Universidade Federal de Pernambuco
Graduação em Ciência da Computação

Centro de Informática

ANÁLISE DE DADOS DE MICRO-ARRAYS: ESTUDO COMPARATIVO ENTRE MÉTODOS DE CLUSTERIZAÇÃO

Trabalho de Graduação

Aluno: Paulo Roberto Figueirôa Amorim (prfa@cin.ufpe.br)
Orientadora: Kátia Silva Guimarães (katiag@cin.ufpe.br)

Dezembro de 2008

Resumo

Esse trabalho tem como proposta mostrar, de uma forma geral, o problema de análise de expressão gênica, assim como alguns conceitos relacionados, como perfis de expressão, micro-arrays, técnicas de agrupamento e o aparecimento dos algoritmos genéticos.

Também como, decorrer sobre os resultados entre algumas das técnicas apresentadas, como K-means, Self-Organizing Maps (SOM) e Incremental Genetic K-means Algorithm (IGKA). Além disso, a realização de um novo estudo das características de comportamento do algoritmo IGKA, tendo como entrada uma nova base de dados.

Agradecimentos

Agradeço a todos que contribuíram direta ou indiretamente com o desenvolvimento e conclusão deste trabalho.

Índice

Índice de Figuras

1. Introdução

Com as novas pesquisas dentro da área de biologia molecular, uma enorme quantidade de dados está disponível para ser analisada. Com o avanço de novas técnicas de extração de informação, como a tecnologia de Micro-array, tornou-se possível observar, simultaneamente, o nível de expressão de milhares de genes, de acordo com o estudo de comportamento das células em determinadas condições ou dentro de processos específicos.

Nos últimos anos, algoritmos de agrupamento estão sendo, de forma bastante efetiva, utilizados na análise de dados referentes à expressão de genes, no campo da biologia molecular. Algoritmos de agrupamento são usados de forma a dividir os genes em grupos baseados nas similaridades entres os seus perfis de expressão. Desse modo, genes que compartilham de uma mesma funcionalidade podem ser identificados.

Dentre os vários algoritmos de agrupamento, o K-means é um dos mais populares métodos usados na análise de dados de expressão gênica devido a sua alta performance computacional. No entanto, um problema, bem conhecido, dessa técnica é o fato de que essa pode chegar a um mínimo local, e o seu resultado estar sujeito ao processo de inicialização do processo que gera, de forma aleatorizada, o primeiro agrupamento. Em outras palavras, diferentes aplicações dessa técnica, com um mesmo dado de entrada, pode produzir diferentes soluções.

Como o número de dados de laboratório em biologia molecular cresce exponencialmente, com o passar dos anos, devido ao avanço de técnicas, novos, eficientes e efetivos, métodos de agrupamento foram desenvolvidos para processar esse crescente valor de dados biológicos.

Na tentativa de resolver tal problema, um grande número de cientistas propôs a noção de algoritmos genéticos para agrupamento. A idéia básica de tal procedimento seria simular o processo de evolução da natureza e evoluir soluções de uma geração para a próxima. Em contraste com o K-means, que pode levar a um mínimo local, esses algoritmos genéticos são insensíveis ao processo de inicialização e sempre convergiriam, eventualmente, ao máximo global. No entanto, tais algoritmos são, na grande maioria dos casos, computacionalmente "caros" o que impede uma ampla utilização dos mesmos em práticas, como a análise de dados de expressão gênica, devido ao grande número de dados a ser trabalhado.

Recentemente, foi proposto um novo método de agrupamento chamado de *Genetic K-means Algorithm (GKA)*, que tem como objetivo formar um híbrido entre algoritmos genéticos e o K-means. Essa abordagem híbrida combina a natureza robusta do algoritmo genético com o alta performance do K-means. Como resultado, tal método sempre irá convergir para um ótimo global de uma forma mais rápida do que as técnicas, antes apresentadas.

A partir desse novo método, foi proposta uma versão mais rápida do mesmo, denominada de *Faster Genetic K-means Algorithm (FGKA)* que implementa a melhoria de várias funcionalidades do GKA, incluindo uma mais eficiente avaliação do valor objeto TWCV (Total Within-Cluster Variation), evitando o overhead de eliminação de strings ilegais e uma simplificação do operador de mutação.

Além desse, método também foi proposta uma nova técnica denominada de *Incremental Genetic K-means Algorithm (IGKA)* que herda todas as vantagens de FGKA incluindo a convergência a um ótimo global, e funciona de uma forma mais eficiente que o mesmo

quando a probabilidade de mutação é pequena. A idéia principal de tal método é calcular o valor objeto TWCV e os centros dos grupos de uma forma incremental.

Com o advento desse novo método, foi proposta a idéia de uma técnica híbrida denominada de *Hybrid Genetic K-means Algorithm (HGKA)* que combina os benefícios do FGKA e IGKA.

2. Perfis de Expressão Gênica

No campo da biologia molecular, o perfil de expressão gênica é a medida da atividade (expressão) de milhares de genes de uma mesma vez, de forma a criar uma representação global do funcionamento de uma célula. Esses perfis podem, por exemplo, distinguir, entre células que estão dividindo ativamente, ou mostrar como as células reagem a um tratamento particular. Vários experimentos desse tipo medem um inteiro genoma simultaneamente, que é cada gene presente em uma célula em particular.

Depois do sequenciamento do genoma, a definição de perfis de expressão de genes foi o próximo passo a ser tomado, a seqüência mostra o que a célula provavelmente poderia fazer, enquanto que o perfil de expressão mostra o que ela realmente está fazendo no momento. Genes contêm as instruções para fazer RNA's mensageiros, mas a cada momento cada célula produz RNA's mensageiros de apenas uma fração dos genes que carrega. Se o gene é usado para produzir tal RNA, é considerado "on", caso contrário "off". Vários fatores determinam se um gente está "on" ou "off", como a hora do dia, se a célula está ou não em processo de divisão, o seu ambiente local, sinais químicos de outras células, etc. Portanto, um perfil de expressão permite a dedução do tipo da célula, seu estado, seu ambiente e assim em diante.

O estudo dos padrões de ativação (expressão) dos genes é realizado sobre diversas condições. Genes que possuem a mesma funcionalidade são comumente ativados pelas mesmas condições. Genes codificadores de enzimas que catalisam um conjunto de reações encadeadas são geralmente co-regulados (e normalmente se localizam próximos no cromossomo). A ativação conjunta também ajuda a inferir funcionalidades de genes dos quais ainda não temos informações. O

padrão de ativação dos genes pode caracterizar doenças e assim gerar novas ferramentas precisas de diagnóstico.

