d. A. R. dos *et al.* A modularity-based measure for cluster selection from clustering hierarchies. In: SPRINGER. **Data Mining: 16th Australasian Conference, AusDM 2018, Bahrurst, NSW, Australia, November 28–30, 2018, Revised Selected Papers 16**. [S.l.], 2019. p. 253–265.

CAMPELLO, R. J. *et al.* A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies. **Data Mining and Knowledge Discovery**, Springer, v. 27, p. 344–371, 2013.

DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. **The Journal of Machine learning research**, JMLR. org, v. 7, p. 1–30, 2006.

FACELI, K. *et al.* **Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina**. [s.n.], 2021. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788521637509>.

GERTRUDES, J. C. *et al.* A unified view of density-based methods for semi-supervised clustering and classification. **Data mining and knowledge discovery**, Springer, v. 33, p. 1894–1952, 2019.

GOLDBARG, M. **Grafos**. [s.n.], 2012. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788595155756>.

HUBERT, L.; ARABIE, P. Comparing partitions. **Journal of classification**, Springer, v. 2, p. 193–218, 1985.

JAIN, A. K.; DUBES, R. C. **Algorithms for clustering data**. [S.l.]: Prentice-Hall, Inc., 1988.

LI, H. *et al.* Classifying drosophila olfactory projection neuron subtypes by single-cell rna sequencing. **Cell**, Elsevier, v. 171, n. 5, p. 1206–1220, 2017.

LI, H. *et al.* Single-cell transcriptomes reveal diverse regulatory strategies for olfactory receptor expression and axon targeting. **Current Biology**, Elsevier, v. 30, n. 7, p. 1189–1198, 2020.

TEITZ, J. *et al.* Potential of dissimilarity measure-based computation of protein thermal stability data for determining protein interactions. **Briefings in Bioinformatics**, Oxford University Press, v. 24, n. 3, p. bbad143, 2023.

ZHAO, P.; ZHANG, C.-Q. A new clustering method and its application in social networks. **Pattern Recognition Letters**, v. 32, n. 15, p. 2109–2118, 2011. ISSN 0167-8655. Disponível em: <https://www.sciencedirect.com/science/article/pii/S016786551100184X>.

Apêndices

APÊNDICE A – Resultado completo obtidos durante a execução dos experimentos

Tabela A.1 – Resultados obtidos com *Single Linkage* e *modknn*

Datasets	**mclsize**				
	4	**8**	**12**	**16**	**20**
ace_ECFP_4	0.1746	0.2454	0.2775	0.2775	0.2775
ace_ECFP_6	0.1993	0.2462	0.3013	0.3013	0.4561
analcatdata_authorship-458	0.5509	0.5442	0.5442	0.5442	0.5442
armstrong2002v1	0.3257	0.1352	0.1352	0.1352	0.1352
autoPrice	0.4628	0.4628	0.4569	0.4497	0.4497
banknote-authentication	0.0667	0.0879	0.1406	0.1426	0.1426
cardiotocography	0.7243	0.9422	0.9468	0.9468	0.9468
chowdary2006	0.9238	0.9238	0.9238	0.9238	0.9238
chscase_geyser1	0.1229	0.3080	0.3080	0.3080	0.3080
cox2_ECFP_6	0.0702	0.0766	0.0660	0.0687	0.0687
dhfr_ECFP_4	0.0609	0.0481	0.0484	0.0460	0.0491
dhfr_ECFP_6	0.0727	0.0663	0.0659	0.0667	0.0726
diggle_table	0.6196	0.7860	0.7925	0.8416	0.8844
fontaine_ECFP_4	0.0794	0.1049	0.1049	0.1042	0.1042
fontaine_ECFP_6	0.0875	0.1075	0.1075	0.1060	0.1060
gordon2002	0.7188	0.7188	0.7188	0.7188	0.0
iris	0.5645	0.6850	0.6850	0.5681	0.5681
m1_ECFP_4	0.0938	0.0234	0.0083	0.0369	0.0057
m1_ECFP_6	0.0583	0.0306	0.0557	0.0557	0.0557
mfeat-factors	0.4993	0.4969	0.4958	0.4958	0.4958
mfeat-karhunen	0.5333	0.5319	0.5235	0.5235	0.5235
seeds	0.2003	0.3773	0.3773	0.3773	0.5042
segmentation-normcols	0.3772	0.4227	0.5404	0.5415	0.5389
semeion	0.2410	0.2089	0.2009	0.1974	0.1982
stock	0.0517	0.0720	0.0801	0.0762	0.1162
transplant	0.1731	0.2052	0.2052	0.3047	0.3047
wdbc	0.2086	0.2769	0.3014	0.3081	0.3081
wine-187	0.5490	0.5636	0.5930	0.5930	0.5930
yeast_Galactose	0.8524	0.8524	0.7826	0.7737	0.7737

