Comunicado Técnico 101
ISSN 1677-8464
Dezembro, 2010
Campinas, SP

Fotos: Embrapa Informática Agropecuária/<http://www.stock.xchng>

Análise de agrupamento de dados de expressão gênica na Rede Genômica Animal

Roberto Hiroshi Higa[1]
Poliana Fernanda Giachetto[2]
Michel Eduardo Beleza Yamagishi[3]
Adriana Mércia Guaratini Ibelli[4]
Luciana Correia de Almeida Regitano[5]
Fernando Flores Cardoso[6]

1 Introdução

O estudo de perfis de expressão gênica relacionadas a manifestações de diferentes fenótipos pode fornecer informações importantes para a compreensão da biologia envolvida nestes processos. Em particular, na agricultura, a identificação dos genes mais relevantes para manifestações de fenótipos de interesse econômico constitui uma etapa importante do processo de melhoramento genético animal e vegetal.

No projeto "Rede Genômica Animal" (Projeto SEG/MP1: 01.06.09.001), a tecnologia escolhida para estudos de expressão gênica é a de microarranjos, que permite mensurar simultaneamente a expressão gênica de milhares de genes de um organismo. Uma parte importante na análise deste tipo de dado consiste em agrupar genes ou amostras em grupos com perfis similares.

Agrupamento de dados é utilizado há décadas em áreas como processamento de imagens e reconhecimento de padrões, sendo popular em análises de microarranjos de expressão gênica. Cada arranjo contém medidas de expressão para milhares de genes que podem ser co-regulados, participando de uma mesma via metabólica, contribuindo para a realização de uma mesma função celular. A análise de agrupamento é utilizada para agrupar genes com padrões de expressão similar, auxiliando especialistas da área biológica a identificar relações funcionais entre eles e, ao mesmo tempo reduzindo a quantidade de informação a ser analisada (BRUN et al., 2005).

A análise de agrupamento também é utilizada para agrupar amostras (arranjos) pela similaridade de seus perfis de expressão. Neste caso, os grupos que resultam da análise de agrupamento podem revelar diferentes mecanismos biológicos contribuindo para a manifestação do fenótipo em análise. Este tipo de análise também pode ser útil para o controle de qualidade dos dados, pois sendo os arranjos réplicas biológicas das populações de interesse, observações que não se agrupam com suas réplicas podem indicar a presença

[1] *Doutor em Engenharia Eletrica, Pesquisador da Embrapa Informática Agropecuária, Campinas, SP, roberto@cnptia.embrapa.br*
[2] *Doutora em Zootécnia, Pesquisadora da Embrapa Informática Agropecuária, Campinas, SP, poliana@cnptia.embrapa.br*
[3] *Doutor em Matemática Aplicada, Pesquisador da Embrapa Informática Agropecuária, Campinas, SP, michel@cnptia.embrapa.br*
[4] *Bolsista CAPESIEmbrapa Informática Agropecuária, Campinas, SP*
[5] *Doutora em Genética e Melhoramento, Pesquisadora da Embrapa Pecuária Sudeste, São Carlos, SP*
[6] *Doutor em Bioinformática, Pesquisador da Embrapa Pecuária Sul, Bagé, RS*

Embrapa

de problemas no desenho experimental e/ou problemas nas fases de pré-processamento ou no processo de hibridização (WIT; MCCLURE, 2004).

O objetivo deste trabalho é apresentar a análise de agrupamento de dados de expressão gênica, conforme realizada no escopo da "Rede Genômica Animal". Na seção 2, são apresentados conceitos básicos sobre análises de agrupamento. Em particular, os algoritmos de agrupamento utilizados nessas análises compreendem aqueles mais conhecidos, como *k*-means, SOM e agrupamento hierárquico. A plataforma de análise escolhida para realização das análises de agrupamento na "Rede Genômica Animal" é o ambiente de análise estatística R (R DEVELOPMENT CORE TEAM, 2010) e o projeto Bioconductor (GENTLEMAN et al., 2004). Assim, na seção 3 é apresentado um exemplo completo de análise de agrupamento e seu respectivo *script* R.

2 Conceitos básicos sobre análise de agrupamento

A análise de agrupamento pode ser vista como um processo composto por três etapas: (i) pré-processamento; (ii) obtenção dos agrupamentos; e (iii) validação dos agrupamentos obtidos. Na etapa (i) são executados procedimentos de seleção e normalização de variáveis e escolhida a distância a ser utilizada na análise de agrupamento. Na etapa (ii) são escolhidos o algoritmo de agrupamento, os valores de seus parâmetros e executado o procedimento de agrupamento, propriamente dito. Finalmente, na etapa (iii) são escolhidas e aplicadas técnicas de validação de agrupamentos, que fornecem uma medida da qualidade dos agrupamentos obtidos. O resultado da análise é criticamente dependente das escolhas feitas nas etapas (i) e (ii), tal que a informação obtida no passo (iii) fornece um indicativo da necessidade ou não de revisão dessas escolhas. Além disso, uma parte importante da análise de agrupamento consiste em fornecer visualizações dos dados que permitam analisar sua estrutura e a coerência com a análise de agrupamento.

Considerando que o número de técnicas que podem ser utilizadas em cada etapa é muito grande, serão abordadas apenas as técnicas efetivamente utilizadas nos scripts. Detalhes sobre as técnicas apresentadas, bem como a apresentação de outras técnicas podem ser encontradas na literatura especializada sobre análise de agrupamento (ANDERBERG, 1973; BRUN et al., 2005; EVERITT et al., 2001; JAIN et al., 1999; XU; WUNSCH, 2005).

