UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia

Programa de Pós-Graduação em Ciência da Computação

Aprendizado semi-supervisionado e não supervisionado para análise de dados de expressão gênica

Fabiana Mari Assao

São Carlos-SP
Maio/2008

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária da UFSCar

A844as

Assao, Fabiana Mari.
Aprendizado semi-supervisionado e não supervisionado para análise de dados de expressão gênica / Fabiana Mari Assao. -- São Carlos : UFSCar, 2008.
117 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2008.

1. Método de agrupamento. 2. Agrupamento semi-supervisionado. 3. Expressão gênica. 4. Aprendizado do computador. 5. Bioinformática. I. Título.

CDD: 004 (20ª)

Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia

Programa de Pós-Graduação em Ciência da Computação

"Aprendizado semi-supervisionado e não supervisionado para análise de dados de expressão gênica"

FABIANA MARI ASSAO

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Membros da Banca:

Profa. Dra. Heloisa de Arruda Camargo
(Orientadora – DC/UFSCar)

Prof. Dr. Mauro Biajiz
(Co-orientador - DC/UFSCar)

Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho (ICMC/USP)

Prof. Dr. Júlio César Nievola (PUC/PR)

São Carlos
Maio/2008

AGRADECIMENTOS

À Profa Heloisa de Arruda Camargo por sua orientação, pelas oportunidades a mim concedidas e principalmente pela possibilidade da “troca” de experiências e aprendizagens.

Ao Prof Mauro Biajiz pelas sugestões valiosas que contribuíram para o aprimoramento deste trabalho.

Aos amigos e incentivadores: Joelle, Rabelo, Bruno, Raphael e Rubens. Aos amigos sempre presentes, em todos os momentos da minha vida... Fabi H., Nina, Glau, Dan, Hideo, Leo, Edu, Akemi, Leone

Aos amigos do trabalho: Gilson, Davi, Marcel, Deodato, Jú e Alê. Aos meus gerentes: César, Mauro e, principalmente, ao Contrucci que proporcionou horários flexíveis para conciliar o trabalho e o mestrado.

Um agradecimento especial aos meus pais FELISBERTO e TEREZA, eternos professores, que compartilharam comigo *todos* os momentos desta jornada, sempre oferecendo palavras e gestos de apoio que me permitiram chegar até aqui. Aos meus irmãos, Tatiana, Mariana e William; ao meu cunhado, Mineo; tias e vovó, pelo apoio e incentivo.

A todos que contribuíram de forma direta ou indireta para a realização deste trabalho.

RESUMO

O agrupamento de dados destacou-se nas últimas décadas como uma importante ferramenta para a análise de dados de expressão gênica. Nos últimos anos, em função do progresso das pesquisas para rotulação de genes, surgiu um interesse pelas técnicas de agrupamento semi-supervisionado, que utilizam o conhecimento prévio disponível sobre a função de alguns genes para descobrir funções de outros genes por meio do agrupamento. Neste trabalho são investigados algoritmos de agrupamento semi-supervisionado e não supervisionados aplicados a dados de expressão gênica. O intuito é realizar uma inspeção das vantagens e desvantagens da utilização destes métodos de agrupamento e, a partir disso, prover subsídios para obtenção de resultados significativos para a área de Biologia. Foram implementados e testados algoritmos de agrupamento com diferentes características, com o objetivo de verificar evidências de eventuais ganhos obtidos com a rotulação parcial dos genes com relação a técnicas não-supervisionadas. Os experimentos realizados consideraram conjuntos de dados do domínio de expressão gênica e de outros domínios mais genéricos. Os resultados obtidos foram avaliados com medidas de validação usualmente aplicadas em contextos semelhantes. Assim, as análises desenvolvidas reforçam o importante papel da computação na análise de dados biológicos, a fim de acelerar o processo de obtenção de resultados e conclusões, na compreensão das estruturas e funções dos genes. Os resultados obtidos neste trabalho justificam o grande investimento na pesquisa do comportamento de técnicas semi-supervisionadas em dados de expressão gênica, como veremos mais adiante.

Palavras chaves: Agrupamento, Agrupamento semi-supervisionado, Expressão Gênica, Aprendizado de Máquina, Bioinformática.

ABSTRACT

Data clustering has been seen, in the last decades, as an important tool for gene expression data analysis. In recent years, due to the progress in gene annotation research, a growing interest has been noticed for the semi-supervised clustering techniques, which use knowledge previously available about some gene functions to discover functions of other genes by means of clustering. This work investigates non-supervised and semi-supervised clustering algorithms applied to gene expression data. The goal is to perform an inspection on strengths and weaknesses of the use of such clustering methods and, based on these findings, to provide ways of obtaining results significant to biology. Algorithms with different characteristics were implemented and tested, with the objective of verifying evidences of eventual gains with the partial labeling, as compared to the non-supervised techniques. The experiments considered data sets from the gene expression domain as well as more generic domains. The obtained results were evaluated with validation measures usually applied in similar contexts. The analysis developed, though, emphasize the important role of computational techniques in biological data analysis, by accelerating the process of deriving results and conclusions, to better understand gene functions and structures. The results of this stydy justify the large investiment in the research of behavior of semi-supervised techniques in gene expression data, as we shall see.

Keywords: Clustering, Semisupervised Clustering, Gene Expression, Machine Learning, Bioinformatics.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE GRÁFICOS

SIGLAS

DNA	Ácido desoxirribonucleico
BHI	Índice de Homogeneidade Biológica
BIRCH	Balanced Iterative Reducing and Clustering using Hierarchies
BSI	Índice de Estabilidade Biológica
cDNA	DNA complementar
EM	Expectation Maximization
EST	Expressed Sequence Tag
HC	Hierarchical Clustering
MI	Mutual Information
MIPS	Munich Information Center for Protein Sequences
mRNA	RNA mensageiro
ORF	Open Reading Frames
RNA	Ácido ribonucleico
rRNA	RNA ribossômico
RT-PCR	Reverse Transcription - Polymerase Chain Reaction
SAGE	Serial Analysis of Gene Expression
SOM	Self-Organization Map
SVM	Máquina de Vetores de Suporte
tRNA	RNA transportador
UCI	University of California - Irvine

SUMÁRIO

Capítulo 1 Introdução

1.1 Contexto

Em linhas gerais, a Bioinformática é a área que investiga, entre outras coisas, a aplicação de metodologias e técnicas computacionais, a problemas da biologia. Os problemas da biologia que mais se beneficiam do uso da computação estão na maioria das vezes relacionados ao armazenamento e análise de dados e à descoberta de novos conhecimentos embutidos nesses dados. Nesse contexto, o aprendizado de máquina oferece um rico conjunto de métodos e técnicas para análise dos dados obtidos experimentalmente.

O surgimento da tecnologia de microarray de DNA, que tem sido utilizada como uma importante metodologia na biologia molecular experimental tornou possível a obtenção de grande volume de dados valiosos relativos ao perfil de expressão dos genes. A identificação de estruturas presentes nesses dados é um importante mecanismo para melhorar a compreensão da genômica funcional. Um experimento de microarray obtém dados de expressão de genes sob condições que podem ser instantes de tempo durante um processo biológico ou diferentes amostras de tecidos. A análise deste tipo de dados apresenta certas particularidades comparando com outras bases de dados, considerando que usualmente o número de amostras é pequeno (na ordem de dezenas de amostras), e o número de genes é muito grande (tipicamente da ordem de milhares de genes).

Entre as técnicas para análise de dados utilizadas neste contexto, destacam-se as técnicas de agrupamento de dados, que são o principal representante da categoria de métodos de aprendizado não supervisionado, isto é, métodos destinados a análise de dados que não possuem informação prévia sobre a classe a que pertencem. Assim, tais métodos têm como objetivo agrupar dados de acordo com uma medida de similaridade, sendo que os grupos encontrados devem refletir a estrutura subjacente dos dados. O método de agrupamento é frequentemente utilizado na área de Biologia para determinar os grupos de dados que têm comportamento similar e que, portanto, contêm elementos com característica de interesses comuns.

Como salientado por Jiang, Tang e Zhang (2004), uma das características dos dados de expressão gênica é que o agrupamento pode ser aplicado tanto a genes quanto a amostras. No agrupamento baseado em genes, os genes são tratados como objetos e as amostras são os

atributos. O agrupamento baseado em amostras, por sua vez, considera as amostras como objetos e os genes como atributos. Este trabalho focaliza o agrupamento baseado em genes, no qual a suposição fundamental é que genes com a mesma função tendem a ter expressões semelhantes, assim, agrupando genes pelos seus perfis de expressão, obtêm-se grupos de genes com funções similares. Isso pode promover um melhor entendimento relativo à função de muitos genes para os quais essa informação ainda não estava disponível. Grande parte da pesquisa recente focaliza a utilização de métodos de agrupamento conhecidos bem como o desenvolvimento de novos algoritmos especificamente projetados para tratar as questões de análise de dados de expressão gênica.

Dados de expressão gênica compartilham das características dos dados de outros domínios que motivaram o crescente interesse no aprendizado semi-supervisionado, observado recentemente, ou seja, uma grande oferta de dados não rotulados e poucos dados rotulados. O aprendizado semi-supervisionado utiliza dados rotulados e não rotulados durante o processo de treinamento para melhorar o resultado obtido e pode ser aplicado tanto a classificação como a agrupamento.

Na classificação semi-supervisionada, supõe-se que os dados não rotulados, que são desconsiderados na classificação convencional, podem trazer informações relevantes e, quando são também explorados, geralmente contribuem para obtenção de uma melhor função de classificação. Quanto ao agrupamento semi-supervisionado, diversos trabalhos relatam melhorias significativas obtidas com o uso de informação previamente disponível sobre o domínio do conhecimento considerado, que pode vir na forma de dados rotulados ou de restrições entre pares.

O agrupamento semi-supervisionado aplicado à análise de dados de expressão gênica tira proveito da informação biológica cada vez mais disponível sobre os genes, especialmente quanto às suas funções, o que pode ser obtido a partir das anotações de genes frequentemente divulgadas em bases de dados públicas disponíveis na web.

1.2 Motivação

Tem sido largamente observado que genes com funções semelhantes ou envolvidos nos mesmos processos biológicos têm grande probabilidade de mostrarem expressões semelhantes, portanto, agrupar genes com base em seus perfis de expressão provê um meio

para descoberta de função de genes. Essa constatação reforça a importância dos métodos de agrupamento como uma das etapas fundamentais da análise de dados de expressão gênica.

Um grande esforço tem sido empregado nos anos mais recentes no sentido de incorporar conhecimento biológico disponível sobre a função dos genes para melhorar o processo de agrupamento e obter assim, grupos compatíveis com a estrutura real dos dados e que possam trazer benefícios na análise posterior do seu significado quanto à descoberta de funções desconhecidas. Esse esforço resulta da constatação frequentemente relatada na literatura de que o uso de tal informação pode levar a resultados melhores quando comparados aqueles obtidos por abordagens convencionais.

O aprendizado semi-supervisionado em geral e o agrupamento semi-supervisionado em particular são tópicos de pesquisa recentes, objeto de intensa e crescente atividade de pesquisa, que ainda apresentam uma série de desafios e questões em aberto que merecem atenção adicional. A investigação dessas técnicas e o inevitável surgimento de outras mais novas certamente trarão benefícios para o campo da expressão gênica.

Muitos algoritmos de agrupamento não supervisionado e semi-supervisionado têm sido propostos para analisar dados de expressão gênica, mas a exemplo do que ocorre em todos os domínios quanto se aplicam esses algoritmos, pouca orientação está disponível para ajudar a escolher entre eles. O mesmo acontece com as possíveis medidas de similaridade utilizadas nesses algoritmos, as quais podem ter influência decisiva nos resultados. No caso específico de agrupamento semi-supervisionado, por ser um tópico de pesquisa mais recente, são ainda desconhecidas indicações a respeito da forma de utilização mais apropriada da informação disponível sobre as classes dos dados.

Assim, a grande variedade de algoritmos, medidas de similaridade e medidas de validação existentes, a intensidade das pesquisas em andamento que sinalizam um grande progresso na área em um futuro próximo, somadas às evidências de que tais algoritmos trazem ganhos significativos a todos os domínios do conhecimento e em particular ao campo de expressão gênica, justifica-se o investimento na pesquisa do comportamento, ainda bastante desconhecido, dos algoritmos aqui abordados.

1.3 Objetivo

O objetivo deste trabalho é investigar o desempenho de diferentes algoritmos de agrupamento semi-supervisionado e não supervisionado aplicados a dados de expressão

gênica, especificamente para a identificação e descoberta de novas funções de genes. Através desta análise, o propósito é contribuir para a criação de mecanismos que propiciem a geração de resultados significativos para a área de Biologia.

A implementação e testes dos algoritmos levou em consideração os diferentes níveis de expressão gênica contida em uma dada seqüência de mRNA (RNA mensageiro) sob diferentes condições. Diversos testes usando conjuntos de dados públicos já existentes foram realizados. Os resultados obtidos foram avaliados pela aplicação de duas medidas de validação comumente utilizadas nesse contexto, que se baseiam em informações previamente conhecidas relativas a classe dos dados.

1.4 Estrutura do Trabalho

Este trabalho está divido da seguinte maneira: No Capítulo 2 são apresentados os conceitos de agrupamento bem como seu processo, medidas de similaridades e os principais algoritmos comumente utilizados na literatura. No Capítulo 3 são apresentados os conceitos de agrupamento semi-supervisionado e os algoritmos estudados neste trabalho. No Capítulo 4 é apresentado o conceito de Expressão Gênica e uma visão geral sobre Biologia Molecular. No Capítulo 5 é apresentada a análise de agrupamento e agrupamento semi-supervisionado para dados de expressão gênica. Já no Capítulo 6 são descritos os métodos estudados e a avaliação experimental realizada. Finalmente no Capítulo 7 é apresentada a conclusão sobre este trabalho.

Capítulo 2 Agrupamento de Dados

2.1 Considerações Iniciais

As técnicas de agrupamento têm como objetivo agrupar padrões semelhantes para refletir a forma como os dados estão estruturados. Estes padrões são agrupados de acordo com suas similaridades ou dissimilaridades. Padrões denotam alguma abstração de um subconjunto dos dados em alguma linguagem descritiva de conceitos.

Neste trabalho os termos *grupos* e *clusters* serão tratados sem distinção de significados.

Neste capítulo são introduzidas definições sobre *cluster*, o processo de agrupamento, a preparação dos dados, e são também descritas as várias medidas de similaridade encontradas na literatura, as técnicas de agrupamento e finalmente a interpretação dos dados agrupados.

2.2 Definições

Cluster é um conjunto de pontos tal que os pontos de um mesmo *cluster* são mais similares de acordo com uma dada medida de similaridade, do que os pontos contidos em *clusters* diferentes.

Segundo (Tan, Steinbach e Kumar, 2005), o termo *cluster*, não tem uma definição precisa. Algumas definições comuns são:

- **Definição de *cluster* bem separado**: Um *cluster* é um conjunto de pontos tal que qualquer ponto em um *cluster* é mais próximo (ou mais similar) a cada outro ponto no *cluster* do que qualquer outro ponto não pertencente ao *cluster*. Algumas vezes, um limite é utilizado para especificar que todos os pontos de um grupo devem estar suficientemente próximos (ou similar) uns dos outros. Esta definição de um *cluster* é satisfeita apenas quando os dados contêm *clusters* naturais que estão bastante distantes uns dos outros.
- **Definição de *cluster* baseado em centro**: Um *cluster* é um conjunto de pontos tal que um objeto em um *cluster* está mais próximo (ou mais similar) ao centro do *cluster* do

que ao centro de qualquer outro *cluster*. O centro de um *cluster* é geralmente um centróide, a média de todos os pontos no *cluster*, mediana, ou um medóide, o ponto mais representativo do *cluster*.

- **Definição de *cluster* contínuo (vizinho mais próximo ou agrupamento transitivo)**: Um *cluster* é um conjunto de pontos tal que um ponto em um *cluster* está mais próximo (ou mais similar) a um ou mais pontos no *cluster* do que qualquer ponto que não pertence ao *cluster*.
- **Definição de *cluster* baseado em densidade**: Um *cluster* é uma região densa de pontos separada de outras regiões de alta densidade por regiões de baixa densidade. Esta definição é geralmente utilizada quando os *clusters* são irregulares ou se cruzam, e na presença de ruídos ou *outliers*. *Outliers* são pontos que estão fora do padrão global de distribuição dos dados.
- **Definição de *cluster* baseado em similaridade:** Um *cluster* é um conjunto de objetos que são similares, enquanto objetos em outros *clusters* não são "similares". Uma variação é definir um *cluster* como um conjunto de pontos que juntos criam uma região com uma propriedade local uniforme, como por exemplo, densidade ou tamanho.

O processo de agrupamento envolve diversas etapas que vão desde a preparação dos padrões até a interpretação dos *clusters* obtidos. Dependendo do objetivo que se deseja atingir com o agrupamento, a etapa de interpretação dos *clusters* pode ser omitida. O processo de agrupamento é detalhado na seção seguinte.

2.3 Componentes do Processo de Agrupamento

Tipicamente uma atividade de agrupamento padrão segue os passos ilustrados na Figura 1 (Jain, Murty e Flynn, 1999):

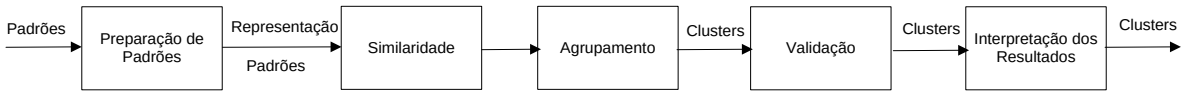


Figura 1: Estágios em agrupamentos

1. **Preparação de padrões** (opcionalmente inclui extração e/ou seleção de características): determina como os padrões serão representados. Em alguns casos, é

necessário aplicar algum tipo de transformação nos dados, como normalizações e seleção e/ou extração de características. Na seção 2.4 é detalhada a representação de padrões.

2. **Medida de similaridade:** consiste em definir uma medida de similaridade apropriada ao domínio da aplicação. Geralmente é fornecida por uma função de distância definida entre pares de padrões. É possível incluir na medida de distância aspectos conceituais (qualitativos) ou numéricos (quantitativos). As principais medidas de similaridades são abordadas na seção 2.5.
3. **Agrupamento**: consiste em aplicar um algoritmo de agrupamento com o intuito de agrupar dados de acordo com um objetivo específico. Existem inúmeros algoritmos que podem ser aplicados nesta etapa que podem ter como resposta um exemplo pertencente ou não a um dado *cluster* (*hard*) ou atribuição de um grau de pertinência a cada exemplo a cada um dos *clusters* (*fuzzy*). Alguns algoritmos de agrupamento são detalhados na seção 2.6.
4. **Validação**: nesta fase ocorre a validação dos resultados obtidos, após aplicado o algoritmo de agrupamento. Tem como objetivo determinar se o resultado é significativo. As principais formas de validação são apresentadas na seção 2.7.
5. **Interpretação dos resultados:** Nesta fase os resultados são examinados com relação a seus exemplos, com o objetivo de descrever a natureza dos grupos.

2.4 Preparação dos Padrões

A preparação dos dados envolve vários aspectos relacionados ao seu pré-processamento e a forma de representação apropriada para a sua utilização por um algoritmo de agrupamento. O pré-processamento pode envolver normalizações, conversão de tipos e redução do número de atributos por meio de seleção ou extração de características. Com isso o número de classes, de padrões disponíveis e o número, tipo e escala das características disponíveis para o algoritmo de agrupamento são informações bastante importantes.

Um padrão pode medir um objeto físico (por exemplo, uma cadeira) ou una noção abstrata (por exemplo, um estilo de escrita). Geralmente os padrões são representados como vetores multidimensionais, onde cada dimensão é uma única característica (Jain, Murty e Flynn, 1999) .

Duas questões importantes no que se refere às características são a escolha das mais relevantes para o agrupamento e a definição do número desejável de atributos nas aplicações do algoritmo de agrupamento. Resumidamente, o problema é encontrar um conjunto de características que melhor representa a similaridade que está sendo trabalhada. Para resolver esta questão podem ser utilizadas técnicas de seleção e/ou extração de características.

A seleção de característica é o processo de identificar o subconjunto mais efetivo das características disponíveis para ser utilizado no agrupamento. A extração de características é o uso de uma ou mais transformações das características de entrada disponíveis para salientar características presentes nos dados (Jain, Murty e Flynn, 1999).

Como mencionado anteriormente, o tipo e a escala das características são informações importantes na escolha da medida de similaridade e do algoritmo de agrupamento a ser empregado. Para cada tipo/escala de atributo existem medidas de similaridade apropriadas. O tipo de um atributo diz respeito ao grau de quantização nos dados. A escala indica a significância relativa dos números. Os possíveis tipos de atributos (Jain, Murty e Flynn, 1999):

1. **Binários**: são atributos que representam apenas dois valores;
2. **Discretos**: representam um número finito de valores;
3. **Contínuos**: podem assumir um número infinito de valores.

Em relação à escala, as características podem ser quantitativas e qualitativas (Jain, Murty e Flynn, 1999):

1. **Qualitativa**
    a. **Nominal**: os valores são apenas nomes distintos. Exemplos: CEP, cores, sexo.
    b. **Ordinal**: os valores apenas refletem uma ordenação. Exemplos: Ruim, Regular, Bom ou cores ordenadas pelo *spectro*.
2. **Quantitativa**
    a. **Intervalo:** a diferença entre os valores tem significado, isto é, existe uma unidade de medida. Exemplos: Em uma escala de 1 a 10 para dar notas as provas de alunos, duração de um evento.
    b. Razão: uma escala que possui zero absoluto. Exemplos: altura, comprimento, largura de um objeto.

Algumas vezes os padrões apresentam atributos de escalas diferentes ou a representação dos dados não é adequada para aplicação do algoritmo de agrupamento. Assim,

é necessária a aplicação de algumas transformações antes de iniciar a utilização dos dados. Quando os limites dos intervalos de valores de atributos são muito diferentes, um atributo pode dominar o resultado do agrupamento. Para solucionar este problema, é comum a padronização dos dados de forma que os atributos estejam na mesma escala.

Para buscar resultados significativamente melhores é necessária uma investigação cuidadosa das características disponíveis e das transformações que podem ser aplicadas aos dados. Um exemplo é o agrupamento dos pontos da Figura 2, onde os padrões formam um *cluster* curvilíneo com distâncias da origem similares.

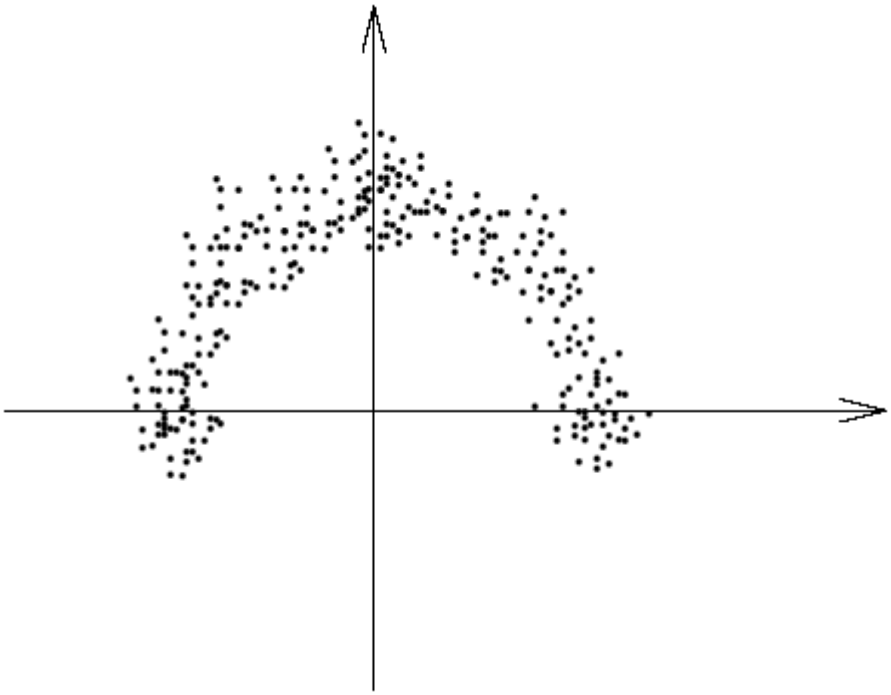


Figura 2: Um *cluster* curvilíneo cujos pontos são aproximadamente eqüidistantes da origem (Jain, Murty e Flynn, 1999)

Se coordenadas cartesianas fossem escolhidas para representar padrões, muitos algoritmos de agrupamento produziriam dois ou mais *clusters*. Entretanto, se fosse escolhida coordenadas polares para representação de padrões, uma solução com um *cluster* seria facilmente obtida (Jain, Murty e Flynn, 1999).

Na maioria dos casos, os dados são representados por uma matriz de padrões $X_{n \times d}$, em que *n* é o número de padrões e *d* é o número de atributos que representam os padrões. Cada elemento desta matriz, X_{ij}, contém o valor da *j-ésima* característica para o *i-ésimo* padrão. Cada padrão pode ser visto como um ponto neste espaço e um grupo como um conjunto de padrões próximos ou que satisfazem uma relação espacial.

Padrões também podem ser representados por uma matriz e um grafo de similaridade ou proximidade. Uma matriz de similaridade $S_{n\times n}$, contém os valores da similaridade/dissimilaridade entre dois padrões *i* e *j*, representados respectivamente na linha *i* e coluna *j* da matriz. Este valor é geralmente calculado por uma medida de similaridade.

2.5 Medida de Similaridade

Uma medida de similaridade indica o grau de semelhança entre dois padrões. Similaridade é fundamental para o conceito de grupos, portanto, a medida de similaridade tem um papel fundamental para a maioria dos algoritmos de agrupamento. A medida de similaridade a ser empregada deve ser cuidadosamente escolhida devido aos diferentes tipos e das diferentes escalas mensuradas dos dados.

Uma vez que agrupamento consiste em agrupar exemplos de tal forma que os padrões pertencentes ao mesmo *cluster* sejam mais semelhantes entre si, do que padrões pertencentes a *clusters* diferentes, de acordo com alguma medida de similaridade, é importante definir antecipadamente qual medida será utilizada.

Segundo (He, 1999), há pelo menos três conceitos de similaridade e distância, que precisam ser consideradas – entre entidades, entre uma entidade e um grupo de entidades, e entre dois grupos de entidades.

Geralmente, as medidas de similaridades devem satisfazer algumas propriedades:

1. Para dissimilaridade: $S_{ii} = 0$, para todo *i* (Os pontos não são diferentes de si próprios). Para similaridade: $S_{ii} \geq max\ S_{ij}$ (Os pontos são mais similares a si próprios), $i \neq j$
2. $S_{ij} = S_{ji}$ (Simetria)
3. $S_{ij} \geq 0$ para todo *i* e *j* (Positividade)
4. $S_{ij} = 0$ somente se $i = j$
5. $S_{ik} \leq S_{ij} + S_{jk}$ para todo *i*, *j* e *k* (Desigualdade triangular)

As medidas que satisfazem todas as propriedades acima são chamadas de métricas. Nem todas as medidas de similaridade empregadas são métricas. Se a medida de similaridade não satisfizer as propriedades 4 e 5, elas não são consideradas métricas.

De acordo com o tipo e escala das características, um conjunto de medidas de similaridade pode ser empregado. Para conjuntos de dados em que todas as características são contínuas e a escala é do tipo relacional, as medidas mais comumente utilizadas são as distâncias baseadas na métrica *Minkowski*, como a distância *Euclidiana*, de *Manhattan* e *supremum*. Para todas as características binárias é comum a utilização da distância de *Manhattan* (distância de *Manhattan* entre dois vetores é chamada de distância de *Hamming*). Para características binárias e nominais, existem coeficientes de casamento (*matching*), como coeficiente de casamento simples e coeficiente de *Jaccard*. (Gordon, 1999) apresenta várias medidas que são mais apropriadas quando as características são de um mesmo tipo. Nas seções seguintes serão apresentadas medidas diferentes para cada tipo de característica.

2.5.1 Medidas de atributos quantitativos

As medidas mais comuns para este tipo de dado são as métricas de *Minkowski*. Outras medidas como correlação de *Pearson* (Jain, Murty e Flynn, 1999) e separação angular são medidas de correlação que medem o coseno do ângulo entre dois vetores. A seguir, são descritas estas e outras medidas para atributos quantitativos.

2.5.1.1 Métricas de *Minkowski*

São métricas derivadas da Equação 1, de acordo com um valor definido para *p*, sendo $1 \leq p < \infty$. Chamadas de L_p medem a dissimilaridade entre padrões. Os valores menores de *p* correspondem a estimativas mais robustas (menos sensíveis a *outliers*). Uma desvantagem das métricas de *Minkowski* (Jain, Murty e Flynn, 1999) é que estas são sensíveis às variações de escala dos atributos (atributos representados em uma escala maior tendem a dominar os outros). Uma solução para este problema é aplicar a normalização dos atributos para um intervalo comum, ou outros esquemas de ponderação (Jain, Murty e Flynn, 1999).

$$S_{ij} = \left(\sum_{k=1}^{d} |x_{ik} - x_{jk}|^p\right)^{1/p} \qquad (1)$$

Equação 1: Métrica de *Minkowshi*

Alguns valores de *p* definem métricas bem conhecidas:

- $p = 1$: Distância de *Manhattan* (Jain, Murty e Flynn, 1999) (também conhecida como bloco-cidade), dada pela Equação 2:

$$S_{ij} = \sum_{k=1}^{d} |x_{ik} - x_{jk}| \qquad (2)$$

Equação 2: Distância de *Manhattan*

- p = 2: Distância Euclidiana tem um significado de variância total entre *clusters*. É uma das distâncias mais utilizadas e é apropriada para conjuntos de dados que possuem *clusters* compactos e isolados. É dada pela Equação 3:

$$S_{ij} = (\sum_{k=1}^{d} |x_{ik} - x_{jk}|^2)^{1/2} \qquad (3)$$

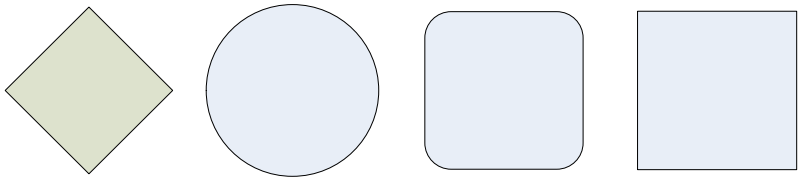
Equação 3: Distância *Euclidiana*

- $p = \infty$: Distância *supremum*, dada pela Equação 4, calcula o máximo da diferença absoluta em coordenadas, ou seja, é a diferença máxima entre quaisquer componentes de vetores.

$$S_{ij} = \max_{1 \le k \le d} |x_{ik} - x_{jk}| \qquad (4)$$

Equação 4: Distância *supremun*

Na Figura 3, é ilustrada a região formada pelos pontos igualmente distante da origem segundo a distância *Minkowski* para vários valores de *p* (Sanches, 2003).



(a) (b) (c) (d)

Figura 3: Efeito da variação de *p* na medida de *Minkowski*:
(a) p = 1, (b) p = 2, (c) p = 4 e (d) p = ∞

2.5.1.2 Métrica de *Camberra*

É uma métrica muito sensível a pequenas mudanças próximas a $x_{ik} = 0 = x_{jk}$. Esta métrica já possui uma padronização embutida e é dada pela Equação 5:

$$S_{ij} = \sum_{k=1}^{d} |x_{ik} - x_{jk}| / (|x_{ik}| + |x_{jk}|) \qquad (5)$$

Equação 5: Métrica de *Camberra*

2.5.1.3 Separação angular ou coseno

É calculada através do ângulo formado entre dois vetores, sendo que o primeiro é calculado a partir da origem até o padrão e outro a partir da média dos dados. A separação angular s assume valores no intervalo [-1, 1]. A distância angular é calculada por $S_{ij} = 1 - s$, fazendo que S_{ij} assuma valores entre 0 e 2. É possível calcular a distância angular através da distância angular absoluta, dada por $S_{aij} = 1 - |s|$. A medida de separação angular é dada pela Equação 6:

$$S_{ij} = \frac{\sum_{K=1}^{d} x_{ik}\, x_{jk}}{\left(\sum_{K=1}^{d} x_{ik}^{2} \sum_{l=1}^{d} x_{il}^{2}\right)^{1/2}} \qquad (6)$$

Equação 6: Separação angular ou coseno

2.5.1.4 Coeficiente de correlação de *Pearson*

É dada pela Equação 7:

$$S_{ij} = \frac{\sum_{K=1}^{d} (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)}{\left(\sum_{K=1}^{d} (x_{ik} - \bar{x}_i)^{2} \sum_{l=1}^{d} (x_{jl} - \bar{x}_j)^{2}\right)^{1/2}} \qquad (7)$$

Equação 7: Coeficiente de correlação de *Pearson*

em que $\bar{x}_i = \sum_{k=1}^{d} x_{ik}/d$, com os valores no intervalo [-1, 1]. O coeficiente de correlação é insensível a diferenças na magnitude dos atributos. É sensível à *outliers* e é menos intuitivo que a distância *Euclidiana*. A correlação de *Pearson (rp)*, é dada por $S_{ij} = 1 - rp$, o que faz com que S_{ij} assuma valores entre 0 e 2. Uma das variações da distância de *Pearson* é a distância absoluta de *Pearson*, dada por $S_{aij} = 1 - |rp|$.

2.5.1.5 Correlação de *Spearman*

É uma alternativa não-paramétrica para o coeficiente de correlação de *Pearson*. Comparada ao coeficiente de correlação de *Pearson*, a correlação de *Spearman* é mais robusta a dados irregulares (*outliers*). Para calcular a correlação de *Spearman (rs)*, é necessário que os atributos dos dados sejam ordenados segundo seus valores. Com isso, é calculada a correlação de *Spearman* para os dados, porém a posição dos atributos ordenados é utilizada no lugar dos seus valores. A distância referente à correlação de *Spearman* é calculada por $S_{ij} = 1 - rs$.

2.5.1.6 Distância de *Mahalanobis*

É dada pela Equação 8, em que C_{kl} é o elemento da *k*-ésima linha e *l*-ésima coluna da inversa da matriz de covariância.

$$S_{ij} = \left(\sum_{K=1}^{d} \sum_{l=1}^{d} (x_{ik} - x_{jk}) C_{kl} (x_{il} - x_{jl}) \right)^{1/2} \qquad (8)$$

Equação 8: Distância de *Mahalanobis*

Esta distância incorpora a correlação entre as características e padroniza cada característica para média zero e variância um. A idéia é associar diferentes pesos à diferentes características com base em suas variâncias e a correlação linear entre pares de padrões (Jain, Murty e Flynn, 1999). Assume-se implicitamente que as densidades condicionais da classe são unimodais e caracterizadas por um espalhamento multidimensional (Jain, Murty e Flynn, 1999). A aplicação dessa medida de distância pode minimizar as distorções causadas pelas medidas de correlações lineares entre características.

Na Figura 4, é ilustrado um exemplo de região formada pelos pontos igualmente distantes da origem segundo a distância *Mahalanobis*. Entretanto, dependendo dos dados, essa medida pode originar *clusters* com formatos elipsoidais com rotação à direita, e até mesmo *clusters* circulares.

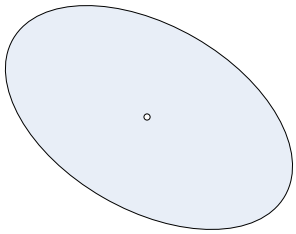


Figura 4: Formato do *cluster* encontrado pela distância *Mahalanobis*

2.5.2 Medidas para atributos binários

Há muitas medidas de similaridades entre vetores binários. Estas medidas são referenciadas como coeficientes de similaridades, e têm valores entre 0 e 1. O valor 1 indica que dois vetores são completamente similares, enquanto o valor 0 indica que os vetores não são de modo algum similares.

A comparação de dois vetores binários, *p* e *q*, conduz a quatro derivações:
N_{01} = número de padrões, em que *p* é igual a 0 e *q* é igual a 1.
N_{10} = número de padrões, em que *p* é igual a 1 e *q* é igual a 0.
N_{00} = número de padrões, em que *p* é igual a 0 e *q* é igual a 0.
N_{11} = número de padrões, em que *p* é igual a 1 e *q* é igual a 1.

2.5.2.1 Coeficiente de Casamento Simples

$$S_{ij} = \frac{a_{00} + a_{11}}{a_{00} + a_{11} + a_{01} + a_{10}} = \frac{a_{00} + a_{11}}{d} \qquad (9)$$

Equação 9: Coeficiente de Casamento Simples

2.5.2.2 Coeficiente de Jaccard

$$S_{ij} = \frac{a_{11}}{a_{11} + a_{01} + a_{10}} = \frac{a_{11}}{d - a_{00}} \qquad (10)$$

Equação 10: Coeficiente de Jaccard

2.5.3 Medidas para atributos nominais e ordinais

Estas medidas focalizam-se na determinação da contribuição de cada variável. As medidas de similaridade entre pares de padrões são obtidas pela soma das contribuições individuais de todas as variáveis.

2.5.3.1 Similaridade nominal/ordinal geral

É dada pela Equação 11, em que S_{ijk} é a contribuição de cada padrão baseada em índices de discordância entre pares de estados dos atributos categóricos.

$$S_{ij} = \sum_{K=1}^{d} S_{ijk} \qquad (11)$$

Equação 11: Similaridade nominal/ordinal geral

2.5.4 Medidas para atributos mistos

A medida para atributos mistos é adequada para obter a similaridade entre padrões que contenham características de tipos diferentes, pois se adequa a qualquer um dos tipos individualmente.

2.5.4.1 Coeficiente geral de similaridade

É dado pela Equação 12, em que s_{ijk} é a contribuição do *k*-ésimo atributo para a similaridade e w_{ijk} é 0 ou 1, dependendo se a comparação para a variável *k* é válida ou não. O valor de s_{ijk} pode ser definido para atributos de tipos diferentes.

$$S_{ij} = \sum_{K=1}^{d} w_{ijk}\, s_{ijk} / \sum_{K=1}^{d} w_{ijk} \qquad (12)$$

Equação 12: Coeficiente geral de similaridade

A seguir, são descritas algumas as medidas de distâncias entre grupos de objetos. Dados n_k pontos de dimensão d em um *Cluster* $C_i = \{x_i \mid i = 1, ..., n_k\}$, algumas medidas de dissimilaridade (distância) entre *clusters* se baseiam nos conceitos de centróide, x_0, raio, R e diâmetro, D, da terminologia de espaço vetorial, dados respectivamente pelas equações: Equação 13, Equação 14 e Equação 15. R é a distância média dos pontos do *cluster* ao centróide e D é a distância média entre pares (*pairwise average distance*) em um *cluster*.

$$x_0 = \frac{\sum_{i=1}^{nk} x_i}{n_k} \qquad (13)$$

Equação 13: Centróide

$$R = \left(\frac{\sum_{i=1}^{nk} (x_i - x_0)^2}{n_k} \right)^{1/2} \qquad (14)$$

Equação 14: Raio

$$D = \left(\frac{\sum_{i=1}^{nk} \sum_{j=1}^{nk} (x_i - x_j)^2}{n_k (n_k - 1)} \right)^{1/2} \qquad (15)$$

Equação 15: Diâmetro

Dados dois *clusters* $C_1 = \{x_i \mid i = 1, 2, ..., n_{k1}\}$ e $C_2 = \{x_j \mid j = n_{k1} + 1, n_{k1} + 2, ..., n_{k1} + n_{k2}\}$, com os respectivos centróides x_{01} e x_{02}, podem ser definidas as seguintes distâncias entre dois *clusters* (Zhang, Ramakrishnan e Livny, 1996).