Tabela A.2 – Resultados obtidos com *Single Linkage* e *stability*

Datasets	**mclsize**				
	4	**8**	**12**	**16**	**20**
ace_ECFP_4	0.1942	0.2454	0.2775	0.2775	0.2775
ace_ECFP_6	0.2038	0.2462	0.3013	0.3013	0.4561
analcatdata_authorship-458	0.5029	0.4951	0.4951	0.4951	0.5442
armstrong2002v1	0.1352	0.1352	0.1352	0.1352	0.1352
autoPrice	0.1002	0.1002	0.1938	0.1896	0.2796
banknote-authentication	0.0117	0.0117	0.0124	0.0124	0.0124
cardiotocography	0.0015	0.1633	0.1633	0.1633	0.1633
chowdary2006	0.6620	0.6708	0.6708	0.9238	0.9238
chscase_geyser1	-0.0044	-0.0044	-0.0044	-0.0044	-0.0044
cox2_ECFP_6	0.0584	0.0766	0.0660	0.0660	0.0687
dhfr_ECFP_4	0.0116	0.0130	0.0081	0.0095	0.0095
dhfr_ECFP_6	0.0116	0.0130	0.0081	0.0095	0.0095
diggle_table	0.9454	0.9590	0.8890	0.9322	0.8844
fontaine_ECFP_4	0.0527	0.0557	0.0743	0.0696	0.0696
fontaine_ECFP_6	0.0997	0.0984	0.0984	0.0957	0.1060
gordon2002	0.7091	0.7188	0.7188	0.7188	0.0
iris	0.5681	0.5681	0.5681	0.5681	0.5681
m1_ECFP_4	-0.0139	0.0335	0.0369	0.0671	0.0057
m1_ECFP_6	-0.0298	0.0222	0.0557	0.0	0.0
mfeat-factors	0.00003	0.4815	0.4958	0.4958	0.4958
mfeat-karhunen	0.00016	0.3729	0.3725	0.3725	0.3725
seeds	0.3593	0.3271	0.3271	0.3271	0.4237
segmentation-normcols	0.2441	0.2442	0.2442	0.2442	0.2442
semeion	0.0011	0.1813	0.0092	0.0096	0.1824
stock	0.1404	0.1404	0.1364	0.1364	0.1364
transplant	0.1618	0.2052	0.2052	0.3047	0.3047
wdbc	0.1139	0.2769	0.3014	0.3081	0.3081
wine-187	0.4234	0.4234	0.4234	0.4234	0.4234
yeast_Galactose	0.8524	0.8524	0.7826	0.7737	0.7737

Tabela A.3 – Resultados obtidos com *Average Linkage* e *modknn*

Datasets	**mclsize**				
	4	**8**	**12**	**16**	**20**
ace_ECFP_4	0.1502	0.1818	0.1969	0.1969	0.1969
ace_ECFP_6	0.1590	0.1590	0.2429	0.2429	0.2429
analcatdata_authorship-458	0.6190	0.6291	0.6291	0.8032	0.8032
armstrong2002v1	0.3534	0.2102	0.2102	0.2102	0.2102
autoPrice	0.3683	0.3683	0.3654	0.3654	0.3654
banknote-authentication	0.0405	0.0535	0.0789	0.0885	0.0913
cardiotocography	0.2516	0.5604	0.6438	0.6519	0.7362
chowdary2006	0.9238	0.9238	0.9238	0.9238	0.9238
chscase_geyser1	0.0985	0.1348	0.1831	0.1956	0.1956
cox2_ECFP_6	0.0167	0.0214	0.0194	0.0308	0.0308
dhfr_ECFP_4	0.0505	0.0679	0.0745	0.0825	0.0878
dhfr_ECFP_6	0.0403	0.0661	0.0663	0.0755	0.0873
diggle_table	0.5195	0.6020	0.6770	0.6770	0.5200
fontaine_ECFP_4	0.0940	0.1110	0.1075	0.1326	0.1891
fontaine_ECFP_6	0.1224	0.1316	0.2011	0.2011	0.2011
gordon2002	0.8040	0.8040	0.8046	0.8046	0.8046
iris	0.4165	0.5912	0.6622	0.6622	0.7592
m1_ECFP_4	0.0244	0.0492	0.0503	-0.0497	-0.0497
m1_ECFP_6	0.0252	0.0696	0.0259	0.0259	0.0259
mfeat-factors	0.6243	0.6870	0.7464	0.7369	0.7571
mfeat-karhunen	0.6208	0.7562	0.7886	0.7880	0.8105
seeds	0.5224	0.5224	0.5224	0.6170	0.6170
segmentation-normcols	0.2024	0.3074	0.3425	0.4606	0.4684
semeion	0.4106	0.5018	0.4885	0.4885	0.4906
stock	0.0376	0.0587	0.0714	0.0836	0.0889
transplant	0.1656	0.2343	0.2733	0.3424	0.3424
wdbc	0.0813	0.0979	0.1184	0.1239	0.1507
wine-187	0.4842	0.8100	0.8100	0.8100	0.8100
yeast_Galactose	0.7692	0.9785	0.9785	0.8725	0.8725