Perfis de expressão provêem excitantes novas informações sobre genes sob variadas condições. No geral, a tecnologia de micro-array produz confiáveis perfis de expressão. A partir dessa informação, podem ser geradas novas hipóteses sobre biologia ou testar aquelas existentes. No entanto, o tamanho e a complexidade desses experimentos geralmente resultam em uma variedade de possíveis interpretações. Em vários casos, analisar resultados de perfis de expressão demanda de um maior esforço do que a realização de experimentos iniciais.

A maioria dos pesquisadores usa múltiplos métodos estatísticos e analise exploratórias de dados antes de publicar os seus resultados de perfis de expressão gênica, coordenando seus esforços com uma estatística biológica ou outra tecnologia perita em micro-array. Um bom projeto de experimento, adequada replicação biológica e experimentos de acompanhamento fazem um papel principal em experimentos de sucesso na definição de perfis de expressão.

A tecnologia de DNA Micro-array mede a atividade relativa de genes alvos previamente identificados. Técnicas baseadas em marcação de genes, como análise serial de expressão gênica (SAGE) também são usadas para definição de perfis de expressão gênica. Tais processos são mais exatos e também podem medir qualquer gene ativo, não apenas um conjunto pré-definido. Uma técnica emergente alternativa a definição de perfis de expressão gênica é caracterizada pelo sequenciamento profundo.

3. Micro-arrays

A tecnologia de micro-array evoluiu de uma técnica chamada de "Shoutern blotting", em que DNA fragmentado é anexado a um substrato e então inquerido com um gene ou fragmento conhecido. O uso de uma coleção de DNA's distintos em arrays para definição de perfis de expressão foi primeiramente descrito em 1987, e tal informação foi utilizada para identificar genes dos quais sua expressão é modulado por "interferon". Esses primeiros arrays de genes foram feito por "machas" de cDNA's em papéis de filtro com um dispositivo de sinalização de "manchas". Uma visão geral de tal processo é mostrado na figura abaixo. O uso de micro-arrays miniaturizados para definição de perfis de expressão de gene foi primeiro reportado em 1995 e um completo genoma eucariona (Saccharomyces cerevisiae) em micro-array foi publicado em 1997.

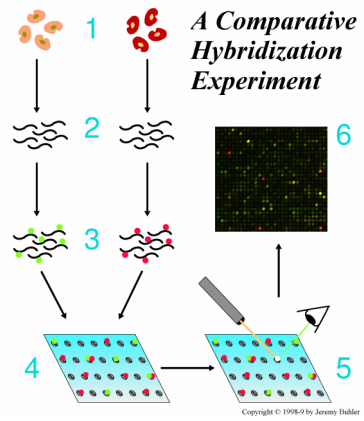

Figura 1: Processamento dos primeiros arrays de genes (visão geral).

O advento da realização de um grande número de experimentos com micro-arrays criou vários desafios específicos na área de bioinformática: os níveis múltiplos de replicação em projetos experimentais (projeto experimental); o número de plataformas e grupos independentes e formato de dados (padronização); o tratamento do dado (análises estatísticas); o que exatamente está

sendo medido (relação entre valor e gene); e o volume de contribuição do dado e a habilidade de compartilhar isso (armazenamento de dados).

Novas tecnologias e aplicações surgiram através de uso de micro-arrays, como:

- Definição de perfis de expressão gênica, em que o experimento de níveis de expressão de milhares de genes que são simultaneamente monitorados para estudar os efeitos de certos tratamentos, doenças e estágios de desenvolvimento em nível de expressão gênica;
- Hibridização comparativa de genomas, em que são avaliados conteúdos de genoma em diferentes células ou organismos estreitamente relacionados;
- Detecção de SNP (Single Nucleotide Polymorphism), identificando o polimorfismo de um único nucleotídeo entre alelos dentro ou entre populações;
- Cromatina de imunoprecipitação em ChIP, em que sequências de DNA ligadas a uma determinada proteína podes ser isolada por imuno-precipitação nessa proteína (ChIP), esses fragmentos podem ser hibridizados então para um micro-array (como uma cobertura de "tilling arrays"), que permite a determinação ocupação de um site de ligação de uma proteína de acordo com o genoma;
- "tilling arrays", que consiste em um sobreposição de "manchas" concebidas para representar densamente uma região de genes de interesse, às vezes tão grande quanto um cromossomo humano inteiro.

Os dados de valores de expressão de gene em um experimento utilizando micro-arrays podem ser tanto representados de uma forma gráfica, através de "manchas" codificadas por cores (Fig.2), ou, de dados numéricos, representados por uma matriz de dados (Fig.3). De outra forma, mais completa, podem ser representados como "mapas de calor" para visualizar o resultado da análise de dados (Fig.4).

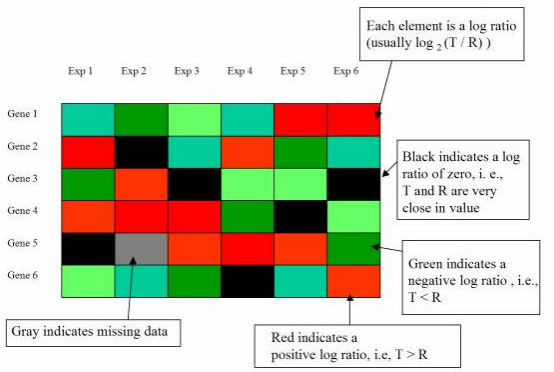

Figura 2: Micro-arrays codificados por "manchas" de cores

	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6
Gene 1	-1.2	-2.1	-3	-1.5	1.8	2.9
Gene 2	2.7	0.2	-1.1	1.6	-2.2	-1.7
Gene 3	-2.5	1.5	-0.1	-1.1	-1	0.1
Gene 4	2.9	2.6	2.5	-2.3	-0.1	-2.3
Gene 5	0.1		2.6	2.2	2.7	-2.1
Gene 6	-2.9	-1.9	-2.4	-0.1	-1.9	2.9