2.5.4.2 Distância *Euclidiana* do centróide

$$D = ((x_{01} - x_{02})^2)^{1/2} \qquad (16)$$

Equação 16: Distância *Euclidiana* do centróide

2.5.4.3 Distância *Manhattan* do centróide

$$D = |x_{01} - x_{02}| \qquad (17)$$

Equação 17: Distância *Manhattan* do centróide

2.5.4.4 Distância *inter*-grupos

$$D = \left(\frac{\sum_{i=1}^{nk_1} \sum_{j=nk_1+1}^{nk_1+nk_2} (x_i - x_j)^2}{n_{k1}\, n_{k2}} \right)^{1/2} \qquad (18)$$

Equação 18: Distância *inter*-grupos

2.5.4.5 Distância *intra*-grupo

$$D = \left(\frac{\sum_{i=1}^{nk_1+nk_2} \sum_{j=1}^{nk_1+nk_2} (x_i - x_j)^2}{(n_{k1} + n_{k2})\,(n_{k1} + n_{k2} - 1)} \right)^{1/2} \qquad (19)$$

Equação 19: Distância *intra*-grupo

2.5.4.6 Distância de variação *intra*-grupo (*variance increase*)

Dados dois *clusters* $C_1 = \{x_i \mid i = 1, 2, ..., n_{k1}\}$ e $C_2 = \{x_j \mid j = n_{k1} + 1, n_{k1} + 2, ..., n_{k1} + n_{k2}\}$, com os respectivos centróides x_{01} e x_{02}, podem ser definidas as seguintes distâncias entre dois *clusters* (Zhang, Ramakrishnan e Livny, 1996)

$$D = \sum_{K=1}^{nk_1+nk_2} \left(x_k - \frac{\sum_{l=1}^{nk_1+nk_2} x_l}{n_{k1} + n_{k2}} \right)^2 - \sum_{i=1}^{nk_1} \left(x_i - \frac{\sum_{l=1}^{nk_1} x_l}{n_{k1}} \right)^2 - \sum_{j=n_{k1}+1}^{nk_1+nk_2} \left(x_j - \frac{\sum_{l=n_{k1}+1}^{nk_1+nk_2} x_l}{n_{k2}} \right)^2 \qquad (20)$$

Equação 20: Distância de variação *intra*-grupo

2.6 Algoritmos de Agrupamento

Um dos algoritmos de agrupamento comumente utilizado é o agrupamento hierárquico. Este tipo de agrupamento define um dendograma (árvore) relacionando objetos similares nas mesmas subárvores. No algoritmo hierárquico aglomerativo, cada objeto é inicialmente considerado uma subárvore (*cluster*). Em cada passo, subárvores similares (*clusters*) são agrupados para formar o dendograma.

Outro método popular para agrupamento de dados de expressão gênica é o *K-means*. *K-means* inicia com o número de *clusters* definido pelo usuário. Depois, os centróides dos *clusters* são iniciados, geralmente randomicamente. É permitido que os dados sejam movidos de *clusters* para os centros de *cluster* mais próximos. Este passo é repetido até convergir.

Self-Organization Maps (SOM) é um outro método bastante popular para agrupamento de dados de expressão gênica. SOM é uma rede de mapa de unidades que funciona como o *K-means*, procurando um ótimo posicionamento do modelo de vetores associados ao mapa de unidade. Como resultado, as unidades formam um mapa para a

representação dos dados agrupados, embora alguns mapas de unidades também possam estar vazios (Toronen, 2004).

Nas próximas seções são descritos, com mais detalhe, os algoritmos *K-means*, algoritmo hierárquico e SOM.

2.6.1 K-means

O algoritmo *K-means* é um típico método de agrupamento baseado em partições (Macqueen, 1967), que consiste em particionar o conjunto de dados em *k* subconjuntos disjuntos com base em uma medida de similaridade, onde *k* é um dado número pré-estabelecido.

O algoritmo começa inicializando um conjunto de *k* centróides para os *clusters*. Cada padrão pertencente ao conjunto de dados é representado por um ponto no espaço *d*-dimensional, onde *d* é o tamanho do vetor de entrada, ou seja, o número de características de cada padrão.

A cada iteração do algoritmo, um vetor média é calculado para cada *cluster* e os pontos são realocados ao *cluster* com o vetor média mais próximo, de acordo com a medida de similaridade escolhida. O algoritmo procede com este processo repetidamente, até que os *clusters* convirjam (o vetor média calculado para cada *cluster* não é alterado) ou até que o número máximo de iterações seja atingido.

A função objetivo é descrita pela Equação 21, em que C_i é o centróide do *cluster* G_i e $D(x_j,C_i)$ é a distância entre um ponto x_j e C_i, sendo que $x_j \in G_i$.

$$E = \sum_{i=1}^{k} \sum_{x_j \in G_i} D(x_j, C_i) \qquad (21)$$

Equação 21: Função objetivo

O centróide pode ser a média, medóide ou mediana dos pontos do *cluster*, dada Equação 13. Assim, o objetivo do algoritmo é minimizar o valor da função *E*, ou seja, diminuir a distância entre cada ponto e o centróide do *cluster* a qual ele pertence (Halkidi, Batistakis e Vazirgiannis, 2001).

O algoritmo *K-means* é simples e rápido. A complexidade do algoritmo é $O(l*k*n)$, onde *l* é o número de iterações e *k* é o número de *clusters*. Tipicamente, o algoritmo converge em um número pequeno de iterações. Entretanto, existem algumas desvantagens. Primeiramente, o número de *clusters* nos conjuntos de dados geralmente é desconhecido previamente. Para obter o número de *clusters* ótimo, os usuários geralmente executam o

algoritmo repetidamente com diferentes valores de *k* e os resultados do agrupamento são comparados. Para um conjunto grande de dados, este processo de ajuste extensivo do parâmetro pode não ser prático. Segundo, o algoritmo *K-means* força cada elemento para um *cluster*, que faz com que o algoritmo seja sensível à ruídos (Jiang, Tang e Zhang, 2004). Em terceiro lugar, o algoritmo é sensível a escolha inicial dos centróides e da sua forma de atualização. Dependendo da escolha dos centróides, o algoritmo pode convergir para um mínimo local.

2.6.2 Agrupamento Hierárquico

Em contraste com os agrupamentos baseados em partição, que tentam decompor diretamente o conjunto de dados em conjuntos de *clusters* disjuntos, o agrupamento hierárquico gera uma série hierárquica de *clusters* aninhados que podem ser representados graficamente por uma árvore, chamada dendograma. Os ramos do dendograma não armazenam somente a formação dos *clusters*, mas também indicam a similaridade entre os *clusters*. Cortando o dendograma em algum nível, pode-se obter um número específico de *clusters* (Jiang, Tang e Zhang, 2004).

Algoritmos de agrupamento hierárquico podem ser divididos em duas abordagens: a abordagem aglomerativa ou abordagem divisiva, que diferem em como o dendograma hierárquico é formado. Algoritmos aglomerativos (*bottom-up*) consideram inicialmente cada padrão como um conjunto individual, e a cada iteração, agrupam o par de padrões mais próximo em um *cluster*. O algoritmo prossegue até que todos os padrões sejam agrupados e estejam fundidos em um único *cluster*. Algoritmos divisivos (*top-down*) iniciam com um *cluster* contendo todos os padrões e, a cada iteração, os *clusters* são divididos até que cada *cluster* contenha somente um padrão. Para métodos aglomerativos, diferentes medidas de similaridade, tal como *single link*, *complete link* e *minimum-variance* (Jain, Murty e Flynn, 1999), derivam várias estratégias de fusão. Para métodos divisivos, o problema essencial é decidir como dividir os *clusters* em cada etapa. Alguns são baseados em métodos heurísticos tais como o algoritmo determinístico de recozimento (Jiang, Tang e Zhang, 2004).

Algumas vantagens do agrupamento hierárquico são sua flexibilidade em relação ao nível de granularidade, a fácil utilização de qualquer medida de similaridade e por sua aplicação a qualquer tipo de atributo. Porém, o critério de terminação é vago e a maioria dos algoritmos não melhoram os *clusters*, uma vez contruídos.

A maioria dos algoritmos hierárquicos utiliza métricas de integração (*linkage metrics*). Porém, existem várias outras implementações de algoritmos de agrupamento hierárquicos que

visam melhorias, por exemplo, na manipulação de *outliers*, obtenção de *clusters* de diferentes formas, tamanhos e escalabilidade. Existe um grande número de algoritmos hierárquicos. Dentre eles, podem ser destacados os algoritmos *BIRCH* (*Balanced Iterative Reducing and Clustering using Hierarchies*) (Zhang, Ramakrishnan e Livny, 1996).

A principal idéia é agrupar os pontos de dados em *sub-clusters* e depois agrupar estes *sub-clusters*. Com isso, o algoritmo precisa de uma única varredura na base de dados. Sua principal contribuição é sua habilidade de lidar com conjuntos de dados muito grandes.

Uma deficiência desse algoritmo é que ele apresenta um desempenho ruim quando os *clusters* não têm tamanho e forma uniformes. Além disso, ele é adequado para dados em espaços de vetor *Euclidiano*, ou seja, os dados devem ser métricos (dados para os quais médias fazem sentido).

2.6.3 SOM

O *Self-Organization Map* (*SOM*) foi desenvolvido por *(Kohonen, 1997),* que é baseado em uma única camada de rede neural. É uma rede neural artificial não supervisionada, frequentemente utilizada em tarefas de agrupamento e visualização. Neste tipo de rede, os neurônios são organizados em um reticulado uni ou bidimensional. A cada padrão de entrada apresentado à rede, os neurônios computam seus valores de ativação, ativando uma região diferente do reticulado. Para cada padrão de entrada, os neurônios de saída da rede competem entre si para serem ativados. O neurônio com maior valor de ativação é considerado o vencedor. Em seguida, é determinada a localização espacial de uma vizinhança de neurônios excitados, centrada no neurônio vencedor. O passo seguinte consiste em adaptar os pesos. Com os ajustes dos pesos, a resposta do neurônio vencedor à aplicação subseqüente de um padrão de entrada similar é melhorada.

Durante o funcionamento do algoritmo, os vetores de entrada direcionam o movimento dos vetores de peso para as áreas mais densas do espaço promovendo uma organização topológica dos neurônios da rede.

Uma das características marcantes do *SOM* é que ele gera um mapa intuitivamente atraente de conjuntos de dados de elevada dimensionalidade em espaços em 2D ou 3D e aproxima conjuntos similares. O processo de treinamento de neurônios do *SOM* é relativamente mais robusto que a abordagem *K-means* para agrupamento de dados com bastante ruídos.

Se o conjunto de dados for abundante em dados irrelevantes, tais como genes com padrões invariantes, *SOM* produzirá uma saída em que este tipo de dados estará contido na maioria dos *clusters*. Neste caso, *SOM* não é eficaz, pois a maioria dos padrões interessantes pode ser agrupada em somente um ou dois *clusters* e não poderão ser identificados (Jiang, Tang e Zhang, 2004).

2.7 Critérios de Validação

As seções precedentes apresentaram alguns algoritmos de agrupamento que dividiram o conjunto de dados baseado em critérios de agrupamento diferentes

Entretanto, diferentes algoritmos de agrupamento, ou até mesmo um único algoritmo de agrupamento usando diferentes parâmetros, geralmente resultam em diferentes conjuntos de *clusters*. Desta forma, é importante comparar os vários resultados gerados e selecionar um que seja adequado a melhor distribuição verdadeira dos dados. A validação dos *clusters* é o processo de avaliar a qualidade e a confiabilidade dos *clusters* gerados dos vários processos de agrupamento (Jiang, Tang e Zhang, 2004).

A avaliação do resultado de um agrupamento deve ser objetiva, tendo como propósito determinar se os *clusters* são significativos, ou seja, se a solução encontrada é representativa para o conjunto de dados analisado.

A validação do resultado de um agrupamento é realizada com base em dados estatísticos, que julgam, de maneira qualitativa, o mérito dos *clusters* encontrados. Um índice quantifica alguma informação a respeito da qualidade do agrupamento. A maneira pela qual um índice é aplicado para validar um agrupamento é dada pelo critério de validação. Existem três tipos de critérios para analisar a qualidade de um agrupamento: internos, externos e relativos.

Critérios internos avaliam o agrupamento com base nos dados originais (matriz de padrões ou matriz de similaridade) sem nenhum conhecimento externo ao agrupamento. Critérios externos avaliam um agrupamento de acordo com uma estrutura pré-estabelecida que reflita a intuição do pesquisador sobre a estrutura presente nos dados. Esta estrutura pré-estabelecida pode ser uma partição construída por um especialista da área com base em conhecimento prévio. Critérios relativos comparam diversos agrupamentos para decidir qual deles é melhor em um determinado aspecto (agrupamento mais estável ou mais adequado, por

exemplo). Podem ser usados para determinar o valor mais apropriado para um parâmetro, como número de *clusters*, ou para comparar diferentes algoritmos de agrupamento.

2.8 Interpretação dos resultados

Nesta etapa os resultados do agrupamento são examinados com base nos seus exemplos, com o objetivo de descrever a natureza do grupo gerado. A interpretação dos *clusters* pode permitir avaliações subjetivas que tenham significados práticos, ou seja, especialistas podem ter interesse em encontrar características semânticas de acordo com os padrões e valores de seus atributos em cada *cluster*.

2.9 Considerações Finais

Neste capítulo foram abordados tópicos relativos ao agrupamento clássico, tais como etapas do processo, algoritmos e medidas de similaridade utilizadas. A abordagem de agrupamento baseado em similaridade tem produzido algoritmos eficientes os quais têm se mostrado úteis em muitas aplicações. Entretanto, a abordagem clássica tem algumas limitações significantes.

No próximo capítulo serão apresentados conceitos e algoritmos de aprendizado semi-supervisionado.

Capítulo 3 Agrupamento Semi-Supervisionado

3.1 Considerações iniciais

Agrupamento de dados é visto tradicionalmente como um método não supervisionado de análise de dados. Na área de aprendizado de máquina, técnicas de agrupamento constituem a principal forma de aprendizado não supervisionado. Entretanto, em alguns casos informações sobre o domínio do problema estão disponíveis, além dos dados propriamente ditos. O agrupamento semi-supervisionado usa um número pequeno de dados rotulados para auxiliar o processo de agrupamento. Neste capítulo serão apresentados conceitos gerais sobre essa abordagem, iniciando com uma discussão sucinta dos conceitos no contexto mais abrangente de aprendizado semi-supervisionado, uma descrição de métodos representativos da classificação semi-supervisionada e por fim, uma apresentação mais detalhada dos algoritmos de agrupamento semi-supervisionado, que são o foco principal deste trabalho.

3.2 Aprendizado Semi-supervisionado

Em muitas tarefas de aprendizado, há uma grande quantidade de dados não rotulados e os dados rotulados são insuficientes, pois a geração de dados rotulados é frequentemente cara e demorada (Amini e Gallinari, 2003; Basu, Banjeree e Mooney, 2004). Algoritmos capazes de aprender a partir de exemplos rotulados e não rotulados têm despertado grande interesse na comunidade científica nos últimos tempos (Blum e Mitchell, 1998; Joachims, 1999; Nigam, Mccallum, Thrun e Mitchell, 2000). Esta nova maneira de aprendizado, chamada de aprendizado semi-supervisionado, se torna interessante quando há poucos exemplos rotulados, o que ocorre na maioria dos casos.

Quando existem exemplos rotulados pode-se utilizar o aprendizado supervisionado para induzir classificadores a partir destes exemplos. Caso contrário, quando os exemplos não estão rotulados, pode-se utilizar o aprendizado não supervisionado com o objetivo de encontrar os *clusters*. Já o aprendizado semi-supervisionado, consiste em utilizar algoritmos que aprendam a partir de exemplos rotulados e não rotulados.

O aprendizado semi-supervisionado é aplicável tanto em tarefas de classificação quanto em tarefas de agrupamento.

Em classificação supervisionada, há um conjunto fixo e conhecido de categorias e dados de treinamento rotulados com a categoria, que é usada para induzir a função de classificação. Em uma classificação semi-supervisionada, a idéia é rotular, com uma certa margem de segurança, alguns dos exemplos no conjunto de exemplos não rotulados, os quais são utilizados posteriormente durante a fase de treinamento do classificador, frequentemente resultando em uma classificação mais precisa (Blum e Langley, 1997) (Ghahramani e Jordan, 1994).

Em agrupamento não supervisionado, uma base de dados não rotulada é particionada em grupos de exemplos similares, tipicamente otimizando uma função objetivo que caracteriza partições boas. Em agrupamento semi-supervisionado, alguns dados rotulados são utilizados com os dados não rotulados para obter um melhor agrupamento servindo geralmente como um conhecimento prévio (Basu, Banerjee e Mooney, 2002).

Se os dados rotulados inicialmente representam todas as categorias relevantes, então ambos os algoritmos de agrupamento semi-supervisionado e classificação semi-supervisionada podem ser utilizados para categorização. No entanto, em muitos domínios, o conhecimento das categorias relevantes é incompleto. Diferentemente da classificação semi-supervisionada, agrupamento semi-supervisionado pode agrupar dados usando as categorias de dados inicialmente rotulados, extendendo e modificando o conjunto de categorias existentes se houver necessidade de refletir outras estruturas nos dados.

Assim, pode-se afirmar que o aprendizado semi-supervisionado ocupa uma posição intermediária entre aprendizado supervisionado e o não supervisionado, sendo o crescente interesse por essa abordagem nos últimos anos motivado principalmente por sua aplicação a domínios nos quais os exemplos não rotulados são fartos como processamento de imagens, mineração de textos e bioinformática.

Vários algoritmos de aprendizado semi-supervisionado têm sido propostos recentemente. A maioria deles tem como base algum algoritmo existente na literatura, o qual é modificado para tratar exemplos rotulados e não rotulados.

Na próxima seção são apresentados resumidamente os principais algoritmos de classificação semi-supervisionada e a seção seguinte aborda, de forma mais completa, os algoritmos de agrupamento semi-supervisionado, foco principal deste trabalho.

3.3 Classificação semi-supervisionada

Como foi explicado na seção anterior, na classificação há um conjunto fixo e conhecido de categorias. O objetivo da classificação semi-supervisionada é rotular exemplos do conjunto de exemplos não rotulados.

Nas próximas seções são apresentados os principais métodos de classificação semi-supervisionada.

3.3.1 CO-training

Co-training é uma técnica que tem por objetivo rotular exemplos automaticamente a partir de um pequeno conjunto de exemplos rotulados. Co-training se baseia na cooperação de dois algoritmos de aprendizado supervisionado. A idéia é fazer com que um classificador rotule exemplos para o outro classificador, aumentando a precisão de classificação dos exemplos.

O algoritmo proposto por (Goldman e Zhou, 2000), utiliza dois algoritmos de aprendizado supervisionado, *A* e *B*, um conjunto *U* de exemplos não rotulados, um conjunto *L* de exemplos rotulados, um conjunto L_A, com os exemplos que o algoritmo *B* rotulou para *A* (inicialmente vazio), e um conjunto L_B, com exemplos que o algoritmo *A* rotulou para *B*.

O algoritmo repete os passos a seguir até que ambos L_A e L_B não se alterem. O algoritmo inicia cada iteração com o treinamento de *A* para os exemplos rotulados $L \cup L_A$ para obter a hipótese H_A. Analogamente, treina-se o conjunto *B* para os exemplos rotulados $L \cup L_B$ para obter H_B. Cada algoritmo considera cada uma das suas classes e decide quais serão utilizadas para rotular exemplos em *U* para o algoritmo. Há dois testes que devem ser satisfeitos antes de se rotular algum dado. O primeiro deve garantir que a classe utilizada possua uma precisão no mínimo tão boa quanto à precisão da hipótese. O segundo teste é para ajudar a prevenir a degradação da performance devido ao aumento de ruídos nos dados. Para todo exemplo em *U* que tenha uma classe equivalente em H_A que passou por ambos os testes, A rotula o exemplo e o coloca em L_B. *B* rotula os exemplos da mesma maneira de *A*. Isto completa a iteração do algoritmo.

3.3.2 Support Vector Machine

Nas Máquinas de Vetores de Suporte (SVMs) (Boser, Guyon e Vapnik, 1992), funções não-lineares de *kernel* mapeiam os vetores de entrada em um espaço de mais alta dimensão (espaço de características), onde um hiperplano de separação é obtido para a resolução de problemas de classificação. As funções de *kernel* são geralmente executadas por funções não-lineares como polinômios, funções radiais e *Perceptron* (Vapnik, 1995). Um conceito fundamental em SVMs é o de margem de separação entre as classes que estão associadas ao erro permitido na classificação. A margem de separação é controlada pelo usuário como um parâmetro adicional de treinamento, além do erro do conjunto de treinamento. O problema de separação é formulado como um problema de programação quadrática com restrições lineares, que dependem dos dados de treinamento, dos parâmetros que definem os *kernels* e da margem de separação. A resolução do problema de programação quadrática resulta em vetores de suporte dentro da margem de separação definida pelo usuário.

Dado o conjunto de treinamento $\{(x_i,y_i)\}^N_{i=1}$ para o problema $W(\alpha) = 1^T\alpha - ½\ \alpha^T H\alpha$ sujeito a $\alpha^T y = 0$ e $0 \leq \alpha \leq C$, onde $H_{ij} = y_i y_i K(x_i,x_j) = y_i,y_j\varphi(x_i)^T\varphi(x_j)$, $K(.,.)$ é a função de kernel e $\alpha = [\alpha_1, \alpha_2, ..., \alpha_n]^T$ contém os Multiplicadores de Lagrange associados com cada vetor de entrada. Vetores C, 1 e 0 possuem dimensão $N \times 1$.

Os valores dos Multiplicadores de Lagrange indicam a posição de cada vetor de entrada em relação à margem de separação. Assim, somente aqueles vetores com valores de α dentro da margem de separação definida pela restrição $0 \leq \alpha \leq C$ interessam para o problema de separação. Estes vetores serão então selecionados como vetores de suporte para a definição da superfície de separação entre classes.

A abordagem de aprendizado semi-supervisionado visa minimizar a discordância entre os vários modelos construídos a partir de cada fonte de informação, utilizando um método de *co-updating* e fazendo uso de ambos dados, rotulados e não rotulados.

Existem muitos trabalhos sobre descoberta de funções gênicas em dados de microarray de expressão e seqüência de dados separadamente, no entanto, alguns trabalhos consideram estas informações juntas em um conjunto heterogêneo. O trabalho (Pavlidis, Cai, Weston e Grundy, 2001) é um dos poucos que combinam diferentes tipos de dados de funções de genes. É apresentando um métodos utilizando SVM para aprender a partir de um classificador as funções gênicasde um conjunto heterogêneo de dados que consistem dados de expressão de microarray e perfis filogenética (Li, Zhu e Ogihara, 2003).

No caso da classificação semi-supervisionada, o SVM é construído utilizando uma mistura de dados rotulados (conjunto de treinamento) e não rotulados (conjunto de trabalho). O objetivo é associar rótulos das classes para o conjunto de trabalho tal que o melhor SVM seja construído. Se o conjunto de trabalho está vazio, o método é o SVM padrão para classificação. Se o conjunto de treinamento está vazio, então o método se torna um método da forma de um aprendizado não supervisionado. O aprendizado semi-supervisionado ocorre quando ambos os conjuntos de treinamento e de trabalho não estão vazios. O aprendizado semi-supervisionado para problemas com pequenos conjuntos de treinamento e grandes conjuntos de trabalho é uma forma de agrupamento semi-supervisionado. Há algoritmos semi-supervisionados bem sucedidos como K-means e C-means fuzzy. Quando o conjunto de treinamento é grande em relação ao conjunto de trabalho, o SVM pode ser encardo como um método para resolver o problema de transdução (Bennett e Demiriz, 1998).

3.4 Agrupamento Semi-supervisionado

Diversos algoritmos com o objetivo de melhorar o agrupamento de dados explorando algum tipo de supervisão foram propostos nos últimos anos. A informação disponível para rotulação dos dados tem sido utilizada em duas abordagens diferentes, chamadas de abordagem baseada em restrições e abordagem baseada em métrica.

Nas abordagens baseadas em restrições o próprio algoritmo de agrupamento é modificado tal que a informação disponível fornecida pelo usuário é usada para guiar o algoritmo a um particionamento dos dados mais apropriado. Isso é feito por meio de mecanismos para:

- Modificar a função objetivo tal que ela inclua satisfação de restrições (Demiriz, Bennett e Embrechts, 1999),
- Reforçar restrições durante o processo de agrupamento (Wagstaff, Cardie, Rogers e Schroedl, 2001) ou
- Inicializar e restringir o agrupamento com base nos exemplos rotulados (Basu, Banerjee e Mooney, 2002).

O conhecimento disponível que permite a incorporação de supervisão parcial nessa classe de métodos pode surgir tanto na forma de restrições entre pares como na forma de rótulos para um subconjunto dos dados. As restrições entre pares podem ser da forma *must-link*, indicando que um par de dados deve pertencer ao mesmo *cluster* ou *cannot link*

indicando que os exemplos do par devem pertencer a *clusters* distintos. A informação na forma de rótulos de classes pode ser traduzida em restrições entre pares envolvendo os dados rotulados, e, reciprocamente, definido restrições consistentes entre pares, pode-se obter grupos de itens que devem pertencer ao mesmo *cluster* (Grira, Crucianu e Boujemaa, 2005).

Nas abordagens baseadas em métricas um algoritmo conhecido que usa métricas de distância é utilizado, mas a métrica é treinada anteriormente para satisfazer os rótulos ou restrições dos dados rotulados. Diversas medidas de distância têm sido usadas para agrupamento semi-supervisionado baseado em métricas incluindo distância Euclidiana treinada por um algoritmo de caminho mais curto (Klein, Kamvar e Manning, 2002), distância edição de cadeia aprendida usando Expectation Maximization (EM) (Bilenko e Mooney, 2003), divergência *KL* adaptada usando gradiente descendente (Cohn, Caruana e Mccallum, 2003) e distância de *Mahalanobis* treinada usando otimização convexa (Xing, Ng, Jordan e Russell, 2003); (Bar-Hillel, Hertz, Shental e Weinshall, 2003).

Outros trabalhos apresentam ainda algoritmos de agrupamento semi-supervisionado que combinam o aprendizado de métrica e o uso de restrições em uma abordagem unificada (Basu, Bilenko e Mooney, 2004).

Na seção seguinte são apresentados alguns dos principais algoritmos baseados em restrições encontrados na literatura, de interesse para este trabalho. As abordagens baseadas em métrica não serão abordadas com mais detalhes já que não fazem parte do escopo das investigações conduzidas nesta dissertação.

3.4.1 Algoritmos de Agrupamento Semi-Supervisionado

3.4.1.1 COP-K-means

O algoritmo é uma variante do algoritmo não supervisionado *K-means* e foi proposto por (Wagstaff, Cardie, Rogers e Schroedl, 2001). A principal diferença está na utilização de conhecimento prévio (*background knowledge*), descrito na forma de relações entre os exemplos, que é utilizado no processo de formação dos *clusters*. Esse conhecimento consiste em fornecer alguns exemplos que podem ou não ser agrupados em um mesmo *cluster*, os quais determinam dois tipos de restrições:

Must-link, que especifica que dois exemplos devem pertencer ao mesmo *cluster*;

Cannot-link, que especifica que dois exemplos não devem pertencer ao mesmo *cluster*.

Os *clusters* encontrados pelo *COP-K-means* devem respeitar todas as relações *must-link* e *cannot-link* impostas pelo usuário nos exemplos rotulados. Durante a construção dos *k clusters*, cada exemplo do conjunto de exemplos não rotulados é associado ao *cluster* mais próximo (Basu, Banerjee e Mooney, 2002).

Algoritmo: **Cop-Kmeans**
Entrada: Conjunto de dados D, *constraints must-link* $Con_{=} \subseteq DxD$, *constraints cannot-link* $Con_{\neq} \subseteq DxD$).

1. Seja $C_1, ..., C_k$ os centroides iniciais
2. Para cada ponto d_i em D, associar este ponto ao cluster mais próximo C_j tal que VIOLATE-CONSTRAINTS (d_i, C_j, $Con_{=}$, $Con_{\neq}$) é falso.
3. Para cada cluster C_i, atualizar os centroides, com a média dos pontos d_j que foram associados a este cluster.
4. Iterar os passos (2) e (3) até convergir.
5. Retornar $\{C_1, ..., C_k\}$.

VIOLATE-CONSTRAINED (ponto d, cluster C, *constraints must-link* $Con_{=} \subseteq DxD$, *constraints cannot-link* $Con_{\neq} \subseteq DxD$).

1. Para cada $\{d,d_{=}\} \in Con_{=}$: Se $d_{=} \notin C$, retornar verdadeiro.
2. Para cada $\{d,d_{\neq}\} \in Con_{\neq}$: Se $d_{\neq} \notin C$, retornar verdadeiro.
3. Senão, retornar falso.

Algoritmo 1: Cop-KMeans

3.4.1.2 SEEDED-K-means

Proposto por (Basu, Banerjee e Mooney, 2002), é um algoritmo variante do *K-means*, também particiona o conjunto de dados em *k clusters*. A diferença mais característica é que o algoritmo *SEEDED-K-means* utiliza exemplos inicialmente rotulados para calcular os centróides iniciais dos *clusters*, isto é, as sementes (*SEED*), e ao invés de escolhê-los aleatoriamente.

Considerando um conjunto de exemplos E, toma-se um subconjunto $S \subset E$ como sendo o conjunto de sementes. Na inicialização do algoritmo, o usuário é responsável por atribuir cada $x_i \in S$ a um dos k *clusters* a serem encontrados, dividindo o conjunto S em k subconjuntos S_l, de tal forma que $S = \cup^{k}_{l=1} S_l$. O algoritmo exige que para cada *cluster* seja atribuído, no mínimo, uma semente. Desta maneira, cada *cluster* terá seu centróide inicializado com a média das sementes atribuídas pelo usuário. A partição definida pelas sementes é usada apenas para inicialização e as sementes não são usadas nos passos seguintes do algoritmo.

Algoritmo: **Seeded-Kmeans**
Entrada: Conjunto de dados $X = \{x_1, ..., x_n\}$ $x_i \in R^d$, número de cluster k, conjunto $S = \bigcup_{i=1}^{k} S_i$ de sementes iniciais.
Saída: k partições disjuntas $\{X_i\}_{i=1}^{k}$ de X tal que a função objetiva Kmeans é otimizada.
Método:
1. Inicialize: $\mu_h^{(0)} \longleftarrow 1/|S_h| \sum_{x \in Sh} x$, para $h = 1, ..., k$; $t \longleftarrow 0$
2. Repetir até convergir
   2a. Associar cluster: Associar cada ponto x para o cluster h^* (isto é, conjunto $X_{h^*}^{(t+1)}$), para $h^* = \arg_h \min ||x - \mu_h^{(t)}||^2$
   2b. Calcular centroides: $\mu_h^{(t+1)} \longleftarrow 1/|X_h^{(t+1)}| \sum_{x \in Xh}^{(t+1)} x$
   2c. $t \longleftarrow (t+1)$

Algoritmo 2: Seeded-KMeans

3.4.1.3 CONSTRAINED-K-means

O algoritmo *CONSTRAINED-K-means*, também proposto por (Basu, Banerjee e Mooney, 2002), é uma melhoria do algoritmo *SEEDED-K-means.* A diferença está nos passos seguintes a inicialização dos centróides, nos quais os exemplos que fazem parte do conjunto das sementes, e que foram inicialmente associados a um dado *cluster* pelo usuário, não poderão ser associados a um outro *cluster*. Assim, apenas os exemplos não selecionados como sementes serão reagrupados, diferentemente do *SEEDED-K-means* em que as sementes podem vir a pertencer a *clusters* diferentes daqueles inicialmente associados. Desta maneira, o *CONSTRAINED-K-means* é mais adequado quando as sementes, relacionados aos exemplos rotulados, estão livres de ruídos.

Algoritmo: **Constrained-Kmeans**
Entrada: Conjunto de dados $X = \{x_1, ..., x_n\}$ $x_i \in R^d$, número de cluster k, conjunto $S = \bigcup_{i=1}^{k} S_i$ de sementes iniciais.
Saída: k partições disjuntas $\{X_i\}_{i=1}^{k}$ de X tal que a função objetiva Kmeans é otimizada.
Método:
1. Inicialize: $\mu_h^{(0)} \longleftarrow 1/|S_h| \sum_{x \in Sh} x$, para $h = 1, ..., k$; $t \longleftarrow 0$
2. Repetir até convergir
   2a. Associar cluster: Para cada $x \in S$, se $x \in S_h$, associar x para o cluster h (isto é, conjunto $X_h^{(t+1)}$). Para $x \notin S$, associar x para o cluster h^* (isto é, conjunto $X_{h^*}^{(t+1)}$), para $h^* = \arg_h \min ||x - \mu_h^{(t)}||^2$
   2b. Calcular centroides: $\mu_h^{(t+1)} \longleftarrow 1/|X_h^{(t+1)}| \sum_{x \in Xh}^{(t+1)} x$
   2c. $t \longleftarrow (t+1)$

Algoritmo 3: Constrained-KMeans

3.4.1.4 PCK-means

Este algoritmo também é uma variante do algoritmo não supervisionado *K-means* e foi proposto por (Basu, Banjeree e Mooney, 2004). Este algoritmo utiliza um cenário de *restrições entre pares dos tipos must-link e cannot-link* no conjunto de dados, assim como o *COP-K-means.* A medida de similaridade utilizada nesse algoritmo é composta pela medida de distância convencional entre dois exemplos, adicionada de dois fatores que avaliam o custo de violação das restrições conhecidas. O custo de violação de uma restrição do tipo *must link* é dado por $w*l[l_i \neq l_j]$ (l_i é o *cluster* que um dado exemplo x_i será associado), ou seja, se os exemplos ligados por *must link* forem associados a dois diferentes *clusters*. Similarmente, o custo de violação de uma restrição do tipo *cannot link* é dado por $w*l[l_i = l_j]$, ou seja, se os exemplos ligados por *cannot link* forem associados ao mesmo *cluster*. O indicador l é a dada pela função: *l*[*true*] = 1 e *l*[b*false*] = 0.

Dado o conjunto de pontos *X*, um conjunto de *must-link constraints* $\mathcal{M}$, um conjunto de *cannot-link constraints* $\mathcal{C}$, o peso das restrições *w* e o número de *clusters k,* o algoritmo inicia com a aplicação do fecho transitivo no conjunto de *must-link constraints*, com isso o conjunto $\mathcal{M}$ é aumentado com pela adição das novas *constraints*.

Seja λ o número de componentes conectados no novo conjunto $\mathcal{M}$, este número é utilizado para criar os λ conjuntos de vizinhanças $\{N_p\}^{\lambda}_{p=1.}$ Para cada par do conjunto de vizinhança N_p e $N_{p'}$ possui pelo menos uma restrição *cannot-link* entre eles, é adicionado uma restrição *cannot-link* entre cada par de pontos in N_p e $N_{p'}$ e com isso o conjunto $\mathcal{C}$ é aumentado com a adição destas restrições.

Após este passo, os λ conjuntos de vizinhanças $\{N_p\}^{\lambda}_{p=1}$ são utilizados para iniciar os centróides dos *clusters*.

Se $\lambda \geq k$ é selecionado os *k* maiores conjuntos de vizinhanças e os *k* centros de *clusters* são iniciados com os centróides destes conjuntos.

Se $\lambda < k$ os λ centros de *clusters* são iniciados com os centróides dos λ conjuntos de vizinhanças. Se houver um ponto $\boldsymbol{x}$ que é conectado por um *cannot-link* em todos os conjuntos de vizinhança, este é utilizado para inicializar os $(\lambda\text{-}1)^{th}$ *clusters*. Se houver mais centróides de *cluster* não inicializados, estes são iniciados randomicamente.

O algoritmo *PCKMeans* alterna entre os passos de associação dos elementos aos *clusters* e o passo de calcular os centróides, como mostrado na figura abaixo.

Algoritmo: **PCKMeans**
Entrada: Conjunto de dados $X = \{x_i\}^n_{i=1}$, conjunto de constraint must-link $M = \{(x_i,x_j)\}$, conjunto de constraint cannot-link $C = \{(x_i,x_j)\}$, número de cluster k, peso da constraint w
Saída: k partições disjuntas $\{X_h\}^k_{h=1}$ de X tal que a função objetivo é minimizada.
Método:

1. Inicializar clusters:
   - 1a. Criar λ vizinhanças $\{N_p\}^{\lambda}_{p=1}$ de M e C
   - 2a. Ordernar o índice p de acordo com o tamanho descrescente de N_p
   - 3a. Se $\lambda \geq k$
     Inicializar $\{\mu_h^{(0)}\}^k_{h=1}$ com os centroides de $\{N_p\}^k_{p=1}$
   - 4a. Se $\lambda < k$
     Inicializar $\{\mu_h^{(0)}\}^{\lambda}_{h=1}$ com os centroides de $\{N_p\}^{\lambda}_{p=1}$
     Inicializar o restante dos clusters randomicamente.
2. Repetir até convergir
   - 2a. Associar clusters: Associar cada ponto x para o cluster h^* (isto é, conjunto $X_{h^*}^{(t+1)}$), para $h^* = \arg_h \min (1/2 \; || \; x - \mu_h^{(t)} \; ||^2 + w \sum(x,x_j) \in M \; I[h \neq l_j] + w \sum(x,x_j) \in C \; I[h = l_j]$
   - 2b. Calcular centroides: $\{\mu_h^{(t+1)}\}^k_{h=1} \longleftarrow \{1/|X_h^{(t+1)}| \; \sum_{x \in Xh}{}^{(t+1)} x\}^k_{h=1}$
   - $t \longleftarrow (t+1)$

Algoritmo 4: PCKMeans

3.4.1.5 Método Huang & Pan

Este método foi proposto por (Huang e Pan, 2006) para agrupar dados considerando funções conhecidas dos genes, explorando esse conhecimento pela incorporação das funções conhecidas em uma nova métrica de distância, que reduz a distância baseada na expressão entre dois genes até zero, apenas quando os dois genes compartilham da mesma função. Esse método é baseado no método *k-medoids*, que por sua vez é baseado no *k-means.*

Nessa proposta, é assumido que cada gene pode ser atribuído a pelo menos um e possívelmente mais de um grupo sendo um grupo formado por genes com funções desconhecidas e cada um dos outros formados por genes que tem a mesma função.