Tabela A.4 – Resultados obtidos com *Average Linkage* e *stability*

Datasets	**mclsize**				
	4	**8**	**12**	**16**	**20**
ace_ECFP_4	0.0981	0.1893	0.1969	0.1969	0.0946
ace_ECFP_6	0.1614	0.1590	0.0946	0.0946	0.0946
analcatdata_authorship-458	0.9734	0.9734	0.9734	0.9734	0.9734
armstrong2002v1	0.2102	0.2102	0.2102	0.2102	0.2102
autoPrice	0.1002	0.1002	0.3654	0.3654	0.3654
banknote-authentication	0.1001	0.1001	0.1001	0.1001	0.1001
cardiotocography	0.0015	0.4393	0.4509	0.4870	0.5270
chowdary2006	0.5340	0.6855	0.6855	0.6855	0.9238
chscase_geyser1	0.3101	0.3101	0.3101	0.3101	0.3101
cox2_ECFP_6	0.0253	0.0255	0.0240	0.0308	0.0308
dhfr_ECFP_4	0.0605	0.0663	0.0745	0.0878	0.0878
dhfr_ECFP_6	0.0662	0.0648	0.0663	0.0755	0.0873
diggle_table	0.1897	0.1897	0.1897	0.1897	0.1897
fontaine_ECFP_4	0.1078	0.0505	0.1075	0.1075	0.1891
fontaine_ECFP_6	0.1061	0.1069	0.1838	0.1812	0.2011
gordon2002	0.7844	0.8046	0.8046	0.8046	0.8046
iris	0.5681	0.5681	0.5681	0.5681	0.5681
m1_ECFP_4	0.0204	0.1038	-0.0543	-0.0497	-0.0497
m1_ECFP_6	-0.0051	-0.0202	-0.0395	-0.0395	-0.0395
mfeat-factors	0.5140	0.4404	0.5805	0.5714	0.6025
mfeat-karhunen	0.4508	0.4586	0.5811	0.5638	0.5249
seeds	0.5149	0.5149	0.5149	0.5149	0.5149
segmentation-normcols	0.1029	0.1029	0.1029	0.1023	0.1023
semeion	0.2469	0.2198	0.3511	0.4044	0.3763
stock	0.1154	0.1154	0.1154	0.1154	0.1154
transplant	0.6646	0.6646	0.6646	0.6646	0.6646
wdbc	0.0881	0.0882	0.0882	0.0882	0.0882
wine-187	0.8069	0.8100	0.8100	0.8100	0.8100
yeast_Galactose	0.1628	0.1628	0.1628	0.8725	0.8725

