Rodrigo Gama Baptista, Walter José Horning Junior

Uma análise experimental de algoritmos de agrupamento por densidade em dados de expressão gênica

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação e Informática Biomédica, setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação, Informática Biomédica*.

Orientador: Eduardo Jaques Spinosa.

Curitiba PR
2017

Resumo

Inúmeros métodos de classificação e seleção de características têm sido estudados para a identificação de genes expressos em dados de microarray. Métodos de classificação como SVM, redes neurais, *random forest*, K-means, entre outros, têm sido utilizados recentemete em diversos estudos. Apesar de existirem diversos métodos de classificação, ainda há uma falta de comparação entre esses métodos para encontrar uma melhor estrutura para classificação, agrupamento e análise de resultados de expressão de genes.

Este trabalho tem como objetivo apresentar uma visão geral sobre o assunto de expressão gênica, revisando conceitos de biologia celular e molecular, bem como, testar a eficiência de algoritmos de agrupamento por densidade em dados de expressão gênica. Devido ao grande aumento de dados gerados na área da bioinformática, torna-se inviável uma análise destes dados sem o auxílio de ferramentas computacionais e algoritmos modernos.

Revisamos pesquisas recentes na área de agrupamento e testamos dois algoritmos propostos de agrupamento por densidade em dados biológicos, a fim de examinar sua performance com dados tão complexos. A abordagem do primeiro algoritmo é baseada na ideia de que os centros de agrupamento são caracterizados por uma densidade maior do que seus vizinhos e por uma distância relativamente grande de pontos com densidades mais altas.

O segundo algoritmo utiliza da mesma base, porém explora uma técnica de computação evolutiva, conhecida como algoritmo genético, no qual a solução vai sendo construída com o passar de gerações (iterações do algoritmo).

Uma análise experimental é executada utilizando quatro *datasets* de expressão gênica, com o intuito de comparar a performance dos dois algoritmos.

A análise de *clusters* visa classificar os elementos em categorias com base na sua similaridade. Suas aplicações variam de astronomia, bioinformática, bibliometria, reconhecimento de padrões, etc.

Palavras-chave: agrupamento por densidade, bioinformática, expressão gênica.

Abstract

Numerous methods of classification and feature selection have been studied for the identification of gene expression in microarray data. Classification methods such as SVM, neural networks, random forest, K-means, among others, have been recently used in several studies.

Although there are several methods of classification, there is still a lack of comparison between these methods to find a better structure for classification, clustering and analysis of gene expression results. This work aims to present an overview on the subject of gene expression, reviewing concepts of cellular and molecular biology, as well as, to test the efficiency of two density-based algorithms in gene expression data. Due to the large increase of data generated in the area of bioinformatics, it becomes impossible to analyze these data without the help of modern computational tools and algorithms.

We have reviewed recent research in the area of clustering and have tested two proposed algorithms for density-based clustering in biological data in order to examine their performance in such complex data. The approach of the first algorithm is based on the idea that the clustering centers are characterized by a higher density than their neighbors and by a relatively large distance of points with higher densities.

The second algorithm uses the same base, but explores an evolutionary computing technique, known as Genetic Algorithm, in which the solution is constructed over generations (iterations of the algorithm).

An experimental analysis is performed using four gene expression datasets, in order to compare the performance of these two algorithms.

The analysis of clusters should classify the elements into categories based on their similarity. Their applications range from astronomy, bioinformatics, bibliometrics, pattern recognition, etc.

Keywords: density clustering, bioinformatics, gene expression.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

DINF	Departamento de Informática
UFPR	Universidade Federal do Paraná
DNA	Ácido desoxirribonucleico
RNA	Ácido ribonucleico
mRNA	Ácido ribonucleico mensageiro
tRNA	Ácido ribonucleico de transferência
rRNA	Ácido ribonucleico ribossômico

Lista de Símbolos

ρ_i	densidade local do algoritmo de Rodriguez
δ_i	distância de pontos com maior densidade do algoritmo de Rodriguez
ρ	fator de flexão do algoritmo do Gong

Capítulo 1

Introdução

O ácido desoxirribonucleico (DNA) é a molécula que carrega as instruções genenéticas de quase todos os seres vivos. Sua química única não só permite que essa informação seja copiada e transmitida aos descendentes de um organismo, mas também possibilita aos cientistas investigar e manipular um organismo a nível molecular. Como resultado, as técnicas de biologia molecular estão na vanguarda da pesquisa científica.

Definimos gene como uma sequência de DNA cromossômico que é necessária para a produção de um produto funcional, seja um polipeptídio ou uma molécula funcional de RNA. [Sperber e Sperber, 2008].

Portanto, os milhares de genes expressos em uma determinada célula determinam o que essa célula pode fazer. Além disso, cada passo no fluxo de informação de DNA para RNA para proteína fornece à célula um "ponto de controle"para auto-regulação de suas funções, ajustando a quantidade e o tipo de proteínas que produz.

Num dado momento, a quantidade de uma proteína particular numa célula reflete o equilíbrio entre as vias bioquímicas sintéticas e degradativas dessa proteína. No lado sintético desse equilíbrio, a produção de proteína começa na transcrição (DNA para RNA) e continua com a tradução (RNA para proteína). Assim, o controle destes processos desempenha um papel crítico na determinação de quais proteínas estão presentes numa célula e em que quantidades. Além disso, a maneira pela qual uma célula processa seus transcritos de RNA e proteínas recentemente produzidas também influencia grandemente os níveis de proteína.

Cluster é um grupo de objetos que pertence à uma mesma classe. Em outras palavras, objetos semelhantes são agrupados em um *cluster* e objetos diferentes são agrupados em outro *cluster*. Métodos de agrupamento são cruciais para a análise de dados de expressão gênica. Podem ajudar a desvendar novos tipos de câncer ou a identificar grupos de genes que respondem de forma semelhante a uma condição experimental específica. Portanto, agrupamento é uma tarefa principal para classificação e mineração de dados, além de uma técnica comum para análise de dados estatísticos, utilizada em diversos campos, incluindo aprendizado de máquina, reconhecimento de padrões, análise de imagens, recuperação de informações, bioinformática, compressão de dados e computação gráfica.

Utilizar a técnica de agrupamento em dados de expressão gênica é de grande valia, pois nos permite, dentre várias outras opções, agrupar genes co-regulados ou agrupar pessoas com uma expressão gênica semelhante. Podemos criar grupos de pessoas com uma doença em comum, por exemplo, e, ao agrupar a expressão gênica de um novo indivíduo, saber se este tem alguma chance de possuir alguma dessas doenças.

1.1 Objetivo geral

Este trabalho tem como objetivo comparar o comportamento de duas técnicas computacionais para agrupamento de dados em *datasets* de expressão gênica.

1.2 Objetivos específicos

- Analisar o comportamento do algoritmo proposto por Alex Rodriguez e Alessandro Laio em *Clustering by fast search and find of density peaks* [Rodriguez e Laio, 2014], cuja abordagem é o agrupamento de dados por densidade.
- Analisar o comportamento do algoritmo proposto por Maoguo Gong, Licheng Jiao, Ling Wang, and Liefeng Bo em *Density-Sensitive Evolutionary Clustering* [Gong et al., 2007], cuja abordagem é agrupar dados também por densidade, mas utilizando uma abordagem evolutiva.
- Testar ambos os algoritmos com *datasets* reais de expressão gênica.
- Comparar os resultados de ambos os algoritmos com dados de expressão gênica a fim de constatar a abordagem mais eficaz e como cada uma comporta-se em cada *dataset*.

1.3 Justificativa

Analisar dados de expressão gênica significa tentar compreender como realmente funciona o código genético "em ação", ou seja, nos permite ter uma visão da verdadeira forma com que o código genético se manifesta nas células em diferentes cenários e condições ambientais. Porém analisar dados de expressão gênica também significa analisar uma grande quantidade de dados. Por não ser uma técnica supervisionada, o agrupamento de dados se apresenta como uma alternativa poderosa para a compreensão de problemas complexos, como é a análise de dados de expressão gênica. Podemos destilar os dados para um nível mais compreensível, subdividindo os genes em um número menor de categorias e depois analisando esses grupos.

Devido à grande dimensionalidade e ao extenso conjunto de características presentes em *datasets* de expressão gênica, algoritmos tradicionais de agrupamento podem apresentar dificuldades na classificação desses tipos de dados. Existem várias abordagens diferentes

para agrupamento, dentre elas, as que utilizam a densidade como métrica de similaridade têm se destacado bastante pelo fato de conseguirem identificar *clusters* com formatos arbitrários, em distribuições bastante complexas, inviabilizando o uso de grande parte dos algoritmos de agrupamento [Gong et al., 2007]. A figura 1.1 mostra 7 exemplos de *datasets*. Distribuições como a (d) e (g), por exemplo, não retornam bons resultados quando agrupadas com métodos e métricas convencionais.

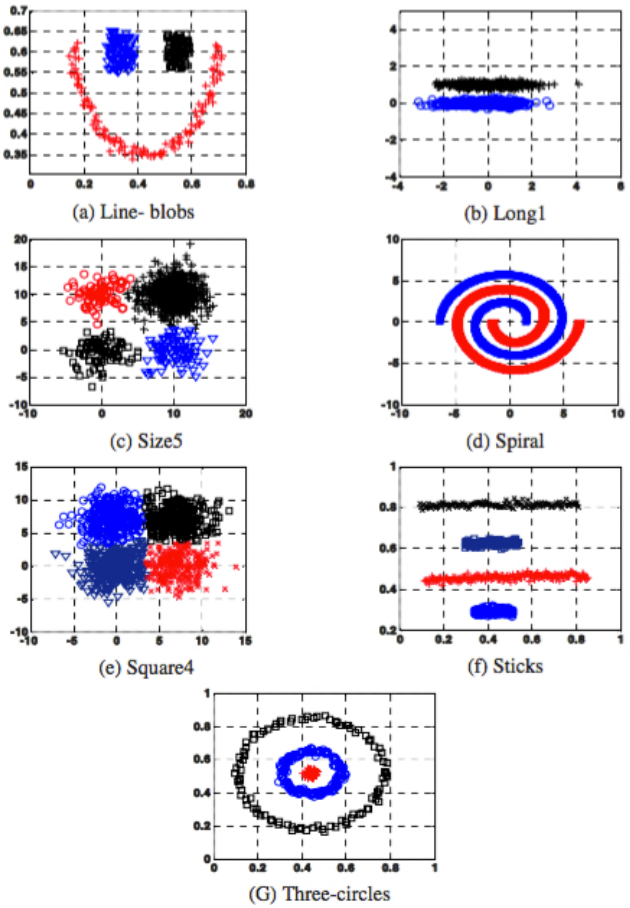

Figura 1.1: Exemplo de 7 *data sets* com diferentes formatos na distribuição dos dados. Fonte: [Gong et al., 2007]

Este trabalho apresenta dois algoritmos de agrupamento de dados, ambos baseados em densidade. Apesar de possuírem a mesma abordagem para a métrica de similaridade, são propostas com diferentes pontos de vista. Enquanto um algoritmo utiliza uma abordagem tradicional, mais aproximada ao descobrimento de *clusters* em grandes bancos de dados espaciais com ruído, o outro algoritmo utiliza uma metodologia bioinspirada evolutiva, a qual tem sua base nas ideias de seleção natural e evolução das espécies. Após, uma avaliação experimental comparativa entre esses dois algoritmos é aplicada, utilizando dados reais de expressão gênica.

1.4 Estrutura do documento

Este documento está estruturado de maneira a respeitar uma ordem com explicações teóricas sobre conceitos de biologia, bioinformática e técnicas de agrupamento de dados, seguido da apresentação de experimentos práticos propostos em duas linhas de pesquisa. Após, os resultados são discutidos e concluídos com possíveis trabalhos futuros.

No capítulo 2, inicia-se a fundamentação teórica e contextualização sobre expressão gênica e a definição de *clustering*. A primeira parte do capítulo explica o processo de expressão gênica, bem como suas etapas de transformação de informação genética em produto funcional, como RNA ou proteína. Após, definimos o que é *clustering* - ou agrupamento - e contextualizamos esta técnica em diversas áreas de estudo, incluindo a bioinformática, tema principal deste trabalho.