Figura 3: Micro-arrays representados por uma matriz de dados

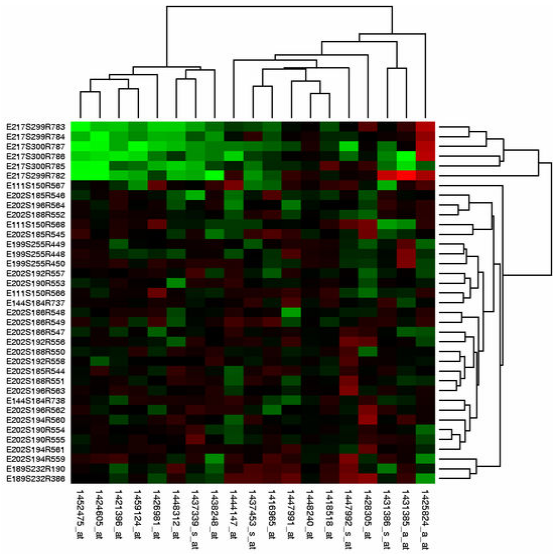

Figura 4: "mapas de calor" para visualizar análise de dados

4. Técnicas de Análise de Dados

Na análise de dados de expressão gênica, diversas técnicas têm sido aplicadas para problemas dessa classe, como a redução de dimensionalidade (PCA), algoritmos de agrupamento (hierárquicos, k-means,...) e SOMs (Self-Organizing Maps).

A co-expressão de genes sugere que eles são relacionados funcionalmente e que eles são possivelmente co-regulados. A função de muitos genes não-caracterizados podem ser descobertas a partir das funções de genes co-expressados conhecidos.

Os principais objetivos de classificação de genes são: organização funcional de genes; interpretar o estado da célula de acordo com um determinado padrão de expressão gênica; deduzir a função de genes desconhecidos; e explorar a regulação transcripcional.

Verificar a expressão gênica relativa a uma condição fisiológica. Classificar doenças utilizando perfis de expressão gênica baseados em micro-array. Agrupar experimentos de acordo com a similaridade dos perfis. Os clusters identificados podem ser analisados diretamente dos padrões de expressão gênica associados sob perspectivas moleculares ou clínicas.

4.1 Redução de Dimensionalidade

Como em alguns casos, os vetores de dados biológicos possuem várias dimensões, métodos que diminuem a dimensionalidade dos dados são utilizados de forma a facilitar a extração de informações.

Uma forma bastante conhecida e eficaz de fazer tal processo é utilizando da técnica de análise do componente principal, ou PCA (Principal Componente Analysis), que é um método estatístico para projetar pontos de uma dimensão M num espaço de dimensão N, em que, logicamente, M é um valor muito maior do que N.

Esse processo encontra a representação, num espaço de dimensionalidade menor, que descreve os pontos dos dados, com o menor erro possível. A Figura 5 abaixo, mostra de que forma pontos de dados de dimensão maior podem ser representados de forma bidimensional.

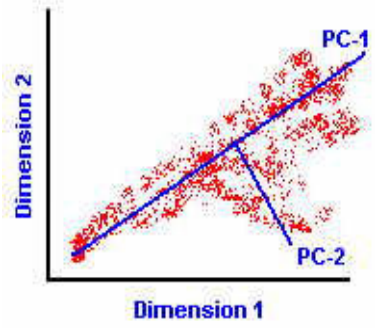

Figura 5: Exemplo gráfico de um PCA

4.2 Algoritmos de Agrupamento

Algoritmos de agrupamento têm como função principal classificar, agrupar ou comprimir uma grande quantidade de dados, que podem estar em formato numérico, relacional ou nominal. Seu procedimento básico é realizado através da similaridade, baseado no cálculo de distâncias, e do número de grupos, no qual o conjunto será dividido.

Tal procedimento pode ser feito de três formais principais, supervisionado, supervisionado, baseado num conjunto de vetores ou classes dados; não-supervisionado, em que não existe nenhum conhecimento prévio de classificação; e métodos híbridos, ou seja, métodos supervisionados utilizando classificações previamente obtidas pela aplicação de um método não-supervisionado.

4.2.1 Classificação Supervisionada

Classificação supervisionada, também chamada de predição ou discriminação, envolve o desenvolvimento de algoritmos para categorias previamente definidas.

Métodos de suporte a vetores é um grupo de métodos de aprendizagem supervisionada usada pra classificação. O tipo mais simples de métodos de suporte a vetores é o de classificação linear que tenta desenhar uma linha reta que separa os dados em duas dimensões. Vários classificadores lineares, também chamados de hiperplanos são capazes de separar dados, no entanto, uma nova proposta, de 1963, em que uma classificação linear é feita através de um algoritmo hiperplano ótimo, atinge uma

separação máxima. Esse método tem como característica principal a substituição de um produto de pontos por uma função de kernel não-linear que permite o algoritmo preencher uma margem máxima do hiperplano em função do espaço. Existem quatros funções de kernel básicas: linear, polinomial, radial e sigmóide.

Outra forma de classificação supervisionada são as estruturas de árvores de decisão, em que folhas representam classificações e os galhos representam conjunções de recursos que levam a tais classificações. Algoritmos de árvores de decisão têm como vantagem serem convertidos facilmente em um conjunto de regras de produção, podem ser usados tanto dados categóricos como numéricos, e não é necessária nenhuma informação a priori da natureza dos dados. No entanto, atributos múltiplos de saída não são permitidos nesse caso, além da instabilidade inerente a algoritmos. Pequenas variações nos dados de treinamento podem resultar em diferentes seleções de atributos em cada ponto de escolha da árvore. Tal efeito pode ser bastante significativo já que a escolha de atributos afeta todos os descendentes das sub-árvores.

Redes Neurais Artificiais é outra forma de classificação supervisionada, são formados por grupos de nós interconectados que usam um modelo computacional de processamento de informação. Sua estrutura pode ser mudada de acordo com informações internas e externas que passam pela rede. Tais redes podem ser usadas para modelas um complexo relacionamento entre entradas e saídas de forma a encontrar padrões nos dados. Dois algoritmos de redes neurais artificiais bastante comuns são

o Multi-layer Perceptron (MLP) e as redes de Radial Basis Function (RBF).