Tabela A.5 – Resultados obtidos com *Complete Linkage* e *modknn*

Datasets	**mclsize**				
	4	**8**	**12**	**16**	**20**
ace_ECFP_4	0.3323	0.3323	0.2582	0.2582	0.2582
ace_ECFP_6	0.1616	0.1616	0.2242	0.2582	0.2582
analcatdata_authorship-458	0.8474	0.8474	0.8474	0.8474	0.8474
armstrong2002v1	0.4069	0.1833	0.1833	0.1833	0.1833
autoPrice	0.6916	0.6916	0.6916	0.6916	0.6916
banknote-authentication	0.0474	0.0640	0.0819	0.0922	0.0954
cardiotocography	0.4536	0.4712	0.4684	0.4866	0.4864
chowdary2006	0.8505	0.8505	0.8505	0.8505	0.8505
chscase_geyser1	0.0993	0.1107	0.1853	0.2467	0.2467
cox2_ECFP_6	0.0185	0.0288	0.0288	0.0260	0.0326
dhfr_ECFP_4	0.0461	0.0798	0.0806	0.0845	0.0845
dhfr_ECFP_6	0.0373	0.0432	0.0899	0.0896	0.0754
diggle_table	0.5474	0.6054	0.5898	0.5870	0.4723
fontaine_ECFP_4	0.0902	0.1236	0.1491	0.1491	0.1471
fontaine_ECFP_6	0.0965	0.1319	0.1285	0.1285	0.1285
gordon2002	0.5327	0.7483	0.7483	0.7483	0.7483
iris	0.3944	0.6023	0.6023	0.6423	0.6423
m1_ECFP_4	0.0080	0.0087	0.0079	0.0142	0.0227
m1_ECFP_6	0.0085	0.0137	0.0042	0.0042	0.0042
mfeat-factors	0.5370	0.5458	0.5351	0.5351	0.5320
mfeat-karhunen	0.4788	0.4786	0.4775	0.4775	0.4939
seeds	0.3026	0.5443	0.5443	0.5443	0.5443
segmentation-normcols	0.2179	0.3086	0.4232	0.4307	0.4245
semeion	0.2384	0.2505	0.2548	0.2567	0.2549
stock	0.0394	0.0580	0.1120	0.1076	0.1430
transplant	0.2080	0.2156	0.3464	0.3464	0.3464
wdbc	0.1116	0.1215	0.1266	0.2197	0.1326
wine-187	0.5126	0.5934	0.5934	0.5911	0.5771
yeast_Galactose	0.8447	0.9572	0.9572	0.8600	0.8600

Tabela A.6 – Resultados obtidos com *Complete Linkage* e *stability*

Datasets	**mclsize**				
	4	**8**	**12**	**16**	**20**
ace_ECFP_4	0.1334	0.1623	0.2582	0.2582	0.2582
ace_ECFP_6	0.1498	0.1616	0.2242	0.0946	0.0946
analcatdata_authorship-458	0.5719	0.4889	0.2714	0.5955	0.9212
armstrong2002v1	0.1219	0.2328	0.1460	0.1833	0.1833
autoPrice	0.5024	0.5024	0.5024	0.5024	0.5024
banknote-authentication	0.0781	0.0781	0.0781	0.0781	0.0781
cardiotocography	0.0015	0.1477	0.1514	0.4508	0.4250
chowdary2006	0.2334	0.5445	0.5774	0.5774	0.5774
chscase_geyser1	0.2953	0.2953	0.2953	0.2953	0.2953
cox2_ECFP_6	0.0260	0.0278	0.0188	0.0354	0.0354
dhfr_ECFP_4	0.0544	0.0801	0.0673	0.0894	0.0845
dhfr_ECFP_6	0.0652	0.0656	0.0635	0.0762	0.0622
diggle_table	0.1897	0.1897	0.1897	0.1897	0.1897
fontaine_ECFP_4	0.0314	0.0450	0.1631	0.1609	0.1833
fontaine_ECFP_6	0.1037	0.1487	0.1560	0.1645	0.1645
gordon2002	0.3077	0.3125	0.5327	0.5327	0.5327
iris	0.4220	0.4220	0.4220	0.4220	0.4220
m1_ECFP_4	0.0073	-0.0148	-0.0118	-0.0142	-0.0110
m1_ECFP_6	0.0131	0.0203	0.0181	-0.0098	-0.0098
mfeat-factors	0.1246	0.1273	0.1300	0.3751	0.4558
mfeat-karhunen	0.1390	0.2149	0.2755	0.3249	0.3370
seeds	0.3239	0.3239	0.3239	0.3239	0.3239
segmentation-normcols	0.2709	0.2388	0.2388	0.2742	0.2742
semeion	0.0831	0.1203	0.1619	0.1862	0.2269
stock	0.1154	0.1154	0.1154	0.1154	0.1154
transplant	0.3967	0.3967	0.3967	0.3967	0.3967
wdbc	0.5351	0.5351	0.5377	0.5377	0.5377
wine-187	0.6095	0.5934	0.6427	0.5771	0.5771
yeast_Galactose	0.1493	0.1493	0.1493	0.8600	0.8600