O *k-medoids* é similar ao *k-means*, mas é considerado mais robusto. A principal diferença é que em vez de utilizar a média de cada *cluster* como centróide do *cluster* como é feito no *k-means*, *o k-medoids* encontra um elemento de cada *cluster* para ser o centróide. Especificamente, dada uma matriz de distância $D = (d_{ij})$, calculada por uma medida de distância como correlação de *Pearson* ou euclidiana, por exemplo, para um número especifico de *cluster*, dito *k*:

Algoritmo: **K-medoids**

1. Selecionar randomicamente k elementos i_k como medoids, $k = 1, ..., k$.
2. Encontrar os membros de cada cluster para cada i $C(i) = \text{argmin}_{\ k}\ d_{i,i_k*}$
3. Atualizar os medoids: o novo medoid para cada cluster k é o elemento i_k*
   $i_k* = \text{argmin}_{\{i:C(i)=k\}} \sum_{C(j)=k}\ d_{ij}$
4. Repetir os passos 1 e 2 até convergir

Algoritmo 5: K-medoids

No método proposto por *Huang & Pan* é definida uma nova métrica de distância d_{ij}* baseada na métrica de distância convencional que considera a expressão gênica d_{ij} *e as funções dos genes*, como mostrado abaixo. F é o conjunto de funções conhecidas dos genes, cada n gene de um genoma pode ser associado a um ou mais dos F + 1 grupos, G_0, ..., G_F; G_0 contém os genes com funções desconhecidas, enquanto G_1, ..., G_F contém genes pertencentes a uma das funções.

$$d_{ij}* = \begin{cases} rd_{ij} \text{ se há uma função } f \text{ tal que } 1 \leq f \leq F \text{ e } i,j \in G_f \\ d_{ij} \text{ por outro lado} \end{cases}$$

onde $0 \leq r \leq 1$ é um parâmetro de redução a ser determinado. Para r =1, conduz para o método padrão, que ignora as funções do gene no processo de agrupamento.

Há dois passos básicos neste método de agrupamento. No primeiro passo, é aplicado o *k-medoids* para os genes em G_1, ..., G_F usando a nova matriz de distância D* = (d_{ij}), obtendo *clusters* para os genes com funções conhecidas. O número de *cluster*, k_0, é fornecido. No segundo passo, é aplicado o *k-medoids* modificado para a matriz de distância D tal que os genes em G_0 podem ser associados a um dos k_0 *clusters* obtidos anteriormente ou para um dos k_1 novos *clusters*, enquanto os medoids e as atribuições dos genes em G_1, ..., G_F aos *clusters* feitas anteriormente permanecem fixos. Os genes com funções desconhecidas podem assim ser agrupados em *clusters* novos, o que permite a descoberta de estruturas desconhecidas correspondentes a novas categorias de funções. Para simplificar as notações, é assumido que os primeiros n_0 genes estão em G_0 e o restante n-n_0 estão em G_1, ..., G_F, indexados por gene 1, ..., gene n.

Algoritmo: Huang e Pan

1. Aplicar o método *k-medoids* para os genes $\{n_0 + 1, ..., n\}$ (isto é, aqueles com funções conhecidas em $G_{1, ..., G_F}$) usando a medida de distância D^*, com k_0 clusters. Supondo que cada k_0 medoids são $\{i^*_k: k = 1, ...k_0\}$ e cada gene i tem cluster $C(i)$ para $i = n_0 + 1, ..., n$.

2. Aplicar o método *k-medoids* modificado para os genes $\{1, ..., n_0\}$ (isto é, aqueles com funções desconhecidas em G_0) usando a matriz de distância baseada em expressão gênica D, criando k_1 novos *clusters*, fixando o número de *cluster* k_0, obtido no passo 1.

2.0 Selecionar randomicamente k_1 genes de $\{1, ..., n_0\}$ como medoids, ditos i^*_k para $k = k_0 + 1, ..., k_0 + k_1$.

2.1 Encontrar os membros dos clusters de cada gene $i \in \{1, ..., n_0\}$

$$C(i) = \text{argmin}_{1 \le k \le k0+k1} d_{i,i^*k}$$

2.2 Atualizar os k_1 medoids: para $k_0 + 1 \le k \le k_0 + k_1$, o novo medoid para o cluster k é o gene i^*_k com

$$i^*_k = \text{argmin}_{\{i:C(i)=k, 1 \le i \le n0\}} \sum_{C(j)=k,\ 1 \le j \le n0} d_{ij}$$

2.3 Repetir os dois passos acima os dois passos acima até convergir.

Algoritmo 6: Método Huang&Pan

3.4.1.6 Método Boratyn

Este algoritmo foi proposto por (Boratyn, Datta e Datta, 2006), é baseado no agrupamento hierárquico para agrupar genes baseados em dados de expressão gênica. Diferentemente do agrupamento hierárquico, que é um método de agrupamento não supervisionado, este algoritmo é considerado um método de agrupamento semi-supervisionado, pois utiliza dados de genes cujas funções são conhecidas. Em cada fase da formação do *cluster*, o algoritmo utiliza a medida de distância baseada no perfil de expressão gênica mais a informação biológica obtida de bases de dados públicas.

Seja $G = \{x_1, x_2, x_3, ..., x_l\}$ o conjunto de todas as expressões gênicas resultantes de um experimento *microarray*, tal que $x_g \in R^p$, para algum p. Seja $F_1, F_2, ..., F_m$ conjuntos, não necessariamente disjuntos, de rótulos correspondentes a genes com funções similares. O algoritmo proposto utiliza informações funcionais pré-conhecidas e encontra *clusters* de genes funcionalmente similares. A principal diferença deste método está na combinação da medida de distância e da informação funcional dos genes. A distância *D(A,B)* entre dois *clusters A* e *B* é composta de duas partes:

1. Distância matemática $d_M(A,B)$ computada entre duas expressões gênicas
2. Distância biológica $d_B(A,B)$ baseada no conhecimento biológico prévio

$$D(A,B) = (1 - \lambda)\, d_M(A,B) + \lambda\, d_B(A,B)$$

Equação 22: Medida de distância – Método Boratyn

onde $\lambda \in [0.1]$ é um coeficiente especificado pelo usuário, representando a relativa importância dos componentes. Considere dois genes com níveis de expressão x_g e $x_{g'}$, $g \neq g'$ pertencentes a dois *clusters* diferentes. A distância matemática é a distância entre cada par de expressões de genes, que pertencem a *clusters* diferentes, normalizados pelo número de elementos nos *clusters*:

$$d_M(A,B) = 1/n(A)n(B) \sum_{x_g \in A,\, x_{g'} \in B} d(x_g, x_{g'})$$

Equação 23: Distância matemática

onde $d(.,.)$ é uma medida de distância (ou dissimilaridade), e $n(.,.)$ é a cardinalidade do conjunto de dados. Por outro lado, a distância biológica é encontrada contando todos os pares de genes cujas expressões pertencem a diferentes *clusters* e não pertencem ao mesmo conjunto funcional, normalizado pelo número de genes em cada *cluster* que tem função é conhecida. Se, $n(A \cap F)\, n(B \cap F) > 0$, então

$$d_B(A,B) = 1/n(A \cap F)n(B \cap F) \sum_{g \in A \cap F,\, g' \in B \cap F} (1 - I(g, g' \in F_k \text{ para algum } k)).$$

Equação 24: Distância biológica

onde $I(.)$ é um indicador do valor lógico. Assume-se que para $n(A \cap F)\, n(B \cap F) = 0$, $d_B(A,B) = 0$.

Capítulo 4 Expressão Gênica

4.1 Considerações Iniciais

A bioinformática (ou biocomputação) combina conhecimentos de química, física, biologia, engenharia genética e ciência da computação para processar dados biológicos. Diz respeito à utilização de técnicas e ferramentas computacionais para resolução de problemas da Biologia (Baldi e Brunak, 1998).

Dados biológicos estão sendo disponibilizados a uma taxa muito elevada, fazendo com que os bancos de dados atuais cresçam exponencialmente (Baldi e Brunak, 1998).

Uma área que tem se mostrado mais frutífera é a Biologia Molecular, principalmente sua parte relacionada à genética, englobando a resolução de problemas em tópicos que vão desde a comparação de seqüências e montagem de fragmentos até a identificação e análise da expressão de genes e determinação da estrutura de proteínas (Setubal e Meidanis, 1997).

A análise de dados de expressão gênica é de grande interesse para Ciências Biológicas. Esse tipo de análise pode fornecer informações importantes sobre as funções de uma célula, uma vez que as mudanças na fisiologia de um organismo são geralmente acompanhadas por mudanças nos padrões de expressão dos genes (Alberts, Bray, Johnson, Lewis, Raff, Roberts e Walter, 1998).

A expressão gênica é o processo de geração de cópias de mRNA (RNA mensageiro) de um gene (uma parte de um DNA de uma célula) e então, o mRNA é transformado em uma seqüência de aminoácido que forma a base de uma proteína. Cada tipo de tecido em um organismo requer diferentes quantidades de diferentes proteínas para desempenhar seu papel. A quantidade de uma proteína específica produzida por uma célula é parcialmente controlada por um número de cópias de mRNA correspondentes. Os relativos níveis de mRNA de cada gene em um tipo de tecido são chamados de perfil de expressão do gene do tecido. Químicos assumem que a grande maioria dos genes no genoma humano só são expressados em um tipo de tecido, e apenas alguns "*housekeeping genes*" são expressados em todas as células, tais como os que controlam a transcrição e tradução (Ng, Sander e Sleumer, 2001).

Este trabalho contém uma descrição de vários aspectos relacionados à análise de expressão gênica. Na seção 4.2 são apresentados os principais conceitos de Biologia Molecular. Na seção 4.3 é descrito o processo de expressão gênica, de forma bastante simplificada. A seção 4.4 contém uma descrição resumida de algumas das técnicas existentes atualmente para a medição da expressão gênica. Essas técnicas também são brevemente comparadas.

4.2 Conceitos de Biologia Molecular

Um problema de fundamental importância na biologia molecular é compreender a estrutura e função das proteínas encontradas em toda a natureza. Atualmente, o crescimento dos dados de seqüência de proteína é muito superior à capacidade de biólogos para caracterizar as proteínas nestas bases de dados (Craven, Mural, Hauser e Uberbacher, 1995).

Nesta seção serão apresentados os principais conceitos de Biologia Molecular. O objetivo é apresentar as definições necessárias para o entendimento das aplicações das técnicas computacionais nesta área da Biologia.

4.2.1 DNA e RNA

As informações genéticas são armazenadas nos ácidos nucléicos – ácido desoxirribonucléico (DNA) e o ácido ribonucléico (RNA).

O DNA carrega informações genéticas em sua longa cadeia de nucleotídeos. Uma cadeia de DNA é um polímero longo, não ramificado, composto de somente quatro tipos de subunidades. Estas são os desoxirribonucleotídeos contendo as bases adenina (A), citosina (C), guanina (G) e timina (T). A molécula de DNA é um polimeral helicoidal composto de duas fitas. A hélice normalmente se formará se cada uma das subunidades vizinhas de um polímero for regularmente orientada. Uma característica essencial do modelo era que todas as bases da molécula de DNA encontram-se no lado de dentro da hélice-dupla, com o açúcar-fosfato no lado de fora. Isto exige que as bases de uma fita estejam extremamente próximas das bases da outra, e o encaixe proposto requer um pareamento específico entre uma grande base purina (A ou G, cada uma com um anel duplo) em uma cadeia, e uma pequena base pirimidina (T ou C, cada uma com um único anel) na outra cadeia. Evidências de experimentos bioquímicos a partir do modelo construído sugeriram que pares de bases

complementares se formam entre A e T e entre G e C (Alberts, Bray, Lewis, Raff, Roberts e Watson, 1997).

A seqüência de bases nitrogenadas em uma molécula de DNA é extremamente variável, mas verificou-se que a quantidade de adenina é igual à quantidade de timina, e que a quantidade de citosina é igual à quantidade de guanina. Isto significa que o pareamento se dá da seguinte maneira: a base A parea-se com a base T e a base G parea-se com a base C (Valafar, 2002).

A Figura 5 ilustra os elementos, a estrutura dupla-hélice e a representação de uma molécula de DNA (Valafar, 2002):

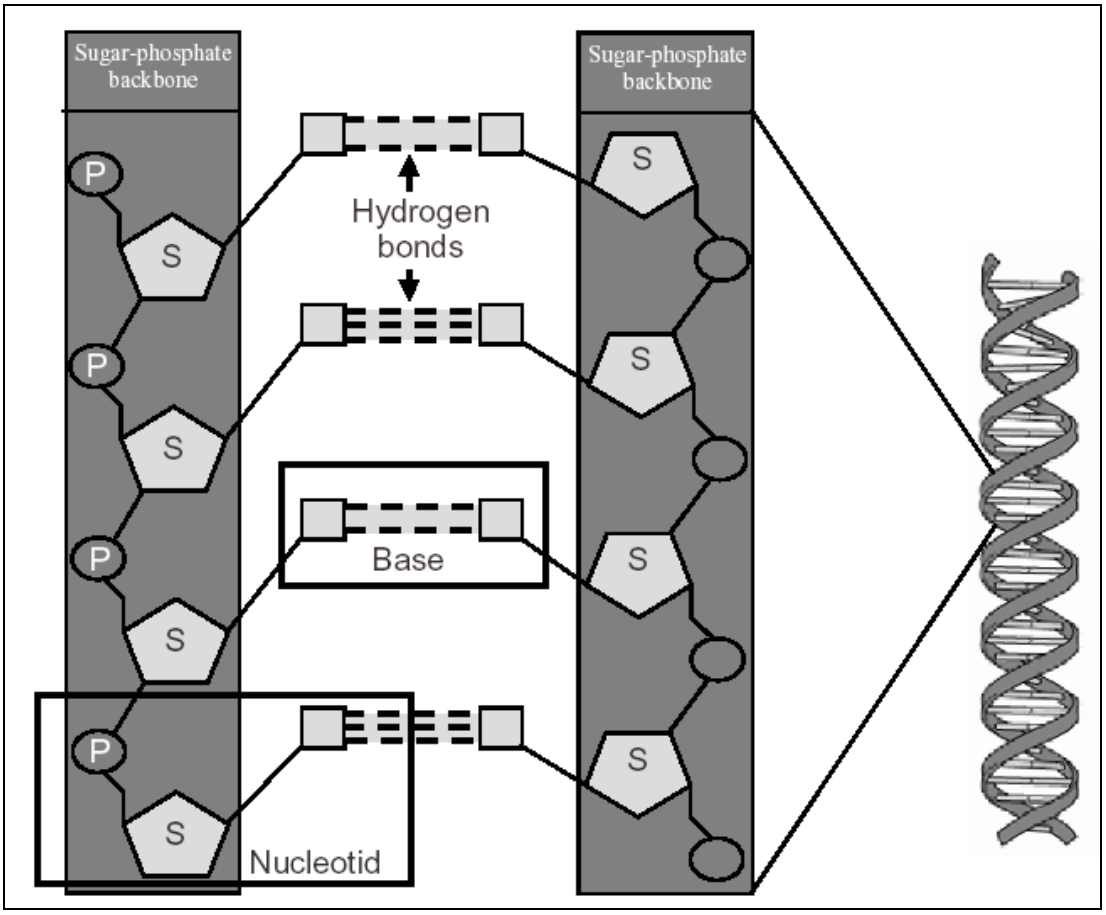


Figura 5: Molécula de DNA, estrutura dupla-hélice

Além da sua capacidade de duplicação, o DNA também é responsável pela síntese de um outro ácido nucléico, o RNA. Assim como o DNA, o RNA também é formado por vários nucleotídeos e possuem os mesmos constituintes: uma molécula de ácido fosfórico, uma molécula de açúcar e uma base nitrogenada. O açúcar do RNA também é uma pentose, mas não é a desoxirribose e sim a ribose. As bases nitrogenadas do RNA são as mesmas do DNA,

com exceção da timina (T). Em vez da timina, possui outra base chamada uracila (U) (Valafar, 2002).

O RNA forma-se no núcleo e migra para o citoplasma. A quantidade de RNA é variável de célula para célula e com atividade celular.

4.2.2 Genes, DNA genômico, cDNA, cromossomos e genoma

A genética cresceu em torno de informações invisíveis, contendo elementos, chamados **genes**, que são distribuídos para cada célula-filha, quando a célula se divide. Portanto, antes de se dividir, a célula precisa fazer uma cópia de seus genes para dividi-los igualmente para suas células-filhas. Os genes nas células-ovo e esperma carregam informações hereditárias de uma geração para a outra.

No final do século 19, biólogos tinham descoberto que os carregadores de informações hereditárias eram os **cromossomos,** que se tornavam visíveis no núcleo quando a célula começava a se dividir. Mas a evidência do DNA, nesses cromossomos, é a substância da qual os genes são formados, apareceu muito tempo depois, a partir dos estudos com bactérias.

Os vários tipos especializados de células, em uma planta ou em um animal, parecem diferentes uns dos outros. Tal fato parece um paradoxo, pois as células, num organismo multicelular, são muito relacionadas e descendem de uma única célula precursora – ovo fertilizado. Linhagens comuns denotam genes similares; como aparecem então as diferenças? Em alguns poucos casos, a especialização celular envolve a perda de material genético. A grande maioria das células de animais e plantas, no entanto, retém toda a informação genética contida num ovo fertilizado. Especialização depende da mudança de *expressão gênica* e não da perda ou aquisição de genes.

Mesmo as bactérias não fabricam todos os seus tipos de proteínas todo o tempo, mas o seu nível de síntese é ajustado de acordo com as condições externas.

As células eucarióticas desenvolveram mecanismos mais sofisticados de controle da expressão gênica, e isto afeta sistemas inteiros de produtos gênicos interativos. Grupos de genes são ativados ou reprimidos em resposta a sinais externos e internos. Composição de membrana, citoesqueleto, produtos de secreção e mesmo metabolismo – todas essas e outras características – devem mudar de maneira coordenada à medida que as células se diferenciam. A diferença radical de caráter entre tipos celulares reflete a mudança estável de expressão gênica.

4.3 Processo de Expressão Gênica

Células produzem as proteínas que necessitam para funcionar corretamente através da transcrição dos correspondentes genes de DNA em RNA mensageiro (mRNA) transciptos e tradução das moléculas de mRNA em proteínas.

Microarrays obtém a imagem da atividade de uma célula através de uma medição do número de cópias de cada tipo de molécula de mRNA (o que também dá uma imagem imperfeita e indireta da atividade da proteína). A chave para esta medição é a propriedade de hibridação de dupla hélice de DNA (e RNA). Quando uma única vertente de DNA é trazida em contato com uma seqüência DNA complementar, recombinará esta seqüência complementar para formar *double-stranded* DNA. Para as quatro bases de DNA, Ademina é complementar a Citosina e Guanina é complementar a Timina, e a seqüência complementar é produzida pela complementaridade das bases de referência iniciando do fim da seqüência e procedendo o restante. Hibridação irá, portanto, permitir uma investigação do DNA para reconhecer uma cópia da sua seqüência complementar obtida a partir de uma amostra biológica.

Um *array* consiste de um padrão reproduzível de diferentes investigações de DNA anexado a um apoio sólido. Após a extração do RNA de uma amostra biológica, DNA complementar (cDNA) ou cRNA etiquetado fluorescentemente é preparado. Esta amostra fluorescente é então hibridizada ao presente DNA no *array*. Graças à fluorescência, a intensidade da hibridação (que está relacionado com o número de cópias de cada espécie de RNA presente na amostra) pode ser medida através de um laser scanner e convertida para uma leitura quantitativa. Desta forma, *microarrays* permitem simultânea medição dos níveis de expressão de milhares de genes em um único ensaio de hibridação.

Estes fragmentos de DNA são normalmente centenas de pares de bases e são frequentemente derivados de coleções de referência de tags de seqüência de expressão (que são subseqüências de um mRNA transcrito que exclusivamente identifica esta transcrição) extraídas de várias fontes de material biológico, de modo a representar o maior número possível de genes. Normalmente cada *spot* representa um único gene. Amostras de cDNA são utilizadas duas amostras independentes: uma de referência e uma de amostra de teste. Um par de amostras de cDNA é independentemente copiado da população correspondente mRNA com a enzima transcriptase reversa e etiquetados usando distintas moléculas fluorescentes (verde e vermelho). Estas amostras de cDNA etiquetadas são, então, agrupadas e hibridizadas

no *array*. Quantidades relativas de um determinado gene transcrito nas duas amostras são determinadas pela medição da intensidade do sinal detectado tanto do comprimento da fluorescência como do cálculo das proporções (aqui, apenas os níveis de expressão relativos são obtidos). Um *microarray* de cDNA é, portanto, que intrinsicamente normaliza parte do ruído experimental. Uma visão geral do processo que pode ser seguido com *microarrays* de cDNA é dado na Figura 6 (Moreau, Smet, Thijs, Marchal e Moor, 2002).

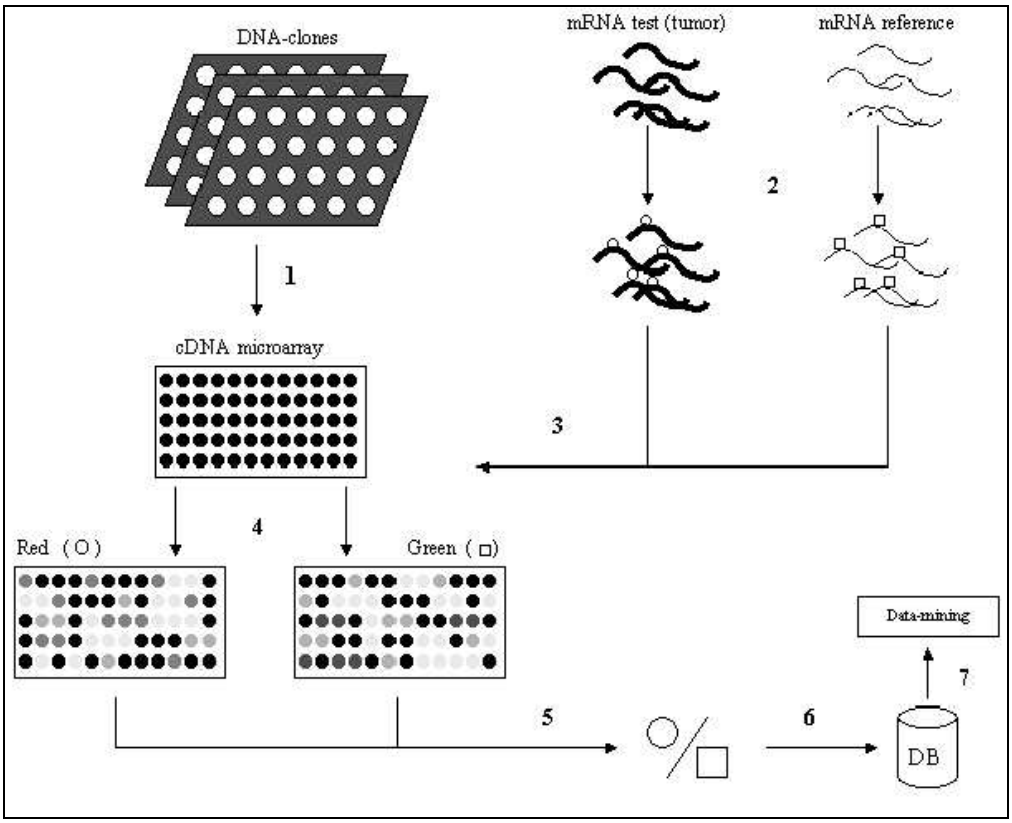


Figura 6: Esquema de um experiment com *microarray* de cDNA

(1) Disponibilizar as sondas de DNA sobre a lâmina de vidro. (2) Etiquetagem (via transcriptase reversa) do total de mRNA do teste de amostra(vermelho) e amostra de referência (verde). (3) Seção das duas amostras e hibridização. (4) Leitura das intensidades vermelhas e verdes separadamente em casa sonda. (5) Cálculo do nível de expressão (intensidade vermelho/ intensidade verde). (6) Armazenamento dos resultados no banco de dados. (7) Data mining.

O processo pelo qual as seqüências de nucleotídeos dos genes são interpretadas na produção de proteínas é denominado **expressão gênica**. A expressão é composta por duas etapas: na primeira, denominada transcrição e a segunda, denominada tradução.

O processo de **transcrição** é gerador do mRNA que carrega a informação para a síntese de proteínas, das moléculas de RNA transportador, ribossomal e de outras moléculas de RNA, que têm funções estruturais ou catalíticas. Todas essas moléculas de RNA são sintetizadas por enzimas, denominadas RNA **polimerase**, que fazem uma cópia de RNA a partir de uma seqüência de DNA.

O RNA retém toda informação da seqüência de DNA da qual foi copiado, assim como as propriedades de pareamento de bases do DNA. Na transcrição de DNA uma das duas fitas de DNA serve como molde, no qual as capacidades de pareamento de novos nucleotídeos são testadas. Quando um bom encaixe com o molde de DNA é obtido, um ribonucleotídeo é incorporado como uma unidade covalentemente é ligada. Desta maneira, a cadeia de RNA em crescimento é alongada a um nucleotídeo por vez.

O RNA resultante não permanece como uma fita ligada ao DNA. Logo após a região, onde os ribonucletídeos estão sendo adicionados, a hélice original de DNA está sendo refeita e liberando a cadeia de RNA. Portanto, moléculas de RNA são fitas simples. Além disso, moléculas de RNA são relativamente curtas se comparadas a moléculas de DNA, uma vez que elas são copiadas de uma região limitada de DNA – o suficiente para fazer uma ou algumas poucas proteínas. Transcritos de RNA, que direcionam a síntese de moléculas de proteínas, são chamados de RNA mensageiros (mRNA), enquanto outros transcritos de RNA servem de RNA transportador (tRNA), ou formam componentes de ribossomos (rRNA), ou ainda partículas menores de ribonucleoproteínas (Alberts, Bray, Lewis, Raff, Roberts e Watson, 1997).

A quantidade de RNA, produzido de uma determinada região de DNA, é controlada por proteínas reguladoras de gene, que se ligam a sítios específicos do DNA perto da seqüência codificadora de um gene. Em qualquer célula, num dado tempo, algum gene está sendo utilizado para produzir RNA em grandes quantidades, enquanto outros genes não estão sendo transcritos. Para um gene ativo, milhares de transcritos de RNA podem ser produzidos do mesmo segmento de DNA em cada geração celular. Como cada molécula de mRNA pode ser traduzida em milhares de cópias de uma cadeia polipeptídica, a informação contida em uma pequena região de DNA pode direcionar a síntese de milhões de cópias de uma proteína específica.

As regras pelas quais a seqüência de nucleotídeos de um gene é traduzido na seqüência de aminoácidos de uma proteína, o assim chamado código genético, foram decifradas no início dos anos 60. Foi descoberto que a seqüência de nucleotídeos na molécula de mRNA, que reage como um intermediário, é lida em série e em grupo de três. Cada trio de nucleotídeos, chamado códon, especifica um aminoácido. Uma vez que RNA é um polímero linear de quatro nucleotídeos diferentes, existem 64 códons possíveis. Todavia, somente 20 diferentes aminoácidos são, em geral, encontrados em proteínas, de forma que a maioria dos aminoácidos é especificada por vários códons; isto é, o código genético é degenerado.

Os códons, em uma molécula de mRNA, não reconhecem diretamente os aminoácidos que eles codificam, como uma enzima reconhece um substrato. A **tradução** do mRNA em proteína depende de moléculas "adaptadoras" que reconhecem tanto o aminoácido como um grupo de três nucleotídeos. Esses adaptadores consistem num grupo de pequenas moléculas de RNA conhecidas como RNAs transportadores (tRNAs), cada um com cerca de 80 nucleotídeos de extensão (Alberts, Bray, Lewis, Raff, Roberts e Watson, 1997).

A função primária do genoma é especificar moléculas de RNA. Porções selecionadas da seqüência nucleotídica do DNA são copiadas numa seqüência nucleotídica de RNA correspondente, a qual codifica uma proteína (se é um mRNA) ou forma um RNA "estrutural", como uma molécula de RNA de transferência (tRNA) ou RNA ribossomal (rRNA). Cada região dupla-hélice de DNA que produz uma molécula de RNA funcional constitui um **gene**.

Em eucariontes superiores, são comuns genes que possuem mais de 100.000 pares de nucleotídeos de extensão. Alguns genes chegam a conter mais de 2 milhões de pares de nucleotídeos, embora apenas em torno de 1.000 pares de nucleotídeos sejam necessários para codificar uma proteína de tamanho médio (contendo de 300 a 400 aminoácidos). A maior parte de extensão adicional consiste de longos segmentos de DNA não-codificante, que interrompe os segmentos relativamente curtos de DNA codificante. As seqüências codificantes são chamadas ***exons,*** as seqüências intervenientes (não-codificantes) são chamadas ***introns***. A molécula de RNA sintetizada a partir de um gene organizado dessa maneira (chamada de transcrito RNA), durante a sua conversão a uma molécula de mRNA, é alterada para a remoção das seqüências dos *introns*, no processo de *splicing* do RNA.

Grandes genes consistem de uma longa cadeia de exons e introns alternados, sendo que a maior parte dos mesmos corresponde a seqüências de introns. Adicionalmente, cada gene está associado a seqüências de DNA regulatórias, que são responsáveis por garantir que o gene seja transcrito no momento adequado e no tipo de célula apropriado. Muitas das seqüências regulatórias estão localizadas *upstream* em relação ao sítio onde a transcrição do RNA inicia, mas elas também podem estar localizadas *downstream* em relação ao sítio onde a transcrição termina, ou mesmo em introns e exons.

Embora o produto final da expressão de um gene seja a proteína, as técnicas baratas e vastamente utilizadas, analisam a expressão de um gene a partir da quantidade de mRNA correspondente a proteína presente na célula. Embora a relação entre a quantidade de mRNA

e a respectiva proteína não seja direta, a abundância de mRNA nas células constitui uma ferramenta importante para os estudos de expressão gênica. O nível de expressão de um gene indica aproximadamente o número de mRNA produzidos em uma célula, que corresponde a quantidade da proteína correspondente (Slonim, Tamayo, Mesirov, Golub e Lander, 2000).

A análise de expressão de um gene é de grande importância para ciências biológicas. Esta análise pode fornecer informações importantes sobre as funções das células, uma vez que as mudanças na fisiologia do organismo são acompanhadas por mudanças no padrão da expressão do gene. Assim, é possível deduzir as características funcionais e taxicológicas de um composto baseado no efeito que este composto terá na expressão gênica (Chan, Hontzeas e Park, 2000).

Algumas das aplicações da análise da expressão gênica:

- Descoberta de genes.
- Determinação de quais genes estão sendo expressos em um determinado tipo de célula em um dado momento e sob certas condições.
- Comparação da expressão dos genes em dois tipos de células diferentes ou duas amostras de tecido diferentes (por exemplo, tecido saudável x tecido doente).
- Observação das mudanças na expressão dos genes em diferentes estágios no ciclo da célula ou durante o desenvolvimento do embrião.
- Estudo dos mecanismos de regulação dos genes.
- Identificação de doenças genéticas complexas.
- Descoberta de medicamentos e estudos de toxicologia.
- Detecção de mutação/polimorfismo nos genes.
- Análise de elemento patogênico.

4.4 Técnicas para Medir a Expressão Gênica

Atualmente, existem diversas técnicas que permitem identificar o nível de expressão de um grande número de genes simultaneamente, gerando a chamada análise de expressão em larga escala. Dentre essas técnicas podem ser citadas SAGE (*Serial Analysis of Gene*

Expression) (Velculescu, Zhang, Vogelstein e Kinzler, 1995), MPSS (*Massively Parallel Signature Sequencing Technology*) (Brenner, Johnson, Bridgham, Golda, Lloyd, Johnson, Luo, Mccurdy, Foy e Ewan, 2000), vários tipos de *microarray* de DNA (*microarrays* de oligonucleotídeos curtos, *microarrays* de cDNA). (Schena, Shalon, Davis e Brown, 1995), *microarrays* de oligonucleotídeos longos, *microarrays* de fibra ótica) e *Real-time RT-PCR* (*Reverse Transcription – Polymerase Chain Reaction*). (Stanton, 2001) faz uma breve revisão de algumas dessas técnicas. Algumas delas são brevemente discutidas a seguir. As pesquisas dos autores estão relacionadas a dados obtidos com as técnicas de *microarray* e SAGE. Assim, algumas informações extras sobre os dados obtidos com essas técnicas são inseridas nas seções seguintes.

4.4.1 *Microarray* de DNA

O princípio básico por trás da tecnologia de *microarray* de DNA (Harrington, Rosenow e Retief, 2000; Murphy, 2002) é a hibridização. Um *microarray* consiste de um conjunto de seqüências de DNA conhecidas (sondas) imobilizadas de forma organizada (com uma configuração conhecida) em uma superfície sólida, em geral, de nylon ou vidro. Conjuntos específicos da superfície, chamados *spots*. Assim, cada *spot* representa um gene conhecido que é identificado pela sua localização no *microarray*.

A escolha das seqüências de DNA a serem colocadas no *microarray* determina quais genes podem ser detectados. Para organismos com genoma completamente seqüenciados pode-se colocar no *microarray* todos os seus genes conhecidos ou ORF (*Open Reading Frames*) suspeitos.

A utilização do *microarray* se dá da seguinte maneira: inicialmente, o mRNA da amostra de interesse é extraído. Em seguida, cada molécula de mRNA é transformada em uma molécula de DNA complementar (cDNA) marcada de forma radioativa ou fluorescente. Um cDNA é uma molécula de DNA construída em laboratório por meio de transcrição reversa a partir de uma molécula de mRNA. Essa molécula é utilizada no lugar do mRNA por ser mais estável. O cDNA marcado é chamado de alvo. A amostra com os alvos é então hibridizada no *microarray*. Cada alvo se liga a uma sonda complementar no spot correspondente a um gene específico. A abundância de cada mRNA na amostra é capturada digitalmente, de acordo com a quantidade de alvo que hibridizou em cada *spot* do *microarray*. A partir do *microarray* com os alvos hibridizados é gerada uma imagem, que é então analisada por meio de um dos vários

softwares disponíveis, para quantificar as informações dessa imagem. Os níveis de expressão são derivados de sinais analógicos de absorvência ou fluorescência.

Os dois tipos principais de *microarray* são os de cDNA (Schena, Shalon, Davis e Brown, 1995) e os de oligonucleotídeos. Cada um deles oferece diferentes possibilidades de análise e também diferentes vantagens e desvantagens. Claverie (Claverie, 1999), Stanton (Stanton, 2001), Murphy (Murphy, 2002) e Nguyen, Arpat, Wang, & Carroll (Nguyen, Arpat, Wang e Carroll, 2002) comparam alguns aspectos desses tipos de *microarray*.

O *microarray* de cDNA é construído pela impressão de material genético pré-sintetizado na superfície sólida. Neste caso, o DNA sonda vem de organismos naturais e podem ser feitos de DNA genômico ou de cDNA.

Os *microarrays* de cDNA construídos em superfícies de vidro permitem que sejam realizados experimentos competitivos, em que alvos de duas amostras marcadas de forma diferente, com corantes fluorescentes, sejam hibridizados simultaneamente em um mesmo *microarray*. Nesse caso, os níveis de expressão são dados como a razão entre os níveis de expressão nas duas amostras (Claverie, 1999). Yang & Speed (Yang, Dudoit, Luu e Speed, 2001) discutem os vários aspectos do projeto de experimentos com *microarrays* de cDNA. Algumas das vantagens e desvantagens do *microarray* de cDNA são:

Vantagens:

- Muito utilizado para quantificação relativa dos níveis de expressão, com a utilização de experimentos competitivos.
- As sondas representam genes de identidade conhecida ou segmentos de DNA funcional, conhecidos como ESTs (*Expressed Sequence Tags*). Pode também ser aplicada para a descoberta de genes, uma vez que não há a necessidade de se conhecer a seqüência dos cDNAs sonda.
- Mede a expressão de dezenas de milhares de genes simultaneamente.
- Capacidade de detectar um grande intervalo de níveis de expressão.
- Embpra tenha alto custo inicial, tanto em termos financeiros, quanto de trabalho, depois de instalada é bastante produtiva.

Desvantagens:

- Necessita de grande quantidade de RNA total para preparar o alvo (50 – 200 μg de RNA total).
- Não é capaz de detectar genes com baixo nível de expressão.
- Apresenta risco de fragmentos do alvo hibridizarem com sondas complementares similares, produzindo detecções falsas.

O *microarray* de oligonucleotídeos é construído pela síntese das seqüências de interesse (oligonucleotídeos) *in situ*, no próprio substrato, sendo cada ponto sintetizado, nucleotídeo por nucleotídeo, utilizando uma máscara de luz. Algumas das vantagens e desvantagens do *microarray* de oligonucleotídeos são:

Vantagens:

- Quantifica níveis absolutos de expressão, permitindo comparações confiáveis dos valores obtidos em experimentos separados.
- Mede a expressão de dezenas de milhares de genes simultaneamente.
- Exige uma quantidade menor de RNA total do que o *microarray* de cDNA (5 μg de RNA total).
- Tem a capacidade de detectar um intervalo de níveis de expressão maior do que o *microarray* de cDNA.
- Apresenta menor risco de fragmentos do alvo hibridizarem com sondas complementares similares do que o *microarray* de cDNA.

Desvantagens:

- É uma tecnologia proprietária, desenvolvida pela *Affymetrix*.
- É menos flexível que o *microarray* de cDNA, pois o usuário não pode selecionar as sondas para compor o *microarray*. Os *microarrays* são comprados prontos. Existe a possibilidade de se especificar os oligonucleotídeos (sondas) desejados, mas o custo é muito alto.

- As sondas são projetadas especialmente para cada gene de interesse, sendo necessário o conhecimento da seqüência dos genes.
- Não possibilita a descoberta de genes novos.
- Tem alto custo

Capítulo 5 Agrupamento não-supervisionado e semi-supervisionado para dados de expressão gênica

5.1 Considerações Iniciais

A tecnologia de *microarray* de DNA é um método poderoso para monitorar níveis de expressão de milhares de genes, ou mesmo genomas inteiros, em um simples experimento (Carmona-Saez, Pascual-Marqui, Tirado, Carazo e Pascual-Montano, 2006). Uma experiência *microarray* avalia tipicamente um número grande de seqüências do DNA (os genes, cDNA clones, ou seqüência tag expressadas [ESTs]) sob circunstâncias múltiplas. Estas circunstâncias podem ser conjuntos de tempo durante um processo biológico (por exemplo, o ciclo da célula do fermento) ou uma coleção de diferentes amostras de tecidos (por exemplo, normal contra tecidos cancerígenos). Essa realidade fez surgir a necessidade de novos métodos para análises desses conjuntos de dados de larga escala. O agrupamento de dados tem sido extensivamente utilizado com essa finalidade. Algoritmos tradicionais foram adaptados ou diretamente aplicados a conjuntos de dados de expressão gênica, e muitos algoritmos novos têm sido propostos especificamente para esse domínio do conhecimento. Os resultados obtidos nos últimos anos provam que técnicas de agrupamento podem ser úteis na descoberta de grupos biologicamente relevantes de genes ou amostras (Jiang, Tang e Zhang, 2004; Priness, Maimon e Ben-Gal, 2007). Neste capítulo são apresentadas noções relativas ao formato e pré-processamento de dados de expressão gênica, os tipos de métodos de agrupamento e uma descrição de abordagens representativas de agrupamento não supervisionado e semi-supervisionado.