Outro formato de classificação supervisionada é a utilização de redes bayesianas que representam a independência entre um conjunto de variáveis em uma dada distribuição de probabilidades conjuntas. Os nodos correspondem a variáveis de interesse e o arcos entre os nodos representam dependências estatísticas entre variáveis. O nome bayesianas se refere ao teorema de Bayes de probabilidade condicional. Esse teorema é um resultado na teoria probabilística, que relaciona a probabilidade marginal e condicional de variáveis aleatórias. Tal método usa todos os atributos e permite que façam contribuições de decisão como se todos fossem de igual importância e independentes um dos outros.

E por fim, um método de classificação supervisionada bastante conhecido é o de agrupamento hierárquico em que os dois elementos mais similares (na matriz de similaridade) se unem criando um novo nó, dessa forma a matriz de similaridade é recalculada, com o novo nó substituindo os dois antigos e com valor igual à média dos anteriores. Com N pontos iniciais, esse processo é repetido N-1 vezes até restar apenas um nó. No entanto, métodos de classificação usando agrupamento hierárquico são imprecisos. A figura 6 abaixo demonstra o resultado de um agrupamento hierárquico.

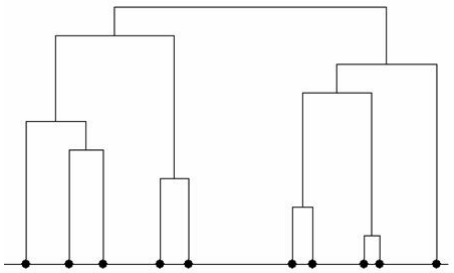

Figura 6: Exemplo de um agrupamento hierárquico

4.2.2 Classificação Não-Supervisionada

Algoritmos de análise de grupos dividem objetos tendo como base algum tipo de métrica de similaridade que é computada. Genes podem ser agrupados em classes com base na similaridade de seus perfis de expressão através de tecidos, casos ou condições. Métodos de agrupamento dividem objetos em um número predeterminados de grupos de forma a maximizar uma função específica.

Um algoritmo de agrupamento não-supervisionado bastante conhecido é o K-means. O algoritmo é inicializado com a escolha de um número K de grupos e a formação de K vetores representantes (núcleos). O próximo passo é que para cada ponto é realizada uma associação ao representante mais próximo. Posteriormente, para cada representante, sua posição é redefinida como sendo a médias das distâncias dos componentes associados a eles previamente. Caso a variação dos representantes seja menor do que um limiar, pré-definido, o procedimento em questão é finalizado, caso contrário, volta-se a passo anterior descrito. Tal algoritmo tem como vantagens, a sua fácil implementação e convergência rápida, e como desvantagens, nem sempre gerar uma divisão ótima (mínimos locais) e má escolhas dos representantes.

4.3 Self-Organizing Maps

Também conhecido como Kohonen Map, foi descrito pela primeira vez como uma rede neural artificial. Parecido com o K-Means, ele preserva as propriedade topológicas dos dados. É um tipo de algoritmo não-supervisionado.

Em seu algoritmo, ele cria um conjunto de nós e mapeie-os aleatoriamente pelo espaço de entrada. A cada iteração há a escolha de um ponto da entrada aleatoriamente e então encontra-se o nó mais próximo a ele. Este nó e seus vizinhos se moverão em direção a este ponto. A influência do nó escolhido nos vizinhos decresce com relação à distância entre eles e a iteração.

Tal método tem como vantagens, o algoritmo convergir para um mapa de classificação e topologia ótimas; possuir uma forma muito conveniente para visualização dos dados; e uma boa manipulação de dados não uniformes e irregulares. E como desvantagens, não ter base teórica para determinar uma dimensão ótima e pode demorar muitas iterações para convergir (20.000 - 50.000).

5. Algoritmos Genéticos

Um algoritmo genético é uma técnica de pesquisa usada em computação para achar soluções exatas, ou aproximadas, para problemas de pesquisa e otimização. Algoritmos genéticos são categorizados como pesquisas heurísticas globais. Fazem parte de uma classe particular de algoritmos evolucionários, também conhecidos como computação evolucionária, que usam técnicas inspiradas pela biologia evolucionária, como herança, mutação, seleção e cross-over, também chamada de recombinação.

Na tentativa de resolver o problema de análise de dados de expressão gênica, tais algoritmos começaram a ser usadas, convergindo de forma eficiente, no entanto, na grande maioria dos casos, tornando-se computacionalmente "caros". Com o intuito de superar tal problema, surgiu o aparecimento de uma nova versão de um algoritmo de agrupamento genético, o *Genetic K-means Algorithm (GKA)*, que tem como objetivo formar um híbrido entre algoritmos genéticos e o K-means.

Essa abordagem híbrida tem como objetivo combinar a natureza robusta do algoritmo genético com o alta performance do K-means. E como resultado, convergir para um ótimo global de uma forma mais rápida do que as técnicas, antes apresentadas.

A partir desse novo método, três novas versões de algoritomos genéticos foram propostas: o *Faster Genetic K-means Algorithm (FGKA),* o *Incremental Genetic K-means Algorithm (IGKA),* e uma versão híbrida denominada de *Hybrid Genetic K-means Algorithm (HGKA)* que combina os benefícios do FGKA e IGKA.

5.1 O Problema

O problema tratado em questão é o agrupamento de dados de expressão gênica, que consiste basicamente de *N* genes e os seus *N* padrões correspondentes. Cada padrão é um vetor de *D* dimensões, gravando o nível de expressão dos genes, sob *D* condições monitoradas ou a cada *D* intervalos de tempo.

O objetivo a ser alcançado é particionar os N padrões em K grupos definidos de forma que tal partição minimize o valor objeto TWCV (Total Within-Cluster Variation), também conhecido por "square-error".

Sendo X_1, X_2,…, X_n os *N* padrões, em que X_{nd} denota a d-ésima característica de padrão X_n(n=1,...,N). Cada partição é representada por uma string, que é uma seqüência de números a_1,...,a_n, em que a_n é o número do grupo do qual o padrão X_n pertence na sua partição. Sendo G_k a denotação do k-ésimo grupo e Z_k o número de padrões em G_k, o centróide c_k = (c_{k1}, c_{k2},...,c_{kn}) do grupo G_k é definido pelo somatório das d-ésimas características de todos os padrões em G_k dividido pelo número de padrões em G_k.