O capítulo 3 apresenta a revisão bibliográfica utilizada neste trabalho. Neste capítulo são apresentados dois algoritmos de agrupamento de dados por densidade que possuem uma variação em suas metodologias. A escolha destes algoritmos originou-se de recente estudo publicado na revista *Nature*, em novembro de 2015, intitulado *Comparing the performance of biomedical clustering methods* [Wiwie et al., 2015]. No artigo, Christian Wiwie apresenta uma comparação entre diversos métodos de agrupamento para dados biomédicos.

O capítulo 4, apresenta os conjuntos de dados de expressão gênica utilizados nos experimentos e a justificativa de escolha dos mesmos. Da mesma forma, também são expostas as métricas de avaliação da qualidade dos *clusters* em ambos os algoritmos. Dessa maneira, é possível variar os parâmetros necessários de ambos os algoritmos e posteriormente compará-los em diferentes esferas dentro de um mesmo critério.

Posteriormente, o capítulo 5 apresenta os resultados obtidos com ambos os algoritmos nos conjuntos de dados escolhidos. Diferentes parâmetros são utilizados e os resultados explanados em forma de tabelas e gráficos.

Por fim, o capítulo 6 indica a parte final do assunto discutido, no qual é feito um resumo final das ideias expostas e possíveis trabalhos futuros.

Capítulo 2

Fundamentação teórica

2.1 Expressão Gênica

A expressão gênica é o processo pelo qual o código genético - a sequência nucleotídica - de um gene é usado para direcionar a síntese protéica e produzir as estruturas da célula. Os genes que codificam as sequências de aminoácidos são conhecidos como "genes estruturais"[Sperber e Sperber, 2008].

O processo de expressão gênica envolve duas etapas principais:

- **Transcrição:** a produção de RNA mensageiro (mRNA) pela enzima RNA polimerase e o processamento da molécula de mRNA resultante.
- **Tradução:** o uso de mRNA para direcionar a síntese proteica e o posterior processamento pós-translacional da molécula proteica.

Alguns genes são responsáveis pela produção de outras formas de RNA que desempenham um papel na tradução, incluindo RNA de transferência (tRNA) e RNA ribossômico (rRNA) [Sperber e Sperber, 2008]. Um gene estrutural envolve uma série de componentes diferentes, como ilustra a Figura 2.1:

Figura 2.1: Estrutura do gene. Fonte: https://www.nature.com/scitable/topicpage/dna-transcription-426

- **Éxons:** Exons codificam para aminoácidos e coletivamente determinam a sequência de aminoácidos da proteína produzida. São estas porções do gene que são representadas na molécula de mRNA final.

- **Íntrons:** o uso de mRNA para direcionar a síntese proteica e o posterior processamento pós-translacional da molécula proteica.
- **Região de início:** Faz parte da região controladora do gene, indicando a posição para o início da transcrição.
- **Região promotora:** É uma região de algumas centenas de nucleotídeos do gene (em direção à extremidade 5'). Não é transcrito no mRNA, mas desempenha um papel no controle da transcrição do gene. Os fatores de transcrição ligam-se às sequências nucleotídicas específicas na região promotora e ajudam na ligação de RNA polimerases.
- **Acentuadores (*enhancers*):** Alguns fatores de transcrição (chamados de ativadores) ligam-se à regiões chamadas "intensificadoras"que aumentam a taxa de transcrição. Estes locais podem ter milhares de nucleotídeos das sequências de codificação ou dentro de um íntron. Alguns acentuadores são condicionais e só funcionam na presença de outros fatores, bem como fatores de transcrição.
- **Silenciadores (*silencers*):** Alguns fatores de transcrição (chamados de repressores) se ligam à regiões chamadas "silenciadoras"que minimizam a taxa de transcrição.

2.1.1 Transcrição

A transcrição é o processo de síntese de RNA, controlado pela interação de promotores e intensificadores. Vários tipos diferentes de RNA são produzidos, incluindo RNA mensageiro (mRNA), que especifica a sequência de aminoácidos na proteína produzida, além de RNA de transferência (tRNA) e RNA ribossômico (rRNA), que desempenham um papel no processo de tradução [Sperber e Sperber, 2008].

O processo de transcrição é constituído de quatro etapas. São elas:

- **Iniciação:** A síntese do RNA começa em regiões do DNA chamadas de regiões promotoras, que são sequências específicas reconhecidas pela RNA polimerase, e direcionam a transcrição de genes. A molécula de DNA desenrola-se e separa-se para formar um pequeno complexo aberto. A RNA-polimerase liga-se ao promotor da fita molde (*template strand*) [Sperber e Sperber, 2008]. A Figura 2.2 ilustra esta etapa.
- **Alongamento:** O RNA recém-sintetizado pareia-se temporariamente com a fita molde de DNA, formando um híbrido curto RNA-DNA. Uma vez iniciada, a transcrição segue numa velocidade de aproximadamente 50 nucleotídeos por segundo, estando a RNA polimerase ligada à fita molde de DNA até encontrar o sinal de término da transcrição. Em procariotos, a RNA polimerase é uma holoenzima constituída por várias subunidades, incluindo um **fator sigma** (fator de transcrição) que reconhece o promotor. Em eucariotos, há três RNA polimerases: I, II e III. O processo inclui um mecanismo de revisão. A Figura 2.3 ilustra esta etapa.

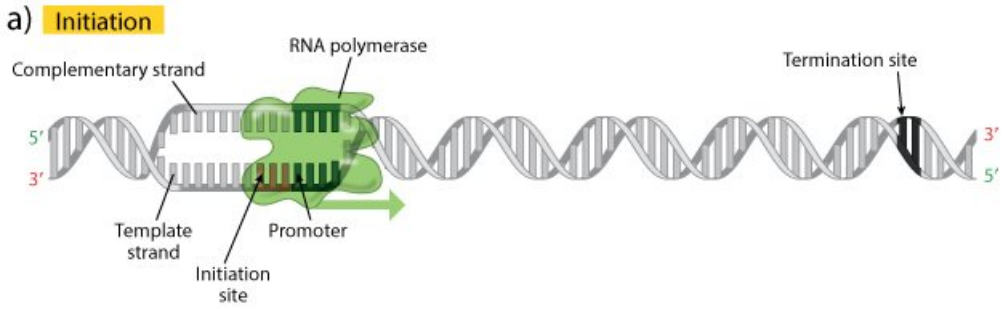

Figura 2.2: O processo de transcrição é iniciado quando a enzima RNA polimerase se liga a um molde de DNA numa região promotora. Fonte: https://www.nature.com/scitable/topicpage/dna-transcription-426

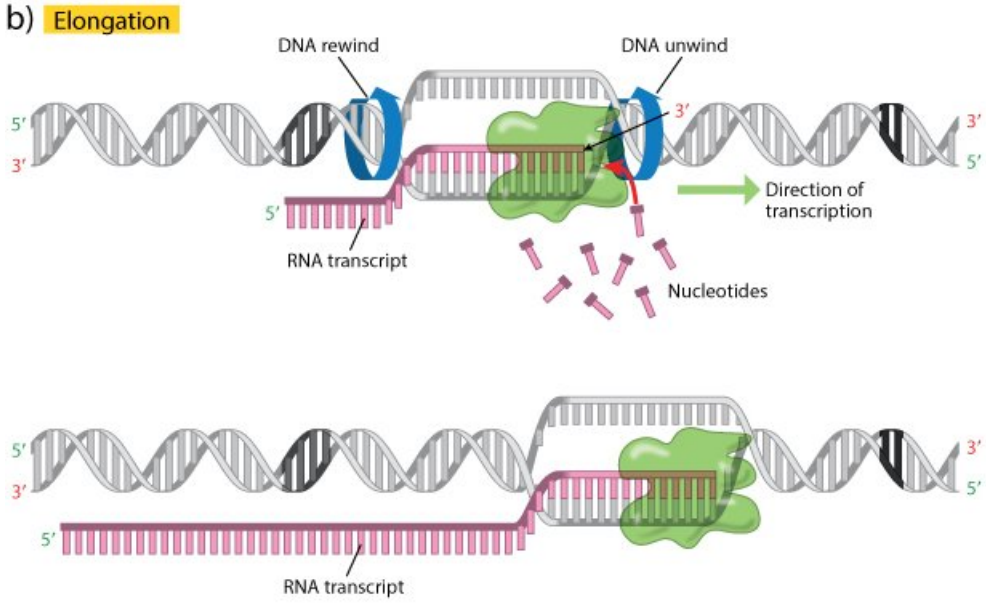

Figura 2.3: Durante o processo de alongamento, a dupla hélice de DNA desenrola-se. A RNA polimerase lê a cadeia da fita molde de DNA e adiciona nucleotídeos à extremidade 3' de um RNA transcrito em crescimento. Fonte: https://www.nature.com/scitable/topicpage/dna-transcription-426

- **Término:** O final da transcrição é um processo bem controlado, determinado pelo surgimento dos códons de parada ou de terminação, finalizando a síntese dessa molécula. Em procariotos, há duas maneiras pelas quais a transcrição é terminada. Na terminação *Rho-dependente*, um fator de proteína chamado "Rho"é responsável por interromper o complexo envolvendo a fita molde, a RNA-polimerase e a molécula de RNA. Em terminação *Rho-independente*, um loop forma no final da molécula de RNA, fazendo com que ele se desprenda. O processo de terminação em eucariotos é mais complicado, envolvendo a adição de nucleotídeos de adenina no 3' do RNA transcrito (um processo conhecido como poliadenilação). A Figura 2.4 ilustra esta etapa.

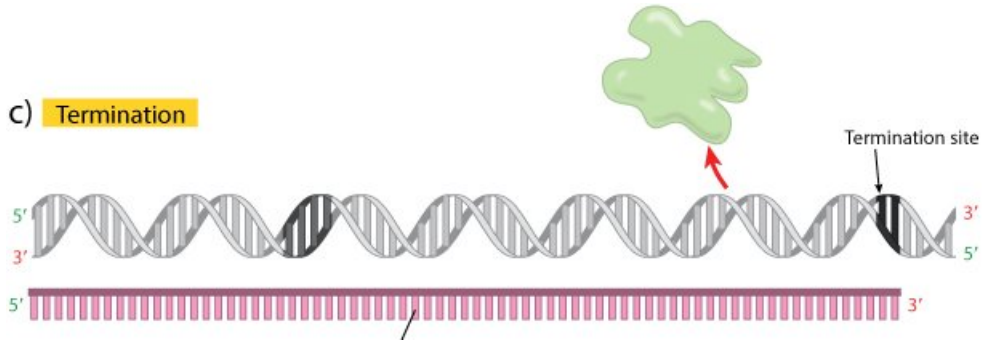

Figura 2.4: Quando a RNA polimerase atinge uma sequência de terminação na fita molde de DNA, a transcrição é terminada e o transcrito de mRNA e RNA polimerase são liberados do complexo. Fonte: https://www.nature.com/scitable/topicpage/dna-transcription-426

- **Processamento:** Após a transcrição, a molécula de RNA é processada de várias maneiras: os íntrons são removidos e os éxons são unidos em conjunto para formar uma molécula de mRNA madura consistindo numa única sequência codificadora de proteína. A síntese de RNA envolve as regras normais de emparelhamento de bases, mas a base timina é substituída pela base uracila.

2.1.2 Tradução

No processo tradução, a molécula de mRNA maduro é utilizada como molde para montar uma série de aminoácidos para, posteriormente, produzir um polipeptídeo com uma sequência de aminoácidos específica. O complexo no citoplasma em que isso ocorre é chamado de ribossomo. Os ribossomos são uma mistura de proteínas ribossômicas e RNA ribossômico (rRNA), e consistem em uma subunidade grande e uma subunidade pequena.