5.2 Dados de expressão gênica

Um conjunto de dados resultante de um experimento de microarray pode ser representado por uma matriz de expressão de valores reais nas quais as linhas representam os padrões de expressão de genes e as colunas representam as amostras. Um elemento w_{ij} da matriz indica o nível de expressão do gene *i* na amostra *j*. A matriz original de expressão dos

genes obtida do processo da exploração contém ruído, valores faltantes, e variações sistemáticas do procedimento experimental. Por isso, o pré-processamento dos dados é uma etapa importante antes de qualquer análise de agrupamento a ser executada. Muitos métodos de agrupamento aplicam um ou mais procedimentos de pré-processamento como filtrar genes com níveis de expressão que não mudam significativamente ao longo das amostras, executam uma transformação logarítmica de cada nível de expressão, ou padronizar cada linha da matriz de expressão dos genes com media zero e variância um. Os procedimentos para pré-processamento dos dados não fazem parte do escopo deste trabalho e não serão tratados com maiores detalhes. Os experimentos foram conduzidos assumindo-se que os dados foram previamente processados e encontravam-se no formato apropriado para aplicação dos métodos. Um exemplo de pré-processamento aplicado a conjuntos dados como os considerados neste trabalho, que tem sido frequentemente citado como referência em trabalhos posteriores, pode ser encontrado em (Abba, Drake, Hawkins, Hu, Sun, Notcovich, Gaddis, Sahin, Baggerly e Aldazcorresponding, 2004).

As técnicas de agrupamento provaram serem úteis na compreensão da função do gene, do regulamento do gene, dos processos celulares, e subtipos de células. Os genes com padrões similares da expressão (genes coexpressados) podem ser agrupados junto com funções celulares similares. Este método pode promover a compreensão das funções de muitos genes para quais a informação não é previamente disponível (Tavazoie, Hughes, Campbell, Cho e Church, 1999) (Eisen, Spellman, Brown e Botstein, 1998) . Além disso, genes coexpressados no mesmo grupo estão provavelmente envolvidos nos mesmos processos celulares, e uma forte correlação de padrões da expressão entre estes genes indica a correlação.

Procurar por seqüências comuns do DNA nas regiões do promoter dos genes dentro do mesmo grupo permite que os motifs regulatórios específicos de cada conjunto do gene sejam identificados e aos elementos cis-regulatórios sejam propostos (Brazma e Vilo, 2000) (Tavazoie, Hughes, Campbell, Cho e Church, 1999). A inferência do regulamento com agrupamento de dados da expressão do gene causa também hipóteses a respeito do mecanismo da rede regulatória do transcriptional (D'haeseleer, Wen, Fuhrman e Somogyi, 1998). Finalmente, as amostras diferentes de agrupamento com base nos perfis correspondentes da expressão podem revelar tipos de subcélulas que são difíceis de identificar pelos métodos tradicionais baseados em morfologia (Alizadeh, Eisen, Davis, Chi Ma, Rosenwald, Boldrick, Sabet, Tran, Yu, Powell, Yang, Marti, Moore, Hudson, Lu, Lewis, Tibshirani, Sherlock, Chan, Greiner, D.Weisenburger, Armitage, Warnke, Levy, Wilson,

Grever, Byrd, Botstein, Brown e Staudt, 2000) (Golub, Slonim, Tamayo, Huard, Gaasenbeek, Mesirov, Coller, Loh, Downing e Caligiuri, 1999).

5.3 Tipos de métodos de agrupamento de expressão gênica

Uma experiência de *microarray* típica pode conter até 10^6genes, enquanto que o número das amostras envolvidas é geralmente menor que 100. Uma das características de dados de expressão dos genes é que é significativo agrupar tanto genes quanto amostras. Por um lado, genes coexpressados podem ser agrupados em conjuntos baseados em seus padrões da expressão (Ben-Dor, Friedman e Yakhini, 2001) (Eisen, Spellman, Brown e Botstein, 1998). Em tais agrupamentos baseados em gene, os genes são tratados como os objetos, enquanto as amostras são as características. Por outro lado, as amostras podem ser divididas em grupos homogêneos. Cada grupo pode corresponder a algum fenótipo macroscópico particular, tal como síndromes clínicas ou tipos de câncer (Golub, Slonim, Tamayo, Huard, Gaasenbeek, Mesirov, Coller, Loh, Downing e Caligiuri, 1999). Os agrupamentos baseados em amostra consideram as amostras como os objetos e os genes como os atributos. Alguns algoritmos de agrupamento, como *K-means* e métodos hierárquicos podem ser utilizados tanto para agrupar genes quanto amostras.

A terceira categoria de análise de agrupamento aplicada a dados de expressão dos genes, que são agrupamento de subespaço, tratam genes e amostras simetricamente tal que tanto os genes quanto as amostras podem ser considerados como objetos ou atributos.

Agrupamento baseado em gene, baseado em amostras e de subespaço apresentam diferentes desafios e diferentes estratégias computacionais são adotadas para cada situação (Jiang, Tang e Zhang, 2004).

Nesta seção, introduziremos resumidamente os três tipos de agrupamento e as principais propostas encontradas na literatura em cada um desses tipos para a análise de dados de expressão gênica. O trabalho desenvolvido aqui abordou o agrupamento baseado em gene, pela abordagem não supervisionada e semi-supervisionada.

5.3.1 Agrupamento baseado em gene

A finalidade do agrupamento baseado em gene é agrupar genes coexpressados que indicam cofunção e corregulação.

Devido às características especiais de dados de expressão dos genes, e às exigências particulares do domínio biológico, agrupamento baseado em gene apresenta diversos desafios novos e é ainda um problema aberto.

Primeiramente, a análise do conjunto é tipicamente a primeira etapa na mineração dos dados e na descoberta do conhecimento. A finalidade do agrupamento dos dados de expressão dos genes é revelar as estruturas naturais dos dados e ganham algumas introspecções iniciais a respeito da distribuição dos dados. Conseqüentemente, um bom algoritmo de agrupamento deve depender tão pouco quanto possível do conhecimento prévio, que geralmente não está disponível antes da análise do conjunto. Por exemplo, um algoritmo de agrupamento que pode exatamente estimar o número "verdadeiro" de grupos no conjunto de dados seria mais favorecido do que um que requer a pré-determinação do número dos conjuntos.

Em segundo, devido aos procedimentos complexos das experiências *microarray*, dados da expressão de genes freqüentemente contêm uma quantidade enorme de ruído. Conseqüentemente, os algoritmos de agrupamento de dados da expressão de genes devem ser capazes de extrair a informação útil de um elevado nível do ruído.

Em terceiro lugar, estudos empíricos demonstraram que os dados da expressão de genes freqüentemente "são altamente conectados" (Jiang, Pei e Zhang, 2003) e os conjuntos podem ser altamente interseccionados com os outros ou encaixado um com o outro (Jiang, Pei e Zhang, 2003). Conseqüentemente, os algoritmos de agrupamento baseado em gene devem segurar eficazmente esta situação.

Finalmente, os usuários de dados *microarray* podem não somente estar interessado nos conjuntos dos genes, mas estão também interessados no relacionamento entre os conjuntos (por exemplo, quais os conjuntos são mais próximos um dos outros e quais conjuntos são remotos dos outros), e relacionamento entre os genes dentro do mesmo conjunto (por exemplo, quais genes podem ser considerados como representantes do conjunto e quais genes estão na área do limite do conjunto). Algoritmo de agrupamento pode não somente dividir o conjunto de dados, mas também fornece alguma representação gráfica da estrutura do conjunto sendo mais favorecida pelos biólogos.

5.3.2 Agrupamento baseado em amostra

Em uma matriz da expressão do gene, há geralmente diversos fenótipos macroscópicos particulares das amostras relacionadas a algumas doenças ou efeitos de droga, tais como amostras doentes, amostras normais, ou amostras tratadas com droga.

O objetivo do agrupamento baseado em amostras é encontrar as estruturas do fenótipo ou subestruturas das amostras.

Estudos anteriores (Golub, Slonim, Tamayo, Huard, Gaasenbeek, Mesirov, Coller, Loh, Downing e Caligiuri, 1999) demonstraram que os fenótipos das amostras podem ser discriminados através somente de um subconjunto pequeno dos genes cujos níveis da expressão correlacionam fortemente com a distinção da classe. Estes genes são chamados genes informativos. O restante dos genes da matriz da expressão do gene é irrelevante à divisão das amostras de interesse e são considerados como ruído do conjunto de dados.

Embora os métodos de agrupamento convencionais, como *K-means*, mapas *self-organizing* (*SOM*), agrupamento hierárquico (HC), podem ser diretamente aplicado em amostras de grupos usando todos os genes como características, a relação *signal-to-noise* (isto é, o número de genes informativos contra os genes irrelevantes) são geralmente menores de 1: 10, que podem seriamente degradar a qualidade e a confiabilidade dos resultados do agrupamento (Xing e Karp, 2001) (Tang, Zhang e Pei, 2003).

Assim, os métodos particulares devem ser aplicados para identificar genes informativos e para reduzir a dimensionalidade do gene para amostras de agrupamento para detectar seus fenótipos.

Os métodos existentes para selecionar genes informativos para amostras do grupo caem nas duas categorias principais: a análise supervisionada (agrupamento baseado na seleção supervisionada informativa do gene) e análise não supervisionada (agrupamento não supervisionado e a seleção informativa do gene).

5.3.3 Agrupamento baseado em subespaço

Os algoritmos de agrupamento discutidos nas seções precedentes são exemplos de agrupamento global, ou seja, para que um conjunto de dados seja agrupado, o espaço de atributos é determinado globalmente e é compartilhado por todos os grupos resultantes, e os grupos resultantes são exclusivos e exaustivos.

Entretanto, é sabido na biologia molecular que somente um pequeno subconjunto de genes participa de um processo celular de interesse e que qualquer processo celular ocorre somente em um subconjunto de amostras.

Além disso, um único gene pode participar de múltiplos *pathways* que podem ou não ser coativos sob todas as circunstâncias, de modo que um gene pode pertencer a múltiplos grupos ou a nenhum grupo.

Métodos de agrupamento de subespaço propostos para capturar a coerência exibida por blocos em matrizes de expressão de gene podem ser encontrados em (Getz, Levine e Domany, 2000) (Cheng e Church, 2000). Neste contexto, um bloco é uma submatriz definida por um subconjunto dos genes em um subconjunto de amostras.

Agrupamento de subespaço foi proposto primeiramente por Agrawal no domínio geral de mineração de dados (Agrawal, Gehrke, Gunopulos e Raghavan, 1998) para encontrar subconjuntos de objetos tais que os objetos aparecem como um grupo em um subespaço formado por um subconjunto dos atributos. Em agrupamento de subespaço, os subconjuntos de atributos para vários grupos de subespaço podem ser diferentes. Dois grupos de subespaço podem compartilhar alguns objetos comuns e características, e alguns objetos podem não pertencer a nenhum grupo de subespaço.

O agrupamento baseado em subespaço não foi abordado neste trabalho. O leitor interessado em informações adicionais pode se reportar ao trabalho de Jiang, Tang & Zhang (2004).

5.4 Abordagens de agrupamento não supervisionado e semi-supervisionado

Devido sua característica de classificar os dados de uma forma não-supervisionada, as técnicas de agrupamento são indicadas na análise de dados de expressão gênica (Golub, Slonim, Tamayo, Huard, Gaasenbeek, Mesirov, Coller, Loh, Downing e Caligiuri, 1999; Slonim, Tamayo, Mesirov, Golub e Lander, 2000; Handl, Knowles e Kell, 2005).

O rápido avanço da escala de sequencialização genômica tem impulsionado o desenvolvimento de métodos para explorar estas informações por categorizar processos biológicos em novas formas. O conhecimento da codificação de seqüências de praticamente todos os genes em um organismo, por exemplo, convida o desenvolvimento de tecnologia para estudar a expressão de todos eles de uma vez, pois o estudo da expressão gênica de genes, um a um, já forneceu uma riqueza biológica. Para este fim, uma variedade de técnicas tem evoluído para acompanhar, a rapidez e eficiência, dos transcritos em abundância de todos os genes de um organismo (Schena, Shalon, Davis e Brown, 1995; Velculescu, Zhang, Vogelstein e Kinzler, 1995; Lockhart, Dong, Byrne, Follettie2, Gallo, Chee, Mittmann, Wang, Kobayashi, Norton e Brown, 1996). Dentro da massa de números produzidos por estas técnicas, que ascendem a centenas de pontos de dados para milhares ou dezenas de milhares

de genes, é uma imensa quantidade de informação biológica (Eisen, Spellman, Brown e Botstein, 1998).

Em (Eisen, Spellman, Brown e Botstein, 1998), Eisen aplica o método de agrupamento hierárquico em dados de expressão gênica, gerando o resultado dos algoritmos graficamente em uma forma intuitiva para os biólogos. Neste trabalho, Eisen utiliza o conjunto de dados de levedura *Saccharomyces cerevisiae* e verifica que dados de expressão gênica são agrupados juntos quando genes possuem funções biológicas similares.

Em (Brazma e Vilo, 2000), Brazma apresenta uma discussão sobre a análise de dados supervisionados e não supervisionados e suas aplicações, tais como predizer a classe das funções dos genes e classificação do câncer. Em seguida, é discutido como a matriz de expressão gênica pode ser usada para prever sinais regulatórios putativos em seqüências genômicas.

Em (Jiang, Tang e Zhang, 2004), Jiang introduz os conceitos da tecnologia de *microarray* e discute os elementos básicos de agrupamento de dados de expressão gênica. Em particular, divide a análise de *cluster* para dados de expressão gênica em três categorias. São apresentados os desafios específicos pertinentes a cada categoria de agrupamento e introduzidas as várias abordagens representantes.

Em (Tang, Zhang e Pei, 2003), Tang propõe um novo problema de *mineração de fenótipos simultaneamente e genes informativos* de dados de expressão gênica. Algumas estatísticas baseadas em métricas são propostas neste trabalho para medir a qualidade dos resultados de mineração. Dois algoritmos interessantes são desenvolvidos: a pesquisa heurística e um método de reforço de ajustamento mútuo. É apresentado um extenso estudo em ambos os conjuntos de dados tanto do mundo real como conjuntos de dados sintéticos.

Em (Jiang, Pei e Zhang, 2003), Jiang propõe um framework interativo de exploração para mineração de padrões de expressão coerente em dados de expressão gênica em séries de tempos. Foi desenvolvida uma nova ferramenta, índice de padrão gráfico coerente, para dar aos utilizadores indicações confiantes da existência de padrões coerentes de forma a resumir as informações necessárias para exploração interativa. Além disso, é apresentando um algoritmo para construir árvores e índices padrão gráfico coerentes em conjuntos de dados de expressão gênica.

Neste contexto, este trabalho é endereçado em analisar e apresentar informação sobre escala genômica para dados de expressão gênica.

5.5 Validação de Clusters

Diferentes algoritmos de agrupamento, ou mesmo um único algoritmo aplicado com parâmetros diferentes podem resultar em conjuntos de *clusters* diferentes. Assim sendo, é fundamental avaliar esses resultados e obter indicadores que permitam constatar a consistência e utilidade das estruturas identificadas nos dados. A validação de *clusters* é o processo de assegurar a qualidade e confiabilidade dos conjuntos de *clusters* derivados dos algoritmos de agrupamento (Jiang, Tang e Zhang, 2004).

Nas diversas áreas do conhecimento em que o agrupamento de dados está presente, nota-se uma grande variedade de técnicas de validação sendo possível distinguir entre as medidas de validação internas e externas. Em (Handl, Knowles e Kell, 2005) pode ser encontrado uma coletânea bastante completa de métodos de validação de agrupamento.

Nas medidas de validação externas, a avaliação dos *clusters* está baseada no conhecimento das classes corretas a que pertencem os dados. Essas medidas são bastante úteis quando se deseja avaliar e comparar algoritmos de agrupamento em conjuntos de dados de *benchmark*, para os quais os rótulos das classes são conhecidos e correspondem a estruturas reais. De acordo com (Handl, Knowles e Kell, 2005), as medidas externas podem ser subdivididas em dois grupos: medidas unárias e medidas binárias. Entre as medidas unárias encontram-se as mais tradicionais como *F-measure*, definida mais adiante e usada neste trabalho. A técnica de *F-measure* permite verificar a qualidade de um resultado de agrupamento no nível de toda a partição e não apenas para grupos individuais. Entre as medidas binárias podemos citar como as mais conhecidas o Índice *Rand*, o coeficiente *Jaccard* e a taxa de *Minkowski*.

Nos casos em que as classes não são conhecidas, as medidas internas, que não utilizam nenhuma informação que não seja intrínseca aos dados, são mais apropriadas. Nessa categoria, as medidas mais conhecidas avaliam características como homogeneidade e separação dos grupos. A homogeneidade avalia o quanto elementos do mesmo *cluster* são similares enquanto a separação avalia o quanto elementos de *clusters* distintos são dissimilares.

A exemplo do que acontece no campo do agrupamento de dados em outros domínios, não parece ser possível formular orientações objetivas a respeito da escolha de algoritmos de agrupamento não supervisionado ou semi-supervisionado a serem aplicados ao agrupamento de genes. Com a intensificação das pesquisas na análise de dados de expressão, além das medidas conhecidas da literatura das áreas de estatística e reconhecimento de padrões, muitas

medidas de validação vêm sendo propostas na tentativa de formular estratégias de validação que sejam adequadas às características específicas dos conjuntos de dados desse domínio visando auxiliar na difícil questão da escolha dos algoritmos a ser aplicado.

Em (Datta e Datta, 2003), os autores propõem três estratégias de validação que podem ser usadas com algoritmos de agrupamento quando observações temporais ou replicações estão presentes.

Em (Priness, Maimon e Ben-Gal, 2007) os autores relatam um estudo empírico de avaliação de diversos resultados de agrupamento usando as medidas de homogeneidade e separabilidade baseadas em *Mutual Information* (*MI*) com comparações com as mesmas medidas usando a distância Euclidiana e coeficiente de correlação de *Pearson*.

O problema de validação de agrupamento é também abordado em (Datta e Datta, 2006) com a proposta de duas medidas de desempenho destinadas a avaliar a habilidade do algoritmo de agrupamento de produzir *clusters* biologicamente signifcativos. A primeira medida é um índice de homogeneidade biológica (*BHI*) que mede o quanto os *clusters* são biologicamente homogêneos. A segunda medida é chamada índice de estabilidade biológica (*BSI*) que avalia a capacidade do algoritmo de produzir resultados estáveis quando aplicado a dados similares.

Outros trabalhos representativos abordando o problema de validação de agrupamento com apresentação de medidas de avaliação e estudos comparativos entre vários métodos podem ser encontrados em (Steuer, Humburg e Selbig, 2006) (Yin, Huang e Ni, 2006) (Pihur, Datta e Datta, 2007).

Capítulo 6 Métodos estudados e avaliação experimental

6.1 Considerações Iniciais

Conforme apresentado, a tecnologia de *microarray* tem gerado uma grande quantidade de dados de expressão gênica. Baseando-se na premissa de que genes que possuem funções correlacionadas tendem a exibir padrões de expressão similares, vários métodos de aprendizado tem sido aplicado para capturar padrões em dados de expressão (Lu, Tian, Liu, Sanchez e Wang, 2007).

Em agrupamento não supervisionado, nenhuma referência pré-definida é utilizada (Eisen, Spellman, Brown e Botstein, 1998). Tem como objetivo organizar o conjunto de dados em grupos, tais que os dados no mesmo grupo sejam mais similares do que outros dados em outros grupos (Grira, Crucianu e Boujemaa, 2005).

Em muitas tarefas de aprendizado, há uma grande quantidade de dados não rotulados e poucos dados rotulados, pois a identificação dos rótulos para cada exemplo é um processo bastante custoso. No agrupamento semi-supervisionado exemplos rotulados são utilizados para agrupar exemplos não rotulados (Basu, Banerjee e Mooney, 2002).

Com base nisso, este trabalho tem por objetivo fazer uma análise da aplicação de algoritmos de agrupamento não supervisionado e algoritmos de agrupamento semi-supervisionado em dados de expressão gênica, para o agrupamento de genes, a fim de verificar o comportamento destes na geração de padrões ou funções similares.

Esta análise pode vir a trazer descobertas sobre funções desconhecidas de genes não rotulados, gerando assim informações biológicas relevantes. As seções seguintes decrevem as características dos conjuntos de dados, algoritmos, medidas de similaridade e medidas de validação utilizadas e resultados obtidos nos experimentos.

6.2 Conjunto de dados e pré-processamento

Uma importante etapa no processo de análise de dados é o pré-processamento dos dados. No contexto de dados de expressão gênica essa fase deve ser feita com a preocupação

de, além de consolidar as informações relevantes que o algoritmo utiliza, manter a variação biológica. Alguns trabalhos que abordam a questão de pré-processamento dos dados para expressão gênica podem ser encontrados em (Borges, Nievola e Pucpr, 2007). Entre os principais problemas tratados no pré-processamento estão os dados faltantes e a identificação de atributos irrelevantes ou redundantes.

Experimentos de *microarray* de expressão gênica podem gerar conjuntos de dados com muitos valores de expressão faltantes. A maioria dos algoritmos para análise de dados de expressão gênica requer uma matriz completa de valores de genes como entrada. Por exemplo, métodos tais como agrupamento hierárquico e *K-means* não são robustos a dados faltantes e pode perder a eficácia, mesmo com poucos valores faltantes. Métodos para inserir dados faltantes são necessários, para minimizar o efeito de conjuntos de dados incompletos nas análises, e para aumentar a gama de conjuntos de dados nos quais estes algoritmos podem ser aplicados (Troyanskaya, Cantor, Sherlock, Brown, Hastie, Tibshirani e Botstein, 2001).

Quando a quantidade de atributos utilizada tem grande influência nos algoritmos de aprendizado, a identificação dos atributos realmente importantes para uma aplicação se torna um fator extremamente importante para se ter sucesso. A exclusão de atributos irrelevantes ou redundantes quase sempre melhora a qualidade do resultado obtido.

A análise de métodos para tratamento de dados faltante não é o foco deste trabalho, portanto, nenhum método para completar os dados faltantes foi utilizado. Como alguns algoritmos utilizados neste trabalho necessitam de um conjunto de dados completos, o conjunto de genes com dados faltantes foram retirados do conjunto original.

Para os experimentos realizados, foram utilizados dois grupos de conjuntos de dados. No primeiro grupo estão dois conjuntos (*wine* e *íris*) que são *benchmarks* comumente utilizados para testar problemas em aprendizado de máquina e encontram-se disponíveis no repositório de dados da UCI (Newman e Asuncion, 2007). Estes dados representam aplicações reais e já foram largamente manipulados e empregados na literatura. Os experimentos conduzidos com esses conjuntos visam criar um quadro mais rico de resultados, que permita avaliar o desempenho dos métodos de agrupamento investigados em domínios com características diferentes daquelas dos domínios de expressão gênica.

O outro grupo consiste de dados de bioinformática, especificamente de expressão gênica, que são variações dos dados de *Saccharomyces cerevisiae*, largamente utilizados na literatura e originalmente apresentados em (Chu, Derisi, Eisen, Mulholland, Botstein, Brown e Herskowitz, 1998). Dois dos conjuntos de dados experimentais desse grupo foram derivados do conjunto original apresentados em (Chu, Derisi, Eisen, Mulholland, Botstein, Brown e

Herskowitz, 1998) e, outros dois foram obtidos de conjuntos de dados apresentados em (Hughes, Marton, Jones, Roberts, Stoughton, Armour, Bennett, Coffey, Dai, He, Kidd, King, Meyer, Slade, Lum, Stepaniants, Shoemaker, Gachotte, Chakraburtty, Simon, Bard e Friend, 2000).

Para os conjuntos de dados *íris* e *wine* nenhum procedimento de pré-processamento foi realizado.

Os dois primeiros conjuntos de dados *yeast Saccharomyces cerevisiae*, foram obtidos dos dados coletados por Chu. Que registra dados de expressão durante a fermentação de *Saccharomyces cerevisiae* em sete instantes de tempo. Inicialmente os genes com dados faltantes foram eliminados e, em seguida, foi utilizada a ferramenta FunCat disponível em http://mips.gsf.de/proj/funcatDB/search_main_frame.html, para classificar os genes em suas respectivas funções biológicas. Foram obtidas 17 funções distintas para o conjunto de dados, sendo que apenas 10 delas foram utilizadas, por serem as que possuíam um número mínimo razoável de genes: *metabolism*, *energy*, *cell cycle and DNA processing*, *transcription*, *protein synthesis*, *protein fate*, *transport*, *defense*, *biogenesis* e *cell differentiation*. Este conjunto deu origem a dois conjuntos distintos, que serão chamados de *yeast1* e *yeast2*. O primeiro conjunto de dados (*yeast1*) contém os genes que possuem somente uma função associada, ou seja, pertencem a apenas uma classe. O segundo conjunto de dados (*yeast2*) é formado pelo conjunto de dados *yeast1* adicionado de 20 genes aleatórios que possuem mais de uma função e, portanto, pertencem a mais de uma classe. Esse conjunto foi usado em experimentos com algoritmos semi-supevisionados que aceitam dados com essa característica.

Os outros dois conjuntos de dados de *yeast Saccharomyces cerevisiae* foram obtidos de um conjunto de dados fornecido, gentilmente, por Wei Pan, que foi utilizado em seu trabalho apresentado em (Huang e Pan, 2006). Esse conjunto de dados é proveniente de um grande conjunto de dados que contém 300 experimentos de microarray com eliminação de genes e tratamentos com drogas apresentados em (Hughes, Marton, Jones, Roberts, Stoughton, Armour, Bennett, Coffey, Dai, He, Kidd, King, Meyer, Slade, Lum, Stepaniants, Shoemaker, Gachotte, Chakraburtty, Simon, Bard e Friend, 2000). Os dados foram processados de tal forma que a média e variância das amostras dos dados de expressão para cada gene é 0 e 1 respectivamente. As funções gênicas foram obtidas da base de dados MIPS (Mewes, Amid, Arnold, Frishman, Güldener, Mannhaupt, Münsterkötter, Pagel, Strack, Stümpflen, Warfsmann e Ruepp, 2004). Os conjuntos de dados usados neste trabalho foram construídos seguindo as mesmas considerações de *Huang & Pan* no seu trabalho (Huang e Pan, 2006), que usaram apenas três funções gênicas: *mitotic cell cycle and cell cycle control*,

mitochondrion e *c-compound and carbohydrate utilization*, chamadas de classe 1, classe 2 e classe 3, respectivamente. O primeiro conjunto de dados usado neste trabalho, obtido daí, foi chamado de yeast3 e contém somente genes da classe 1 e classe 2. O segundo conjunto de dados, chamado de *yeast4*, contém genes das três classes. Essa definição dos conjuntos foi adotada em função de características do algoritmo proposto por *Huang & Pan* (2006), um dos estudados neste trabalho, que permite a construção de *clusters* além daqueles definidos previamente, para identificar estruturas inesperadas nos dados agrupados. Maiores explicações sobre o algoritmo e o conjunto de dados são encontradas na seção 3.4.1.5.

A Tabela 1 contém as informações das características dos conjuntos de dados. Nesta tabela *n* é o número de objetos presentes no conjunto de dados, *d* é a dimensionalidade dos dados (número de atributos) e *c* é o número de classes.

Domínio	Conjunto de Dados	*n*	*D*	*c*
Não Expressão Gênica	*Íris*	150	4	3
	Wine	175	13	3
Expressão Gênica	*Yeast1*	545	80	10
	Yeast2	585	80	10
	Yeast3	630	300	2
	Yeast4	845	300	3

Tabela 1: Características dos conjuntos de dados

6.3 Medida de Similaridade

Muitos algoritmos de agrupamento dependem muito da medida de "similaridade" ou "distância", que quantifica o grau de associação entre os perfis de expressão. A definição de medida de distância é um fator chave para o sucesso da identificação da relação entre os genes e redes de genes (D'haeseleer, Liang e Somogyi, 2000) (Priness, Maimon e Ben-Gal, 2007).

A escolha de uma medida de distância – usada para quantificar a diferença nos perfis de expressão entre dois genes – talvez seja tão importante como a escolha do algoritmo de agrupamento. Medidas de distância podem ser divididas em pelo menos três classes, enfatizando diferentes regularidades presentes nos dados: (a) similaridade de acordo com correlações positivas, que identificam regulações positivas e negativas; (b) similaridade de acordo com correlações positivas e negativas, que também auxiliam na identificação de processos de controle que regulam antagonicamente os *pathways;* (c) similaridade de acordo com a informação mútua, que detecta relacionamentos mais complexos (D'haeseleer, Liang e Somogyi, 2000).

Em muitos estudos na literatura de agrupamento de expressão gênica, medidas como distância Euclidiana e correlação de Person entre perfis de expressão são utilizadas como medida de distância (D'haeseleer, Liang e Somogyi, 2000).

Neste trabalho foi utilizada a medida chamada de Distância Euclidiana, que é uma das medidas mais comumente utilizadas na literatura para agrupamento de dados de expressão gênica e por isso foi escolhida para ser utilizada neste trabalho. A descrição dessa e de outras medidas de similaridade foi apresentada na seção 2.5. Alguns dos algoritmos de agrupamento semi-supervisionado investigados aqui utilizam uma versão expandida da medida de similaridade, que passa a considerar o conhecimento disponível sobre os dados, seja na forma de rótulos ou de restrições. Nesses casos, a medida de similaridade é definida acrescentando-se um novo parâmetro que, combinado com a medida de distância convencional, reforça a similaridade final entre dois exemplos que, de acordo com o conhecimento disponível, pertencem a mesma classe e diminui a similaridade dos que pertencem a classes diferentes.

6.4 Algoritmos utilizados

Neste trabalho foram utilizados algoritmos de agrupamento não supervisionados e semi-supervisionados. Os algoritmos de agrupamento não supervisionado utilizados foram o *K-means* e o agrupamento hierárquico aglomerativo *bottom-up*, apresentados na seção 2.6. Os algoritmos semi-supervisionados considerados, descritos no capítulo 23, foram:

- ✓ baseados em sementes, como *Seeded K-means* e *Constrained K-means (Basu, Banerjee e Mooney, 2002),* que são extensões do *K-means* e foram propostos como algoritmos gerais de agrupamento, no contexto de aprendizado de máquina;
- ✓ baseados em restrições, como *Cop-Kmeans* (Wagstaff, Cardie, Rogers e Schroedl, 2001) (extensão do K-means) e *PCKMeans (Basu, Banjeree e Mooney, 2004)* (extensão do *K-medoides*) e também propostos como algoritmos gerais de agrupamento, no contexto de aprendizado de máquina;
- ✓ propostos especificamente para domínios de expressão gênica, como método de *Huang & Pan* (Huang e Pan, 2006) e método de *Boratyn* (Boratyn, Datta e Datta, 2006), este último uma extensão do algoritmo de agrupamento hierárquico.

Estes algoritmos foram utilizados para realizar uma análise do comportamento dos diferentes algoritmos aplicados à conjuntos de dados de domínio de expressão gênica e de não

expressão gênica. Todos os algoritmos foram implementados em Java, a seguir é descrita a metodologia utilizada para cada um dos algoritmos estudados.

6.5 Medida de Validação

Uma das medidas de validação utilizadas foi o *F-measure* baseado em pares, que é definido como a média harmônica de precisão e *recall* baseadas em pares, medidas da área de recuperação de informação tradicional, adaptadas para avaliar agrupamento considerando pares de pontos. Essa é uma medida externa, pois conforme explicado na seção 5.5, é necessário conhecer previamente as classes dos dados agrupados. O valor da medida é calculado considerando que, para cada par de pontos, a decisão de agrupar este par em um mesmo *cluster* ou em *clusters* diferentes é considerada como sendo correta, se esse agrupamento corresponde às classes conhecidas dos pontos. A medida *Pairwise F-measure* é definida como:

$$Precision = \frac{\#PairsCorrectlyPredictedInSameCluster}{\#TotalPairsPredictedInSameCluster}$$

$$Recall = \frac{\#PairsCorrectlyPredictedInSameCluster}{\#TotalPairsActuallyInSameCluster}$$

$$F - measure = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall}$$

Uma outra medida de validação chamada *Biological Homogeneity Index*, BHI, proposta mais recentemente no domínio de expressão gênica com o objetivo de avaliar o resultado de algoritmos de agrupamento e sua habilidade de produzir *clusters* biologicamente significativos (Datta e Datta, 2006) também foi empregada para os métodos semi-supervisionados: *Seeded-K-Means, Constrained-K-Means, Huang & Pan, Boratyn.* Conforme explicado na seção 5.5, essa medida avalia a homogeneidade dos *clusters*. Essa medida não foi aplicada aos algoritmos baseados em restrições de pares, por necessitar explicitamente do rótulo do dado para ser calculada.

De acordo com Datta & Datta, essa medida é assim definida: Considere dois genes *x, y* que pertencem ao mesmo *cluster D*. Seja *C*(*x*) a classe funcional que contém o gene *x*.

Similarmente, $C(y)$ a classe que contém o gene y. A função $I(C(x) = C(y))$ tem valor 1 se $C(x)$ e $C(y)$ forem iguais (nos casos em que o gene pertence a mais de uma classe funcional, será igual a 1 se pelo menos uma das funções dos genes x e y for igual). A função matemática para medir o valor da similaridade biológica é:

$$BHI = \frac{1}{k}\sum_{j=1}^{k}\frac{1}{nj(nj-1)}\sum_{x\neq y\in Dj} I(C(x) = C(y))$$

onde k é o número de *clusters* e para um *cluster* D_j, $n_j = n(D_j \cap C)$ é o número de genes rotulados em D_j, e para um conjunto A, $n(A)$ é o tamanho do conjunto.

Ambas as medidas de validação, F-measure e BHI, foram escolhidas neste trabalho, pois são medidas bastante utilizadas na literatura.

Outras medidas, inicialmente consideradas, como BSI (Datta e Datta, 2006) e *hard classification* (Huang e Pan, 2006), não foram utilizadas por exigirem procedimentos diferentes na condição dos experimentos.

6.6 Resultados Experimentais

A metodologia utilizada nos experimentos, na maioria dos casos, foi a validação cruzada com 5 partições – os conjuntos de dados foram particionados aleatoriamente em 5 partições de tamanho aproximadamente igual e, em cada uma das 5 execuções, uma das partições foi usada como conjunto de teste e as outras 4 como conjunto de treinamento. Em cada execução foi calculado o valor de *F-measure* ou BHI considerando apenas os dados do conjunto de teste. Os gráficos apresentados nessa seção foram construídos com cada ponto do gráfico representando a média dos valores de *F-measure* ou BHI para cada configuração dos conjuntos de dados, obtidas variando a porcentagem de sementes ou de pares de restrições.

6.6.1 Conjunto de dados: *Íris*

O Gráfico 1 mostra o resultado obtido para o conjunto de dados *Íris* pela aplicação da medida de validação *F-measure*, com os algoritmos *K-means* e os algoritmos baseados em sementes, variando de 0 a 100% o número de sementes no conjunto de dados. Para a primeira execução foram rotulados aleatoriamente 20% dos dados, para a segunda 40%, para a terceira 60%, para a quarta 80% e para a quinta 100%. O algoritmo *K-means*, que é não-

supervisionado, foi executado cinco vezes para cada partição ignorando os rótulos considerados pelos outros algoritmos.

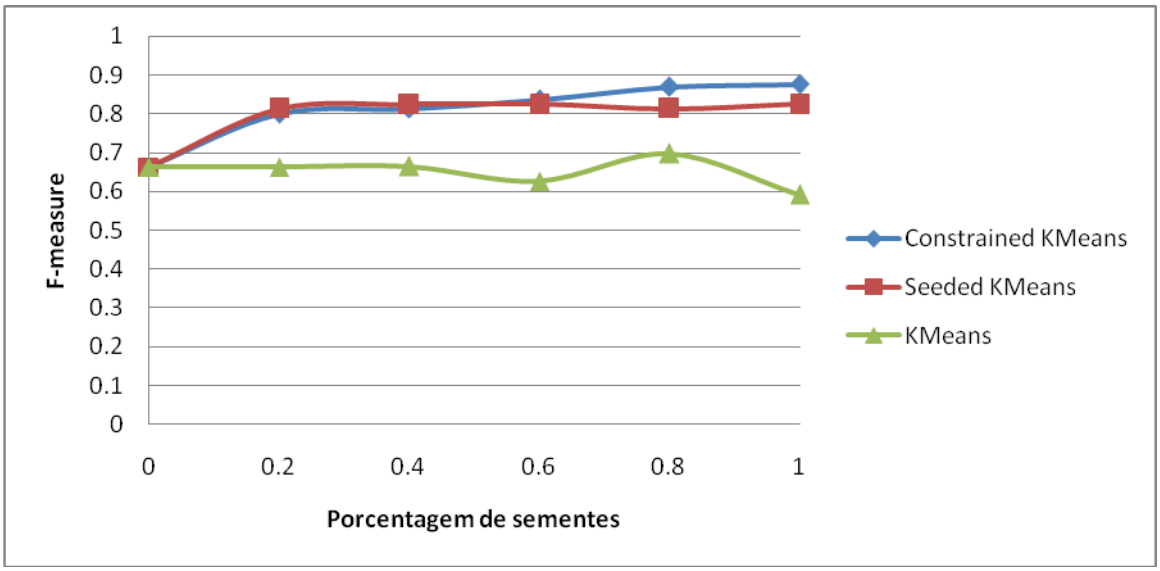


Gráfico 1: Conjunto Íris – F-measure – Comparação K-Means, Seeded-Kmeans e Constrained-Kmeans

A Tabela 2 apresenta o desvio padrão para o Kmeans.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.648107902	0.035742468

Tabela 2: Conjunto Íris – Kmeans - Desvio Padrão

A Tabela 3 apresenta o desvio padrão para o Seeded-Kmeans.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.812181219	0.024535872
0.82369201	0.031369885
0.82369201	0.031369885
0.812181219	0.024535872
0.82369201	0.031369885

Tabela 3: Conjunto Íris – Seeded-Kmeans - Desvio Padrão

A Tabela 4 apresenta o desvio padrão para o Constrained-Kmeans.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.80041368	0.040061281
0.8127633	0.035254938
0.83628278	0.037944271
0.86895607	0.038630202
0.87647256	0.027238374

Tabela 4: Conjunto Íris – Constrained-Kmeans - Desvio Padrão

O Gráfico 2 apresenta os resultados do mesmo conjunto de dados para a medida de validação BHI.