A idéia abordada é que exista uma população (conjunto) de Z soluções codificadas, em que Z é um parâmetro pré-definido. Cada solução, também chamada de "cromossomo", é definida por uma string a1,...,an de tamanho n, em que cada an, chamado de "alelo", corresponde a um padrão de dado de expressão gênica e toma um valor (1,2...,k) representando o número do grupo do qual padrão correspondente pertence. Por exemplo, a1a2a3a4a5 = "33212", encoda uma porção de cinco padrões em que, os padrões X1 e X2 pertencem ao grupo 3, os

padrões X3 e X5, ao grupo 2, e o padrão X4, ao grupo 1.

5.2 Visão Geral

Uma visão geral do algoritmo é mostrada, através de um fluxograma, apresentado na figura abaixo. Ele começa com uma fase de inicialização, que gera a população inicial P_o. A população na próxima geração P_{i+1} é obtida a partir da aplicação de operadores genéticos na atual população P_i. A evolução acontece até que uma condição de parada é alcançada. Os operadores genéticos usados no algoritmo são: seleção, mutação e o operador K-means.

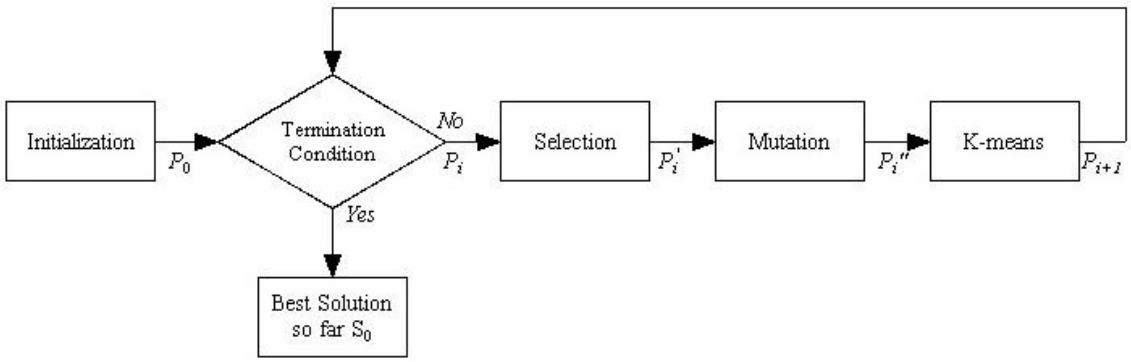

Figura 7: Fluxograma de um algoritmo genético

Uma nova noção apresentada nesse problema é a noção de strings legais e ilegais. Dada uma partição Sz = a1,...,an, e(Sz) é o número de grupos não-vazios em Sz dividido por K, também conhecido por "taxa de legalidade". Uma string Sz é chamada de legal se e(Sz) = 1, e ilegal, caso contrário. Dessa forma, uma string ilegal representa uma partição em que alguns grupos estão vazios. Por exemplo, dado um K=3, a string a1a2a3a4a5 = "23232" é ilegal porque o grupo está vazio.

5.2.1 Operador de Seleção

O operador de seleção funciona usando um tipo de seleção proporcional, em que a população de próxima geração é determinada por Z independentes experimentos aleatórios. Cada experimento seleciona aleatoriamente uma solução para a atual população (S_1,S_2,...,S_k) de acordo com uma distribuição de probabilidade (p_1,p_2,...p_k) definida por um função que denota o valor de melhor ajuste da solução S_z a respeito da população atual.

Várias funções de melhor ajuste têm sido definidas pela literatura em que o valor de ajuste de cada solução na população atual reflete o seu mérito de sobreviver na próxima geração. No contexto atual, o objetivo é minimizar o *Total Within-Cluster Variation (TWCV)*. Portanto, soluções com menores *TWCVs* devem ter maiores probabilidades de sobrevivência e devem ser relacionadas com maiores valores de ajuste. Em adição a isso, strings ilegais são menos desejáveis e devem ter probabilidades de sobrevivência baixas logo, relacionada a menores valores de ajuste.

A idéia por trás da função de melhor ajuste é que, cada solução vai ter uma probabilidade de sobrevivência ao ser assinalada a um valor positivo de ajuste, mas uma solução com menor *TWCV* vai ter um valor de ajuste maior e, portanto, uma maior probabilidade de sobrevivência. Soluções ilegais são permitidas a sobreviver, mas com um valor de ajuste menor de que todas as soluções legais na população atual. Strings ilegais que tem mais grupos vazios são assinaladas com um menor valor de ajuste e, portanto, menores probabilidades de sobrevivência. A razão pela qual soluções ilegais sobrevivem com pequena probabilidade é pela crença de que soluções ilegais podem mutar em uma boa solução e o custo de manter tal solução é pequeno.

5.2.2 Operador de Mutação

Dada uma solução (cromossomo) representada por $a_1,...,a_n$, o operador de mutação muda cada alelo $a_n(n=1,...,N)$ para um novo valor $na_,$ que pode ser igual ao anterior, com uma probabilidade *MP* independente, em que *0 < MP < 1,* é uma parâmetro chamado de *probabilidade de mutação* especificado pelo usuário.

O operador de mutação é muito importante para ajudar na procura de melhores soluções. Da perspectiva da teoria evolucional, descendentes produzidos por mutações podem ser superiores aos seus antecessores. Mais importante, o operador de mutação performa a função de agitar o algoritmo fora de um ótimo local, movendo para um ótimo global.

O operador de mutação é definido por três propriedades de tal forma que: cada padrão possa re-assinalado aleatoriamente para cada grupo com uma probabilidade positiva; a probabilidade de mudança de um alelo de um grupo é maior se o padrão está localizado próximo ao centróide de tal grupo; e grupos vazios são vistos como os grupos mais próximos do padrão. A primeira propriedade assegura que uma solução arbitrária, incluindo um ótimo global, pode ser gerada pela mutação de uma solução atual com uma probabilidade positiva. A segunda probabilidade encoraja que cada padrão estão se movendo em direção ao grupo mais próximo com uma grande probabilidade. E a terceira probabilidade promove a probabilidade de converter uma solução ilegal em uma legal. Essas propriedades são essenciais para garantir que o algoritmo vai convergir eventualmente para um ótimo global de forma rápida.

5.2.3 Operador K-means

De forma a apressar o processo de convergência, um passo do algoritmo clássico K-means, chamado de *K-means Operator (KMO)* é introduzido. Dada uma solução representada por $a_1,...,a_n$, os alelos são substituídos simultaneamente pelo número dos grupos de cujo centróide está mais próximo do padrão, através da distância euclidiana.