O processo de tradução é constituído de três etapas [Sperber e Sperber, 2008]. São elas:

- **Iniciação:** A subunidade pequena do ribossomo liga-se à extremidade 5’ da molécula de mRNA e move-se na direção 3’ até encontrar um códon de início (AUG). Forma-se, então, um complexo com a grande subunidade do complexo de ribossomo e a iniciação da molécula de tRNA.

- **Alongamento:** Os códons subsequentes na molécula de mRNA determinam qual molécula de tRNA ligada a um aminoácido se liga ao mRNA. Uma enzima peptidil-transferase liga os aminoácidos em conjunto utilizando ligações peptídicas. O processo continua, produzindo uma cadeia de aminoácidos à medida que o ribossomo se move ao longo da molécula de mRNA.

- **Terminação:** O processo de tradução termina quando o complexo ribossômico atingiu um ou mais códons de parada (UAA, UAG, UGA). O complexo ribossômico em eucariotos é maior e mais complicado do que em procariotos. Além disso, os processos de transcrição

e tradução são divididos, em eucariotos, entre o núcleo (transcrição) e o citoplasma (tradução), o que proporciona mais oportunidades para a regulação da expressão gênica.

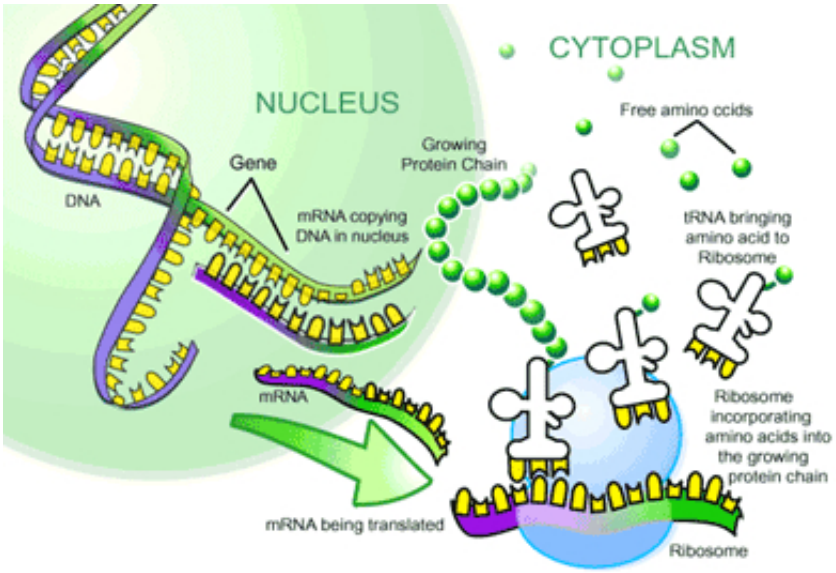

Figura 2.5: Etapas do processo de tradução. Fonte: https://www.nature.com/scitable/topicpage/dna-transcription-426

2.2 *Clustering*

Uma definição menos formal de agrupamento ou *clustering* pode ser "o processo de organização de objetos em grupos cujos membros são semelhantes de alguma forma"[Rashid, 2016]. Um *cluster* é, portanto, uma coleção de objetos que são "semelhantes"entre eles e são "dissimilares"aos objetos pertencentes a outros *clusters*. A Figura 2.6 ilustra como uma coleção de objetos semelhantes estão agrupados em diferentes *clusters*.

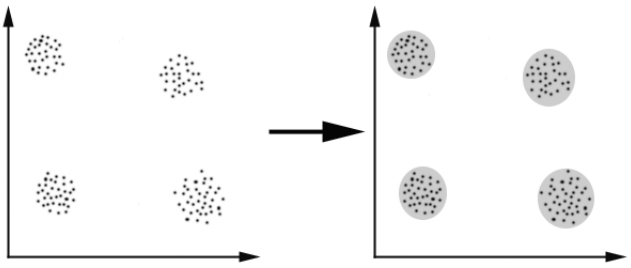

Figura 2.6: Objetos semelhantes agrupados em *clusters*. Fonte: [Rashid, 2016]

Neste caso, identificamos facilmente quatro *clusters* nos quais os dados podem ser divididos. A regiões sombreadas são os grupos (*clusters* em que as amostras foram divididas. O critério de similaridade, neste exemplo, é a distância: dois ou mais objetos pertencem ao

mesmo *cluster* se estiverem "próximos"de acordo com uma distância dada (neste caso, distância geométrica). Outro tipo de agrupamento é o agrupamento conceitual: dois ou mais objetos pertencem ao mesmo *cluster* se este define um conceito comum a todos os objetos. Em outras palavras, os objetos são agrupados de acordo com seu ajuste a conceitos descritivos, não de acordo com simples medidas de similaridade [Jain et al., 1999].

Portanto, o objetivo do agrupamento é determinar o agrupamento intrínseco em um conjunto de dados não marcados. Mas como decidir o que constitui um bom agrupamento? Pode-se demonstrar que não há nenhum critério absoluto melhor que seria independente do objetivo final do agrupamento. Consequentemente, é o utilizador que deve fornecer este critério, de tal forma que o resultado do agrupamento satisfará as suas necessidades [Hassanien et al., 2013].

Algoritmos de agrupamento podem ser utilizados em diversas áreas, por exemplo [Zaïane, 1999]:

- **Marketing:** encontrar grupos de consumidores com comportamento similar, dado um grande banco de dados contendo consumidores e suas propriedades (características), bem como registros de compras.
- **Biologia:** classificação de plantas, animais, genes, dado as suas respectivas características.
- **Bibliotecas:** ordenação de livros
- **Seguros:** identificação de grupos de segurados de seguros de automóvel com um custo médio elevado; identificação de fraudes;
- **Planejamento de cidades:** identificação de grupos de residências de acordo com seu tipo, valor e localização geográfica.
- **Web:** classificação de documentos; classificação de diferentes tipos de usuários baseado em seus acessos.
- **Estudos de terremotos:** identificação de epicentros de terremotos observados para identificar zonas de perigo.

2.2.1 Tipos de *Clusters*

Técnicas de agrupamento visam encontrar grupos úteis de objetos (*clusters*), onde um grupo útil é definido pelo objetivo da análisa feita em cima daquele grupo de dados. Não surpreendentemente, existem diversos tipos de *clusters* que são rotulados como úteis na prática. Abaixo estão os métodos de clusterização mais comuns [Jain et al., 1999].

- **Particionamento:** Os algoritmos particionais dividem a base de dados em k-grupos, onde o número k é dado pelo usuário. O algoritmo escolhe k objetos como sendo os centros dos k clusters. Os objetos são divididos entre os k clusters de acordo com a medida de

similaridade adotada, de modo que cada objeto fique no cluster que forneça o menor valor de distância entre o objeto e o centro do mesmo. Então, o algoritmo utiliza uma estratégia iterativa de controle para determinar que objetos devem mudar de cluster, de forma que a função objetivo usada seja otimizada. O algoritmo mais conhecido é o *K-Means*.

- **Hierárquico:** algoritmos de clusterização baseados no método hierárquico (HC) organizam um conjunto de dados em uma estrutura hierárquica de acordo com a proximidade entre os indivíduos. Os resultados de um algoritmo hierárquico são normalmente mostrados como uma árvore binária ou dendograma, que é uma árvore que iterativamente divide a base de dados em subconjuntos menores. A raiz do dendograma representa o conjunto de dados inteiro e os nós folhas representam os indivíduos. O resultado da clusterização pode ser obtido cortando-se o dendograma em diferentes níveis de acordo com o numero de cluster k desejado.

- **Baseado em densidade:** clusters são definidos como regiões densas, separadas por regiões menos densas que representam os ruídos. As regiões densas podem ter uma forma arbitrária e os pontos dentro de uma região podem também estar distribuídos arbitrariamente e, por isso, os métodos baseados em densidade são adequados para descobrir clusters com forma arbitrária, tais como elíptica, cilíndrica, espiralada, por exemplo. Até os completamente cercados por outro "cluster"e são especialistas em identificar e filtrar ruídos. Os métodos baseados em densidade diferem-se pela forma com que crescem os clusters: uns determinam os clusters de acordo com a densidade da vizinhança dos objetos, outros, trabalham de acordo com alguma função de densidade.

- **Baseado em modelo:** utilizam um modelo de referência para cada cluster. Eles tentam otimizar a curva entre os objetos dados e algum modelo matemático. Um algoritmo baseado em modelo pode descobrir clusters construindo uma função de densidade que reflete a distribuição espacial dos pontos de dados. Ele também conduz a um modo de determinar automaticamente o número de clusters baseado na estatística padrão, identificando ruídos no relatório e assim produzindo métodos de clusterização robustos. Estes modelos são, frequentemente, baseados na suposição que os dados são gerados por uma mistura de distribuições de probabilidades.

- **Baseado em kernel:** é uma técnica desenvolvida especialmente para problemas não linearmente separáveis serem resolvidos de forma "mais elegante". Algoritmos Kernel usam do espaço de características para permitir uma separação não-linear no espaço de entrada. Clusterização baseada em kernel realiza o agrupamento implicitamente por um método Kernel, que executa um mapeamento não linear apropriado dos dados de entrada para um espaço de características de alta dimensão, ao substituir o produto interno entre as variáveis não-lineares por um determinado kernel. Portanto, são capazes de produzir uma separação não linear entre os hiperespaços dos clusters.

- **Baseado em grafos:** buscam representar um conjunto de dados em um grafo, onde cada vértice representa um elemento do conjunto de dados e a existência de uma aresta conectando dois vértices é feita com base na proximidade entre os dois dados. A maneira mais simples de estabelecer as ligações entre os vértices é conectar cada vértice aos vértices restantes, onde o peso indica a similaridade entre os dois dados e um cluster é definido como um subgrafo do grafo inicial. Para tal, adota-se uma medida de similaridade no processo de agrupamento, o que pode fazer com que o algoritmo apresente alguma dificuldade em determinar clusters de formas variadas. Os algoritmos de clusterização baseados em grafos são fortemente relacionados com os algoritmos hierárquicos e particionais. Isso significa que o resultado obtido pode ser uma partição ou uma hierarquia de partições.

- **Baseado em computação evolutiva:** compreende um conjunto de técnicas de busca e otimização baseados em mecanismos da evolução biológica, tais como reprodução, mutação, recombinação e seleção natural e estão sendo utilizados amplamente pela comunidade de inteligência artificial para obter modelos de inteligência computacional. Em tais abordagens, o conjunto de dados representa a população sob evolução e seu comportamento é simulado através de repetidas operações associadas aos princípios de mutações genéticas e de seleção natural, comuns na evolução biológica. O mais popular é o algoritmo genético, no qual busca a solução através de operações, como mutação, sobre cadeias de números, geralmemente binários.

Dentre todas as técnicas de agrupamento de dados apresentadas, sobressaem-se aquelas baseadas em densidade por causa de sua capacidade de identificar *clusters* em formatos e dimensões variadas - além de serem distribuições bastante complexas - que tornariam inviável o uso de uma grande parte dos algoritmos de agrupamento [Gong et al., 2007].

Neste contexto, dois algoritmos de agrupamento baseados em densidade têm se destacado. Apesar de ambos os algoritmos usarem como base a análise de densidade, eles apresentam propostas com diferentes pontos de vista: um baseado em algoritmos de *cluster* tradicionais e outro baseado nas ideias de seleção natural e evolução das espécies.

Neste projeto, propomos uma avaliação experimental comparativa entre essas duas abordagens, aplicando-as com dados de expressão gênica. Este processo será descrito em detalhes nas próximas seções.

2.3 Algoritmos Genéticos

Algoritmos Genéticos (AGs) são técnicas de busca global e otimização utilizadas para combinar as características de possíveis soluções que obtiveram bom desempenho, com o objetivo de construir soluções melhores [Balakrishnan e Honavar, 1995]. Baseiam-se na concepção darwiniana de sobrevivência do mais apto e na teoria da seleção natural das espécies [Beasley et al., 1993].