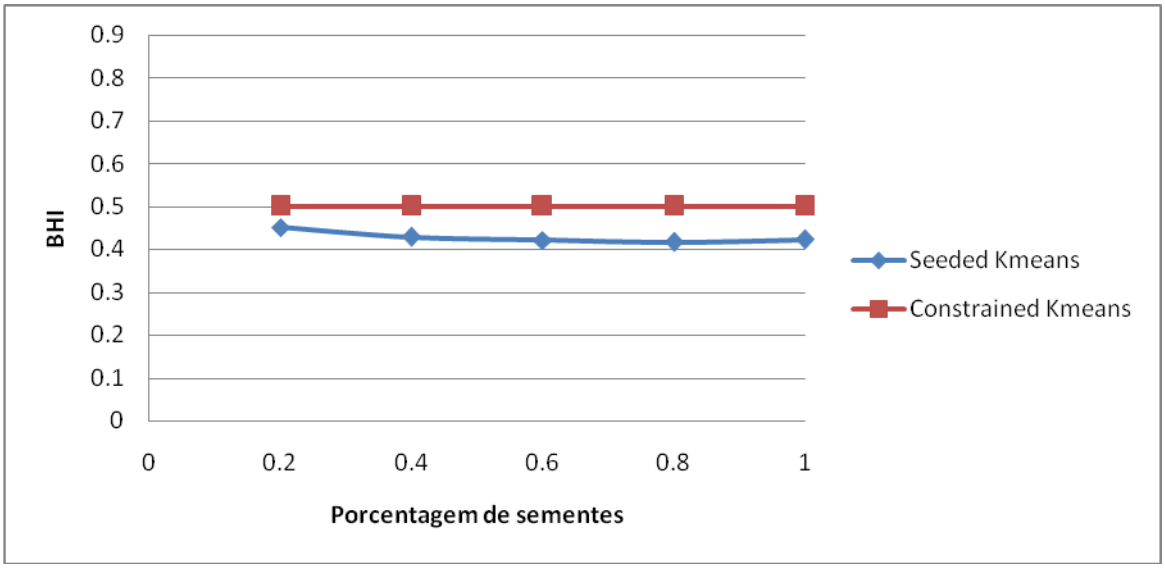


Gráfico 2: Conjunto Íris – BHI - Comparação Seeded-K-Means e Constrained-K-Means

A Tabela 5 apresenta o desvio padrão para o Seeded-Kmeans.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.451736412	0.017214808
0.428241523	0.026476767
0.421166111	0.017571283
0.416422817	0.007462889
0.422615479	0.005292944

Tabela 5: Conjunto Íris – Seeded-Kmeans - Desvio Padrão

A Tabela 6 apresenta o desvio padrão para o Constrained-Kmeans.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.5	0
0.5	0
0.5	0
0.5	0
0.5	0

Tabela 6: Conjunto Íris – Constrained-Kmeans - Desvio Padrão

O Gráfico 3 mostra o desempenho dos algoritmos baseados em restrições, juntamente com o *K-means*, para comparações, utilizando o *F-measure*, variando o número de restrições

dos conjuntos *must-link* e *cannot-link*. O número de restrições foi variado de 20 a 100 restrições, na primeira execução foi consideradas 20 restrições, na segunda 40 restrições, na terceira 60 restrições, na quarta 80 restrições e na quinta 100 restrições entre pares.

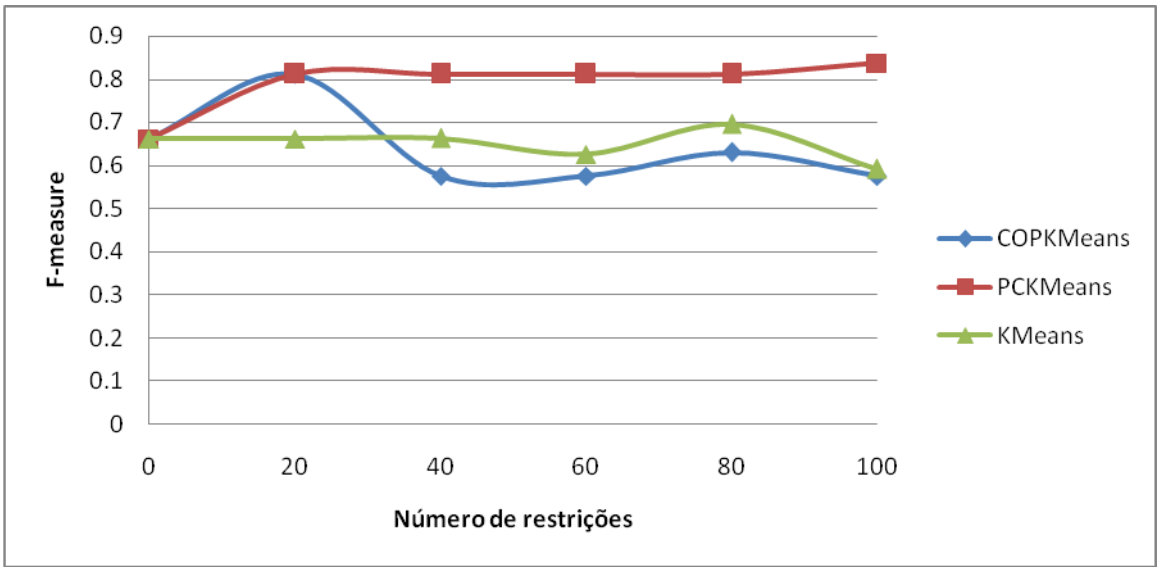


Gráfico 3: Conjunto Íris – F-measure – Comparação K-Means, Cop-Kmeans e PCKMeans

A Tabela 7 apresenta o desvio padrão para o Cop-Kmeans.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.812181219	0.024535872
0.575162561	0.109980841
0.575471044	0.113107087
0.630965096	0.124132346
0.575471044	0.113107087

Tabela 7: Conjunto Íris – Cop-Kmeans - Desvio Padrão

A Tabela 8 apresenta o desvio padrão para o PCKmeans.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.812181219	0.024535872
0.812181219	0.024535872
0.812181219	0.024535872
0.812181219	0.024535872
0.836744425	0.051958982

Tabela 8: Conjunto Íris – PCKmeans - Desvio Padrão

O algoritmo proposto por *Huang & Pan*, que utiliza informação sobre a classe de alguns genes, foi executado variando o número de classes conhecidas no conjunto de dados, de 0 a 100%. A primeira execução considerou que nenhum novo *cluster* seria gerado na segunda parte do algoritmo, além daqueles definidos para a primeira parte, ou seja, $k_1 = 0$, no passo 2 do método de *Huang & Pan*, descrito na seção 3.4.1.5. Foram utilizados os valores 0.8 e 1 para o parâmetro *r*.

O Gráfico 4 mostra o resultado do desempenho deste algoritmo, utilizando a medida *F-measure* e o gráfico 5 mostra os resultados para a medida BHI.

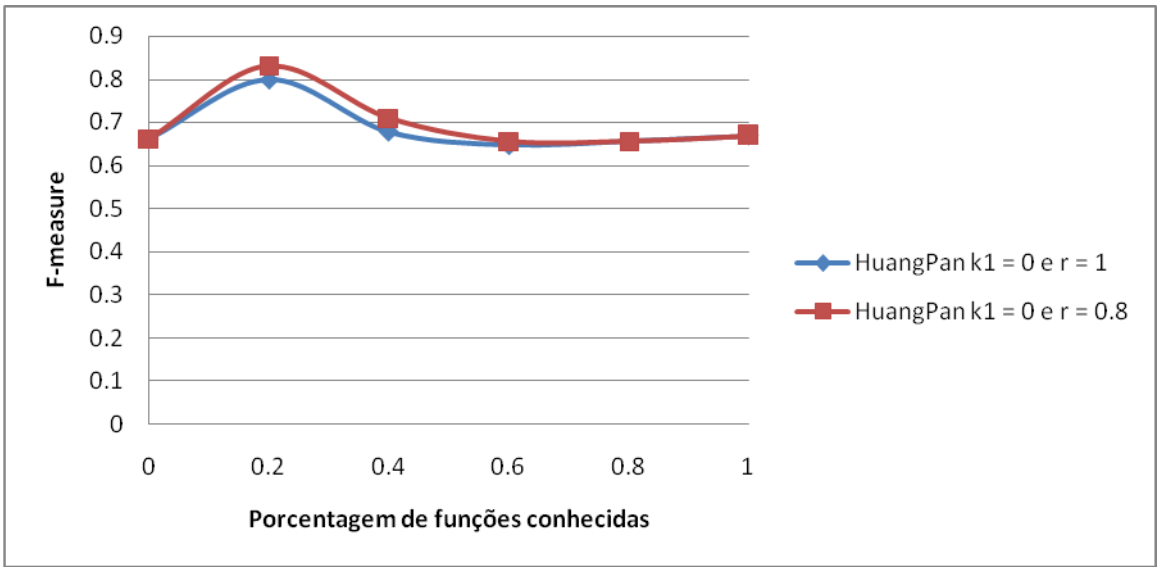


Gráfico 4: Conjunto Íris – F-measure – Método Huang & Pan para $k_1 = 0$

A Tabela 9 apresenta o desvio padrão para o Método Huang & Pan para $k_1 = 0$ e $r = 1$.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.79871075	0.024926385
0.678639988	0.061475836
0.647446722	0.021023117
0.655950255	0.012692103
0.669567057	0.037760937

Tabela 9: Conjunto Íris – Huang & Pan - Desvio Padrão

A Tabela 10 apresenta o desvio padrão para o Método Huang & Pan para k_1 = 0 e r =0.8.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.812181219	0.024535872
0.812181219	0.024535872
0.812181219	0.024535872
0.812181219	0.024535872
0.836744425	0.051958982

Tabela 10: Conjunto Íris – Huang & Pan - Desvio Padrão

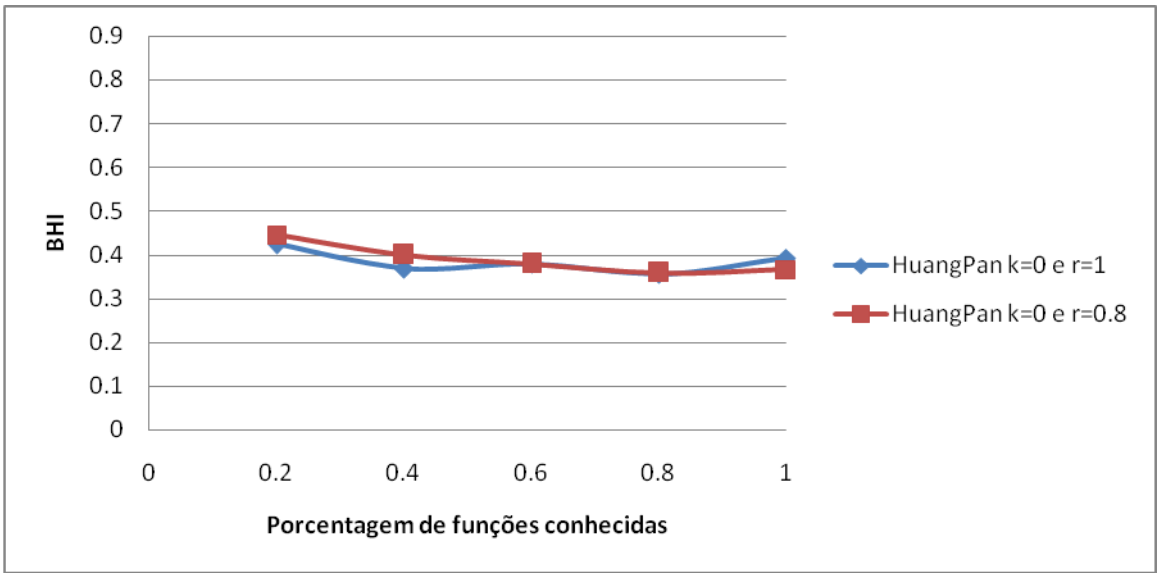


Gráfico 5: Conjunto Íris – BHI - Método Huang & Pan para k_1 = 0

A Tabela 11 apresenta o desvio padrão para o Método Huang & Pan para k_1 = 0 e r = 1.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.424390054	0.01195108
0.368873404	0.059156736
0.379431457	0.028520943
0.356879248	0.065151176
0.392756862	0.025023591

Tabela 11: Conjunto Íris – Huang & Pan - Desvio Padrão

A Tabela 12 apresenta o desvio padrão para o Método Huang & Pan para k_1 = 0 e r =0.8.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.4452405	0.024911182
0.400158491	0.063080505
0.379431457	0.028520943
0.359048496	0.067857007
0.366559144	0.067529104

Tabela 12: Conjunto Íris – Huang & Pan - Desvio Padrão

A segunda execução considerou que um novo *cluster* poderia ser gerado na segunda parte do algoritmo, ou seja, k_1 = 1, no passo 2 do método de *Huang & Pan*. Foram utilizados, novamente, os valores 0.8 e 1 para o parâmetro r. O Gráfico 6 mostra o resultado do desempenho deste algoritmo, utilizando a medida *F-measure* e o gráfico 7 mostra o resultado para a medida BHI.

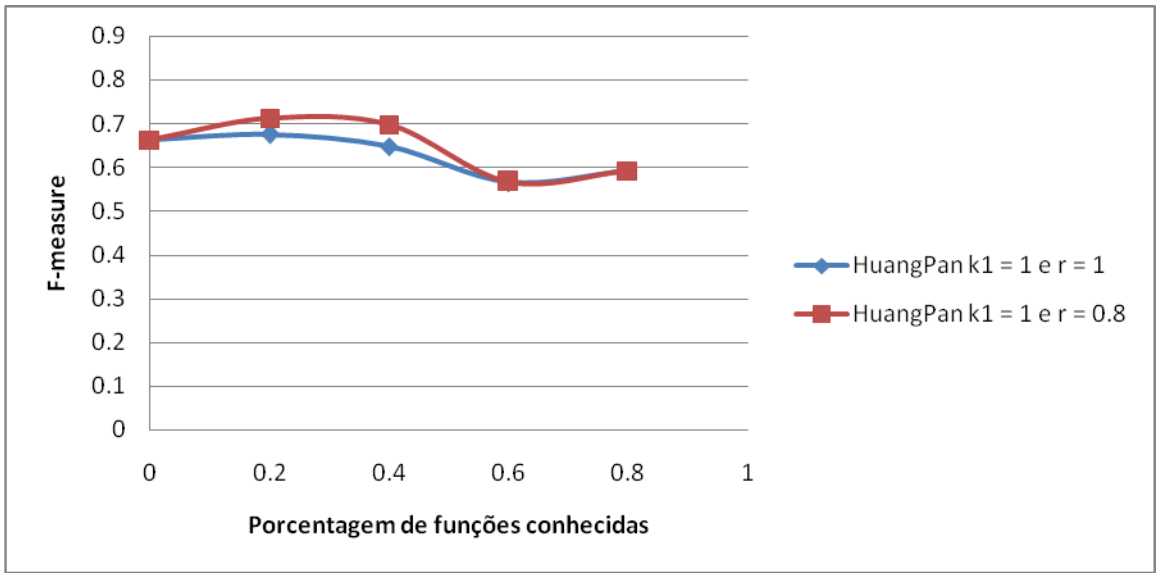


Gráfico 6: Conjunto Íris – F-measure - Método Huang & Pan para k_1 = 1

A Tabela 13 apresenta o desvio padrão para o Método Huang & Pan para k_1 = 1 e r = 1.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.675568158	0.103121118
0.647701112	0.070879013
0.566135972	0.055859042
0.592120484	0.062941793

Tabela 13: Conjunto Íris – Huang & Pan - Desvio Padrão

A Tabela 14 apresenta o desvio padrão para o Método Huang & Pan para k_1 = 1 e r =0.8.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.71265946	0.062794785
0.698171837	0.070425971
0.566871039	0.055614528
0.592120484	0.062941793

Tabela 14: Conjunto Íris – Huang & Pan - Desvio Padrão

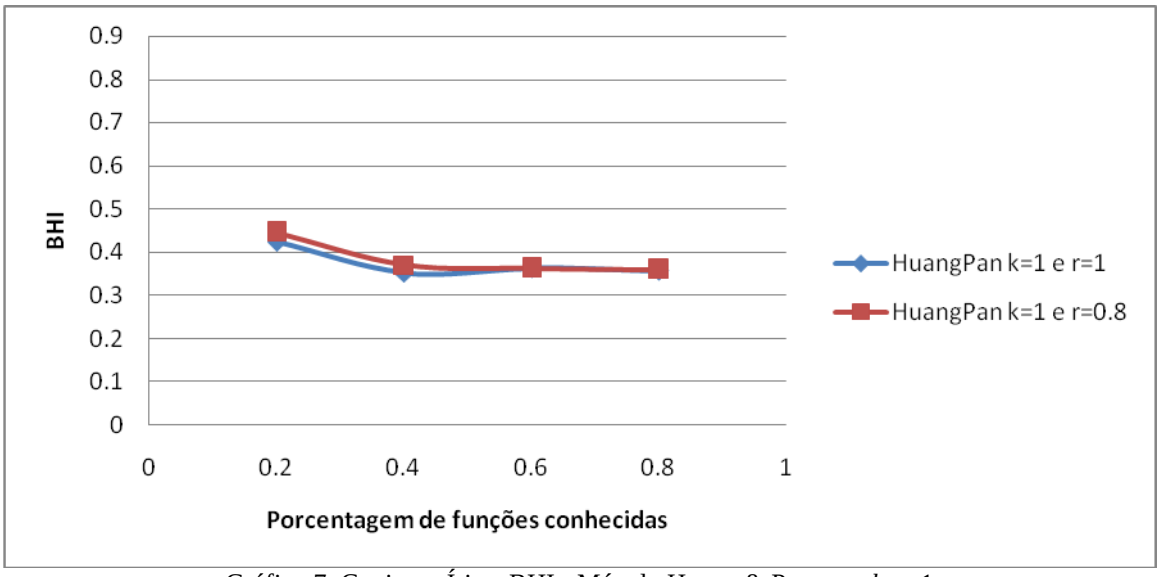


Gráfico 7: Conjunto Íris – BHI - Método Huang & Pan para k_1 = 1

A Tabela 15 apresenta o desvio padrão para o Método Huang & Pan para k_1 = 1 e r = 1.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.424390054	0.01195108
0.352333	0.079635323
0.362934386	0.051543191
0.356879248	0.065151176

Tabela 15: Conjunto Íris – Huang & Pan - Desvio Padrão

A Tabela 16 apresenta o desvio padrão para o Método Huang & Pan para k_1 = 1 e r =0.8.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.4452405	0.024911182
0.370333929	0.06221636
0.362934386	0.051543191
0.359048496	0.067857007

Tabela 16: Conjunto Íris – Huang & Pan - Desvio Padrão

O Gráfico 8 mostra o resultado obtido de acordo com a medida *F-measure* para os algoritmos hierárquico e *Boratyn* para comparação, uma vez que o algoritmo *Boratyn* é uma extensão do hierárquico. A porcentagem de funções conhecidas variou de 0 a 100% e o parâmetro assumiu os valores 0,8 e 0,5.

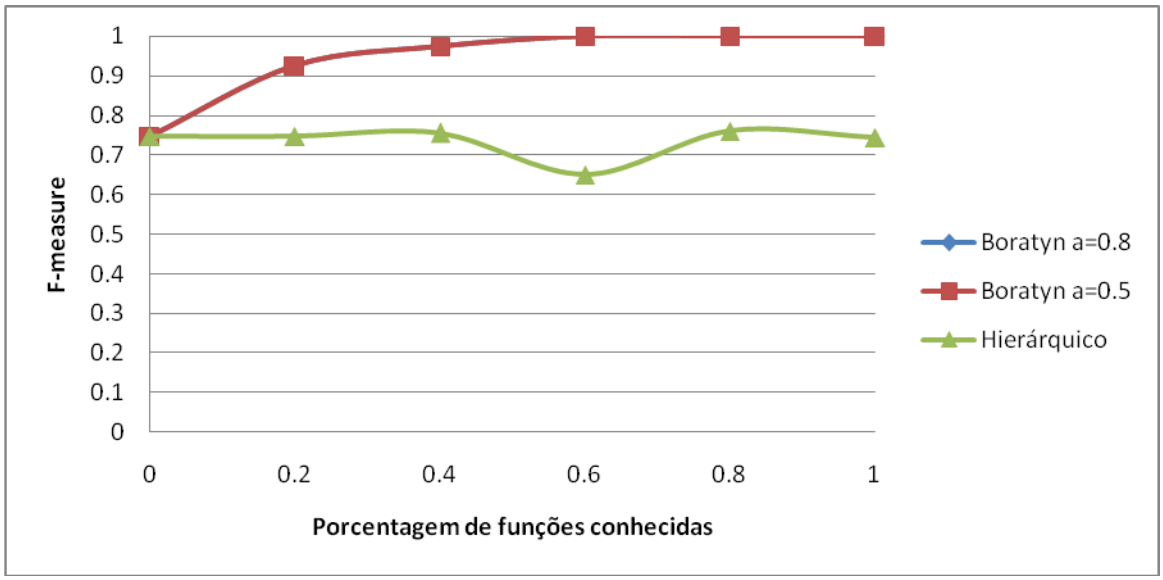


Gráfico 8: Conjunto Íris – F-measure - Método Boratyn

Para o algoritmo Boratyn, nas condições λ = 0.5 e λ = 0.8, as curvas no gráfico tiveram comportamento similares apresentando uma sobreposição de valores.

A Tabela 17 apresenta o desvio padrão para o Método Boratyn para $\lambda = 0.8$.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.925712924	0.044855186
0.97394958	0.032153154
1	0
1	0
1	0

Tabela 17: Conjunto Íris – Boratyn - Desvio Padrão

A Tabela 18 apresenta o desvio padrão para o Método Boratyn para $\lambda = 0.5$.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.925712924	0.044855186
0.97394958	0.032153154
1	0
1	0
1	0

Tabela 18: Conjunto Íris –Boratyn - Desvio Padrão

O gráfico 9 apresenta os resultados para a medida de validação, BHI:

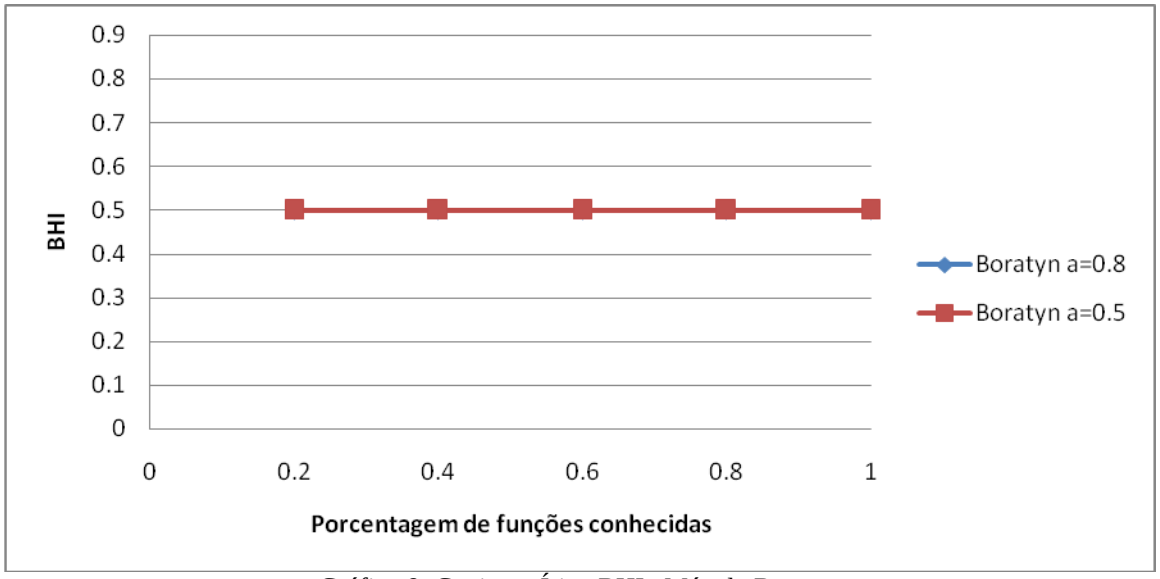


Gráfico 9: Conjunto Íris - BHI– Método Boratyn

Para o algoritmo Boratyn, nas condições $\lambda = 0.5$ e $\lambda = 0.8$, as curvas no gráfico tiveram comportamento similares apresentando uma sobreposição de valores.

A Tabela 19 apresenta o desvio padrão para o Método Boratyn para $\lambda = 0.8$.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.5	0
0.5	0
0.5	0
0.5	0
0.5	0

Tabela 19: Conjunto Íris – Boratyn - Desvio Padrão

A Tabela 20 apresenta o desvio padrão para o Método Boratyn para $\lambda = 0.5$.

Conjunto de dados *Íris*	
Média	Desvio Padrão
0.5	0
0.5	0
0.5	0
0.5	0
0.5	0

Tabela 20: Conjunto Íris –Boratyn - Desvio Padrão

Como pode ser observado no Gráfico 1, ambos os algoritmos semi-supervisionado *Seeded-K-Means* e *Constrained-K-Means*, tiveram uma melhor performance em relação ao algoritmo não supervisionado, *K-Means*. Sendo que o *Contrained-K-Means* desempenhou um pouco melhor do que o *Seeded-K-Means* (Gráficos 1 e 2).

Os algoritmos semi-supervisionados baseados em restrições também tiveram um melhor desempenho em relação ao *K-Means*, sendo que o *PCKMeans* teve uma melhor performance que o *COPKmeans* (Gráfico 3). O mesmo aconteceu com o algoritmo proposto por *Huang & Pan* (Gráficos 4 a 7).

O algoritmo proposto por *Boratyn*, também apresentou um melhor desempenho em relação ao agrupamento hierárquico (Gráfico 8).

6.6.2 Conjunto de dados: *Wine*

O gráfico 10 mostra o resultado obtido pela aplicação da medida *F-measure* aos algoritmos baseados em sementes e o *K-means* e o Gráfico 11 mostra o resultado da medida BHI para os mesmos algoritmos.

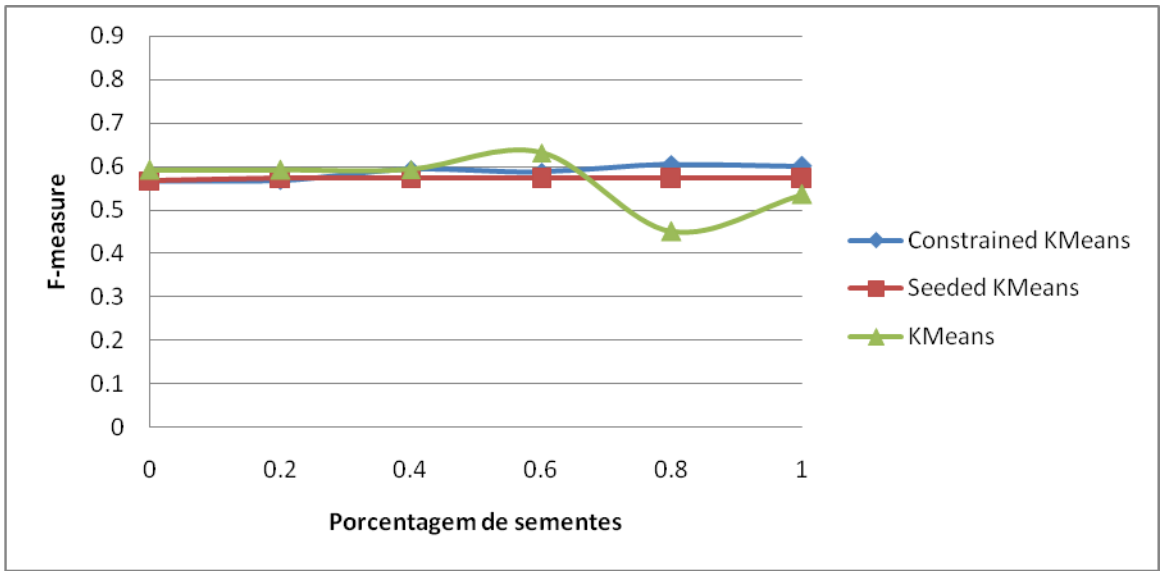


Gráfico 10: Conjunto Wine – F-measure - Comparação K-Means, Seeded-Kmeans e Constrained-Kmeans

Nota-se que o *K-means* teve um melhor desempenho para o conjunto de dados Íris quando comparado ao conjunto de dados Wine. Já os algoritmos semi-supervisionados, *Seeded-K-Means* e *Constrained-K-Means*, tiveram desempenho similares nos dois cojuntos de dados.

A Tabela 21 apresenta o desvio padrão para o Kmeans.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.560718831	0.062970069

Tabela 21: Conjunto Wine – Kmeans - Desvio Padrão

A Tabela 22 apresenta o desvio padrão para o Seeded-Kmeans.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.572151844	0.016876815
0.572151844	0.016876815
0.572151844	0.016876815
0.572151844	0.016876815
0.572151844	0.016876815

Tabela 22: Conjunto Wine – Seeded-Kmeans - Desvio Padrão

A Tabela 23 apresenta o desvio padrão para o Constrained-Kmeans.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.568977241	0.016203214
0.593224332	0.042278221
0.587119026	0.043733556
0.604843261	0.030472514
0.600090419	0.027231993

Tabela 23: Conjunto Wine – Constrained-Kmeans - Desvio Padrão

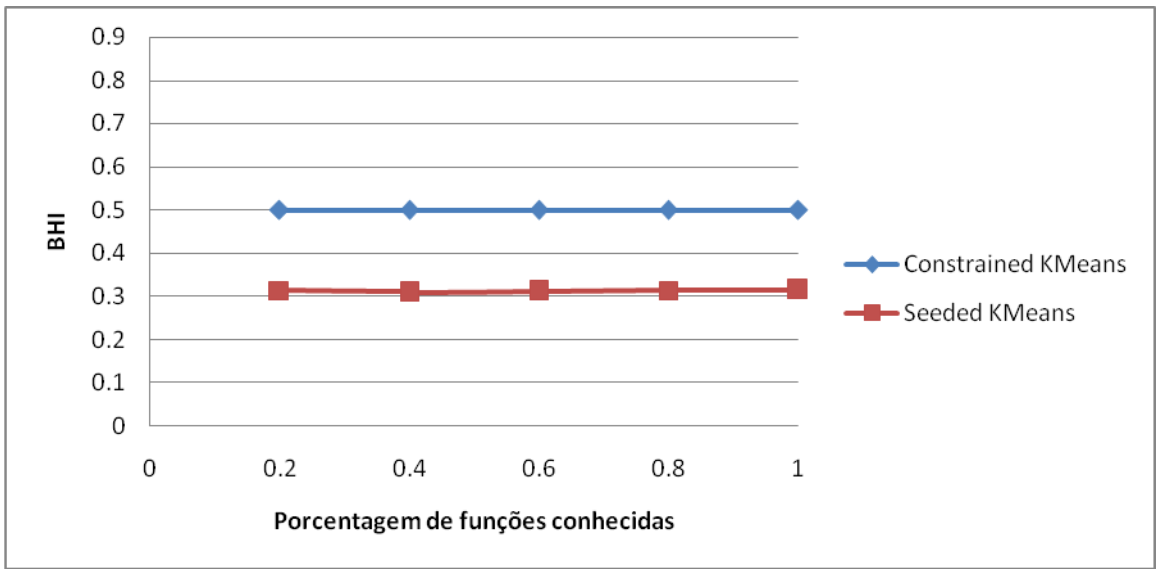


Gráfico 11: Conjunto Wine – BHI - Comparação Seeded-Kmeans e Constrained-Kmeans

Na análise da medida de validação BHI, para os conjuntos de dados *Íris* e *Wine*, ambos os algoritmos tiveram resultados similares, sendo que o *Constrained-K-Means* apresentou um melhor desempenho em relação ao *Seeded-K-Means*.

A Tabela 24 apresenta o desvio padrão para o *Seeded-Kmeans*.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.313350723	0.012767846
0.309915177	0.006382075
0.311855257	0.006911598
0.313332792	0.005370401
0.315224543	0.005076008

Tabela 24: Conjunto Wine – Seeded-Kmeans - Desvio Padrão

A Tabela 25 apresenta o desvio padrão para o *Constrained-Kmeans*.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.5	0
0.5	0
0.5	0
0.5	0
0.5	0

Tabela 25: Conjunto Wine – Constrained-Kmeans - Desvio Padrão

O Gráfico 12 mostra o desempenho dos algoritmos baseados em restrições, utilizando o *F-measure*.

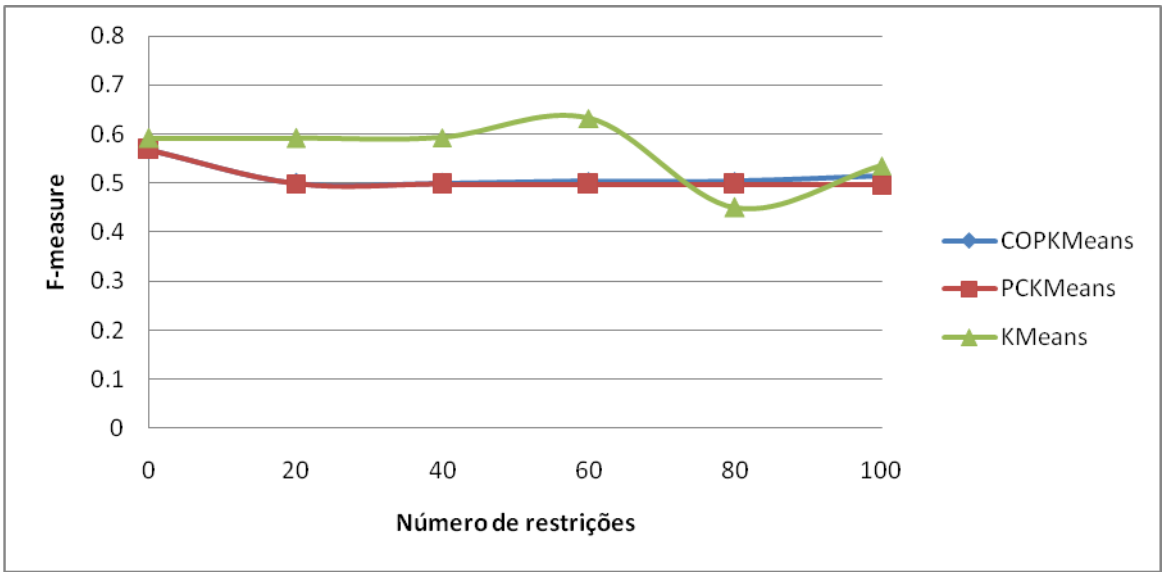


Gráfico 12: Conjunto Wine – F-measure – Comparação K-Means, Cop-Kmeans e PCKMeans

Na análise da medida de validação *F-measure*, para o conjunto de dados *Íris*, o algoritmo semi-supervisionado *PCK-Means* teve um melhor desempenho e, para o conjunto de dados *Wine*, o algoritmo não supervisionado teve melhor desempenho.

A Tabela 26 apresenta o desvio padrão para o Cop-Kmeans.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.49992672	0.054840425
0.49992672	0.054840425
0.503862531	0.053188996
0.503862531	0.053188996
0.514713239	0.052516264

Tabela 26: Conjunto Wine – Cop-Kmeans - Desvio Padrão

A Tabela 27 apresenta o desvio padrão para o PCKmeans.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.498359755	0.056003708
0.497521235	0.056737536
0.497521235	0.056737536
0.497521235	0.056737536
0.496007234	0.05796423

Tabela 27: Conjunto Wine – PCKmeans - Desvio Padrão

O algoritmo proposto por *Huang & Pan* foi executado utilizando as mesmas condições do conjunto de dados *Íris*. Os Gráfico 13 e 14 mostram o desempenho deste algoritmo com k_1 = 0, utilizando as medidas *F-measure* e BHI, respectivamente.

Note que para ambos os conjuntos de dados (*Íris* e *Wine*) e para ambas as medidas de validação, o algoritmo de *Huang & Pan* teve comportamento similar.

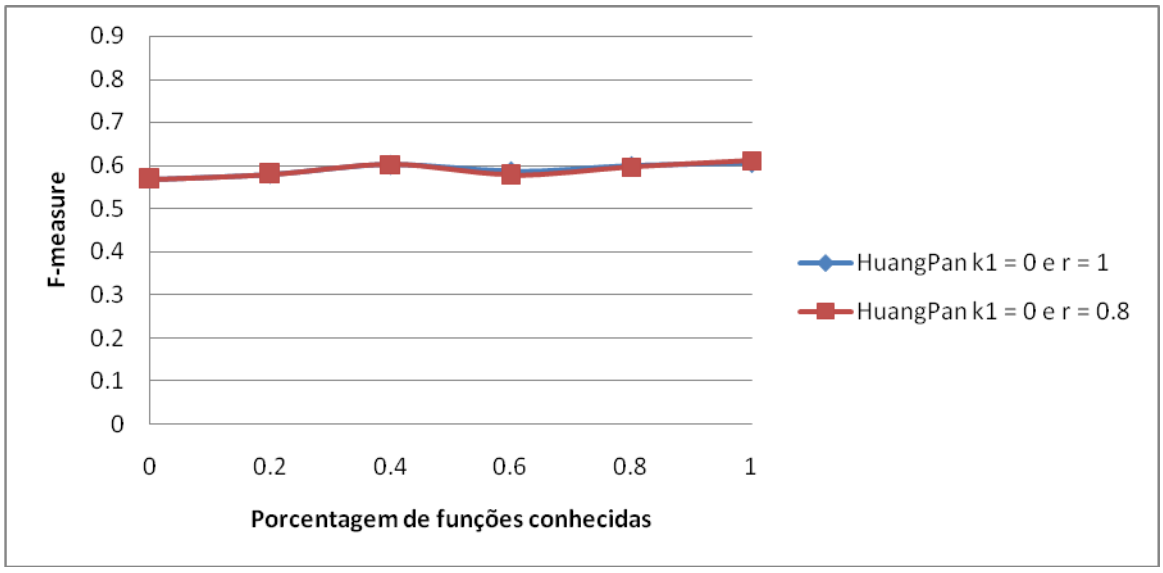


Gráfico 13: Conjunto Wine – F-measure – Método Huang & Pan para k_1 = 0

A Tabela 28 apresenta o desvio padrão para o Método *Huang & Pan* para $k_1 = 0$ e $r = 1$.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.579824975	0.059008724
0.601548662	0.037761953
0.586621985	0.034448568
0.598435964	0.040273198
0.604799343	0.035725204

Tabela 28: Conjunto Wine – Huang & Pan - Desvio Padrão

A Tabela 29 apresenta o desvio padrão para o Método *Huang & Pan* para $k_1 = 0$ e $r = 0.8$.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.57922345	0.060701165
0.602858017	0.036052567
0.577719605	0.040240587
0.596201288	0.038414624
0.610583735	0.027945513

Tabela 29: Conjunto Wine – Huang & Pan - Desvio Padrão

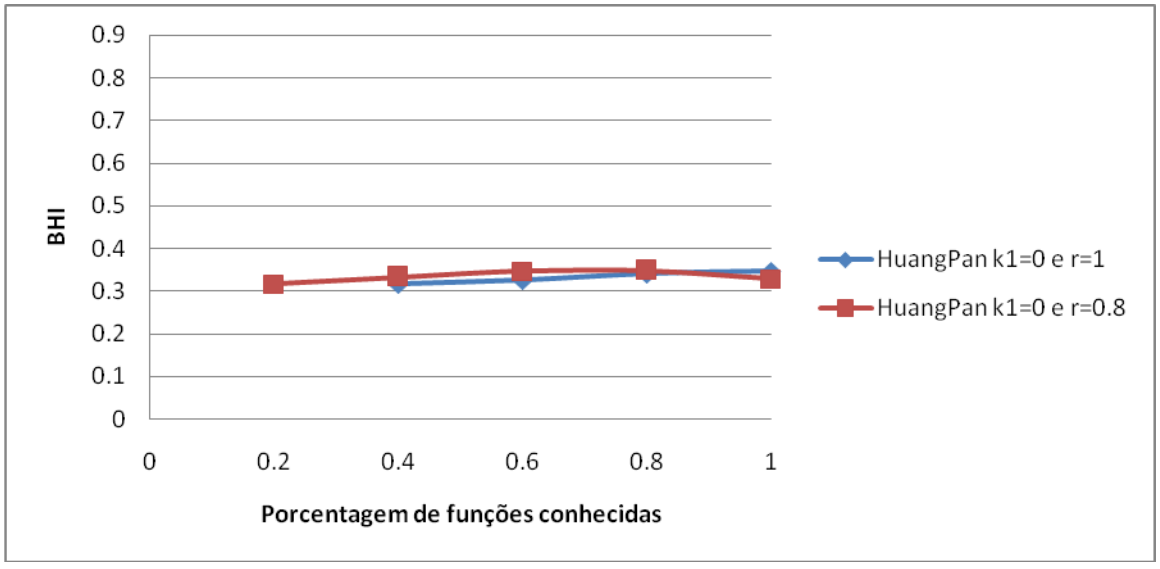


Gráfico 14: Conjunto Wine – BHI – Método Huang & Pan para $k_1 = 0$

A Tabela 30 apresenta o desvio padrão para o método *Huang & Pan* para $k_1 = 0$ e $r = 1$.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.316218981	0.080041577
0.325924562	0.061189252
0.341202824	0.02997134
0.346260986	0.026908546
0.333779247	0.022966678

Tabela 30: Conjunto Wine – Huang & Pan - Desvio Padrão

A Tabela 31 apresenta o desvio padrão para o método *Huang & Pan* para $k_1 = 0$ e $r = 0.8$.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.316218981	0.080041577
0.332538773	0.06085317
0.347105668	0.02117884
0.34846993	0.025310123
0.328160137	0.006099932

Tabela 31: Conjunto Wine – Huang & Pan - Desvio Padrão

Os Gráficos 15 e 16 mostram o desempenho do método de *Huang & Pan* para $k_1 = 1$, utilizando as medidas de *F-measure* e BHI, respectivamente.