A motivação para essa nova definição é a tentativa de evitar o re-assinalar todos os padrões para grupos vazios. No entanto, tal processo não trata o problema de strings ilegais, após tal procedimento, strings ilegais continuarão ilegais.

5.3 Fast Genetic K-Means Algorithm

O *Fast Genetic K-Means Algorithm (FGKA)* compartilha do fluxograma apresentado na Figura 7. Começa com a inicialização da população P_0 com Z soluções. Para cada geração P_i, são aplicados os três operadores, seleção, mutação e KMO sequencialmente que gera as populações P_i', P_i'' e P_{i+1}, respectivamente. Esse processo é repetido por G iterações, cada uma da qual corresponde a uma geração de soluções. A melhor solução até o momento, é observada e guardada em S_0 antes do operador de seleção. S_0 é retornada como solução de saída ao término do algoritmo.

5.4 Incremental Genetic K-Means Algorithm

O *Incremental Genetic K-Means Algorithm (IGKA)* tem como principal característica principal o cálculo incremental dos centróides e TWCV. Por exemplo, se um padrão qualquer é re-assinalado de uma grupo k para um grupo k', apenas o centróides e WCVs desses dois grupos precisam ser recalculados. Além do mais, os centróides desses dois grupos podem ser calculados incrementalmente já que os membros de outros padrões não foram mudados. O TWCV também pode ser calculado incrementalmente, já que WCVs de outros grupos também não mudaram.

De forma a obter o novo centróide, é calculado vetor de diferença de valores entre a solução antiga e a nova quando ocorre a mudança dos alelos. Com esses dois valores, a atualização incremental, o novo centróide, da nova solução pode ser alcançado. Similarmente, de forma a obter o novo TWCV, é calculada a diferença de valor entre o velho TWCV e o novo para uma solução. No entanto o WCV_k precisa ser recalculado desde o começo já que o centróide do grupo k foi mudado. Desse modo, o TWCV pode ser atualizado incrementalmente, também.

Como o cálculo de TWCV domina todas as iterações, a atualização incremental do TWCV terá um melhor desempenho quando a probabilidade de mutação é menor, que implica que um número menor de alelos mudou. No entanto, se a probabilidade de mutação é alta, ou seja, muitos alelos mudam sua localização de grupo, a manutenção de tal cálculo torna-se bastante "cara". Por esse motivo, o IGKA, em casos em que a probabilidade de mutação é muito baixa, supera o FGKA, porque calcular centróides e TWCV do início pode ser muito mais "caro" do que calculá-los de forma incremental. Em casos contrários, em que a mutação é muito alta, o FGKA ainda é a melhor escolha.

5.5 Hybrid Genetic K-Means Algorithm

O *Hybrid Genetic K-Means Algorithm(HGKA)*, surgiu do dilema de ambos o FGKA e IGKA se superarem em determinados casos. Quando a probabilidade de mutação é pequena de que algum limite, o IGKA supera o FGKA, caso contrário, o FGKA supera o IGKA.

A idéia central do HGKA é combinar os benefícios do FGKA e IGKA. No entanto, é muito difícil derivar um valor limite para tal processo, já que isso depende do grupo de dados a ser analisado. Adicionalmente, o tempo de processamento de todas as iterações irá variar ao passo que as soluções convergem para um ótimo. A proposta desse algoritmo é, periodicamente, rodar uma iteração do FGKA seguida de uma iteração do IGKA, monitorando o tempo de duração de cada, e então rodando o algoritmo vencedor nas seguintes iterações até que um novo ponto de competição seja alcançado.

De uma forma geral, como o FGKA e IGKA chegam num ótimo global por utilizar-se do mesmo fluxograma e operações, o HGKA também irá convergir para um ótimo global.

6. Resultados

6.1 Conjunto de Dados

O conjunto de dados usados para conduzir o experimento realizado por [1] foram dados referentes a soros, denominado por *fig2data,* e referente a leveduras, denominados por *chodata*. O dado *fig2data* contêm a expressão de 517 genes. Cada gente tem 19 faixas de expressão de dado de 15 minutos até 24 horas. E esses 517 genes podem ser divididos em 10 grupos. O dado *chodata* é conjunto de dados de levedura composto por dados de expressão de 2907 genes e o dado de expressão de cada gene numa faixa de 0 a 160 minutos, representados por 15 intervalos. E esses genes podem ser divididos em 30 grupos. Como o IGKA é um algoritmo estocástico, para cada experimento, foi obtido o resultado da média de 10 independentes usos do algoritmo. A probabilidade de mutação, os números de gerações e os números de populações são fatores que afetam o desempenho e a convergência dos algoritmos FGKA e IGKA. De forma a simplificar tal problema o número de população foi fixado como sendo 50 e o número de gerações, 100.

6.2 Tempo de Perfomance

Foi realizada uma comparação de impacto de desempenho de algoritmo IGKA com o seu predecessor FGKA. De forma óbvia, foi constatado que com o aumento da probabilidade de mutação, o tempo de processamento de cada algoritmo aumentou, de formas diferentes, de acordo com a variação de informação dos dados, como pode ser visto na figura 8. A figura 8 mostra que, quando a probabilidade de

mutação é menor do que certo limite (0.005 para *fig2data* e 0.0005 para *chodata*), o algoritmo IGKA tem um melhor desempenho.