Algoritmos genéticos operam em um conjunto de indivíduos conhecido como população. Cada indivíduo é uma representação de uma solução do problema e é chamado de cromossomo [Hassanien et al., 2013]. Durante o processo evolutivo, cada indivíduo da população é avaliado com uma função de aptidão (*fitness*) que reflete a qualidade da solução que este indivíduo representa. Essa função deve ser desenhada para o problema em questão e está totalmente limitada pela representação do problema.

Seguindo a teoria da seleção natural, os indivíduos mais aptos serão selecionados para recombinação (ou cruzamento). Na recombinação, o código genético de cada indivíduo pai escolhido na seleção é combinado e são gerados novos filhos. Esses filhos podem ou não sofrer mutação, que é a alteração de alguma parte da solução que não foi transmitida por nenhum dos pais, mas sim alterada aleatoriamente, inspirado na mutação biológica. Esse processo se repete durante várias gerações, até que algum critério de parada seja satisfeito. [Beasley et al., 1993] A visão geral de um AG é apresentada na figura 2.7.

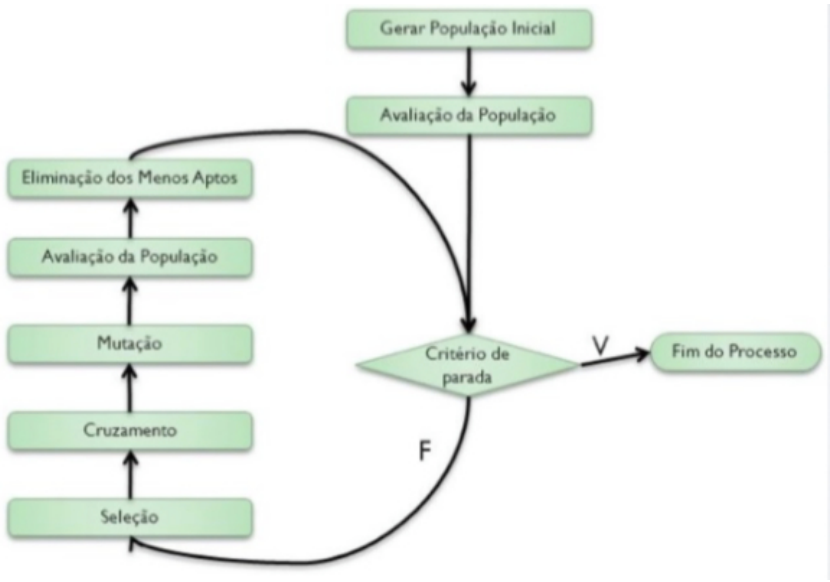

Figura 2.7: Visão Geral de um Algoritmo Genético. Fonte: http://slideplayer.com.br/slide/359174/

Existem diferentes critérios de paradas. O mais comum deles é quando um número máximo de gerações é atingido. Quando se conhece a resposta máxima da função de aptdão, pode-se utilizar esse valor como critério de parada. Outra forma, é parar quando não se nota aumento da aptdão durante um determinado número de gerações [Naldi, 2006].

2.4 *Clustering* usando Algoritmos Genéticos

Abordagens evolutivas têm se mostrado muito eficientes para a obtenção de soluções para problemas de agrupamento [Jain et al., 1999]. Dentre essas abordagens, AGs são bastante

utilizados, principalmente em problemas de agrupamento em k grupos em que o valor de k é previamente conhecido [Belew e Brooker, 1991].

Segundo [Cowgill et al., 1998, Jain et al., 1999], AGs têm se destacado na solução de problemas de agrupamento porque:

1. Executam uma busca global,
2. Utilizam procedimentos probabilísticos ao invés de regras determinísticas,
3. Procuram por uma população de soluções em paralelo, evitando máximos locais
4. Os princípios evolutivos permitem a evolução para o ponto ótimo
5. Podem obter não apenas uma, mas um grupo de potenciais soluções.

Dentre as semelhanças entre algoritmos de agrupamento e AGs, merece destaque a relação entre as funções objetivo e de aptidão [Naldi, 2006]. A função objetivo de um algoritmo de agrupamento pode ser utilizada como função de aptidão do AG.

2.5 *Clustering* em Bioinformática

Com o avanço das tecnologias recentes, uma grande quantidade de dados biológicos está sendo gerada. À medida que os bancos de dados aumentam seu tamanho, um dos desafios atuais na biologia é ser capaz de inferir algumas das funções críticas de tais dados complexos. Para analisar sistemas biológicos complexos, os pesquisadores geralmente visam identificar alguns padrões que co-ocorrem na forma de grupos. [Jiang et al., 2004] Análise de *clusters* é uma técnica exploratória que descobre padrões ricos de dados vastos e, portanto, tornou-se uma ferramenta indispensável para várias tarefas de descoberta de conhecimento no campo da biologia computacional. *Clustering* é uma técnica poderosa e amplamente utilizada que organiza e elucida a estrutura de dados biológicos. Dados de agrupamento de uma grande variedade de experimentos biológicos provou ser imensamente útil para derivar uma variedade de *insights*, como a regulação compartilhada ou a função dos genes.

Ao analisar esses dados biológicos complexos, pode-se observar que as atividades dos genes não são independentes umas das outras. Tem sido demonstrado que os genes com a mesma função (ou genes envolvidos no mesmo processo biológico) são suscetíveis de ser co-expressa. Assim, é importante estudar grupos de genes em vez de realizar uma única análise genética. Em outras palavras, é crucial identificar subconjuntos de genes que sejam relevantes para o problema biológico em estudo. A análise desses subconjuntos de dados fornece informações cruciais sobre os processos biológicos e as funções celulares. Assim, agrupar os perfis de expressão gênica pode proporcionar *insights* sobre a função gênica, a regulação de genes e os processos celulares.

Também foi demonstrado que proteínas de funções conhecidas tendem a se agrupar em conjunto [Schwikowski et al., 2000]. A distância da rede é correlacionada com a distância funcional, e as proteínas que estão mais próximas umas das outras tendem a ter funções biológicas semelhantes [Sharan e Shamir, 2000]. Assim, o agrupamento das redes de interação proteína-proteína é crucial na descoberta das funções das proteínas e, portanto, na compreensão do

funcionamento interno das células. Os blocos de construção mais importantes de organismos vivos, tais como DNA, RNA, mRNA, polipeptídeos e proteínas têm estrutura linear e podem ser representados como sequências. O agrupamento de dados de sequências biológicas pretende agrupar as sequências biológicas que estão relacionadas. Os clusters identificados podem ajudar a proporcionar uma melhor compreensão do genoma.

2.5.1 *Clustering* microarrays

A ampla análise de expressão do genoma com a tecnologia de microarrays de DNA tornou-se uma ferramenta indispensável na pesquisa de genômica. Devido à sua alta variabilidade intrínseca, a extração de características biológicas em experiências com microarrays continua a ser um grande desafio. Com base na hipótese de que genes funcionalmente relacionados tendem a exibir padrões correlatos de expressão gênica, a análise de agrupamento emergiu como uma abordagem frutífera para mecanismos reveladores subjacentes a vários processos moleculares e celulares. O objetivo do agrupamento é identificar grupos de genes que mostram padrões de expressão correlacionados em uma série de condições experimentais [Jiang et al., 2004].

A maioria das abordagens de cluster implementadas hoje são baseadas em distância, como agrupamento hierárquico *K-means* e mapas auto-organizáveis. Embora simples e visualmente atraente, os desempenhos desses métodos são sensíveis ao ruído, que é extenso em dados de microarrays. Além disso, eles têm dificuldade em fornecer informações úteis, como número total de *clusters* e medidas de confiança para *clusters* individuais, e não são flexíveis o suficiente para acomodar dados faltantes, que são comuns na análise de dados de microarrays [Monti et al., 2003].

Capítulo 3

Revisão bibliográfica

3.1 *Clustering by fast search and find of density peaks*

A análise de clusters visa classificar os elementos em categorias com base na sua similaridade. Suas aplicações vão da astronomia à bioinformática, bibliometria e reconhecimento de padrões. Como citado por Christian Wiwie em sua pesquisa *Comparing the performance of biomedical clustering methods* [Wiwie et al., 2015], o método proposto por Alex Rodriguez e Alessandro Laio, obteve notoriedade na classificação de dados biomédicos dentre outros 13 métodos submetidos às mesmas condições de testes e métricas. "Na média, métodos baseados por densidade, liderados pelo algoritmo CDP[1], foram os mais performáticos entre todos os conjuntos de dados", afirma Christian Wiwie.

O estudo realizado por Alex Rodriguez e Alessandro Laio propõe uma abordagem baseada na ideia de que os centros de agrupamento são caracterizados por uma densidade maior do que seus vizinhos e por uma distância relativamente grande de pontos com densidades mais altas (Figura 3.1).

Essa ideia é a base de um procedimento de agrupamento no qual o número de clusters surge intuitivamente, os *outliers* são automaticamente manchados e excluídos da análise, e os clusters são reconhecidos independentemente da sua forma e da dimensionalidade do espaço em que estão inseridos [Rodriguez e Laio, 2014].

O algoritmo tem sua base nos pressupostos de que os centróides de agrupamento são rodeados por vizinhos com menor densidade local e que estão a uma distância relativamente grande de quaisquer pontos com maior densidade local. Para cada ponto de dados i, calculamos duas quantidades: sua densidade local ρ_i e sua distância δ_i de pontos com maior densidade. Ambas as quantidades dependem apenas das distâncias $d_{i,j}$ entre os pontos de dados. A densidade local ρ_i de um ponto i é definida por:

$$\rho_i = \sum_j X(d_{ij-d_c})$$

[1]CDP - Clusterdp - é como Christian Wiwie denomina o algoritmo proposto por Alex Rodriguez e Alessandro Laio

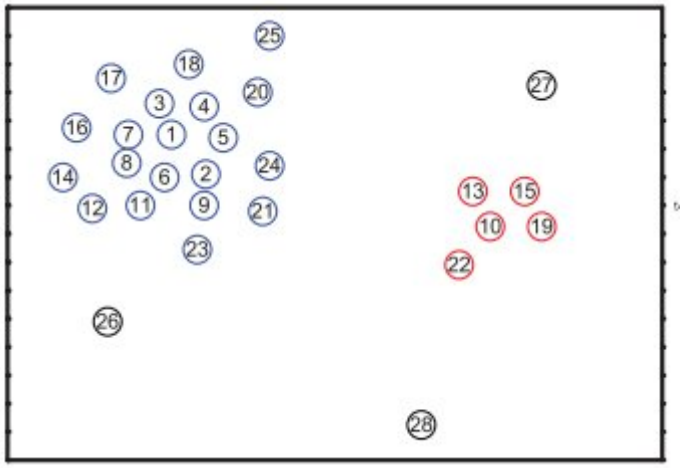

Figura 3.1: Distribuição dos dados e clusterização utilizando o algoritmo proposto.

onde $X(x) = 1$ se $x < 0$ e $X(x) = 0$ caso contrário; d_c é o distância de corte (*cutoff distance*). Já δ_i é medido calculando a distância mínima entre o ponto i e qualquer outro ponto com maior densidade:

$$\delta_i = \min_{j:\rho_i > \rho_j}$$

3.2 *Density-Sensitive Evolutionary Clustering*

Um dos grandes desafios na modelagem de um algoritmo para agrupamento de dados é a elaboração de uma boa maneira para comparar esses dados, ou seja, uma métrica que possa nos dizer o quão similar essas amostras são e após isso agrupá-las conforme essa similaridade.

A medida de similaridade mais comum na literatura é a Distância Euclidiana, definida abaixo:

$$dist(a, b) = \sqrt{|(a - b)|^2}$$

A utilização dessa métrica tem uma boa performance em data sets com uma distribuição hiper-esférica, mas tende a falhar em data sets com uma distribuição mais complexa e com formas indefinidas [Gong et al., 2007].

A figura 3.2 ilustra uma situação em que, se utilizássemos a distância Euclidiana como métrica, os pontos 1 e 2 seriam muito mais similares do que os pontos 1 e 3. Contudo, pode-se perceber que os pontos 1 e 2 pertencem a clusters diferentes e 1 e 3 ao mesmo cluster.