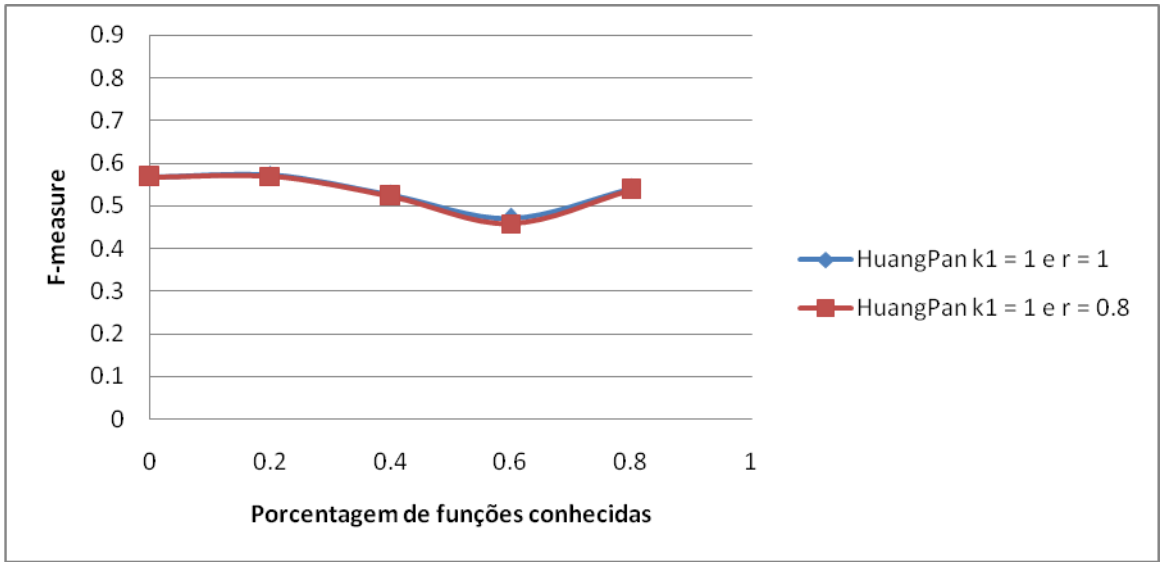


Gráfico 15: Conjunto Wine – F-measure - Método Huang & Pan para $k_1 = 1$

A Tabela 32 apresenta o desvio padrão para o método *Huang & Pan* para k_1 = 1 e r = 1.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.571897242	0.048093373
0.525249801	0.10605724
0.470737738	0.076460718
0.5400276	0.064346836

Tabela 32: Conjunto Wine – Huang & Pan - Desvio Padrão

A Tabela 33 apresenta o desvio padrão para o método *Huang & Pan* para k_1 = 1 e r = 0.8.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.568900111	0.051908506
0.522719653	0.113917858
0.458143661	0.064642969
0.536911532	0.074640351

Tabela 33: Conjunto Wine – Huang & Pan - Desvio Padrão

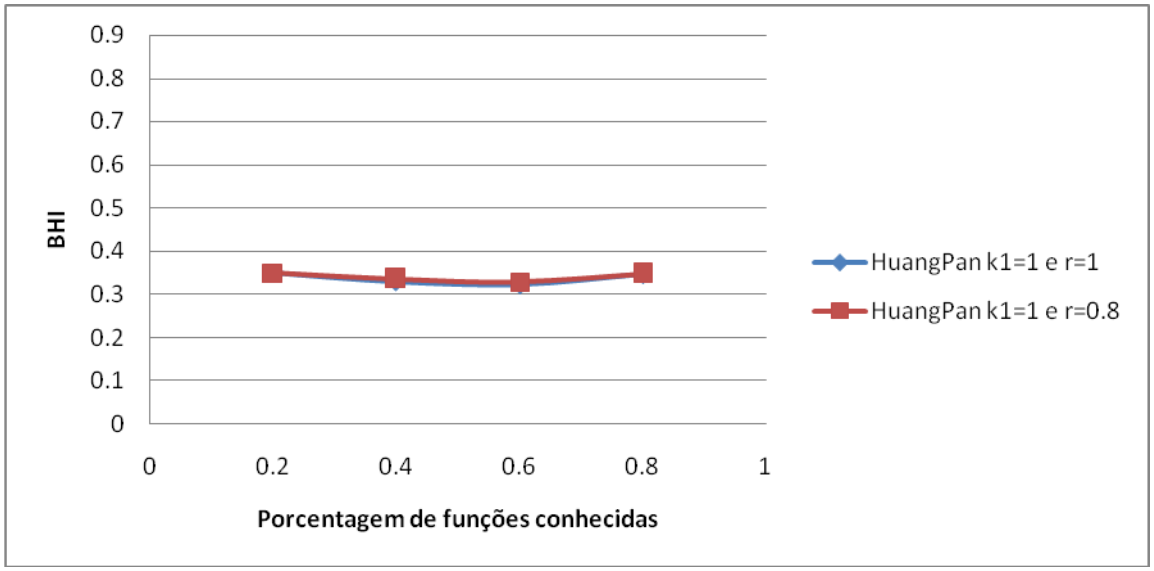


Gráfico 16**:** Conjunto Wine – BHI - Método Huang & Pan para k_1 = 1

A Tabela 34 apresenta o desvio padrão para o método *Huang & Pan* para $k_1 = 1$ e $r = 1$.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.349552315	0.080001181
0.329815032	0.033671395
0.322352249	0.062679512
0.346260986	0.026908546

Tabela 34: Conjunto Wine – Huang & Pan - Desvio Padrão

A Tabela 35 apresenta o desvio padrão para o método *Huang & Pan* para $k_1 = 1$ e $r = 0.8$.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.349552315	0.080001181
0.335234466	0.03367539
0.328255093	0.055856357
0.34846993	0.025310123

Tabela 35: Conjunto Wine – Huang & Pan - Desvio Padrão

Os Gráficos 17 e 18 mostram o desempenho dos algoritmos hierárquico e *Boratyn* utilizando as medidas *F-measure* e BHI, respectivamente.

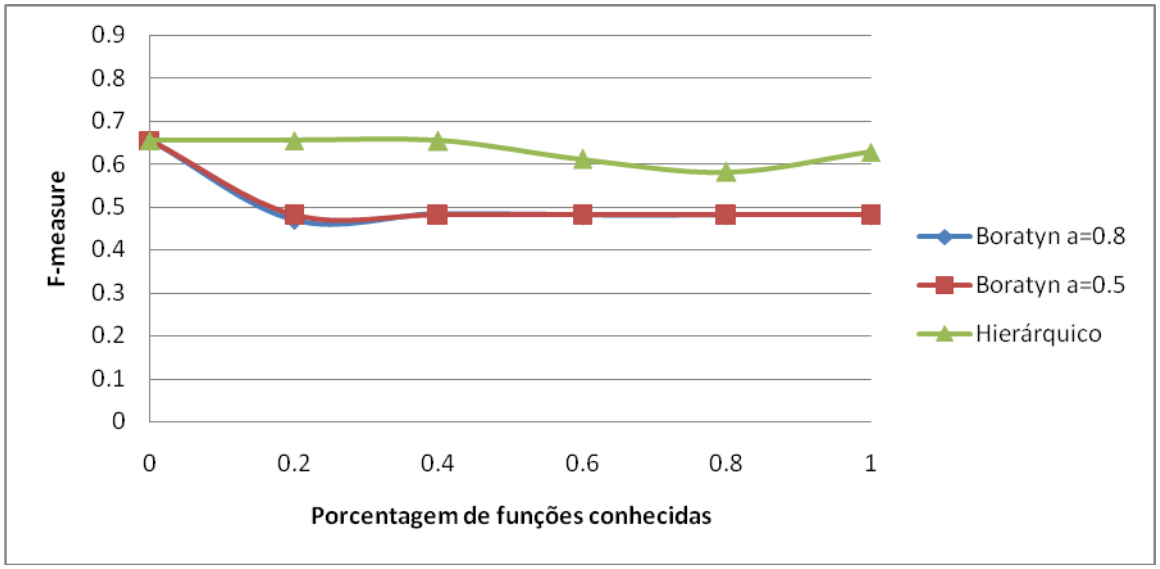


Gráfico 17: Conjunto Wine – F-measure – Método Boratyn

Para o algoritmo Boratyn, nas condições $\lambda = 0.5$ e $\lambda = 0.8$, as curvas no gráfico tiveram comportamento similares apresentando uma sobreposição de valores.

Na subseção anterior foi apresentado que para o conjunto de dados *Íris*, o algoritmo semi-supervisionado, *Boratyn*, apresentou melhor desempenho em relação ao algoritmo não supervisionado, hierárquico. Já para o conjunto de dados *Wine*, o resultado foi o inverso, umas vez que o algoritmo não supervisionado apresentou melhor desempenho em relação ao algoritmo semi-supervisionado.

A Tabela 36 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.8$.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.469456784	0.027783253
0.483640876	0.030751151
0.480261437	0.027544499
0.480261437	0.027544499
0.480261437	0.027544499

Tabela 36: Conjunto Wine – Boratyn - Desvio Padrão

A Tabela 37 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.5$.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.480261437	0.027544499
0.480261437	0.027544499
0.480261437	0.027544499
0.480261437	0.027544499
0.480261437	0.027544499

Tabela 37: Conjunto Wine –Boratyn - Desvio Padrão

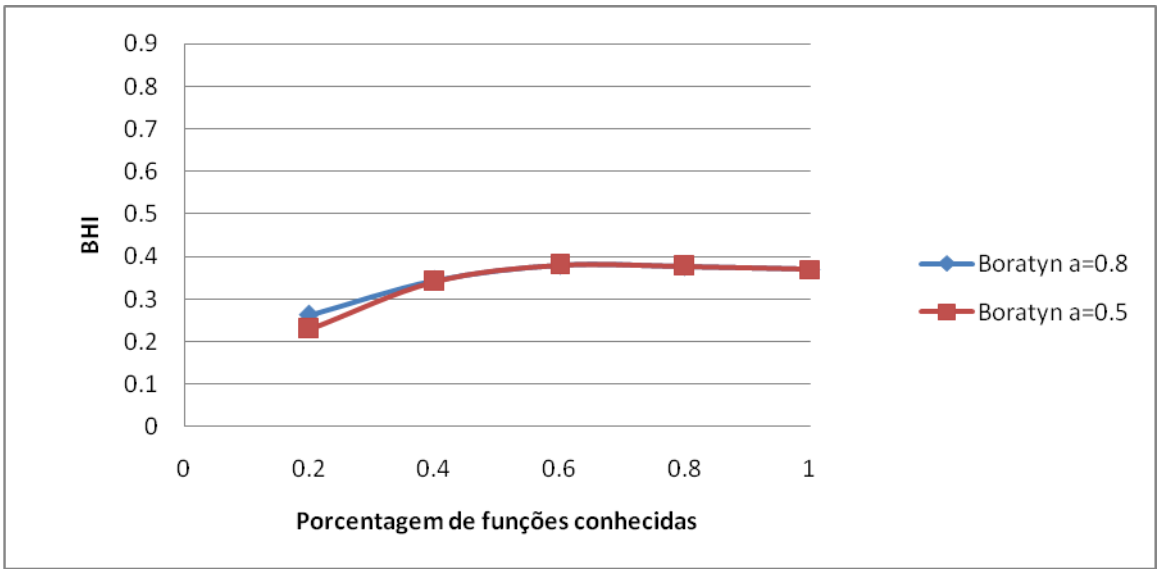


Gráfico 18: Conjunto Wine – BHI – Método Boratyn

A Tabela 38 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.8$.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.263153718	0.141594982
0.344141357	0.068188702
0.380645455	0.011617829
0.377452008	0.019176548
0.369357984	0.025658422

Tabela 38: Conjunto Wine – Boratyn - Desvio Padrão

A Tabela 39 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.5$.

Conjunto de dados *Wine*	
Média	Desvio Padrão
0.229820384	0.122792916
0.341735481	0.066383505
0.380645455	0.011617829
0.377452008	0.019176548
0.369357984	0.025658422

Tabela 39: Conjunto Wine –Boratyn - Desvio Padrão

Como pode ser observado no Gráfico 10, ambos os algoritmos semi-supervisionado *Seeded-K-Means* e *Constrained-K-Means*, tiveram uma melhor performance em relação ao algoritmo não supervisionado, *K-Means*. Sendo que o *Contrained-K-Means* desempenhou um pouco melhor do que o *Seeded-K-Means* (Gráficos 10 e 11).

Os algoritmos semi-supervisionados baseados em restrições, no geral, não tiveram um melhor desempenho em relação ao *K-Means* (Gráfico 12). O algoritmo proposto por *Huang & Pan* (Gráficos 13 a 14) teve melhor desempenho quando comparado ao *K-Means*, para $k_1 = 0$, para o caso em que $k_1 = 1$, o algoritmo não supervisionado apresentou melhor desempenho.

O algoritmo proposto por *Boratyn*, também não apresentou um melhor desempenho em relação ao agrupamento hierárquico (Gráfico 17).

6.6.3 Conjunto de dados: *Yeast1*

Os Gráficos 19 e 20 mostram o desempenho obtido pelos algoritmos baseados em sementes e algoritmo *K-means* para este conjunto de dados, aplicando as medidas *F-measure* e BHI, respectivamente.

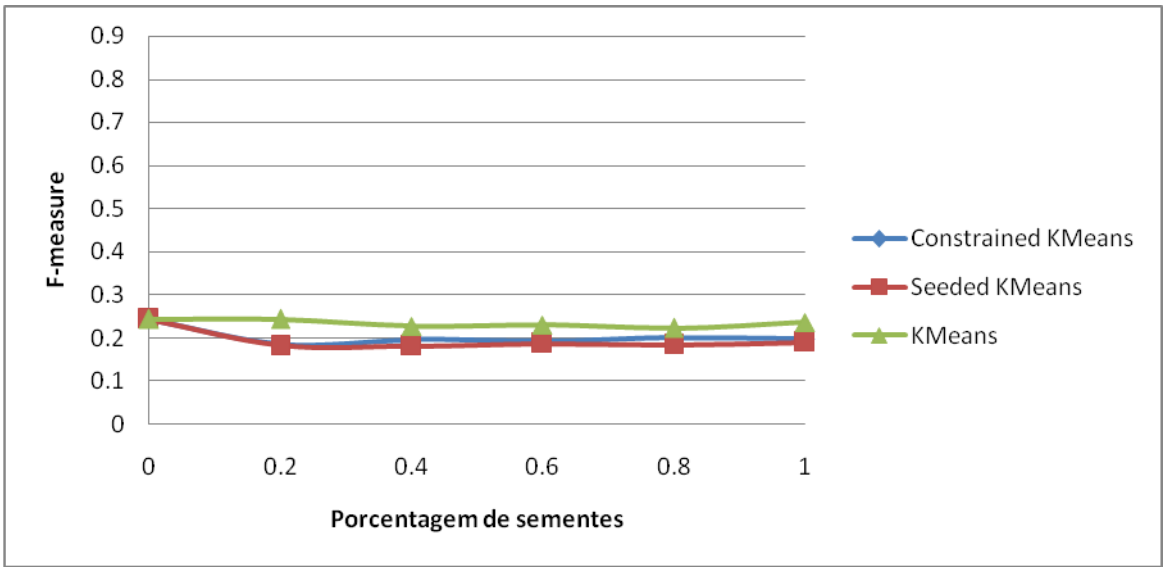


Gráfico 19: Conjunto Yeast1 – F-measure - Comparação K-Means, Seeded-Kmeans e Constrained-Kmeans

A Tabela 40 apresenta o desvio padrão para o *Kmeans*.

Conjunto de dados *Yeast1*	
Média	Desvio Padrão
0.232104063	0.007574261

Tabela 40: Conjunto Yeast1 – Kmeans - Desvio Padrão

A Tabela 41 apresenta o desvio padrão para o *Seeded-Kmeans*.

Conjunto de dados *Yeast1*	
Média	Desvio Padrão
0.182164697	0.017485006
0.180831206	0.019931387
0.186496629	0.016199823
0.184819148	0.006904154
0.190474485	0.020646269

Tabela 41: Conjunto *Yeast1 – Seeded-Kmeans* - Desvio Padrão

A Tabela 42 apresenta o desvio padrão para o *Constrained-Kmeans*.

Conjunto de dados *Yeast1*	
Média	Desvio Padrão
0.185162462	0.014646251
0.195962686	0.034218255
0.192600989	0.023521256
0.200809307	0.027569614
0.197991265	0.030980075

Tabela 42: Conjunto *Yeast1 – Constrained-Kmeans* - Desvio Padrão

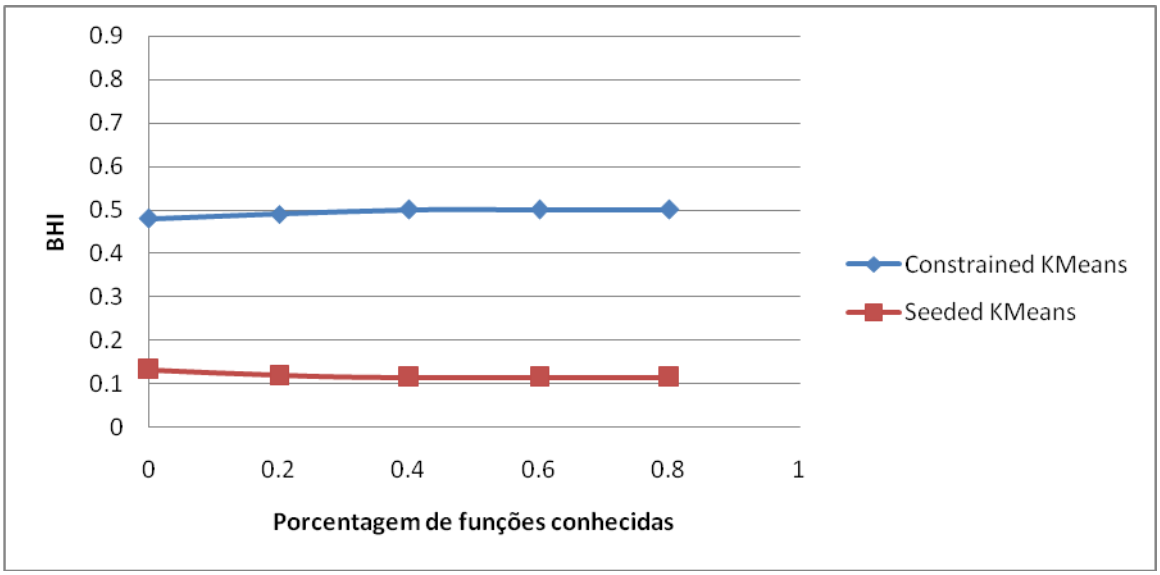


Gráfico 20: Conjunto *Yeast1* – BHI - Comparação *Seeded-Kmeans* e *Constrained-Kmeans*

A Tabela 43 apresenta o desvio padrão para o *Seeded-Kmeans*.

Conjunto de dados *Yeast1*	
Média	Desvio Padrão
0.132888836	0.019812311
0.118837962	0.007463612
0.114461688	0.004580918
0.115462684	0.006739011
0.115059014	0.003975169

Tabela 43: Conjunto *Yeast1* – *Seeded-Kmeans* - Desvio Padrão

A Tabela 44 apresenta o desvio padrão para o *Constrained-Kmeans*.

Conjunto de dados *Yeast1*	
Média	Desvio Padrão
0.48	0.04
0.49	0.02
0.5	0
0.5	0
0.5	0

Tabela 44: Conjunto *Yeast1* – *Constrained-Kmeans* - Desvio Padrão

O Gráfico 21 mostra o desempenho dos algoritmos baseados em restrições, utilizando o *F-measure*:

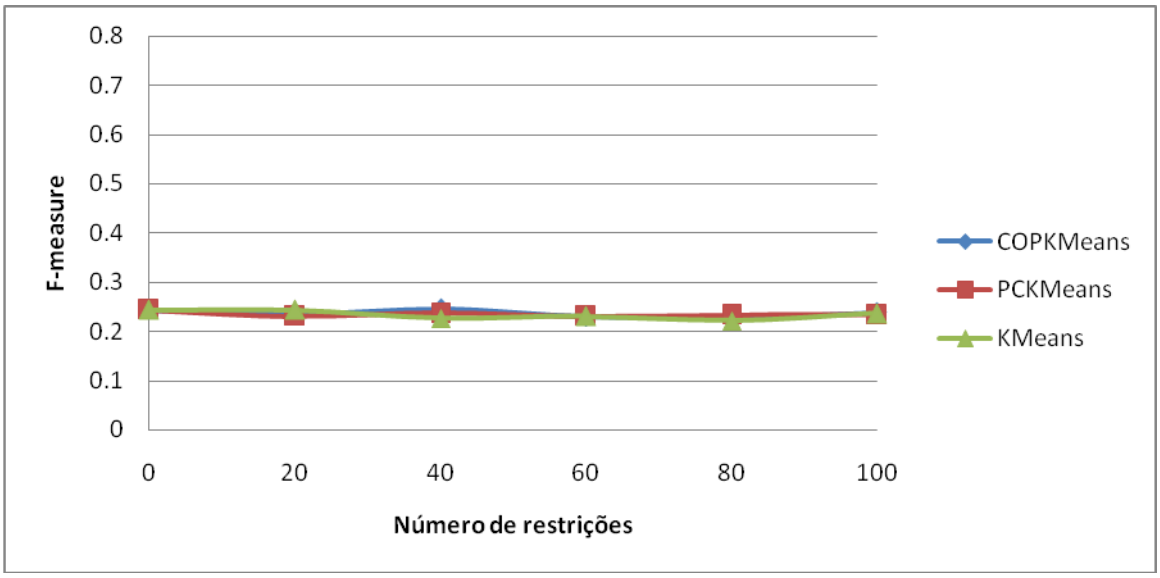


Gráfico 21: Conjunto *Yeast1* – *F-measure* - *Comparação K-Means, Cop-Kmeans* e *PCKMeans*

A Tabela 45 apresenta o desvio padrão para o *Cop-Kmeans*.

Conjunto de dados *Yeast1*	
Média	Desvio Padrão
0.234578696	0.010190598
0.246108331	0.019516872
0.229499957	0.010998573
0.228077052	0.008033719
0.238301594	0.020257739

Tabela 45: Conjunto *Yeast1* – *Cop-Kmeans* - Desvio Padrão

A Tabela 46 apresenta o desvio padrão para o *PCKmeans*.

Conjunto de dados *Yeast1*	
Média	Desvio Padrão
0.230058666	0.013764941
0.23666973	0.008787635
0.230146903	0.007092904
0.233561541	0.003266838
0.233926814	0.013351536

Tabela 46: Conjunto *Yeast1* –*PCKmeans* - Desvio Padrão

A Tabela 47 mostra os resultados obtidos pela execução do algoritmo proposto por *Huang & Pan* com $k_1 = 0$ e $k_1 = 1$, aplicando as medidas de *F-measure* e BHI.

Conjunto de dados *Yeast1*			
		$r = 1$	$r = 0{,}8$
F-measure	$k_1 = 0$	0.243601	0.239713
	$k_1 = 1$	0.241374	0.236614
BHI	$k_1 = 0$	0.090506	0.239895
	$k_1 = 1$	0.090506	0.225401

Tabela 47: Conjunto Yeast1 – Método Huang & Pan

Os Gráficos 22 e 23 mostram o desempenho obtido pela execução dos algoritmos hierárquico e *Boratyn*, aplicando as medidas de validação *F-measure* e BHI, respectivamente.

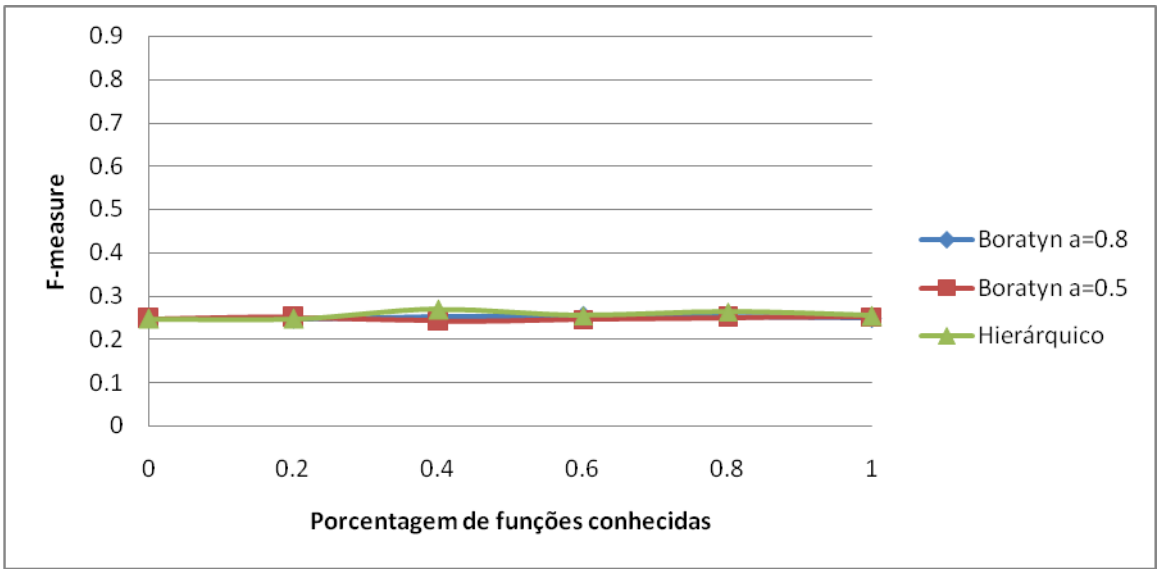


Gráfico 22**:** Conjunto *Yeast1* – *F-measure* - Método *Boratyn*

A Tabela 48 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.8$.

Conjunto de dados *Yeast1*	
Média	Desvio Padrão
0.246791491	0.014448132
0.252005306	0.008527318
0.253016862	0.015940827
0.253089684	0.015408053
0.24918337	0.012298373

Tabela 48: Conjunto *Yeast1* – *Boratyn* - Desvio Padrão

A Tabela 49 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.5$.

Conjunto de dados *Yeast1*	
Média	Desvio Padrão
0.250568579	0.010336444
0.242386343	0.019508625
0.245313819	0.016086954
0.250677518	0.013280191
0.251773329	0.00751301

Tabela 49: Conjunto *Yeast1* –*Boratyn* - Desvio Padrão

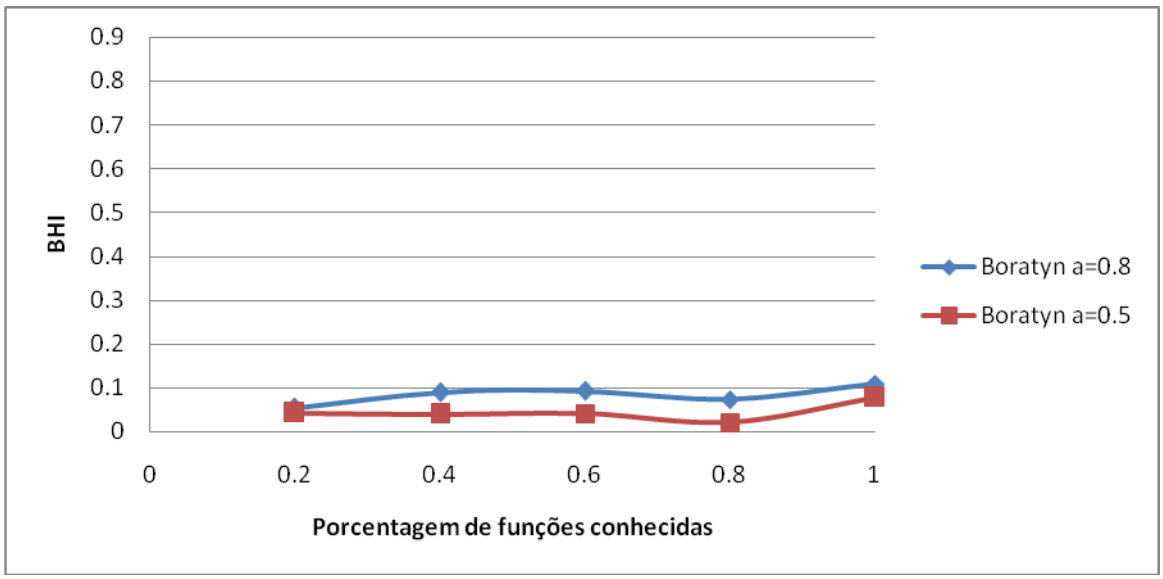


Gráfico 23: Conjunto *Yeast1* – BHI - Método *Boratyn*

A Tabela 50 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.8$.

Conjunto de dados *Yeast1*	
Média	Desvio Padrão
0.054066336	0.08100309
0.089936228	0.085512382
0.092312683	0.052876731
0.0732576	0.04457384
0.109024585	0.068220646

Tabela 50: Conjunto *Yeast1* – *Boratyn* - Desvio Padrão

A Tabela 51 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.5$.

Conjunto de dados *Yeast1*	
Média	Desvio Padrão
0.043372315	0.037673608
0.03986959	0.009537032
0.040910313	0.024979309
0.021236606	0.01335456
0.077217029	0.069888962

Tabela 51: Conjunto *Yeast1 –Boratyn* - Desvio Padrão

Como pode ser observado no Gráfico 19, ambos os algoritmos semi-supervisionado *Seeded-K-Means* e *Constrained-K-Means*, tiveram performance similares em relação ao algoritmo não supervisionado, *K-Means*. Sendo que o *Contrained-K-Means* desempenhou um pouco melhor do que o *Seeded-K-Means* (Gráficos 20).

Os algoritmos semi-supervisionados baseados em restrições (Gráfico 21) e o algoritmo proposto por *Boratyn* (Gráfico 22) tiveram desempenhos bem similares em relação ao *K-Means.*

6.6.4 Conjunto de dados: *Yeast2*

Neste conjunto de dados foram aplicados somente os dois algoritmos que permitem que um dado gene pertença a mais de uma classe. São eles: método de *Huang & Pan* e método *Boratyn.* As condições dos experimentos foram as mesmas utilizadas para os outros conjuntos de dados.

A Tabela 52 mostra os resultados obtidos pela execução do algoritmo proposto por *Huang & Pan* com $k_1 = 0$ e $k_1 = 1$, aplicando as medidas de *F-measure* e BHI.

Conjunto de dados *Yeast2*			
		$r = 1$	$r = 0{,}8$
F-measure	$k_1 = 0$	0.242268	0.230349
	$k_1 = 1$	0.239594	0.227733
BHI	$k_1 = 0$	0.093469	0.234695
	$k_1 = 1$	0.094251	0.227733

Tabela 52: Conjunto *Yeast2* – Método *Huang & Pan*

Os Gráficos 24 e 25 mostram os resultados obtidos com as execuções do algoritmo *Boratyn* aplicando as medidas *F-measure* e BHI, respectivamente.

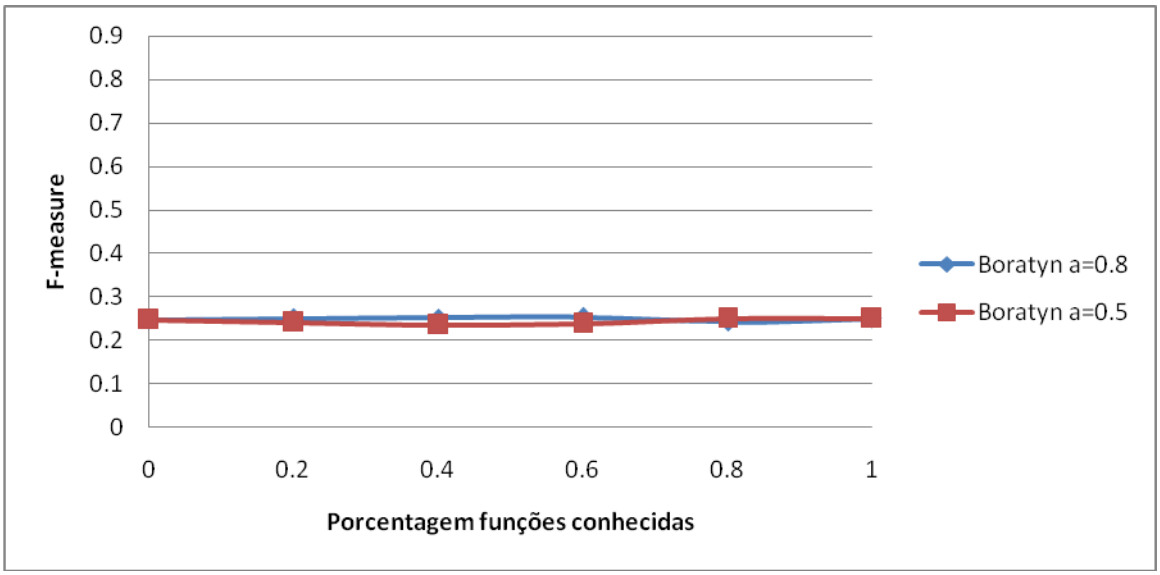


Gráfico 24: Conjunto *Yeast2* – *F-measure* - Método *Boratyn*

A Tabela 53 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.8$.

Conjunto de dados *Yeast2*	
Média	Desvio Padrão
0.250122099	0.013721309
0.251463848	0.013840679
0.252703981	0.009287892
0.242544391	0.019191895
0.249668545	0.010723132

Tabela 53: Conjunto *Yeast2* – *Boratyn* - Desvio Padrão

A Tabela 54 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.5$.

Conjunto de dados *Yeast2*	
Média	Desvio Padrão
0.241946865	0.018329227
0.2353305	0.017283799
0.238488103	0.022182576
0.250722328	0.011991938
0.250722328	0.011991938

Tabela 54: Conjunto *Yeast2* –*Boratyn* - Desvio Padrão

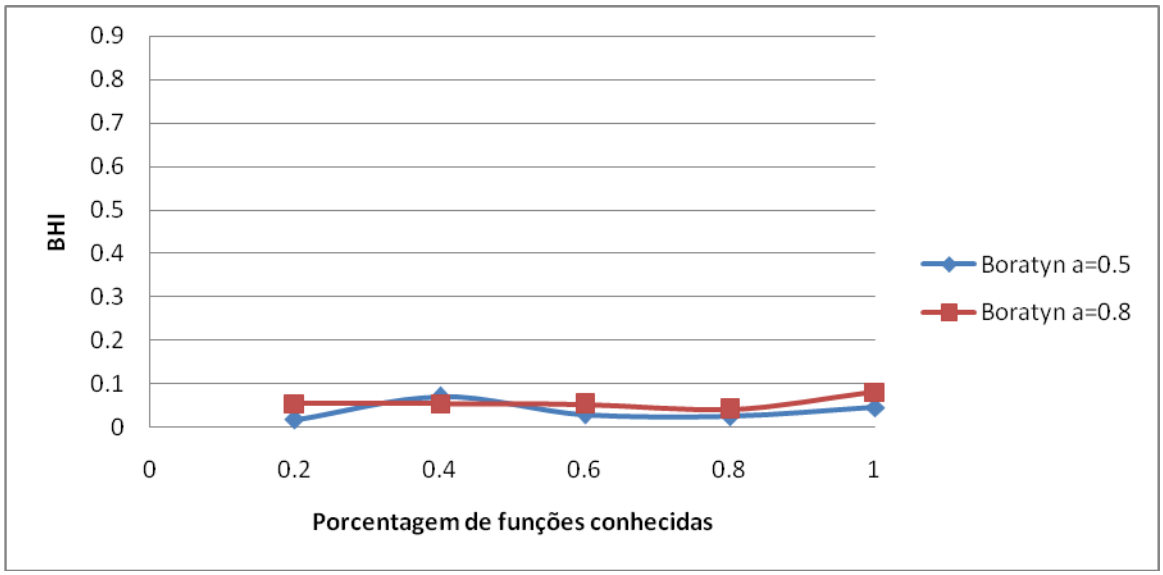


Gráfico 25: Conjunto *Yeast2* – *F-measure* - Método *Boratyn*

A Tabela 55 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.8$.

Conjunto de dados *Yeast2*	
Média	Desvio Padrão
0.054587412	0.048928219
0.054811593	0.054027539
0.053628545	0.022348248
0.041043655	0.00989664
0.081703966	0.053035416

Tabela 55: Conjunto *Yeast2* – *Boratyn* - Desvio Padrão

A Tabela 56 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.5$.

Conjunto de dados *Yeast2*	
Média	Desvio Padrão
0.017889104	0.006943767
0.070726545	0.044240337
0.029100242	0.014418218
0.024958256	0.013859839
0.046232012	0.035586977

Tabela 56: Conjunto *Yeast2* –*Boratyn* - Desvio Padrão

6.6.5 Conjunto de dados: *Yeast3*

Os Gráficos 26 e 27 mostram o desempenho obtido pelos algoritmos baseados em sementes e algoritmo *K-means* para este conjunto de dados, aplicando as medidas *F-measure* e BHI, respectivamente.