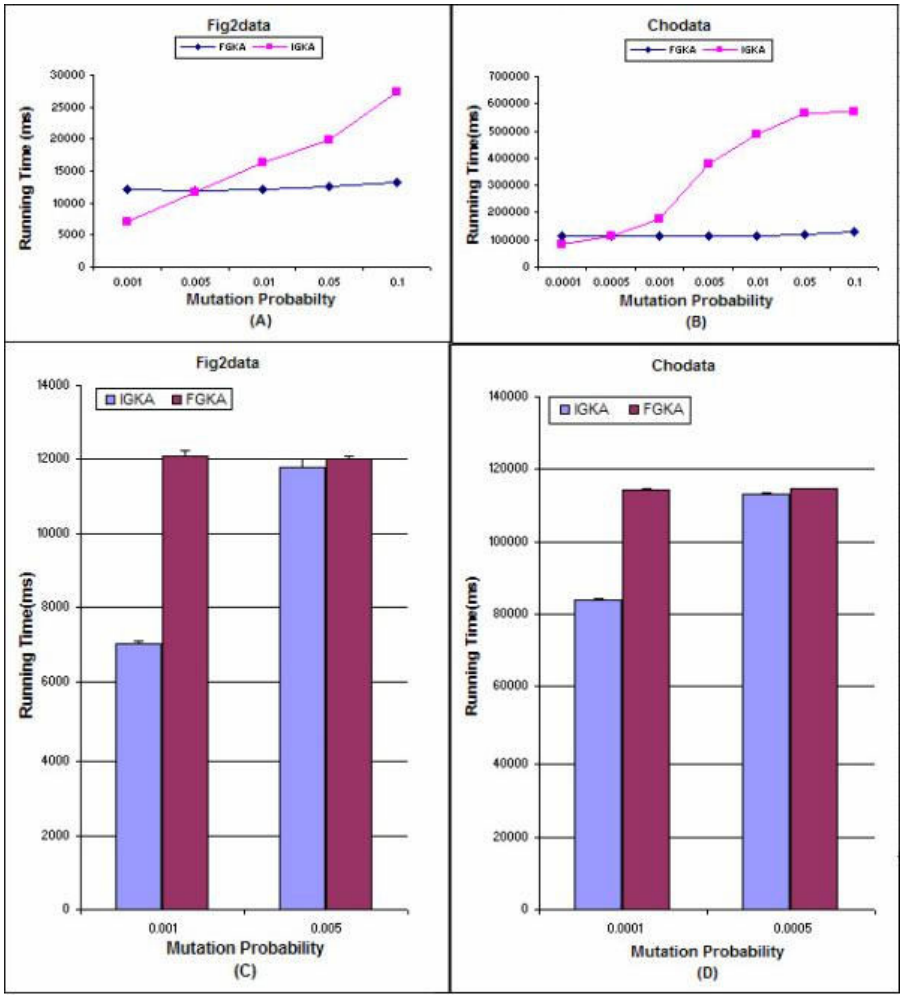

Figura 8: Tempo de Performace x Probabilidade de Mutação, nos dois conjuntos de dados apresentados.

É fácil de ver que o valor limite varia de um conjunto de dados para o outro. O conjunto de dados maior, *chodata*, deve ter uma probabilidade de mutação menor ou igual a 0.0005 para o IGKA superar o FGKA, enquanto que o conjuntos de dados menor, *fig2data*, deve ter uma probabilidade mutação menor ou igual a 0.005 para o IGKA superar o FGKA. De forma geral, o valor de limite depende do número de padrões e do

número de exemplos de cada conjunto de dados, devido a fato de que a perfomance do IGKA está totalmente ligada em quantos padrões são mudados dentro dos componentes de seus grupos. Em conjunto de dados maior, mesmo um número pequeno na probabilidade de mutação pode causar vários membros mudarem de padrão dentro dos seus grupos de participação.

6.3 Combinação

Foi também realizada, uma comparação entre os algoritmos FGKA e IGKA, juntamente com uma combinação de ambos, o HGKA, baseado no tempo de processamento de 100 iterações. A probabilidade de mutação considerada foi de 0.0001 para os três algoritmos.

A figura 9 mostra que, no conjunto de dados *chodata*, o tempo de processamento de cada interação do FGKA é mais estável do que os outros. Em outra perspectiva, o tempo de processamento do IGKA é muito maior do que o FGKA no começo porque existe um maior número de padrões mudam o seu grupo de participação durante o operador de K-means, o que faz com que o IGKA gaste um maior tempo computacional. No entanto, o tempo de processamento de cada iteração do IGKA decresce rapidamente nas iterações finais.

O HGKA combina as vantagens dos dois algoritmos. O ponto de mudança em que o HGKA usa o IGKA em vez do FGKA depende totalmente do conjunto de dados. O resultado mostra que o ponto de mudança nesse caso é na 30ª iteração.

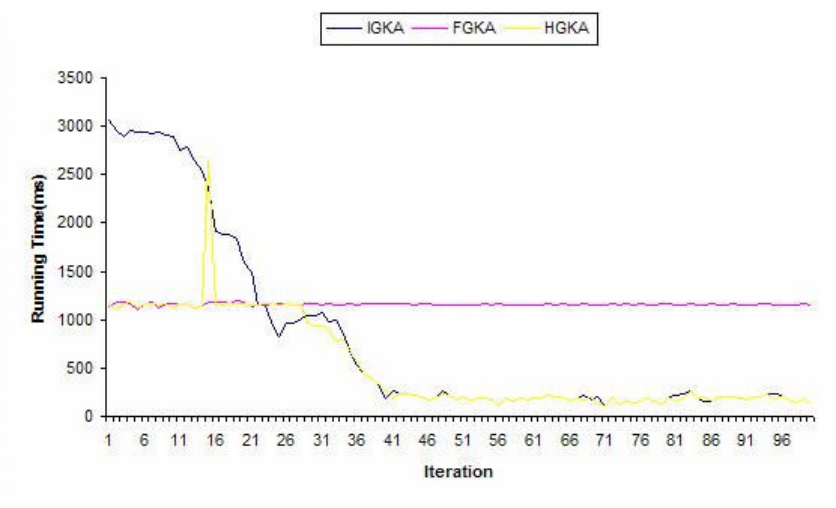

Figura 9: Tempo de perfomance x Iteração,

no algoritmos IGKA, FGKA e HGKA

6.4 Comparação de Convergência do IGKA com FGKA, K-means e SOM

A figura 10 mostra a convergência do IGKA versus FGKA de acordo com diferentes probabilidades de mutação baseadas nos conjuntos de dados fig2data (A), e chodata (B), respectivamente. Ambos os algoritmos tiveram resultados de convergência semelhantes. Quando a probabilidade de mutação muda nesses dois conjuntos de dados, ocorre um pequeno impacto na convergência dos dois algoritmos durante o intervalo mostrado, exceto no caso em que a probabilidade de mutação é muito alta, no FGKA, no conjunto de dados *fig2data*, devido, provavelmente, ao seu menor número de padrões, o que demonstra a segurança de melhor escolher o IGKA com melhor desempenho, sem perder o benefício de convergência.