Partindo dessa observação, Maoguo Gong, Lincheng Jiao, Ling Wang e Liefeng Bo apresentaram um novo algoritmo evolutivo baseado em densidade chamado de *Density-Sensitive Evolutionary Clustering (DSEC)*.

O *data set* é transformado em um grafo *G=(V, E)*, onde cada vértice *V* é um *data point* (uma amostra) e cada aresta *E* reflete a afinidade entre cada par de data points. Um par de data

Figura 3.2: Uma ilustração de que a distância euclidiana pode não refletir a consistência global. Fonte: [Gong et al., 2007]

points têm alta afinidade se eles podem ser ligados por um caminho em uma região de alta densidade [Gong et al., 2007].

Como pode-se ver na figura 3.3, para representar a consistência global do *data set*, um caminho entre dois *data points* não é sempre o caminho mais curto. Para isso é necessário que o tamanho do caminho conectado por arestas curtas seja menor do que o caminho conectado diretamente, isto é:

$$\overline{af} + \overline{fe} + \overline{ed} + \overline{dc} + \overline{cb} < \overline{ab}$$

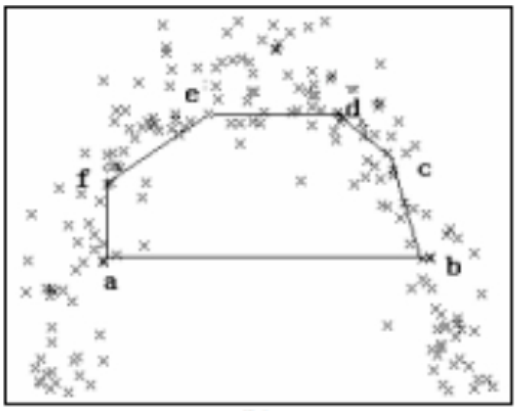

Figura 3.3: A consistência da solução nem sempre satisfaz a desigualdade triangular utilizando a distância euclidiana. Fonte: [Gong et al., 2007]

Baseado nessa propriedade, é definido um comprimento ajustado por densidade do segmento de linha (x_i,x_j),

$$\mathrm{L}(x_i,x_j) = \rho^{dist(x_i,x_j)} - 1$$

onde, $dist(x_i,x_j)$ é a distância Euclidiana entre x_i,x_j e $\rho > 1$ é o fator de flexão. Com isso, um segmento de linha pode ser encurtado ou alongado ajustando o fator de flexão.

Utilizando desse comprimento ajustado, uma nova métrica é introduzida, chamada de Métrica de Distância Sensível a Densidade (*Density-Sensitive Distance Metric*), que mede a distância entre um par de pontos procurando pelo menor caminho no grafo. Essa métrica é definida abaixo.

$$D(x_i,x_j) = \min_{p \in P_{i,j}} \sum_{k=1}^{|p|-1} L(p_k, p_{k+1})$$

Onde os *data points* são os nós do grafo *G=(V,E)*, p é um caminho concetando os nós p_1 e $|p|$, $P_{i,j}$ é o conjunto de todos os caminhos que conectam i e j e $(p_k, p_{k+1}) \in E$.

Como resultado, teremos a menor distância obtida através do comprimento ajustado por densidade do segmento de linha entre x_i e x_j. Dois pontos em uma mesma região de alta densidade são conectados por várias arestas curtas, enquanto que dois pontos em diferentes regiões de alta densidade são conectados por uma aresta longa através de uma região de baixa densidade [Gong et al., 2007]. Essa métrica será utilizada pela função objetivo do algoritmo evolutivo proposto.

3.2.1 Representação

Cada indivíduo (cromossomo) é uma sequência de números inteiros reais com tamanho k (número de clusters), representando os número de sequencia de cada representante dos k clusters. Por exemplo, considere um data set com tamanho N = 100 e 5 clusters diferentes (k = 5). Então o indivíduo (6,19,91,38,64) é uma solução onde o ponto 6 representa o primeiro cluster, o ponto 19 representa o segundo e assim por diante. Seguindo esta representação, o espaço de busca tem tamanho N^k.

3.2.2 Operadores

Crossover

O método utilizado para a recombinação foi o crossover uniforme. Um exemplo desse método é apresentado a seguir: Sejam dois pais A e B. A = (6, 19, 91, 38, 64) e B = (3, 29, 17, 61, 6). Uma máscara randômica é gerada. Essa máscara informa quais posições devem ser trocadas (posições em 1). Para esse exemplo supomos (1, 0, 0, 1, 0). Após aplicação do crossover uniforme, dois filhos serão gerados, são eles: filho C = (6, 29, 17, 38, 64) e D = (3, 19, 91, 61, 64). Pode-se notar que o filho C deveria ter a posição 5 alterada, porém como o ponto 6 já estava na solução, mantém-se o original.

Mutação

Após a recombinação, é aplicado o operador de mutação com uma probabilidade p_m. Duas funções são propostas, conforme descrito a seguir:

$$M = C + \text{floor}((N\text{-}C)*\text{random}+1))$$

ou

$$M = C - \text{floor}((C\text{-}1)*\text{random}+1)$$

Onde M representa o novo cluster, C é o cluster atual, N é a quantidade de pontos e random é um número aleatório entre 0 e 1.

Seleção

Para seleção dos pais que sofrerão recombinação, o método escolhido foi a seleção por roleta com elitismo. No método de seleção por roleta, os pais são selecionados de acordo com sua adequação. Quanto melhores são os cromossomos, maiores são as chances de serem selecionados. Segundo [Mitchel, 1999], a utilização do elitismo melhora significativamente o desempenho do algoritmo genético.

Função Objetivo

Cada ponto é atribuído para o cluster que possua a menor distância sensível a densidade entre ele e o representante do cluster. É computada da seguinte maneira:

$$Dev(\mathrm{C}) = \sum_{C_k \in C} \sum_{i \in C_k} D(\mathrm{i}, \mu_k)$$

Onde C é o conjunto de todos os clusters, μ_k é o respresentante do cluster C_k, e D(i, μ_k) é a distância sensível a densidade entre o i-ésimo ponto do cluster C_k e μ_k.

Algoritmo

As operações descritas acima são executadas por um numero máximo de gerações definido abaixo como *Gmax*. O melhor indivíduo da última geração é retornado como solução do problema. A população inicial é gerada randomicamente. A figura 3.4 detalha o pseudo-código do algoritmo.

Algorithm 1. Density-Sensitive Evolutionary Clustering (DSEC)

```
Begin
1.  t=0
2.  random initialize population P(t)
3.  assign all points to clusters according to the density-
    sensitive dissimilarity measure and compute the objective
    function values of P(t)
4.  t=t+1
5.  if t< G_max
6.     select P(t) from P(t-1)
7.     crossover P(t)
8.     mutate P(t)
9.     go to step 3
10. end if
11. output best and stop
end
```

Figura 3.4: Algoritmo do DSEC.

Capítulo 4

Proposta

Este capítulo tem como objetivo apresentar as características dos *datasets* de expressão gênica utilizados no Capítulo 5, assim como as técnicas e métricas escolhidas para a análise e avaliação dos algoritmos anteriormente discutidos.

4.1 *Datasets* de Expressão Gênica

Para ressaltar a eficiência individual e as diferenças entre os algoritmos de agrupamento escolhidos, optamos por utilizar quatro conjuntos de dados de expressão gênica que são muito comuns em experimentos como o deste trabalho.

O conjunto de dados de Alizadeh [Alizadeh et al., 2000] consiste de 45 amostras, 24 delas são do grupo "centro germinativo *B-like* DLBCL"enquanto 21 são do grupo "ativados *B-like* DLBCL". Cada amostra é descrita por 4026 genes. O dataset é representado por distintos tipos de linfoma difuso de grandes células B (*Distinct types of diffuse large B-cell lymphoma* - DLBCL) usando dados de expressão gênica. O conjunto de dados Alizadeh é um dos mais populares e amplamente estudado com dados de linfoma. Diversos outros trabalhos originaram-se com este dataset.

O conjunto de dados de Golub [Golub et al., 1999] consiste de 47 pacientes com leucemia linfoblástica aguda (LLA) e 25 pacientes com leucemia mielóide aguda (LMA). Cada um dos 72 pacientes tinha amostras de medula óssea obtidas no momento do diagnóstico. Conforme discutido em sua pesquisa, “a leucemia linfoblástica aguda surge de dois tipos diferentes de linfócitos (linfócito-T e linfócito-B)”, por isso podemos considerar os dados em termos de três classes: LMA, LLA-T e LLA- B. O conjunto de dados Golub é possivelmente o mais amplamente estudado e citado microarray conjunto de dados.

O conjunto de dados tcga (*The Cancer Genome Atlas*) é um projeto que mantém um banco de dados armazenando informações moleculares de células cancerígenas, incluindo expressão gênica, metilação de DNA ou aberração de números de cópias. Ele inclui dados de muitos tipos diferentes de câncer, permitindo sua comparação em um nível molecular. Um

conjunto de dados foi derivado integrando os níveis de expressão genética, metilação do DNA e aberração do número de cópias dos três diferentes tipos de câncer, nomeadamente Carcinoma Invasivo de Peito (*Breast Invasive Carcinoma* - BRCA, 207 amostras), Glioblastoma Multiforme (*Glioblastoma Multiforme* - GBM, 67 amostras) e Carcinoma de Células Escamosas Pulmonares (*Lung Squamous Cell Carcinoma* - LUSC , 19 amostras). Para cada tipo de informação molecular, os autores [Speicher, 2012] calcularam semelhanças em pares entre as amostras usando a correlação de Spearman. Isso resultou em três semelhanças para cada par de amostras, que foram combinadas tomando sua média aritmética.

O último conjunto de dados, *bone_marrow*, contém níveis de expressão gênica de microarrays de 999 genes para 38 amostras de pacientes com leucemia que sofrem de três subtipos diferentes de leucemia aguda. Este é um *dataset* público fornecido pelo Broad Institute[1].

Dataset	Nº de amostras	Nº de genes	Nº de classes	Referência
Linfoma	45	4026	9	[Alizadeh et al., 2000]
Leucemia	72	7129	2	[Golub et al., 1999]
tcga	293	2212	3	[Speicher, 2012]
bone_marrow (Medula)	38	999	3	[Monti et al., 2003]

Tabela 4.1: Conjunto de dados de expressão gênica

4.2 Métricas de avaliação

Os métodos de agrupamento, em geral, formalizam o objetivo de obter alta similaridade intra-*cluster* e baixa similaridade extra-*cluster*, ou seja, os objetos dentro do *cluster* são semelhantes, enquanto os objetos de outros *clusters* são diferentes [Schulte, 2009]. Este é um critério normalmente utilizado para avaliar a qualidade de um *cluster*. Entretanto, boas pontuações em um critério como este não traduzem, necessariamente, em uma boa eficácia do algoritmo. Uma alternativa a esse critério é a avaliação direta dos resultados do *dataset* de interesse [Schulte, 2009]. Por exemplo, em determinada aplicação de classificação de dados de expressão gênica, podemos querer medir o tempo que o algoritmo leva para efetuar a classicação em relação à outros algoritmos de agrupamento. Esta é uma medida mais direta, porém é dispendiosa, especialmente se também forem avaliadas outras características para medir a qualidade de um agrupamento.

Avaliar o desempenho de um algoritmo de agrupamento não é uma tarefa tão trivial quanto contar o número de erros ou a precisão e *recall* de um algoritmo de classificação supervisionado. Em particular, qualquer métrica de avaliação não deve levar em conta os valores absolutos dos rótulos do *cluster*, mas sim se esse *cluster* definir separações dos dados semelhantes a um conjunto de classes ou satisfazer alguma suposição, de modo que os membros pertençam à mesma classe sejam mais parecidos alguma métrica de similaridade. Por este motivo, neste

[1]Mais informações em `https://www.broadinstitute.org/`

trabalho optamos por utilizar duas métricas de similaridade para a avaliação de algoritmos de agrupamento: F1-*score* e Rand Index.