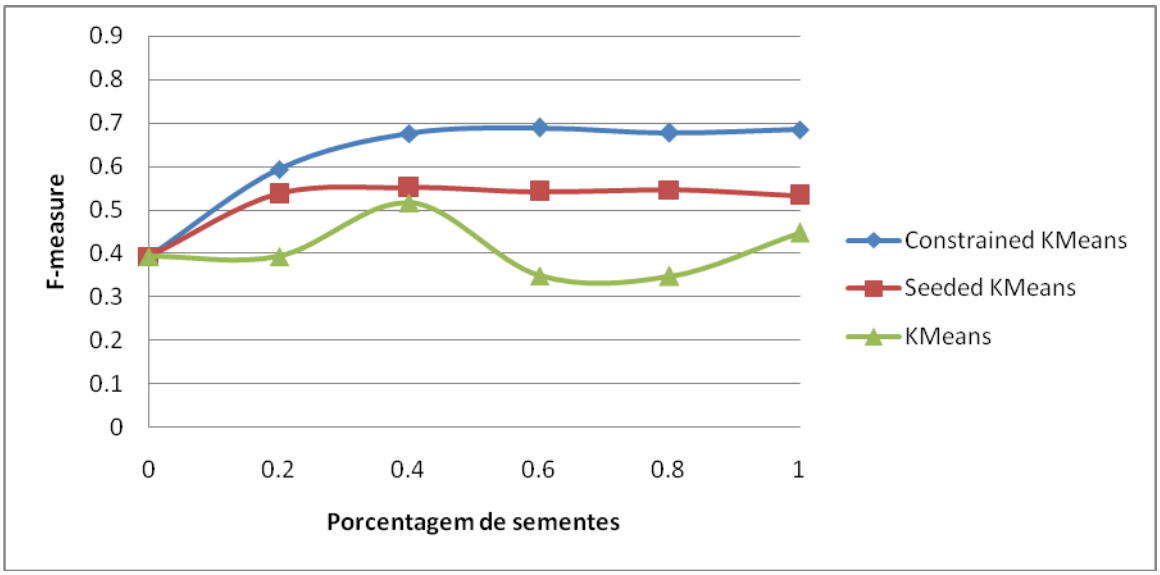


Gráfico 26: Conjunto *Yeast3 – F-measure* - Comparação *K-Means*, *Seeded-Kmeans* e *Constrained-Kmeans*

A Tabela 57 apresenta o desvio padrão para o *Kmeans*.

Conjunto de dados *Yeast3*	
Média	Desvio Padrão
0.411225791	0.641268892

Tabela 57: Conjunto Yeast3 – Kmeans - Desvio Padrão

A Tabela 58 apresenta o desvio padrão para o *Seeded-Kmeans*.

Conjunto de dados *Yeast3*	
Média	Desvio Padrão
0.536877696	0.03535656
0.55085589	0.031878696
0.541733514	0.03940877
0.546777245	0.030366403
0.532788058	0.033701343

Tabela 58: Conjunto *Yeast3 – Seeded-Kmeans* - Desvio Padrão

A Tabela 59 apresenta o desvio padrão para o *Constrained-Kmeans*.

Conjunto de dados *Yeast3*	
Média	Desvio Padrão
0.593811539	0.044996104
0.676008758	0.079898605
0.689263432	0.077003847
0.677977583	0.041983105
0.684217004	0.049928667

Tabela 59: Conjunto *Yeast3 – Constrained-Kmeans* - Desvio Padrão

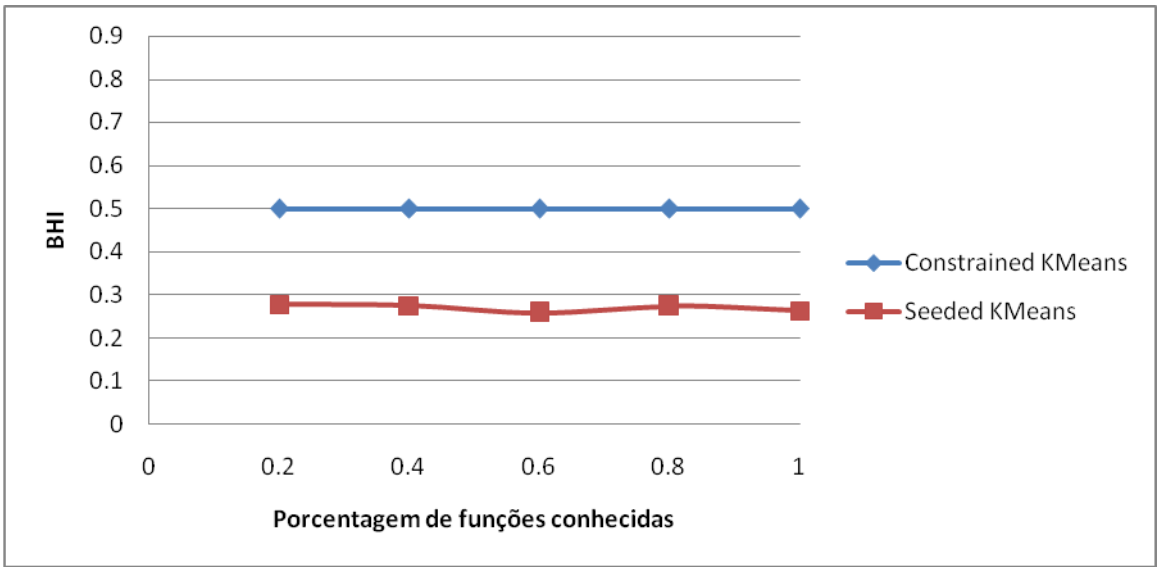


Gráfico 27: *Yeast3* – BHI - Comparação *Seeded-Kmeans* e *Constrained-Kmeans*

A Tabela 60 apresenta o desvio padrão para o *Seeded-Kmeans*.

Conjunto de dados *Yeast3*	
Média	Desvio Padrão
0.278226446	0.026427455
0.274558193	0.01855102
0.258777746	0.009235553
0.273723825	0.021413
0.263713672	0.017642306

Tabela 60: Conjunto *Yeast3* – *Seeded-Kmeans* - Desvio Padrão

A Tabela 61 apresenta o desvio padrão para o *Constrained-Kmeans*.

Conjunto de dados *Yeast3*	
Média	Desvio Padrão
0.5	0
0.5	0
0.5	0
0.5	0
0.5	0

Tabela 61: Conjunto *Yeast3* – *Constrained-Kmeans* - Desvio Padrão

O Gráfico 28 mostra o desempenho dos algoritmos baseados em restrições, utilizando o *F-measure*:

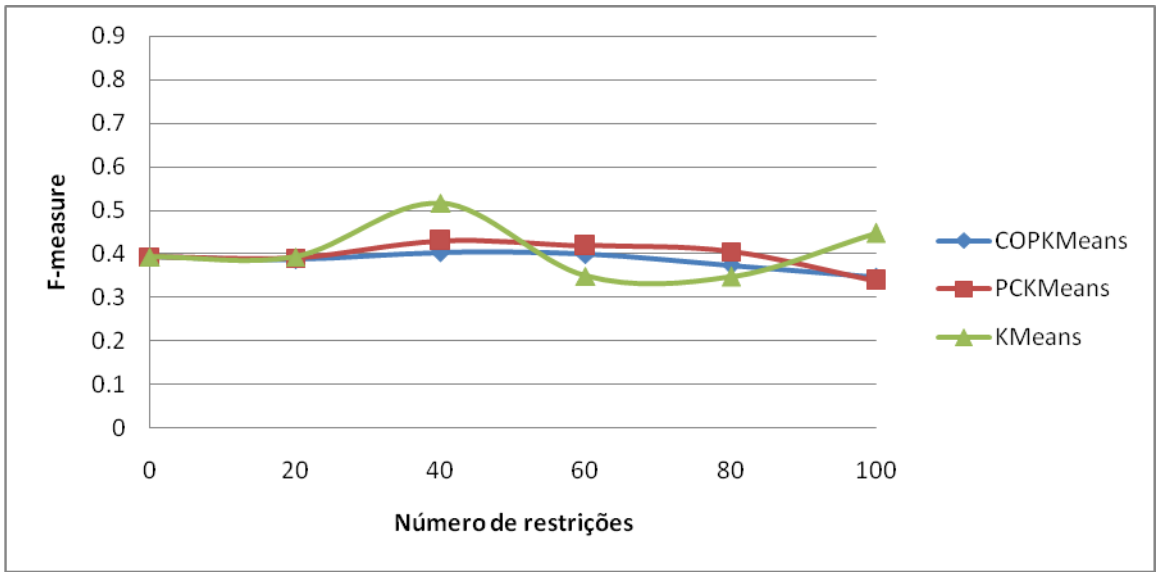


Gráfico 28: Conjunto *Yeast3* – *F-measure* - Comparação *K-Means*, *Cop-Kmeans* e *PCKMeans*

A Tabela 62 apresenta o desvio padrão para o *Cop-Kmeans*.

Conjunto de dados *Yeast3*	
Média	Desvio Padrão
0.387842071	0.060036051
0.40348735	0.049298136
0.399373885	0.061762547
0.373712305	0.025163565
0.345402764	0.108215434

Tabela 62: Conjunto *Yeast3* – *Cop-Kmeans* - Desvio Padrão

A Tabela 63 apresenta o desvio padrão para o *PCKmeans*.

Conjunto de dados *Yeast3*	
Média	Desvio Padrão
0.390631557	0.046635489
0.429827469	0.076077293
0.418653727	0.07926774
0.405669359	0.049700717
0.338626976	0.1119048

Tabela 63: Conjunto *Yeast3* – *PCKmeans* - Desvio Padrão

A Tabela 64 mostra os resultados obtidos pela execução do algoritmo proposto por *Huang & Pan* com $k_1 = 0$ e $k_1 = 1$, aplicando as medidas de *F-measure* e BHI.

Conjunto de dados *Yeast3*			
		$r = 1$	$r = 0{,}8$
F-measure	$k_1 = 0$	0.35466	0.378261
	$k_1 = 1$	0.337984	0.364626
BHI	$k_1 = 0$	0.278004	0.416318
	$k_1 = 1$	0.278004	0.416318

Tabela 64: Conjunto *Yeast3* – Método *Huang & Pan*

Os Gráficos 29 e 30 mostram os resultados obtidos com as execuções do algoritmo *Boratyn* aplicando as medidas *F-measure* e BHI, respectivamente.

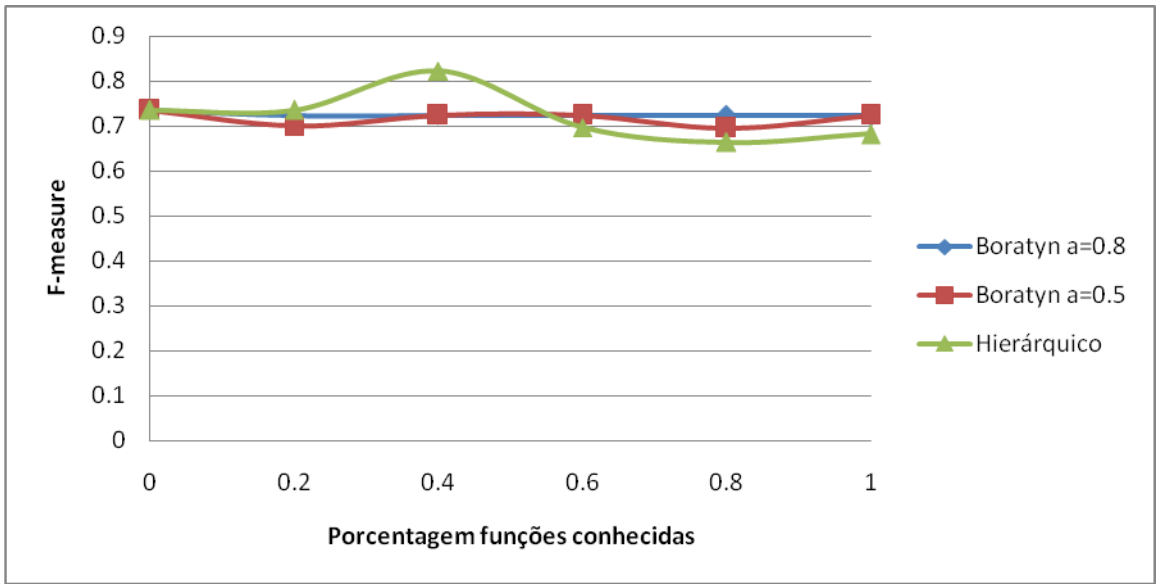


Gráfico 29: Conjunto *Yeast3* - *F-measure* – Método *Boratyn*

A Tabela 65 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.8$.

Conjunto de dados *Yeast3*	
Média	Desvio Padrão
0.723080635	0.058395524
0.723296262	0.058241271
0.723296262	0.058241271
0.723296262	0.058241271
0.721933244	0.058015502

Tabela 65: Conjunto *Yeast3* – *Boratyn* - Desvio Padrão

A Tabela 66 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.5$.

Conjunto de dados *Yeast3*	
Média	Desvio Padrão
0.699615638	0.087123933
0.723296262	0.058241271
0.723296262	0.058241271
0.695238299	0.094116689
0.723296262	0.058241271

Tabela 66: Conjunto *Yeast3* –*Boratyn* - Desvio Padrão

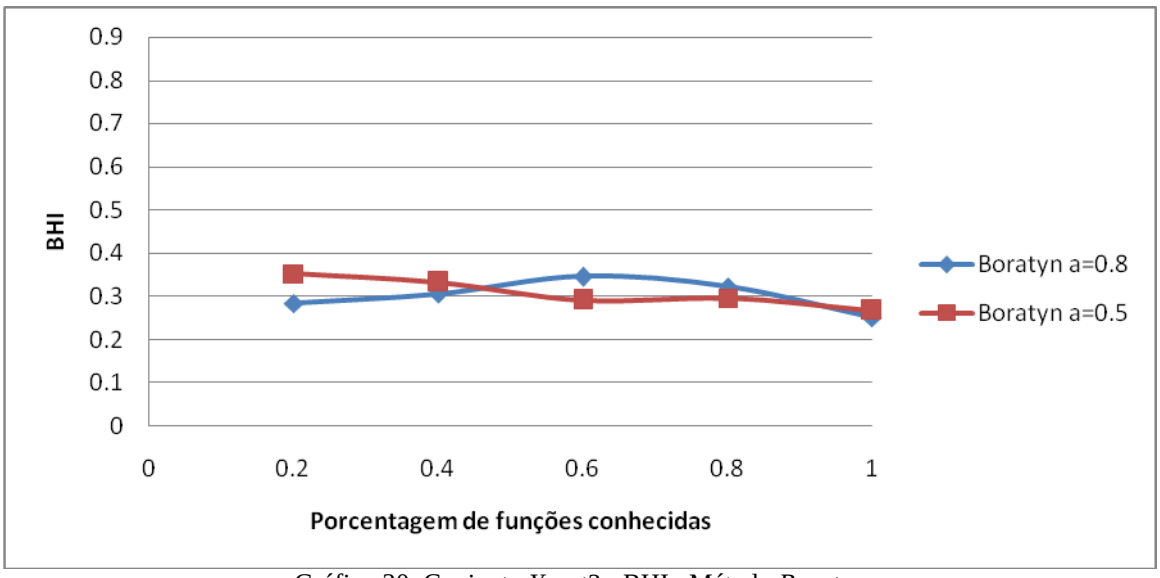


Gráfico 30: Conjunto *Yeast3* - BHI– Método *Boratyn*

A Tabela 67 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.8$.

Conjunto de dados *Yeast3*	
Média	Desvio Padrão
0.283953579	0.109971696
0.305539072	0.065008353
0.345699647	0.039558592
0.322851723	0.047824212
0.251233575	0.036374624

Tabela 67: Conjunto *Yeast3* – *Boratyn* - Desvio Padrão

A Tabela 68 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.5$.

Conjunto de dados *Yeast3*	
Média	Desvio Padrão
0.352671814	0.105556616
0.332485126	0.039963449
0.291913016	0.028526635
0.295873396	0.049095024
0.267043458	0.019865337

Tabela 68: Conjunto *Yeast3* –*Boratyn* - Desvio Padrão

Como pode ser observado no Gráfico 26, ambos os algoritmos semi-supervisionado *Seeded-K-Means* e *Constrained-K-Means*, tiveram uma melhor performance em relação ao algoritmo não supervisionado, *K-Means*, sendo que o *Contrained-K-Means* desempenhou um pouco melhor do que o *Seeded-K-Means* (Gráficos 26 e 27).

Os algoritmos semi-supervisionados baseados em restrições tiveram um desempenho mais constante em relação ao *K-Means* que teve resultados mais inconstante, sendo que o *PCKMeans* teve uma melhor performance que o *COPKmeans* (Gráfico 28).

O algoritmo proposto por *Boratyn*, também apresentou desempenho mais constante em relação ao agrupamento hierárquico (Gráfico 29).

6.6.6 Conjunto de dados: *Yeast4*

Os Gráficos 31 e 32 mostram o desempenho obtido pelos algoritmos baseados em sementes e algoritmo *K-means* para este conjunto de dados, aplicando as medidas *F-measure* e BHI, respectivamente.

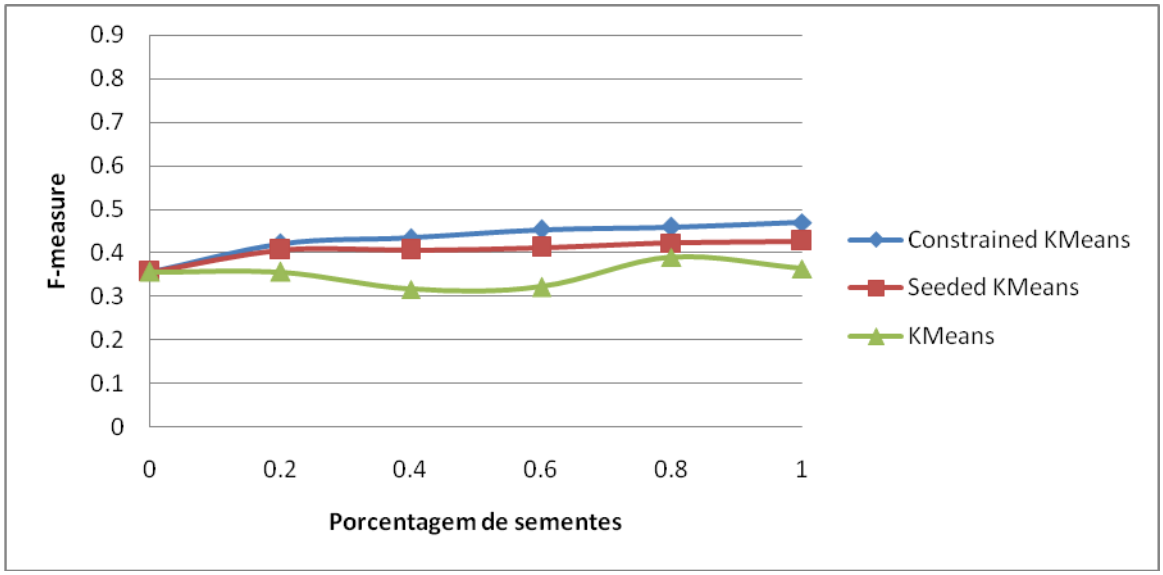


Gráfico 31: Conjunto *Yeast4* – *F-measure* - Comparação *K-Means*, *Seeded-Kmeans* e *Constrained-Kmeans*

A Tabela 69 apresenta o desvio padrão para o *Kmeans*.

Conjunto de dados *Yeast4*	
Média	Desvio Padrão
0.349958342	0.027383363

Tabela 69: Conjunto *Yeast4* – *Kmeans* - Desvio Padrão

A Tabela 70 apresenta o desvio padrão para o *Seeded-Kmeans*.

Conjunto de dados *Yeast4*	
Média	Desvio Padrão
0.407048921	0.03750568
0.406550353	0.037068212
0.412007174	0.032626811
0.422959753	0.048516905
0.426549072	0.047252793

Tabela 70: Conjunto *Yeast4* – *Seeded-Kmeans* - Desvio Padrão

A Tabela 71 apresenta o desvio padrão para o *Constrained-Kmeans*.

Conjunto de dados *Yeast4*	
Média	Desvio Padrão
0.420885462	0.034672245
0.43372251	0.043294173
0.453777304	0.036258757
0.459939539	0.04298096
0.469292388	0.053261246

Tabela 71: Conjunto *Yeast4* – *Constrained-Kmeans* - Desvio Padrão

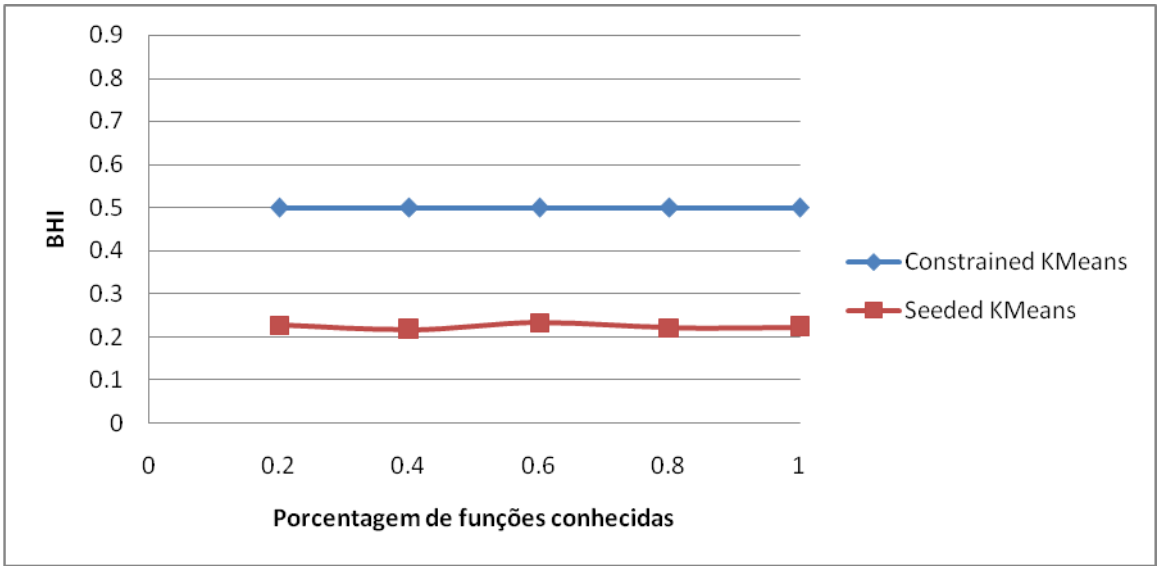


Gráfico 32: Conjunto *Yeast4* – BHI - Comparação *Seeded-Kmeans* e *Constrained-Kmeans*

A Tabela 72 apresenta o desvio padrão para o *Seeded-Kmeans*.

Conjunto de dados *Yeast4*	
Média	Desvio Padrão
0.227096257	0.021417855
0.21695684	0.024946172
0.233979456	0.014991665
0.221628841	0.021609847
0.222132203	0.01914508

Tabela 72: Conjunto *Yeast4* – *Seeded-Kmeans* - Desvio Padrão

A Tabela 73 apresenta o desvio padrão para o *Constrained-Kmeans*.

Conjunto de dados *Yeast4*	
Média	Desvio Padrão
0.5	0
0.5	0
0.5	0
0.5	0
0.5	0

Tabela 73: Conjunto *Yeast4* – *Constrained-Kmeans* - Desvio Padrão

O Gráfico 33 mostra o desempenho dos algoritmos baseados em restrições, utilizando o *F-measure*:

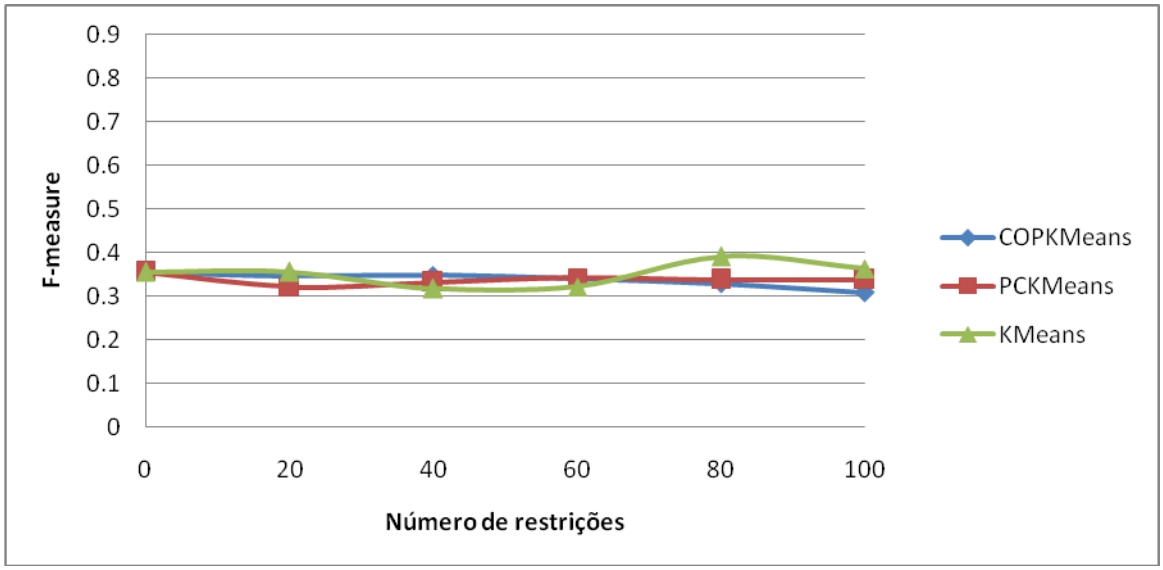


Gráfico 33: Conjunto *Yeast4* – *F-measure* - Comparação *K-Means, Cop-Kmeans* e *PCKMeans*

A Tabela 74 apresenta o desvio padrão para o *Cop-Kmeans*.

Conjunto de dados *Yeast4*	
Média	Desvio Padrão
0.346812121	0.032810153
0.348573162	0.035229406
0.340119697	0.020782685
0.32905127	0.033954005
0.308815934	0.02430939

Tabela 74: Conjunto *Yeast4* – *Cop-Kmeans* - Desvio Padrão

A Tabela 75 apresenta o desvio padrão para o *PCKmeans*.

Conjunto de dados *Yeast4*	
Média	Desvio Padrão
0.321425719	0.030164939
0.331909422	0.021755335
0.342063832	0.031553628
0.337556889	0.019254087
0.338548114	0.031674138

Tabela 75: Conjunto *Yeast4* – *PCKmeans* - Desvio Padrão

A Tabela 76 mostra os resultados obtidos pela execução do algoritmo proposto por *Huang & Pan* com $k_1 = 0$ e $k_1 = 1$, aplicando as medidas de *F-measure* e BHI.

Conjunto de dados *Yeast4*			
		$r = 1$	$r = 0{,}8$
F-measure	$k_1 = 0$	0.308114	0.313662
	$k_1 = 1$	0.300958	0.303513
BHI	$k_1 = 0$	0.213044	0.311071
	$k_1 = 1$	0.213044	0.311071

Tabela 76: Conjunto *Yeast4* – Método *Huang & Pan*

Os Gráficos 38 e 39 mostram os resultados obtidos com as execuções do algoritmo *Boratyn* aplicando as medidas *F-measure* e BHI, respectivamente.

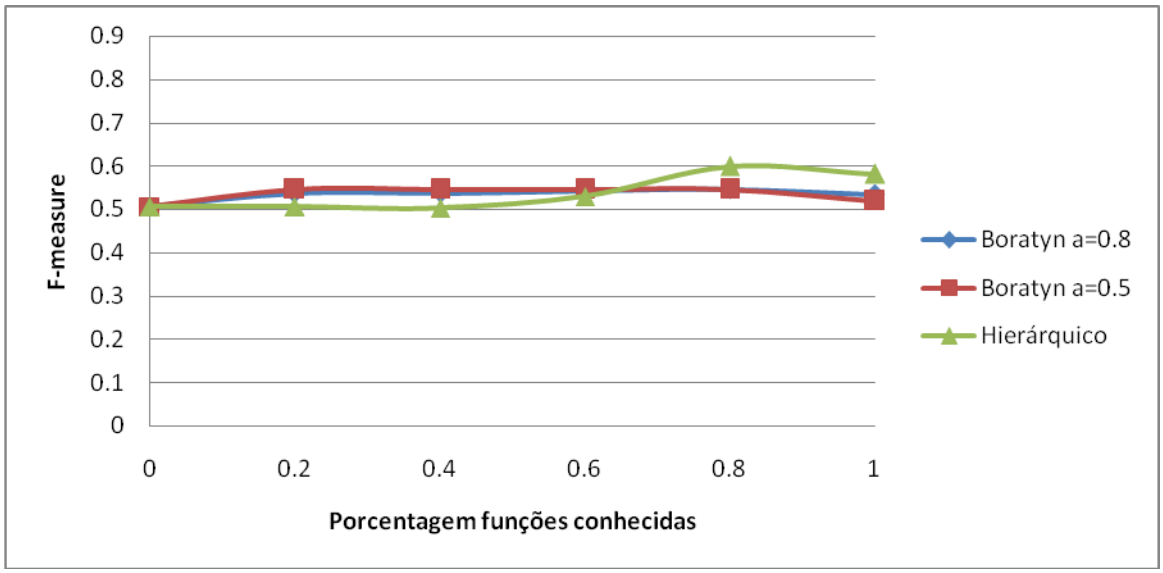


Gráfico 34: Conjunto Yeast4 - F-measure - Método Boratyn

A Tabela 77 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.8$.

Conjunto de dados *Yeast4*	
Média	Desvio Padrão
0.536820051	0.036971157
0.537022814	0.039258481
0.542899751	0.038801556
0.546154349	0.041247113
0.534856301	0.055579639

Tabela 77: Conjunto *Yeast4* – *Boratyn* - Desvio Padrão

A Tabela 78 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.5$.

Conjunto de dados *Yesat4*	
Média	Desvio Padrão
0.546154349	0.041247113
0.546154349	0.041247113
0.546154349	0.041247113
0.546154349	0.041247113
0.519526568	0.036413045

Tabela 78: Conjunto *Yeast4* –*Boratyn* - Desvio Padrão

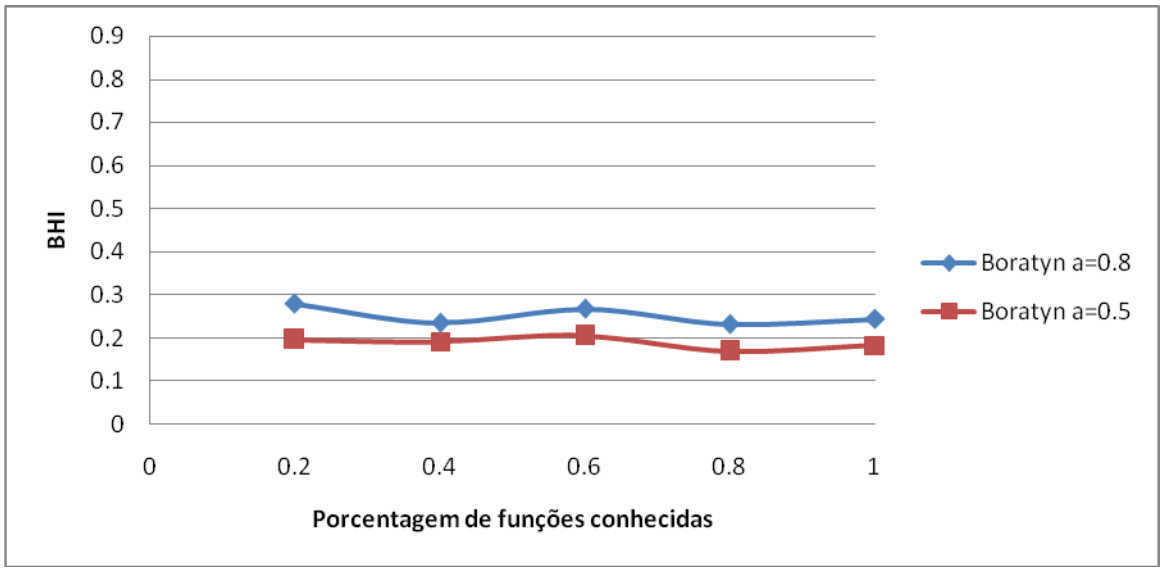


Gráfico 35: Conjunto *Yeast4* – BHI – Método *Boratyn*

A Tabela 79 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.8$.

Conjunto de dados *Yeast4*	
Média	Desvio Padrão
0.28007328	0.112213909
0.235350947	0.083817107
0.267398625	0.051482996
0.232365985	0.086138579
0.243775557	0.04463135

Tabela 79: Conjunto *Yeast4* – *Boratyn* - Desvio Padrão

A Tabela 80 apresenta o desvio padrão para o método *Boratyn* para $\lambda = 0.5$.

Conjunto de dados *Yesat4*	
Média	Desvio Padrão
0.196660713	0.076880148
0.191687609	0.023683538
0.20493163	0.025407403
0.16879264	0.017019592
0.183118634	0.042291875

Tabela 80: Conjunto *Yeast4* –*Boratyn* - Desvio Padrão

Como pode ser observado no Gráfico 31, ambos os algoritmos semi-supervisionado *Seeded-K-Means* e *Constrained-K-Means*, tiveram uma melhor performance em relação ao algoritmo não supervisionado, *K-Means*, sendo que o *Contrained-K-Means* desempenhou um pouco melhor do que o *Seeded-K-Means* (Gráficos 31 e 32).

Os algoritmos semi-supervisionados baseados em restrições tiveram desempenhos similares em relação ao *K-Means* (Gráfico 33).

O algoritmo proposto por *Boratyn*, também apresentou, no geral, um melhor desempenho em relação ao agrupamento hierárquico (Gráfico 38).

6.6.7 Consolidação dos dados

Nesta seção apresentaremos um resumo consolidado dos resultados apresentados anteriormente, agrupados por algoritmos.