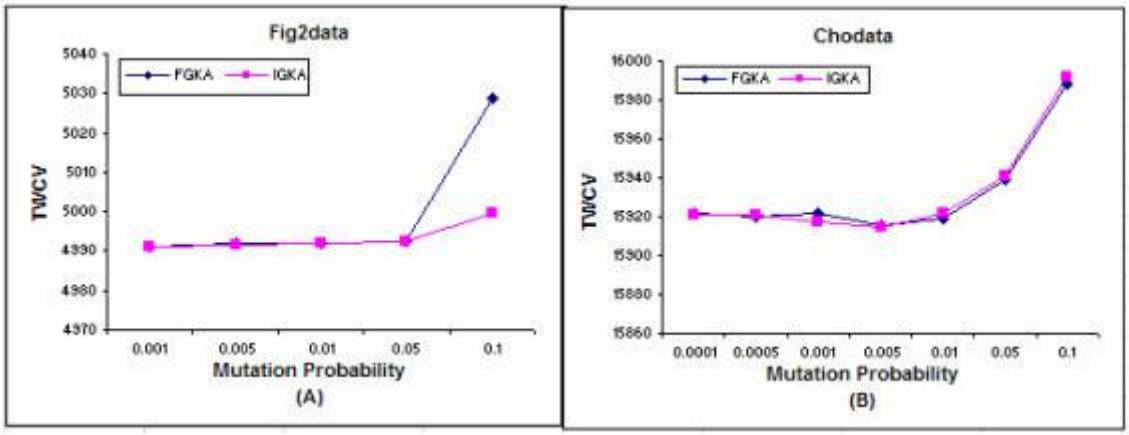

Figura 10: Convergência x Probabilidade de Mutação, entre FGKA e IGKA

Foi também realizada uma comparação de convergência entre o IGKA com o FGKA, K-means e SOM. Cada algoritmo foi tratado com uma caixa preta, e ambos os conjuntos de dados foram alimentados aos algoritmos e o resultados de agrupamento reportados, com o cálculo de TWCVs para cada resultado. Os experimentos do algoritmo K-means e SOM foram realizados a partir de códigos abertos.

A figura 11 mostra uma tabela que demonstra que o IGKA e FGKA possuem resultados de convergência praticamente similares, e uma melhor convergência do que o K-means. A convergência do SOM se mostrou pior em relação aos outros apesar desses quatro algoritmos usarem a distância euclidiana como forma de medição.

Algorithms	Fig2data	Chodata
IGKA (Average of 10 individual runs with generation 100, population 50, mutation probability 0.005 in *fig2data*, and 0.0005 in *chodata*)	4991.53889	16995.7
FGKA (Average of 10 individual runs with generation 100, population 50, mutation probability 0.005 in *fig2data*, and 0.0005 in *chodata*)	4992.13889	16995.4
K-means (Average of 20 individual runs)	5154.21434	17374.6758
SOM (Average of 8 individual runs with different setting)	24805.3661	21660.9049

Figura 11: IGKA x FGKA x K-means x SOM, em convergência.

6.5 Outro Conjunto de Dados

Um novo conjunto de dados foi usado para estudo do algoritmo disponível, IGKA, aqui tratado. Tal conjunto faz parte de uma pesquisa de classificação de dado suplementar, descoberta de subtipos, predição de desfecho em leucemia linfoblástica pediátrica através de perfis de expressão gênica do Hospital de Pesquisa Infantil de St. Jude.

O conjunto de dados final foi formado pela escolha de 100 exemplos de dados de perfis de expressão gênica em micro-arrays de 6 grupos de diagnóstico da doença. Previamente ao processamento do algoritmo, os dados foram devidamente normalizados e reorganizados de forma aleatorizada de forma a obter um resultado mais confiável do procedimento.

O algoritmo funciona tendo como entrada um arquivo de dos, o número de grupos, o número de populações, o número máximo de gerações e a probabilidade de mutação, e como saída, a divisão dos dados em grupo, o Total Within-Cluster Variation (TWCV) e o tempo de processamento. Os valores de população e número máximo de gerações foram fixados. Como mostrado anteriormente, o algoritmo teve um melhor resultado de convergência, menor TWCV, com valores menores de probabilidade de mutação (0.0001 – 0.1). E com valores fixados de número de população e probabilidade de mutação, o algoritmo apresentou, do mesmo modo, uma redução do tempo de processamento das interações finais, com a variação do máximo de gerações (50-100).

7. Conclusão

Com as novas pesquisas dentro da área de biologia molecular, uma enorme quantidade de dados disponível para ser analisada, e como o número de dados de laboratório em biologia molecular cresce exponencialmente, com o passar dos anos, devido ao avanço de técnicas, novos, eficientes e efetivos, métodos de agrupamento foram desenvolvidos para processar esse crescente valor de dados biológicos.

Paralelamente ao avanço de novas técnicas de extração de informação, como a tecnologia de Micro-array, a utilização de algoritmos de agrupamento se mostrou necessária para análise de dados referentes à expressão de genes, no campo da biologia molecular.

Com uma combinação de um dos mais populares métodos de agrupamento existentes e a nova noção de algoritmos genéticos para agrupamento, novos algoritmos surgiram para resolver, de forma bastante eficiente, tal problema.

Referências

[1] Lu, Y., Lu, S., Fotouhi, F., Deng, Y., Brown, S.(2008). *Incremental genetic K-means algorithm and its application in gene expression data analysis.*

[2] Wikipédia. Gene Expression Profiling. Em: http://en.wikipedia.org/wiki/Gene_expression_profiling

[3] Pirooznia M, Yang JY, Yang MQ, Deng Y (2008). *A comparative study of different machine learning methods on microarray gene expression data*.

[4] Wikipédia. DNA Micro-array. Em: http://en.wikipedia.org/wiki/DNA_micro-array

[5] D. Amaratunga, J. Cabrera and V. Kovtun (2008). *Microarray learning with* ABC.

[6] Handbook of Computational Molecular Biology - S Aluru - 2006 - Chapman and Hall/CRC, Cambridge, MA, USA;

[7] Wikipédia. Genectic Algorithm. Em: http://en.wikipedia.org/wiki/Genetic_Algorithm

[8] St. Jude Children's Research Hospital. *Suplemental Data for Classification, Subtype Discovery, and Prediction of Outcome in Pediatric Lymphoblastic Leukemia by Gene Expression Profiling Data Files*.Em: http://www.stjuderesearch.org/data/ALL1/all_datafiles.html

Assinaturas

Kátia Silva Guimarães
(Orientadora)

Paulo Roberto Figueirôa Amorim
(Aluno)