4.2.1 F1-*score*

O F1-*Score* é originário da classificação binária, onde temos apenas duas classes que queremos distinguir: **positivo** e **negativo** [Wikipedia, 2017a]. Neste cenário, existem quatro possíveis resultados:

- VP (Verdadeiro Positivo): o objeto pertence à classe positivo e foi classificado como positivo,
- FP (Falso Positivo): o objeto pertence à classe negativo e foi classificado como positivo,
- VN (Verdadeiro Negativo): o objeto pertence à classe negativo e foi classificado como negativo,
- FN (Falso Negativo): o objeto pertence à classe positivo e foi classificado como negativo

O F1-*Score* pode ser interpretado como uma média harmônica da precisão e do *recall* (sensitividade), onde sua melhor pontuação atinge o valor em 1 e a pior pontuação atinge 0. A contribuição relativa de precisão e *recall* para o F1-*Score* é igual [Sokolova et al., 2006]. A fórmula do F1-*Score* é dada por:

$$F_1 = 2 \times \frac{\text{Precisão} \times \textit{recall}}{\text{Precisão} + \textit{recall}}$$

Onde **precisão** é dada por:

$$\frac{\text{VP}}{\text{VP} + \text{FP}}$$

Enquanto *recall* é dada por:

$$\frac{\text{VP}}{\text{VP} + \text{FN}}$$

4.2.2 *Rand index*

O Rand Index é a proporção de pares de objetos corretamente agrupados de todos os pares possíveis. Esta medida estima a probabilidade de um elemento ser corretamente classificado. Esta medida é baseada na abordagem *pairwise* para calcular VP, VN, FP e FN. Para cada possível par de elementos no conjunto considerado, o Rand Index avalia quão semelhante as duas partições os tratam, penalizando decisões falso-positivas e falso-negativas durante o agrupamento [Wikipedia, 2017b]. Sua fórmula é dada por:

$$R = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{FP} + \text{VN} + \text{FN}}$$

4.3 Seleção de características

Um dos principais problemas com conjuntos de dados de expressão gênica é seu alto número de características, um fator que pode comprometer a classificação dos objetos, independente do método utilizado para o agrupamento. Uma das formas para evitar este problema é a seleção de características [Dy e Brodley, 2004].

A seleção de características é um processo em que seleciona-se automaticamente características que mais contribuem para a predição ou saída de interesse em um determinado conjuntos de dados. Conjuntos de dados de expressão gênica costumam apresentar milhares de características e é possível que algumas dessas características sejam irrelevantes e diminuam a precisão de alguns modelos, especialmente algoritmos lineares, como regressão linear e logística [Dy e Brodley, 2004].

Alguns benefícios da seleção de características são [Dy e Brodley, 2004]:

- Redução de *overfitting*: menos dados redundantes significa menos oportunidades de tomar decisões com base em ruídos
- Melhora a precisão: dados menos enganosos significam que a precisão da modelagem pode melhorar
- Reduz o tempo de treinamento: menos dados significa que os algoritmos treinam mais rápido

Diante de tantas possibilidades e possíveis melhorias, a seleção de características é um campo de estudo bastante amplo e com aplicações em diversas áreas, incluindo a bioinformática.

Utilizamos o *Weka* [Hall et al., 2009], para fazer uma seleção de características. *Weka* é uma coleção de algoritmos de aprendizagem de máquina para tarefas de mineração de dados. Contém ferramentas para pré-processamento de dados, classificação, regressão, agrupamento, regras de associação e visualização.

Resumidamente, o processo de seleção de características no *Weka* é dividido em duas partes: *Attribute Evaluator* e *Search Method* [Thornton et al., 2013]. Cada seção tem várias técnicas para escolher. O *Attribute Evaluator* (avaliador de atributos) é a técnica pela qual cada atributo no conjunto de dados é avaliado no contexto da variável de saída (por exemplo, a classe). O *Search Method* (método de busca) é a técnica através da qual se pode tentar ou navegar diferentes combinações de atributos no conjunto de dados para chegar a uma pequena lista de recursos escolhidos.

Uma técnica popular para selecionar os atributos mais relevantes em seu conjunto de dados é usar *correlation* (correlação). Por se tratar de um método tradicional de seleção de características, optamos por utilizá-lo neste trabalho [Dy e Brodley, 2004].

Capítulo 5

Resultados

O algoritmo proposto no artigo *Clustering by fast search and find of density peaks* possui implementações disponíveis em diversas linguagens de programação, tais como em Python, R, Matlab, C++, entre outras. Como mencionado no Capítulo 3, o algoritmo depende de dois parâmetros importantes: ρ_i e δ_i. Com estes valores a critério do usuário, o algoritmo torna-se extremamente sensível a estes valores; forçando alguns testes em cada tipo de dataset até encontrar valores satisfatórios ou ter um conhecimento prévio do conjunto de dados. Neste projeto, optamos por utilizar a implementação disponível em seu website[1] pelos próprios autores do artigo, em Matlab, para avaliarmos seu desempenho nos *datasets* selecionados de expressão gênica.

Para os testes com o algoritmo proposto por [Gong et al., 2007], utilizamos uma implementação em *Python* disponibilizada por [Igarashi e Tibães, 2016]. Essa implementação necessitou algumas alterações. Primeiro na forma em que o *dataset* era lido e segundo na adaptação para tratamento de problemas com mais de duas dimensões. Como mencionado no Capítulo 3, o algoritmo necessita de um parâmetro referente a taxa de flexão ρ, parâmetro este tratado com foco nos testes.

5.1 Experimentos

Para os testes com o algoritmo proposto por Rodriguez, cada conjunto de dados foi rodado com diferentes valores de d_c para cada conjunto de dados.

Na implementação do algoritmo de Rodriguez, o parâmetro d_c é calculado conforme:

```
percent=2.0; % media percentual de vizinhos

position = round(N*percent/100);
sda = sort(xx(:,3));
```

[1]Download disponível em `http://people.sissa.it/~laio/Research/Clustering_source_code/cluster_dp.tgz`

```
dc = sda(position);
```

Como este é o parâmetro principal do algoritmo e, a partir dele, são calculados sua densidade local ρ_i e sua distância δ_i de pontos com maior densidade, intensificamos a variação do mesmo para ter uma gama de resultados. Rodriguez afirma em sua pesquisa que, em média, o valor de d_c deve variar entre 1% a 2% do número total de pontos do conjunto de dados. Com base nessa afirmação, decidimos variar os experimentos de 0.5% até 2.5% para o cálculo de d_c. Em busca de testar ainda mais exaustivamente o algoritmo, também variamos o número de *clusters* que o algoritmo deveria encontrar. Com isso garantimos que as divisões em grupos são mais próximas da realidade do conjunto de dados e conseguimos provar que os melhores agrupamentos são, de fato, com o número total de grupos previamente conhecidos.

Para os testes com o algoritmo evolutivo, cada conjunto de dados foi rodado para quatro diferentes valores para a taxa de flexão ρ, são eles (1, 10, 50.5 e 100).

A tabela abaixo mostra os demais parâmetros usados para os testes com o algoritmo evolutivo.

Parâmetro	Valor
Número de Gerações (G_{max})	100
Tamanho da População	50
Probabilidade de Crossover	0.8
Probabilidade de Mutação	0.1

Tabela 5.1: Parâmetros fixos usados no algoritmo evolutivo

Para cada resultado foram utilizadas as funções f1_score e adjusted_rand_score da biblioteca scikit-learn[2] - conforme as métricas definidas no Capítulo 4 - a fim calcular a taxa de acerto de cada algoritmo. Ambas as funções recebem como parâmetros um vetor com as classes do conjunto de dados original e outro vetor com as classes que o algoritmo produziu, respectivamente.

Optamos por focar os testes nos conjuntos de dados **tcga** e **bone_marrow**, visto que estes foram os conjuntos de dados escolhidos por Christian Wiwie em seu estudo de comparação de métodos de agrupamento [Wiwie et al., 2015]. Além destes dois, utilizamos os *datasets* **Linfoma** e **Leucemia** visto que são dois dos *datasets* mais amplamente estudados na literatura.

Para cada *dataset*, apresentaremos os resultados de F1-Score e Randon Index resultantes do agrupamento com os algoritmos abordados. Diferentes valores para o parâmetro d_c e ρ de cada algoritmo foram testados, nas tabelas apresentamos o valor da execução com melhor resultado para cada algoritmo (sem e com seleção de atributos) no *dataset* correpondente. Os resultados completos, de todos os parâmetros testados, podem ser encontrados nos apêndices A e B.

[2] http://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

Resultados conjunto de dados Linfoma

A tabela abaixo mostra os resultados aplicados sobre a base Linfoma. Para esse *dataset*, o melhor valor de d_c foi 0.33 e de ρ foi 100.

Algoritmo	Seleção de Características	F1-*Score*	*Rand Index*
Rodriguez	Não	0.8031178	0.7984832
Rodriguez	Sim	0.8159896	0.8030217
Gong	Não	0.8312867	0.8097356
Gong	Sim	0.8227360	0.8012734

Tabela 5.2: Resultados do *dataset* Linfoma

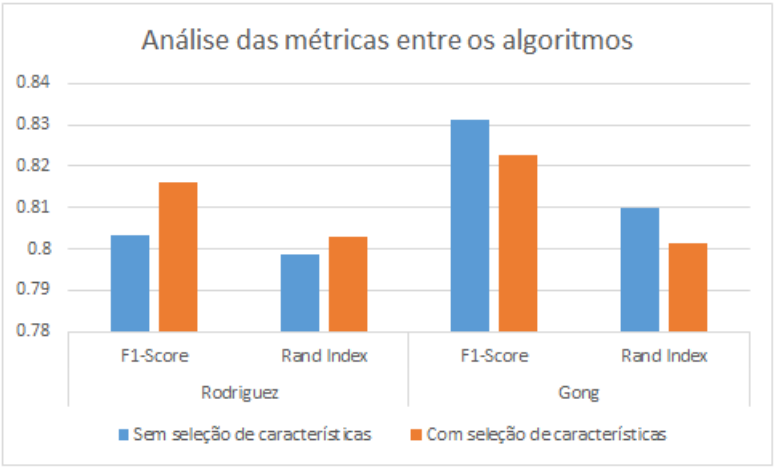

Figura 5.1: Comparação dos resultados entre os algoritmos com o dataset de linfoma

Nesse *dataset*, os dois algoritmos apresentaram bons resultados, com uma leve vantagem do algoritmo de Gong, sem a utilização da seleção de características.

Resultados conjunto de dados Leucemia

A tabela abaixo mostra os resultados aplicados sobre a base Leucemia. Para esse *data set*, o melhor valor de d_c foi 0.13 e de ρ foi 100.

Algoritmo	Seleção de Características	F1-*Score*	*Rand Index*
Rodriguez	Não	0.8028413	0.7843274
Rodriguez	Sim	0.8276123	0.8143117
Gong	Não	0.8328167	0.8287231
Gong	Sim	0.8481637	0.8303254

Tabela 5.3: Resultados do *dataset* Leucemia

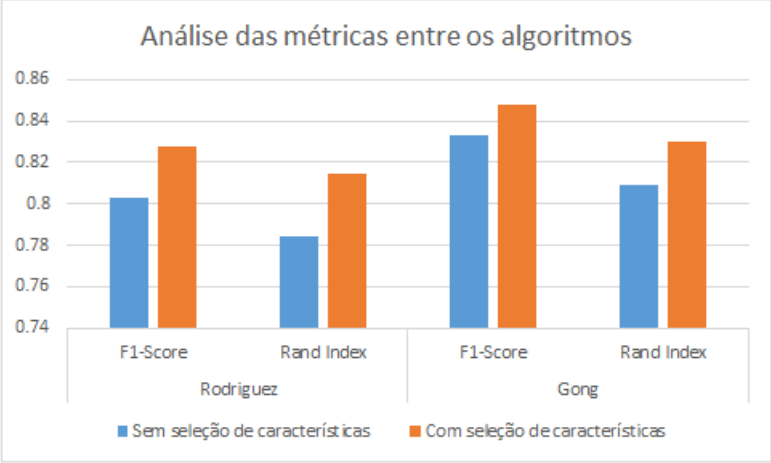

Figura 5.2: Comparação dos resultados entre os algoritmos com o dataset de Leucemia

Novamente ambos os algoritmos apresentaram bons resultados, sendo o algoritmo de Gong com a seleção de características um pouco melhor.