6.6.7.1 F-measure

O Gráfico 36 representa os resultados obtidos para os todos os conjuntos de dados que foram aplicados o algoritmo *Seeded-Kmeans:*

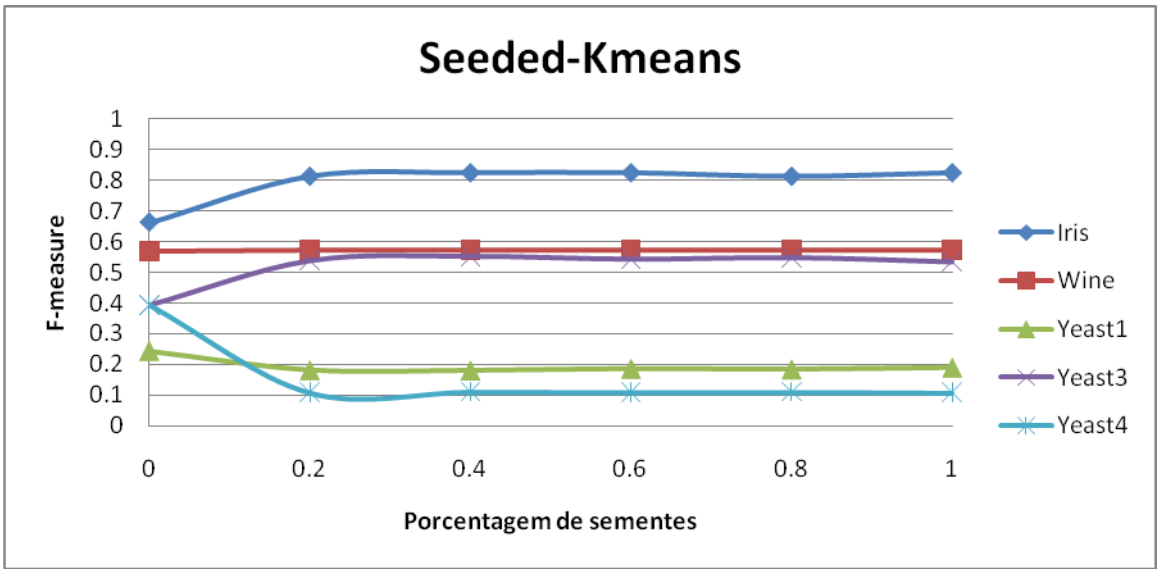


Gráfico 36: Resultados consolidados – Fmeasure - Seeded-K-Means

O Gráfico 37 representa os resultados obtidos para os todos os conjuntos de dados que foram aplicados o algoritmo *Constrained-Kmeans:*

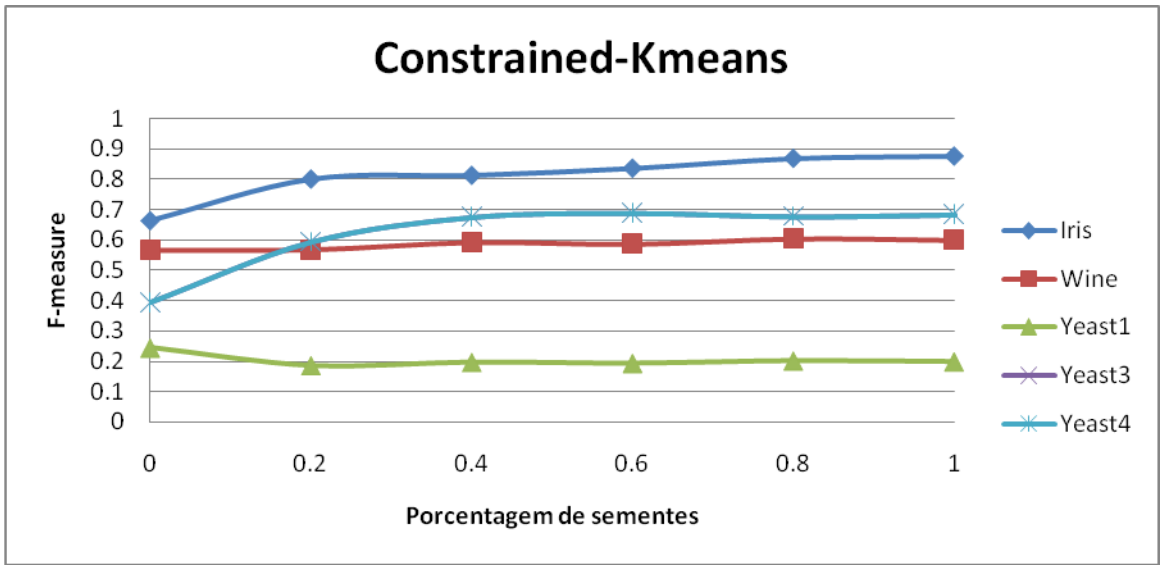


Gráfico 37: Resultados consolidados – Fmeasure - Constrained-K-Means

O Gráfico 38 representa os resultados obtidos para os todos os conjuntos de dados que foram aplicados o algoritmo *COPKmeans:*

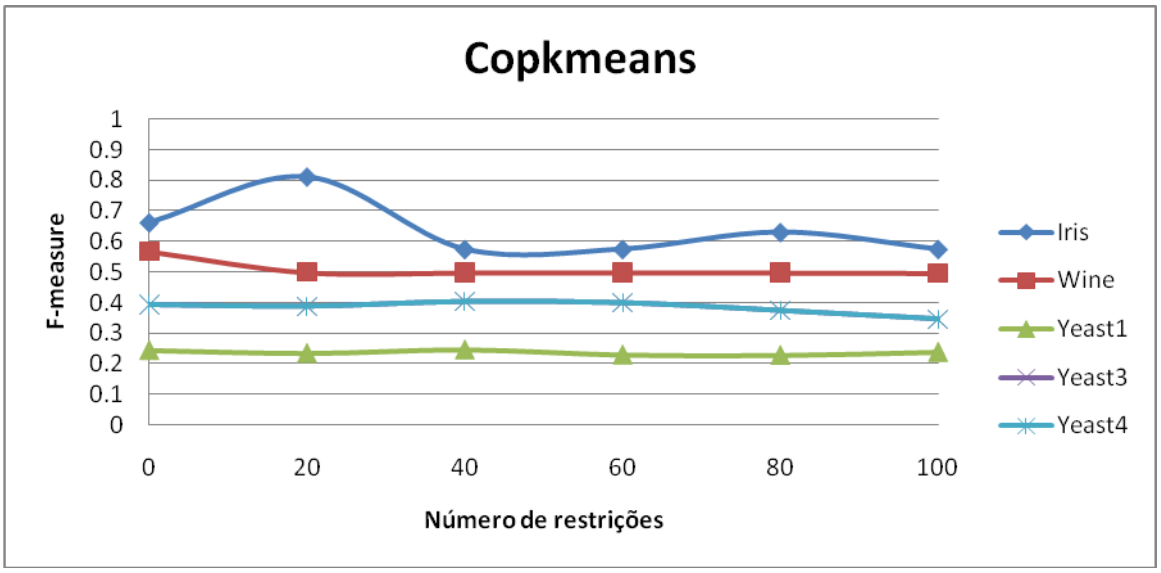


Gráfico 38: Resultados consolidados – Fmeasure - COPKMeans

O Gráfico 39 representa os resultados obtidos para os todos os conjuntos de dados que foram aplicados o algoritmo *PCKmeans:*

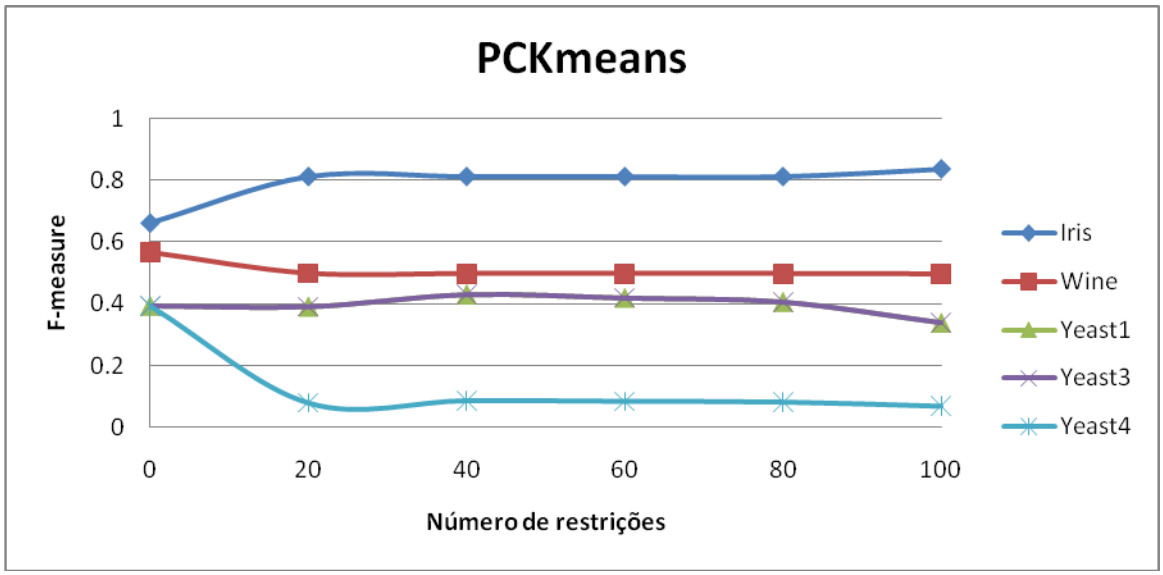


Gráfico 39: Resultados consolidados – Fmeasure - PCKMeans

O Gráfico 40 representa os resultados obtidos para os todos os conjuntos de dados que foram aplicados o algoritmo *Boratyn* quando r = 0.8:

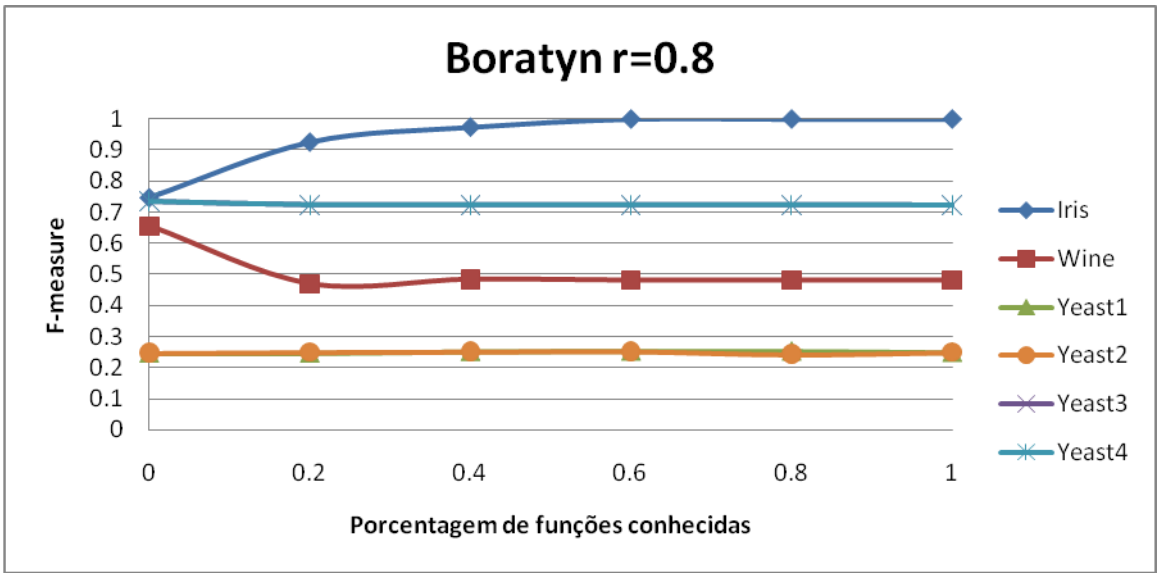


Gráfico 40: Resultados consolidados – Fmeasure - Boratyn (r = 0,8)

O Gráfico 41 representa os resultados obtidos para os todos os conjuntos de dados que foram aplicados o algoritmo *Boratyn* quando r = 0.5:

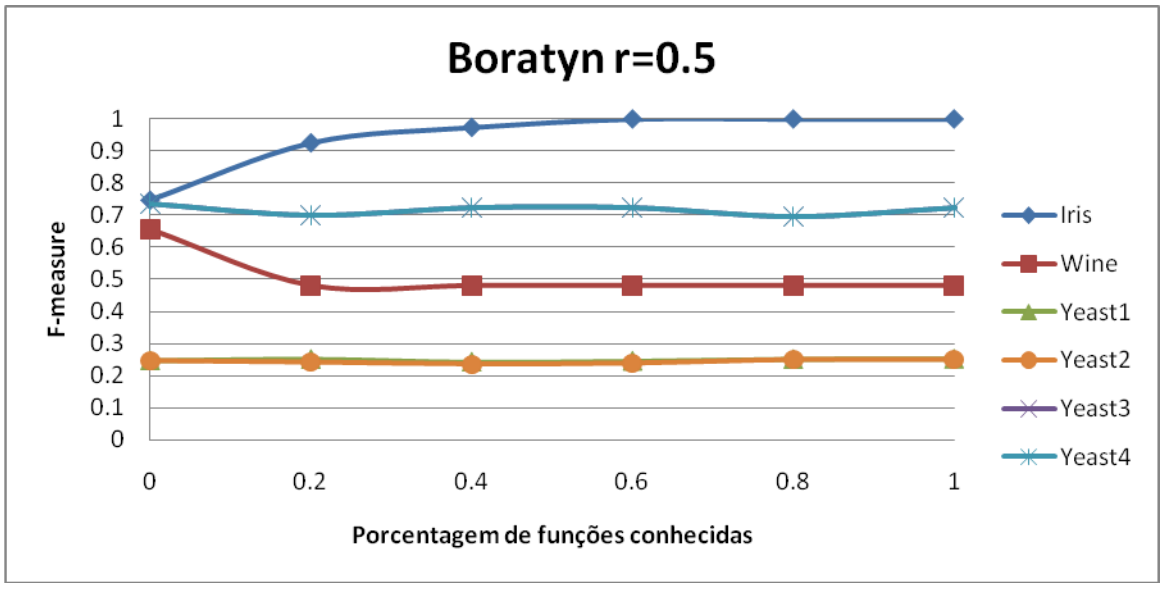


Gráfico 41**:** Resultados consolidados – Fmeasure - Boratyn (r = 0,5)

6.6.7.1 BHI

O Gráfico 42 representa os resultados obtidos para os todos os conjuntos de dados que foram aplicados o algoritmo *Seeded-K-Means*:

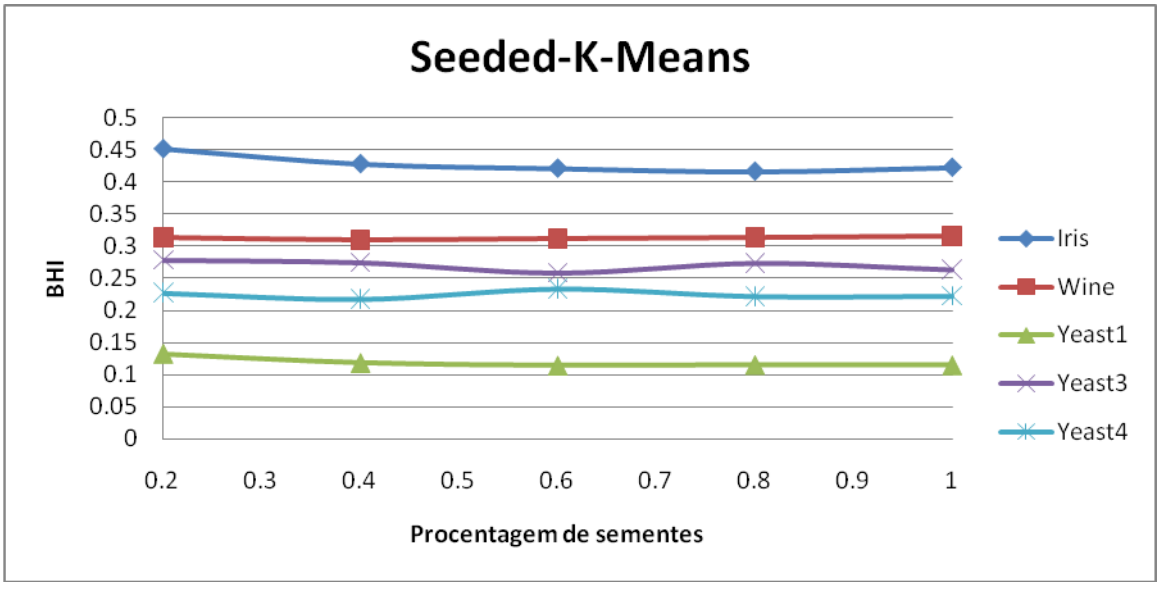


Gráfico 42: Resultados consolidados – BHI - Seeded-K-Means

O Gráfico 43 representa os resultados obtidos para os todos os conjuntos de dados que foram aplicados o algoritmo *Constrained-K-Means*:

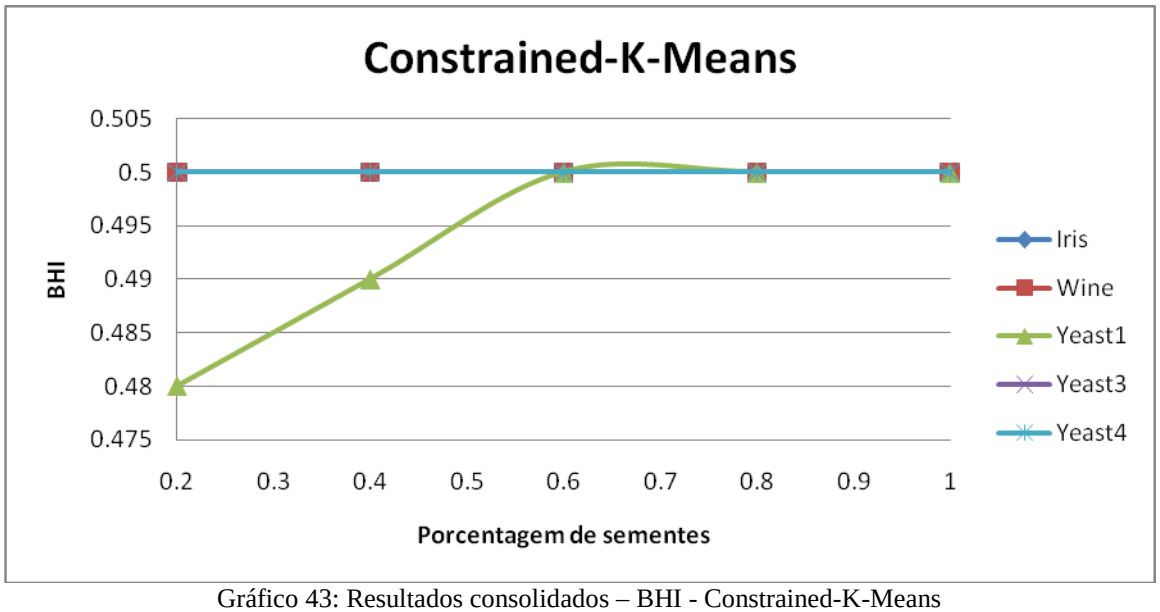


Gráfico 43: Resultados consolidados – BHI - Constrained-K-Means

O Gráfico 44 representa os resultados obtidos para os todos os conjuntos de dados que foram aplicados o algoritmo *Boratyn* quando r = 0.8:

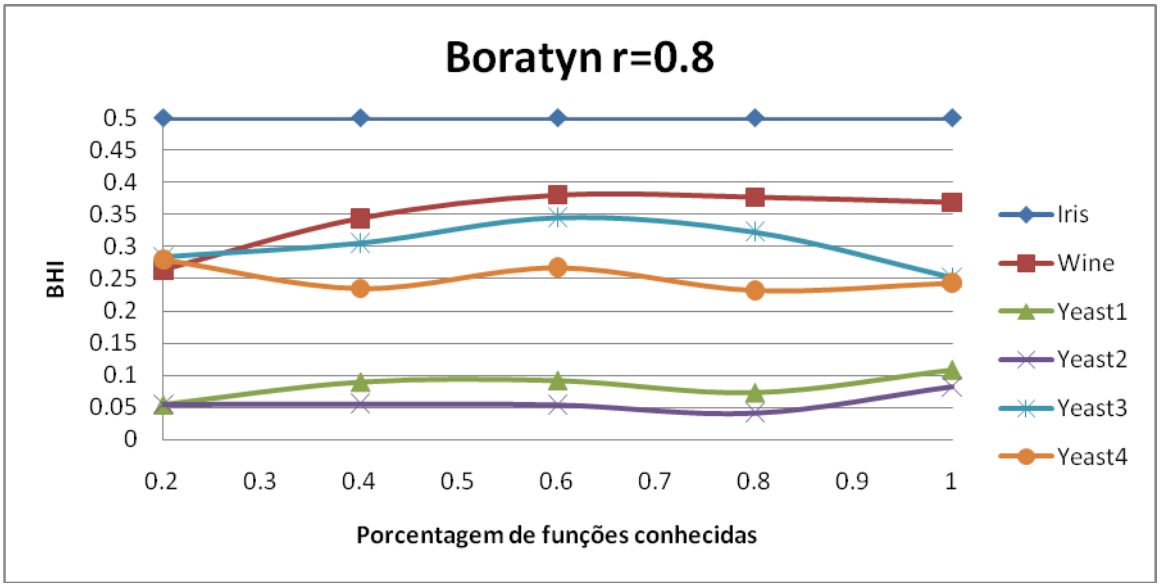


Gráfico 44: Resultados consolidados – BHI – Boratyn (r = 0,8)

O Gráfico 45 representa os resultados obtidos para os todos os conjuntos de dados que foram aplicados o algoritmo *Boratyn* quando r = 0.5:

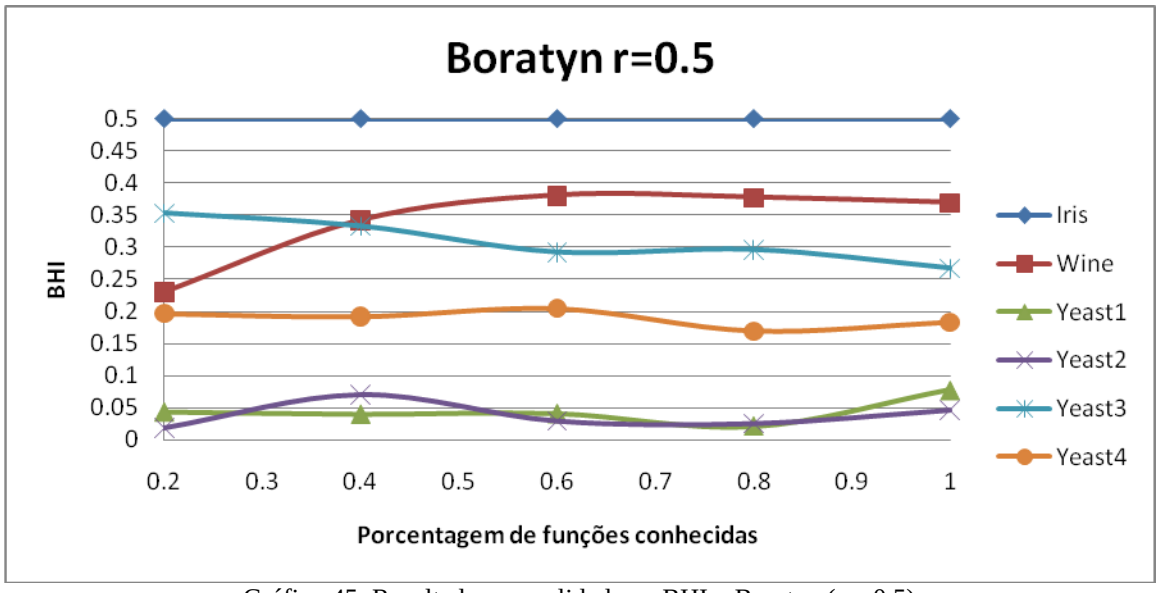


Gráfico 45: Resultados consolidados – BHI – Boratyn (r = 0,5)

Capítulo 7 Conclusões

Neste trabalho foram estudados diversos métodos para agrupamento de dados de expressão gênica. Algoritmos de aprendizado não supervisionado e semi-supervisionado foram pesquisados com o intuito de analisar seus desempenhos em diversos conjuntos de dados, seja de expressão gênica ou de outro domínio.

Nos experimentos realizados, seis conjuntos diferentes foram utilizados, sendo dois deles conjuntos de dados da base de dados UCI (*íris* e *wine*) e os demais, conjuntos de dados de expressão gênica de *yeast Saccharomyces cerevisiae.*

Os resultados obtidos mostraram que os algoritmos semi-supervisionados *Seeded-Kmeans* e *Constrained-Kmeans* tiveram, no geral, melhor desempenho quando comparado ao algoritmo não supervisionado *K-means*.

O mesmo acontece quando o *K-means* é comparado aos algoritmos *COP-Kmeans* e *PCKmeans*. No geral, estes algoritmos semi-supervisionados, tiveram um desempenho igual ou melhor quando comparado ao algoritmo não supervisionado.

Os resultados do método proposto por *Huang & Pan* mostram que quando se tem informações prévias sobre as funções do conjunto, o processo de agrupamento apresenta resultados mais constantes do que quando nenhuma informação prévia é conhecida.

O método proposto por *Boratyn* teve desempenho melhor ou igual ao algoritmo hierárquico para alguns os conjuntos de dados, e pior desempenho para outros conjuntos de dados, conforme apresentado nas seções anteriores.

Em resumo, os resultados mostraram um melhor desempenho dos algoritmos semi-supervisionados em relação aos algoritmos não supervisionados, o que justifica os estudos recentes neste sentido, em que cada vez mais métodos baseados em algum conhecimento prévio têm sido propostos na literatura, como forma de solucionar o problema do alto custo para rotular dados.

7.1 Trabalhos futuros

Muitas são as possibilidades de projetos futuros e recomendados que podem advir deste trabalho, conforme já discutido na seção anterior em que diversas questões abertas foram apontadas.

- ✓ Uma investigação de agrupamento e agrupamento semi-supervisionado baseada em diferentes medidas de similaridades, disponíveis na literatura e apresentadas na seção 2.5.
- ✓ Neste trabalho não foram aplicadas técnicas de pré-processamento de dados para diminuir a dimensionalidade. Assim, uma possível sugestão de trabalho futuro seria utilizar os métodos aqui estudados, após um pré-processamento dos dados e compará-los com os resultados obtidos com os dados originais.
- ✓ Realização de experimentos com auxílio de um especialista, utilizando conjuntos de dados com nenhum exemplo rotulado. A idéia consiste em utilizar um especialista do domínio que seja responsável por rotular os dados mais significativos.
- ✓ Expansão dos estudos de medidas de similaridade que considerem também o conhecimento biológico nos dados, em algoritmos conhecidos na literatura como, por exemplo, *COPKMeans, PCKMeans*, *MPCKMeans*, entre outros.

Referências Bibliográficas

Abba, M. C., J. A. Drake, K. A. Hawkins, Y. Hu, H. Sun, C. Notcovich, S. Gaddis, A. Sahin, K. Baggerly e C. M. Aldazcorresponding. Transcriptomic changes in human breast cancer progression as determined by serial analysis of gene expression. PubMed Central - Journal List, v.6, n.5, p.R499–R513. 2004.

Agrawal, R., J. Gehrke, D. Gunopulos e P. Raghavan. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. SIGMOD International Conference on Management of Data. Seattle, Washington, USA: ACM Press, 1998. 94-105 p.

Alberts, B., D. Bray, J. Lewis, M. Raff, K. Roberts e J. D. Watson. Biologia Molecular da Célula. Porto Alegre: Artes Médicas. 1997

Alberts, B., D. A. Bray, A. A. Johnson, J. A. Lewis, M. A. Raff, K. A. Roberts e P. A. Walter. Essential Cell Biology: An Introduction to the Molecular Biology of the Cell: Garland New York:. 1998. 630 p.

Alizadeh, A. A., M. B. Eisen, R. E. Davis, I. S. L. Chi Ma, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D.Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown e L. M. Staudt. Distinct Types of Diffuse Large B-Cell Lymphoma Identified by Gene Expression Profiling. Nature. 403: 503-511 p. 2000.

Amini, M.-R. e P. Gallinari. Semi-Supervised Learning with Explicit Misclassification Modeling. Proceedings of the 18th International Joint Conference on Artificial Intelligence. Acapulco, Mexico, 2003. 555-560 p.

Baldi, P. e S. Brunak. Bioinformatics: The machine learning approach: MIT Press Cambridge, USA. 1998 (Adaptative Computation and Machine Learning)

Bar-Hillel, A., T. Hertz, N. Shental e D. Weinshall. Learning Distance Functions using Equivalence Relations. Proceedings of 20th International Conference on Machine Learning (ICML-2003), 2003. 11-18 p.

Basu, S., A. Banerjee e R. J. Mooney. Semi-supervised clustering by seeding. Nineteenth ICML - Internacional Conference on Machine Learning Sidney, Australia, 2002. 19-26 p.

Basu, S., A. Banjeree e E. Mooney. Active Semi-Supervision for Pairwise Constrained Clustering. In Proceedings of the 2004, 2004. 333--344 p.

Basu, S., M. Bilenko e R. J. Mooney. A probabilistic framework for semi-supervised clustering. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. Seattle, WA, USA: ACM: 59 - 68 p. 2004.

Ben-Dor, A., N. Friedman e Z. Yakhini. Class discovery in gene expression data. Proceedings of the fifth annual international conference on Computational biology Montreal, Quebec, Canada: ACM 2001

Bennett, K. P. e A. Demiriz. Semi-Supervised Support Vector Machines. In: (Ed.). Advances in Neural Information Processing Systems. Denver: MIT Press, 1998. Semi-Supervised Support Vector Machines, p.368 - 374

Bilenko, M. e R. J. Mooney. Adaptive Duplicate Detection Using Learnable String Similarity Measures. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003}. Department of Computer Sciences - University of Texas at Austin, 2003. 39-48 p.

Blum, A. e P. Langley. Selection of Relevant Features and Examples in Machine Learning. Artificial Intelligence, v.97, n.1-2, p.245-271. 1997.

Blum, A. e T. Mitchell. Combining labeled and unlabeled data with co-training. Eleventh Annual Conference on Computational Learning Theory – COLT. New York, NY, USA: ACM Press, 1998. 92-100 p.

Boratyn, G. M., S. Datta e S. Datta. Biologically supervised hierarchical clustering algorithms for gene expression data. Conf Proc IEEE Eng Med Biol Soc, v.1, p.5515--5518. 2006.

Borges, H. B., J. C. Nievola e B. Pucpr. Gene-finding as an Attribute Selection Task. International Conference on Computer and Information Science - ICIS: IEEE/ACIS 2007. 537-542 p.

Boser, B., I. Guyon e V. Vapnik. A training algorithm for optimal margin classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning theory. Pittsburgh, Pennsylvania, United States, 1992. 144-152 p.

Brazma, A. e J. Vilo. Gene expression data analysis. Federation of European Biochemical Letters - FEBS, v.480, n.1, p.17-24. 2000.

Brenner, S., M. Johnson, J. Bridgham, G. Golda, D. H. Lloyd, D. Johnson, S. Luo, S. Mccurdy, M. Foy e M. Ewan. Gene expression analysis by massively parallel signature sequencing (MPSS) on microbead arrays. Nature Biotechnology 18: 630-634 p. 2000.

Carmona-Saez, P., R. D. Pascual-Marqui, F. Tirado, J. M. Carazo e A. Pascual-Montano. Biclustering of gene expression data by non-smooth non-negative matrix factorization. BMC Bioinformatics: BioMed Central. 7: 78 p. 2006.

Chan, V., N. Hontzeas e V. Park. Gene Expression. University of Waterloo, Ontario, Canada 2000.

Cheng, Y. e G. M. Church. Biclustering of expression data. Eighth International Conference of Intelligent Systems for Molecular Biology - ISMB: AAAI Press, 2000. 93-103 p.

Chu, S., J. Derisi, M. Eisen, J. Mulholland, D. Botstein, P. O. Brown e I. Herskowitz. The Transcriptional Program of Sporulation in Budding Yeast Science Magazine. 282: 699 - 705 p. 1998.

Claverie, J.-M. Computational methods for the identification of differential and coordinated gene expression. Human Molecular Genetics v.8, n.10, p.1821-1832. 1999.

Cohn, D., R. Caruana e A. Mccallum. Semi-supervised clustering with user feedback. Cornell University, p.183–190. 2003

Craven, M. W., R. J. Mural, L. J. Hauser e E. C. Uberbacher. Predicting protein folding classes without overly relying on homology. Proc. of the 3rd International Conference on Intelligent Systems for Molecular Biology. Menlo Park, CA, 1995. p.

D'haeseleer, P., X. Wen, S. Fuhrman e R. Somogyi. Mining the gene expression matrix: inferring gene relationships from large scale gene expression data. Proceedings of the second international workshop on Information processing in cell and tissues. Sheffield, United Kingdom Plenum Press 1998.

D'haeseleer, P., S. Liang e R. Somogyi. Genetic network inference: from co-expression clustering to reverse engineering. Bioinformatics, v.16, n.8, p.707-726 2000.

Datta, S. e S. Datta. Comparisons and validation of statistical clustering techniques for microarray gene expression data. Oxford Journals, v.19, n.4, p.459-466. 2003.

Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes. BMC Bioinformatics, v.7, n.397. 2006.

Demiriz, A., K. P. Bennett e M. J. Embrechts. Semi-supervised clustering using genetic algorithms. Rensselaer Polytechnic Institute. Troy, New York, p.809–814. 1999

Eisen, M. B., P. T. Spellman, P. O. Brown e D. Botstein. Cluster analysis and display of genome-wide expression patterns. Natural Academy of Science, v.95, n.25, p.14863-14868. 1998.

Getz, G., E. Levine e E. Domany. Coupled Two-Way Clustering Analysis of Gene Microarray Data. National Academy of Sciences, 2000. 12079-12084 p.

Ghahramani, Z. e M. I. Jordan. Supervised learning from incomplete data via an EM approach. Advances in Neural Information Processing Systems: Morgan Kaufmann Publishers, Inc., 1994. 120-127 p.

Goldman, S. e Y. Zhou. Enhancing Supervised Learning with Unlabeled Data. Proc. 17th International Conf. on Machine Learning: Morgan Kaufmann, San Francisco, CA, 2000. 327-334 p.

Golub, T. R., D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing e M. A. Caligiuri. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science, v.286, n.5439, October, p.531-537. 1999.

Gordon, A. D. Classification: Chapman & Hall/CRC. 1999

Grira, N., M. Crucianu e N. Boujemaa. Unsupervised and Semi-supervised Clustering: a Brief Survey. A Review of Machine Learning Techniques for Processing Multimedia Content. Le Chesnay Cedex, France 2005. p.

Halkidi, M., Y. Batistakis e M. Vazirgiannis. On Clustering Validation Techniques. Intelligent Information Systems, v.17, n.2, p.107-145. 2001.

Handl, J., J. Knowles e D. B. Kell. Computational cluster validation in post-genomic data analysis. Bioinformatics, v.21, n.15, p.3201-3212. 2005.

Harrington, C. A., C. Rosenow e J. Retief. Monitoring gene expression using DNA microarrays. Current Opinion in Microbiology, v.3, n.3, p.285-291. 2000.

He, Q. A review of clustering algorithms as applied in ir. University of Illinois. 1999

Huang, D. e W. Pan. Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data. Oxford Journals, v.22, n.10, p.1259-1268. 2006.

Hughes, T. R., M. J. Marton, A. R. Jones, C. J. Roberts, R. Stoughton, C. D. Armour, H. A. Bennett, E. Coffey, H. Dai, Y. D. He, M. J. Kidd, A. M. King, M. R. Meyer, D. Slade, P. Y. Lum, S. B. Stepaniants, D. D. Shoemaker, D. Gachotte, K. Chakraburtty, J. Simon, M. Bard e S. H. Friend. Functional discovery via a compendium of expression profiles. Cell, v.102, n.1, p.109-126. 2000.

Jain, A. K., M. N. Murty e P. J. Flynn. Data clustering: a review. ACM Computing Surveys, v.31, n.3, p.264-323. 1999.

Jiang, D., J. Pei e A. Zhang. DHC: a density-based hierarchical clustering method for time series gene expression data. Bioinformatics and Bioengineering, 2003. Proceedings. Third IEEE Symposium on. Dept. of Comput. Sci., State Univ. of New York, Buffalo, NY, USA;: IEEE, 2003. 393- 400 p.

_____. Interactive exploration of coherent patterns in time-series gene expression data Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining Washington, D.C.: ACM 2003

Jiang, D., C. Tang e A. Zhang. Cluster analysis for gene expression data: a survey. IEEE Transactions on Knowledge and Data Engineering v.16, n.11, p.1370-1386. 2004.

Joachims, T. Transductive Inference for Text Classification using Support Vector Machines. Proceedings of ICML-99, 16th International Conference on Machine Learning. San Francisco, US: Morgan Kaufmann Publishers, 1999. 200-209 p.

Klein, D., S. D. Kamvar e C. D. Manning. From Instance-level Constraints to Space-Level Constraints: Making the Most of Prior Knowledge in Data Clustering. Proceedings of the Nineteenth international Conference on Machine Learning. San Francisco, CA: Morgan Kaufmann Publishers, 2002. 307-314 p.

Kohonen, T. Self-Organizing Maps. Springer, Berlin. 1997 (Springer Series in Information Sciences)

Li, T., S. Zhu e Q. L. A. M. Ogihara. Gene functional classification by semi-supervised learning from heterogeneous data. Proceedings of the 2003 ACM symposium on Applied computing. Melbourne, Florida: ACM, 2003. 78 - 82 p.

Lockhart, D. J., H. Dong, M. C. Byrne, M. T. Follettie2, M. V. Gallo, M. S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Norton e E. L. Brown. Expression monitoring by hybridization to high-density oligonucleotide arrays. Nature biotechnology, v.14, p.1675 - 1680. 1996.

Lu, Y., Q. Tian, F. Liu, M. Sanchez e Y. Wang. Interactive Semisupervised Learning for Microarray Analysis. Transactions on computacional biology and bioinformatics, v.4, n.2, p.190-203. 2007.

Macqueen, J. B. Some methods of classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967. 281-297 p.

Mewes, H. W., C. Amid, R. Arnold, D. Frishman, U. Güldener, G. Mannhaupt, M. Münsterkötter, P. Pagel, N. Strack, V. Stümpflen, J. Warfsmann e A. Ruepp. MIPS: analysis and annotation of proteins from whole genomes. Oxford Journals, v.32, n.D41-D44. 2004.

Moreau, Y., F. D. Smet, G. Thijs, K. Marchal e B. D. Moor. Functional bioinformatics of microarray data: from expression to regulation: IEEE, 2002. 1722-1743 p.

Murphy, D. Gene Expression Studies Using Microarrays: Principles, Problems, and Prospects. Advances in Physiology Education: The American Physiological Society. 26: 256-270 p. 2002.

Newman, D. e A. Asuncion. UCI Machine Learning Repository: Irvine, CA: University of California, School of Information and Computer Science. 2007.

Ng, R. T., J. Sander e M. C. Sleumer. Hierarchical cluster analysis of SAGE data for cancer profiling. Workshop on Data Mining in Bioinformatics - BIOKDD01 (2001), 2001. 65-72 p.

Nguyen, D. V., A. B. Arpat, N. Wang e R. J. Carroll. DNA Microarray Experiments: Biological and Technological Aspects. Biometrics: Blackwell Synergy. 58: 701-717 p. 2002.

Nigam, K., A. K. Mccallum, S. Thrun e T. M. Mitchell. Text Classification from Labeled and Unlabeled Documents using EM. Machine Learning, v.39, n.2/3, p.103-134. 2000.

Pavlidis, P., J. Cai, J. Weston e W. N. Grundy. Gene functional classification from heterogeneous data. Proceedings of the 5th International Conference on Computational Modelcular Biology (RECOMB}, 2001. 249 - 255 p.

Pihur, V., S. Datta e S. Datta. Weighted rank aggregation of cluster validation measures: a Monte Carlo cross-entropy approach. Bioinformatics, v.23, n.13, p.1607-1615. 2007.

Priness, I., O. Maimon e I. Ben-Gal. Evaluation of gene-expression clustering via mutual information distance measure. BMC Bioinformatics, v.8, n.111. 2007.

Sanches, M. K. Aprendizado de máquina semi-supervisionado: proposta de um algoritmo para rotular exemplos a partir de poucos exemplos rotulados. Instituto de Ciências Matemáticas e de Computação - ICMC, USP, São Carlos, 2003.

Schena, M., D. Shalon, R. W. Davis e P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science. 270: 467-470 p. 1995.

Setubal, J. C. e J. Meidanis. Introduction to Computational Molecular Biology. Boston: PWS Publishing Company. 1997

Slonim, D. K., P. Tamayo, J. P. Mesirov, T. R. Golub e E. S. Lander. Class prediction and discovery using gene expression data. 4th Annual International Conference on Computational Molecular Biology - RECOMB. Tokyo, Japan: ACM Press New York, NY, USA, 2000. 263-272 p.

Stanton, L. W. Methods to profile gene expression. Trends in Cardiovascular Medicine. 11: 49-54 p. 2001.

Steuer, R., P. Humburg e J. Selbig. Validation and functional annotation of expression-based clusters based on gene ontology. BMC Bioinformatics, v.7. 2006.

Tan, P.-N., M. Steinbach e V. Kumar. Introduction to Data Mining: Addison Wesley. 2005

Tang, C., A. Zhang e J. Pei. Mining phenotypes and informative genes from gene expression data Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. Washington, D.C. : ACM 2003

Tavazoie, S., J. D. Hughes, M. J. Campbell, R. J. Cho e G. M. Church. Systematic determination of genetic network architecture. Nature genetics, v.22, p.281 - 285. 1999.

Toronen, P. Analysis of gene expression data using clustering and functional classifications. Department of Neurobiology, University of Kuopio, Kuopio, 2004.

Troyanskaya, O., M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani e D. Botstein. Missing value estimation methods for DNA microarrays. Bioinformatics: Oxford Univ Press. 17: 520-525 p. 2001.

Valafar, F. Pattern Recognition Techniques in Microarray Data Analysis: A Survey. Annals of the New York Academy of Sciences, v.980, p.41-64. 2002.

Vapnik, V. The nature of statistical learning theory. New York, NY, USA: Springer-Verlag. 1995

Velculescu, V. E., L. Zhang, B. Vogelstein e K. W. Kinzler. Serial analysis of gene expression. Science. 270: 368-9 p. 1995.

Wagstaff, K., C. Cardie, S. Rogers e S. Schroedl. Constrained k-means clustering with background knowledge. Eighteenth International Conference on Machine Learning - ICML. Williamstown, Massachusetts, USA: ACM Press, 2001. 577–584 p.

Xing, E. P. e R. M. Karp. Cliff: Clustering of High-Dimensional Microarray Data via Iterative Feature Filtering Using Normalized Cuts. Bioinformatics. 17: 306-315 p. 2001.

Xing, E. P., A. Y. Ng, M. I. Jordan e S. Russell. Distance Metric Learning with Application to Clustering with Side-Information. Oxford Journals, p.505-512. 2003.

Yang, Y. H., S. Dudoit, P. Luu e T. P. Speed. Normalization for cDNA microarray data. The International Biomedical Optics Symposium - SPIE BiOS San Jose, California: Oxford Univ Press, 2001. e15 p.

Yin, L., C.-H. Huang e J. Ni. Clustering of gene expression data: performance and similarity analysis. BMC Bioinformatics, v.7. 2006.

Zhang, T., R. Ramakrishnan e M. Livny. BIRCH: an efficient data clustering method for very large databases. SIGMOD International Conference on Management of Data. Montreal, Quebec, Canada: ACM Press New York, NY, USA, 1996. 103-114 p.