Resultados conjunto de dados tcga

A tabela abaixo mostra os resultados aplicados sobre a base tcga. Para esse *data set*, o melhor valor de d_c foi 0.10 e de ρ foi 100.

Algoritmo	Seleção de Características	F1-*Score*	*Rand Index*
Rodriguez	Não	0.9226169	0.9046941
Rodriguez	Sim	0.9230970	0.9047012
Gong	Não	0.9039847	0.8719002
Gong	Sim	0.9103944	0.8849021

Tabela 5.4: Resultados do *dataset* tcga

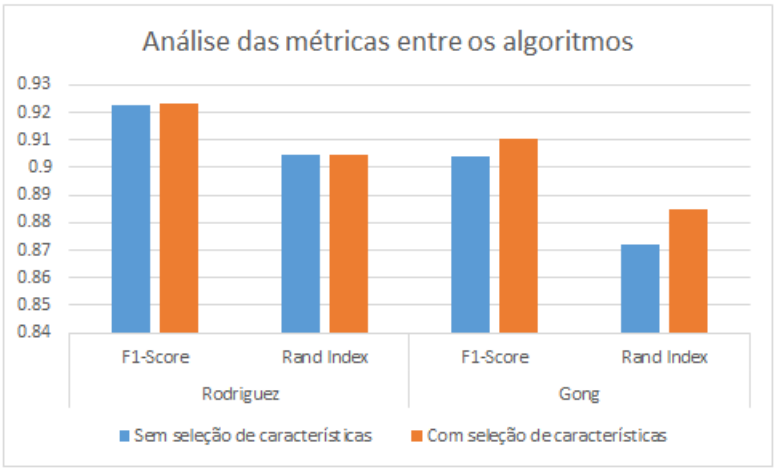

Figura 5.3: Comparação dos resultados entre os algoritmos com o dataset de tcga

A seleção de características, como no *dataset* anterior, melhorou levemente o resultado, porém para esse *dataset* o algoritmo de Rodriguez de agrupamento teve uma pequena vantagem.

Resultados conjunto de dados bone_marrow

A tabela abaixo mostra os resultados aplicados sobre a base bone_marrow. Para esse *data set*, o melhor valor de d_c foi 0.04 e de ρ foi 100.

Algoritmo	Seleção de Características	F1-*Score*	*Rand Index*
Rodriguez	Não	0.9148201	0.9046995
Rodriguez	Sim	0.9149333	0.9062458
Gong	Não	0.9098264	0.8903844
Gong	Sim	0.9173942	0.9039203

Tabela 5.5: Resultados do *dataset* bone_marrow

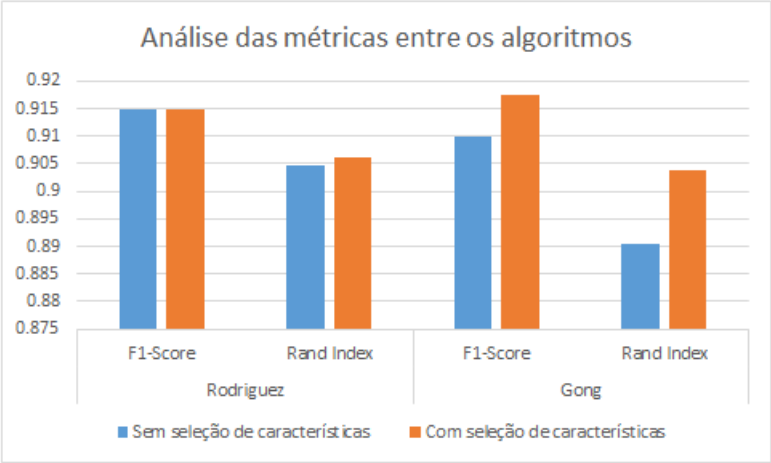

Figura 5.4: Comparação dos resultados entre os algoritmos com o dataset de bone_marrow

Neste *dataset* os resultados foram bem similares. Segundo a métrica *F1-Score*, o algoritmo de Gong com seleção de características obteve o melhor resultado. Já para a métrica *Rand Index*, o melhor algoritmo foi o de Rodriguez, também com seleção de características. Porém todos os resultados foram muito próximos.

Capítulo 6

Conclusão

Este projeto buscou analisar experimentalmente dois algoritmos de agrupamento por densidade em quatro conjuntos de dados populares de expressão gênica. Apesar de ambos utilizarem da mesma métrica, apresentam abordagens diferentes (um tradicional e outro evolutivo). De modo geral, constatou-se que ambos os algoritmos foram eficientes na classificação dos dados de expressão gênica. Embora os resultados tenham sido satisfatórios, ainda há espaço para melhorias e otimização.

No algoritmo de Rodriguez, a adversidade maior foi a escolha manual de uma porcentagem para calcular a limiar de corte d_c e, consequentemente, encontrar os *clusters* em um conjunto de dados. Além disso, se definirmos d_c com uma porcentagem muito grande, pode haver sobreposição de *clusters* levando a uma classificação ineficiente. Entretanto, a questão da distância entre pontos nos algoritmos de agrupamento tem sido amplamente tratada na literatura e ainda é um campo de pesquisa ativo. O problema é que há diversas distâncias diferentes para as aplicações de agrupamento, portanto, torna-se uma limitação a escolha automática deste parâmetro, visto que ainda é necessário calcular uma distância de acordo com um conhecimento prévio do conjunto de dados ou testar inúmeras variações até alcançar resultados satisfatórios.

Os parâmetros principais dos dois algoritmos (ρ, d_c) foram alterados para a realização dos testes para cada *dataset*. Os resultados completos de todos os textes são demonstrados nos apêndices A e B.

O algoritmo de Gong apresenta a mesma dificuldade com a seleção do fator de flexão ρ, contudo pode-se perceber que fatores de flexão maiores retornaram melhores classificação dos dados, visto que o caminho entre pontos em diferentes regiões de alta densidade é alongado utilizando o parâmetro ρ.

Uma das grandes diferenças entre os dois algoritmos foi no quesito tempo de execução. Enquanto o algoritmo proposto por [Rodriguez e Laio, 2014] retornava resultados quase que de maneira imediata, o algoritmo proposto por [Gong et al., 2007] levava vários minutos para execução. Os parâmetros tamanho da população e quantidade de gerações (G_{max}) influenciam fortemente no tempo de execução. Por se fazer necessário o uso de várias gerações para construir uma solução boa, o algoritmo evolutivo perde no que se refere a tempo de execução.

A seleção de características contribuiu na redução do tempo de execução do algoritmo evolutivo, mas ainda assim demorava bem mais que o algoritmo tradicional. Do ponto de vista do resultado do agrupamento, a seleção de características contribuiu com uma leve melhora na maioria dos casos. A escolha de um algoritmo de seleção de características escolhido com base na distribuição dos dados de expressão gênica é um ponto interessante a ser estudado.

6.1 Trabalhos futuros

Para trabalhos futuros, seria interessante investigar o comportamento dos algoritmos em mais conjuntos de dados de expressão gênica.

Outro ponto interessante seria o desenvolvimento de métodos de calibração dos parâmetros necessários, sem que haja um conhecimento prévio do conjunto de dados. Normalizar as métricas pode contrubuir para essa calibração de parâmetros.

A seleção de características mostrou ser uma técnica apta a ser explorada mais profundamente, considerando os resultados obtidos neste trabalho com um método popular de seleção de características. Explorar formas específicas de seleção de atributos em conjuntos de dados de expresão gênica pode agregar resultados ainda mais acertivos.

Referências Bibliográficas

[Alizadeh et al., 2000] Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., Powell, J. I., Yang, L., Marti, G. E., Moore, T., Hudson, J., Lu, L., Lewis, D. B., Tibshirani, R., Sherlock, G., Chan, W. C., Greiner, T. C., Weisenburger, D. D., Armitage, J. O., Warnke, R., Levy, R., Wilson, W., Grever, M. R., Byrd, J. C., Botstein, D., Brown, P. O. e Staudt, L. M. (2000). Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511.

[Balakrishnan e Honavar, 1995] Balakrishnan, K. e Honavar, V. (1995). Evolutionary design of neural architectures: A preliminary taxonomy and guide to literature. *Technical report, Departament of Computer Science, Iowa State University, Ames, Iowa*.

[Beasley et al., 1993] Beasley, D., R., D. e Martin, R. R. (1993). An overview of genetic algorithms: Part 2, research topics. *University Computing*, 15(4):170–181.

[Belew e Brooker, 1991] Belew, R. K. e Brooker, L. B. (1991). Solving partitioning problems with genetic algorithms. *Morgan Kaufmann*.

[Cowgill et al., 1998] Cowgill, M. C., Harvey, R. J. e Watson, L. T. (1998). A genetic algorithm approach to cluster analysis. *Technical report, Virginia Polytechnic Institute State University, Blacksburg, VA, USA*.

[Dy e Brodley, 2004] Dy, J. G. e Brodley, C. E. (2004). Feature selection for unsupervised learning. *J. Mach. Learn. Res.*, 5:845–889.

[Golub et al., 1999] Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D. e Lander, E. S. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537.

[Gong et al., 2007] Gong, M., Jiao, L., Wang, L. e Bo, L. (2007). *Density-Sensitive Evolutionary Clustering*, páginas 507–514. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Hall et al., 2009] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. e Witten, I. H. (2009). The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18.

[Hassanien et al., 2013] Hassanien, A. E., Al-Shammari, T. E. e Ghali, I. N. (2013). Computational intelligence techniques in bioinformatics. *Computational Biology and Chemistry*, 47:37–47.

[Igarashi e Tibães, 2016] Igarashi, A. T. S. e Tibães, J. H. (2016). Implementação, em Python, do artigo Density-Sensitive Evolutionary Clustering.

[Jain et al., 1999] Jain, A. K., M., N. e Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, 31:264–323.

[Jiang et al., 2004] Jiang, D., Tang, C. e Zhang, A. (2004). Cluster analysis for gene expression data: A survey. *IEEE Trans. on Knowl. and Data Eng.*, 16(11):1370–1386.

[Mitchel, 1999] Mitchel, M. (1999). An introduction to genetic algorithms. *MIT Press*.

[Monti et al., 2003] Monti, S., Tamayo, P., Mesirov, J. e Golub, T. (2003). Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52(1):91–118.

[Naldi, 2006] Naldi, M. C. (2006). Agrupamento híbrido de dados utilizando algoritmos genéticos. ICMC-USP.

[Rashid, 2016] Rashid, T. (2016). Clustering. Acessado em dezembro de 2016.

[Rodriguez e Laio, 2014] Rodriguez, A. e Laio, A. (2014). Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496.

[Schulte, 2009] Schulte, S. (2009). Clustering algorithms and evaluations: Chapter 4.

[Schwikowski et al., 2000] Schwikowski, B., Uetz, P. e Fields, S. (2000). A network of protein-protein interactions in yeast. *Nat Biotech*, 18(12):1257–1261.

[Sharan e Shamir, 2000] Sharan, R. e Shamir, R. (2000). CLICK: a clustering algorithm with applications to gene expression analysis. *Proc Int Conf Intell Syst Mol Biol*, 8:307–316.

[Sokolova et al., 2006] Sokolova, M., Japkowicz, N. e Szpakowicz, S. (2006). Beyond accuracy, f-score and roc: A family of discriminant measures for performance evaluation. Em *Proceedings of the 19th Australian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence*, AI'06, páginas 1015–1021, Berlin, Heidelberg. Springer-Verlag.

[Speicher, 2012] Speicher, N. (2012). Towards the identification of cancer subtypes by integrative clustering of molecular data. Master's thesis, Universität des Saarlandes.

[Sperber e Sperber, 2008] Sperber, G. H. e Sperber, S. M. (2008). Thompson and thompson genetics in medicine. 7th edition. *The Cleft Palate-Craniofacial Journal*, 45(1).

[Thornton et al., 2013] Thornton, C., Hutter, F., Hoos, H. H. e Leyton-Brown, K. (2013). Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. Em *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, páginas 847–855, New York, NY, USA. ACM.

[Wikipedia, 2017a] Wikipedia (2017a). F1 score — wikipedia, the free encyclopedia.

[Wikipedia, 2017b] Wikipedia (2017b). Rand index — wikipedia, the free encyclopedia.

[Wiwie et al., 2015] Wiwie, C., Baumbach, J. e Rottger, R. (2015). Comparing the performance of biomedical clustering methods. *Nat Meth*, 12(11):1033–1038. Analysis.

[Zaïane, 1999] Zaïane, O. R. (1999). Principles of knowledge discovery in databases - chapter 8: Data clustering. Acessado em dezembro de 2016.

Apêndice A

Experimentos com variações de d_c no algoritmo de Rodriguez

A.1 *Dataset* Linfoma

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
7	Não	0.4851851	0.4692734
7	Sim	0.4860467	0.4788314
8	Não	0.6333238	0.6285748
8	Sim	0.6482692	0.6417014
9	Não	0.4061649	0.3999896
9	Sim	0.4214894	0.4114946

Tabela A.1: Resultados do *dataset* de linfoma com d_c=0.22

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
7	Não	0.5747734	0.5602807
7	Sim	0.5888889	0.5777415
8	Não	0.7417218	0.7368790
8	Sim	0.7506315	0.7454545
9	Não	0.5124384	0.5085743
9	Sim	0.5248812	0.5188706

Tabela A.2: Resultados do *dataset* de linfoma com d_c=0.25

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
7	Não	0.5259003	0.5188704
7	Sim	0.5290725	0.5259894
8	Não	0.8023881	0.7989275
8	Sim	0.8098035	0.8034664
9	Não	0.5430353	0.5339335
9	Sim	0.5506956	0.5468984

Tabela A.3: Resultados do *dataset* de linfoma com d_c=0.28

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
7	Não	0.5315526	0.5107017
7	Sim	0.5323965	0.5268484
8	Não	0.8031178	0.7984832
8	Sim	0.8159896	0.8030217
9	Não	0.5725687	0.5506956
9	Sim	0.5709334	0.5696631

Tabela A.4: Resultados do *dataset* de linfoma com d_c=0.33

A.2 *Dataset* Leucemia

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.7085527	0.6883484
2	Sim	0.7144746	0.7033016
3	Não	0.5548604	0.5334417
3	Sim	0.5551613	0.5424969
4	Não	0.4785749	0.4544837
4	Sim	0.4817593	0.4670519

Tabela A.5: Resultados do *dataset* de leucemia com d_c=0.05

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.7901754	0.7896142
2	Sim	0.8005889	0.7995356
3	Não	0.5159163	0.5069131
3	Sim	0.5195176	0.5093254
4	Não	0.4854584	0.4676794
4	Sim	0.4856263	0.4796455

Tabela A.6: Resultados do *dataset* de leucemia com d_c=0.09

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.8028413	0.7843274
2	Sim	0.8276123	0.8143117
3	Não	0.5362165	0.5287415
3	Sim	0.5372393	0.5353898
4	Não	0.4796436	0.4562645
4	Sim	0.4874195	0.4728988

Tabela A.7: Resultados do *dataset* de leucemia com d_c=0.13

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.7632644	0.7479365
2	Sim	0.7655217	0.7548664
3	Não	0.5725764	0.5676794
3	Sim	0.5796055	0.5708117
4	Não	0.4417618	0.4334417
4	Sim	0.4424969	0.4358141

Tabela A.8: Resultados do *dataset* de leucemia com d_c=0.17

A.3 *Dataset* tcga

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.5943543	0.5837838
2	Sim	0.5985911	0.5898630
3	Não	0.9226169	0.9046941
3	Sim	0.9230970	0.9047012
4	Não	0.4895924	0.4790043
4	Sim	0.4852490	0.4725647

Tabela A.9: Resultados do *dataset* bone_marrow com d_c=0.10

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.5705207	0.5663892
2	Sim	0.5705207	0.5663892
3	Não	0.8927638	0.8219155
3	Sim	0.8927638	0.8219155
4	Não	0.4701111	0.4691442
4	Sim	0.4701111	0.4591442

Tabela A.10: Resultados do *dataset* tcga com d_c=0.15

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.5643529	0.5609191
2	Sim	0.5643529	0.5674191
3	Não	0.7691593	0.6736874
3	Sim	0.7659335	0.6536467
4	Não	0.4978747	0.7539155
4	Sim	0.4962645	0.4775213

Tabela A.11: Resultados do *dataset* tcga com d_c=0.20

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.4705207	0.4663892
2	Sim	0.4862759	0.4597245
3	Não	0.6705562	0.6687737
3	Sim	0.6863784	0.6512894
4	Não	0.4691926	0.4372983
4	Sim	0.4692477	0.4420995

Tabela A.12: Resultados do *dataset* tcga com d_c=0.25

A.4 *Dataset* bone_marrow

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.5950084	0.5730441
2	Sim	0.5792548	0.5665971
3	Não	0.9120035	0.9046941
3	Sim	0.9094164	0.9052015
4	Não	0.4695423	0.4341593
4	Sim	0.4692101	0.4757412

Tabela A.13: Resultados do *dataset* bone_marrow com d_c=0.02

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.5951184	0.5730482
2	Sim	0.5955230	0.5795584
3	Não	0.9148201	0.9046995
3	Sim	0.9149333	0.9062458
4	Não	0.5695608	0.5561593
4	Sim	0.5694089	0.5559305

Tabela A.14: Resultados do *dataset* bone_marrow com d_c=0.04

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.5943543	0.5837838
2	Sim	0.5980463	0.5906811
3	Não	0.8821879	0.8620147
3	Sim	0.8905787	0.8721687
4	Não	0.4695423	0.4661593
4	Sim	0.4604852	0.4498537

Tabela A.15: Resultados do *dataset* bone_marrow com d_c=0.06

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.5943102	0.5837120
2	Sim	0.6096351	0.6000387
3	Não	0.8156489	0.8032899
3	Sim	0.8315970	0.8209716
4	Não	0.4355423	0.4361475
4	Sim	0.4465482	0.4397185

Tabela A.16: Resultados do *dataset* bone_marrow com d_c=0.08

Apêndice B

Experimentos com variações de ρ no algoritmo de Gong

B.1 *Dataset* Linfoma

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
7	Não	0.4072136	0.3874664
7	Sim	0.4084763	0.3897635
8	Não	0.5175467	0.5072637
8	Sim	0.5187163	0.5096624
9	Não	0.4272634	0.4028163
9	Sim	0.4217378	0.4116136

Tabela B.1: Resultados do *dataset* Linfoma com ρ=1

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
7	Não	0.5021637	0.4961234
7	Sim	0.5072365	0.4973253
8	Não	0.6412836	0.6228136
8	Sim	0.6397243	0.6163598
9	Não	0.5182983	0.5076364
9	Sim	0.5098476	0.4986348

Tabela B.2: Resultados do *dataset* Linfoma com ρ=10

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
7	Não	0.4918267	0.4806231
7	Sim	0.4983267	0.4836420
8	Não	0.7634257	0.7439841
8	Sim	0.7528934	0.7341864
9	Não	0.5647238	0.5415379
9	Sim	0.5512345	0.5316324

Tabela B.3: Resultados do *dataset* Linfoma com ρ=50.5

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
7	Não	0.5186472	0.4946539
7	Sim	0.5037826	0.4912434
8	Não	0.8312867	0.8097356
8	Sim	0.8227360	0.8012734
9	Não	0.5223641	0.5063125
9	Sim	0.5363154	0.5082456

Tabela B.4: Resultados do *dataset* Linfoma com ρ=100

B.2 *Dataset* Leucemia

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.5086321	0.4886424
2	Sim	0.5176345	0.5017735
3	Não	0.4286318	0.4100938
3	Sim	0.4378274	0.4199727
4	Não	0.3757351	0.3597913
4	Sim	0.3897913	0.3709827

Tabela B.5: Resultados do *dataset* Leucemia com ρ=1

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.6194723	0.5987364
2	Sim	0.6282347	0.6063143
3	Não	0.5075445	0.4918374
3	Sim	0.5187133	0.4826456
4	Não	0.4071253	0.3875534
4	Sim	0.4186134	0.4002763

Tabela B.6: Resultados do *dataset* Leucemia com ρ=10

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.7621789	0.7451237
2	Sim	0.7753409	0.7602133
3	Não	0.5121567	0.4982760
3	Sim	0.5191762	0.4997123
4	Não	0.5228371	0.5081237
4	Sim	0.5327366	0.5149723

Tabela B.7: Resultados do *dataset* Leucemia com ρ=50.5

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.8328167	0.8287231
2	Sim	0.8481637	0.8303254
3	Não	0.5617286	0.5481348
3	Sim	0.5761728	0.5601378
4	Não	0.5931268	0.5712876
4	Sim	0.6083165	0.5900723

Tabela B.8: Resultados do *dataset* Leucemia com ρ=100

B.3 *Dataset* tcga

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.3912745	0.3690238
2	Sim	0.4017263	0.3792786
3	Não	0.5976324	0.5806437
3	Sim	0.6086423	0.5823645
4	Não	0.4138476	0.4006238
4	Sim	0.4284270	0.4088175

Tabela B.9: Resultados do *dataset* tcga com ρ=1

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.4471286	0.4297612
2	Sim	0.4511386	0.4383671
3	Não	0.6717654	0.6607361
3	Sim	0.6976215	0.6772654
4	Não	0.4982347	0.4712576
4	Sim	0.5027313	0.4862303

Tabela B.10: Resultados do *dataset* tcga com ρ=10

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.4682319	0.4448329
2	Sim	0.4726543	0.4607126
3	Não	0.8301234	0.8027363
3	Sim	0.8473261	0.8321278
4	Não	0.5282437	0.5068371
4	Sim	0.5353742	0.5192814

Tabela B.11: Resultados do *dataset* tcga com ρ=50.5

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.5121934	0.4902349
2	Sim	0.5192768	0.5062373
3	Não	0.9039847	0.8719002
3	Sim	0.9103944	0.8849021
4	Não	0.4963283	0.4723651
4	Sim	0.5034267	0.4873642

Tabela B.12: Resultados do *dataset* tcga com ρ=100

B.4 *Dataset* bone_marrow

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.3684932	0.3383029
2	Sim	0.3702319	0.3590237
3	Não	0.5408476	0.5283749
3	Sim	0.5529821	0.5303429
4	Não	0.4098327	0.3839021
4	Sim	0.4193282	0.4007123

Tabela B.13: Resultados do *dataset* bone_marrow com ρ=1

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.4082032	0.3943522
2	Sim	0.4194578	0.4067382
3	Não	0.7091820	0.6783902
3	Sim	0.7194384	0.6897233
4	Não	0.4352632	0.4142230
4	Sim	0.4392763	0.4086125

Tabela B.14: Resultados do *dataset* bone_marrow com ρ=10

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.5028396	0.4828374
2	Sim	0.5147369	0.4971823
3	Não	0.8532721	0.8403929
3	Sim	0.8618743	0.8412387
4	Não	0.5392318	0.5065324
4	Sim	0.5438126	0.5209342

Tabela B.15: Resultados do *dataset* bone_marrow com ρ=50.5

Nº de *Clusters*	Seleção de características?	F1-*Score*	*Rand Index*
2	Não	0.5798744	0.5492849
2	Sim	0.5963264	0.5625638
3	Não	0.9098264	0.8903844
3	Sim	0.9173942	0.9039203
4	Não	0.6072823	0.5836476
4	Sim	0.6097378	0.5897371

Tabela B.16: Resultados do *dataset* bone_marrow com ρ=100