2.1 Notação e definições

Os dados de um experimento com microarranjos são representados por uma matriz X de dimensão n x m, onde n representa o número de genes e m é o número de amostras utilizadas no experimento. A linha i da matriz X contém os valores de expressão para o gene gi em cada uma das amostras do experimento, enquanto a coluna j corresponde aos valores de expressão para cada um dos genes considerados no experimento para a amostra j.

Em análise de agrupamento, um grupo corresponde a um conjunto de objetos, que no caso de experimentos de microarranjos é um grupo de genes, $C = \{g_1, .., g_n\}$ ou um grupo de amostras, $C = \{a_1, ..., a_m\}$. O objetivo da análise de agrupamento é particionar o conjunto de objetos sendo analisado em K grupos de objetos, $L = \{C_1, \ldots, C_K\}$, onde cada objeto pertence a apenas um dos K grupos. Note que, quando os objetos considerados na tarefa de agrupamento são genes, cada objeto é caracterizado por uma linha da matriz X, enquanto que no caso de agrupamento de amostras, cada objeto é caracterizado por uma coluna de X.

2.2 Pré-processamento

Na fase de pré-processamento decide-se quais variáveis serão consideradas pelo algoritmo de agrupamento e se elas devem ou não passar por um processo de normalização. A principal decisão refere-se à definição da medida de dissimilaridade, denotada por δ_{ij}, a ser utilizada para comparar os objetos, sendo esta escolha determinada pela aplicação e tipo de dado a ser analisado (categórico, binário ou contínuo). No caso da medida escolhida satisfazer à desigualdade (1) para todos os pares de objetos (i,j), (i,k) e (j,k), ela constitui uma métrica e recebe a denominação de distância.

$$\delta_{ij} + \delta_{ik} >= \delta_{jk}$$

A Tabela 1, lista as medidas de dissimilaridade utilizadas nos *scripts* apresentados na seção 3. Uma lista mais abrangente pode ser encontrada nas referências (ANDERBERG, 1973; BRUN et al., 2005; EVERITT et al., 2001; MARDIA et al., 1979).

2.3 Algoritmos de Agrupamento

Algoritmos de agrupamento podem ser classificados segundo diferentes critérios. Se o critério considerado é a maneira como os grupos são formados, os algoritmos

Tabela 1. Medidas de dissimilaridade utilizadas no scripts apresentados na seção 3.

Medida de dissimilaridade	Fórmula
distância euclidiana	$d_{ij} = \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2}$
distância manhattan	$d_{ij} = \sum_{k=1}^{p} \lvert x_{ik} - x_{jk} \rvert$
dissimilaridade baseda na correlação de Pearson	$d_{ij} = \frac{1}{2}(1 - c_{ij})$, onde: $c_{ij} = \frac{\sum_{k=1}^{p} (x_{ik} - x_{i.})(x_{jk} - x_{j.})}{\sqrt{\sum_{k=1}^{p} (x_{ik} - \overline{x}_{i.})^2 \sum_{k=1}^{p} (x_{jk} - \overline{x}_{j.})^2}}$; e $\overline{x}_{i.} = \frac{1}{p} \sum_{k=1}^{p} x_{ik}$

são classificados em hierárquicos e particionais. Os algoritmos hierárquicos, por sua vez, se subdividem em métodos divisivos, que formam grupos particionando-os sucessivamente, e métodos aglomerativos, que formam grupos pela fusão de grupos menores. Já os algoritmos particionais formam os grupos otimizando uma medida global de qualidade dos grupos. Outra forma de classificação de algoritmos de agrupamento baseia-se na saída que elas produzem. No particionamento hard a saída é uma partição dos dados, com cada objeto pertencendo a um único grupo, enquanto no particionamento *soft* a saída considera um valor de probabilidade ou grau de pertinência para associar objetos com grupos. Os algoritmos utilizados nos *scripts* apresentados na seção 3 pertencem à classe *hard* e incluem os métodos hierárquicos aglomerativo e os particionais *k*-means e *Self Organizing Maps* (SOM). Os conceitos básicos associados a cada um desses algoritmos são apresentados a seguir. Ao leitor interessado em descrições mais detalhadas sobre esses algoritmos, sugere-se os estudos de (ANDERBERG, 1973; BRUN et al., 2005; EVERITT et al., 2001; HAYKIN, 1999; JAIN et al., 1999; XU; WUNSCH, 2005).

2.3.1 Agrupamento hierárquico aglomerativos

Algoritmos de agrupamento hierárquicos aglomerativos constroem agrupamentos fundindo grupos menores de forma iterativa e podem ser sumarizados da seguinte forma (XU; WUNSCH, 2005):

1. Inicialmente, considere N grupos formados por um único objeto (*singleton*), onde N representa o número de objetos. Calcule a matriz de dissimilaridades para os N grupos.
2. Encontre a dissimilaridade mínima entre dois grupos

$$d(c_i, c_j) = \min_{1 \le m,\, l \le N,\, m \ne l} \mathrm{d}(c_m, c_l)$$

   onde d(.,.) é a função dissimilaridade entre grupos; combine os grupos *i* e *j* (*linkage*); Note que a função dissimilaridade pode utilizar diferentes medidas de dissimilaridade (ou distâncias) entre objetos, como as apresentadas na seção 1.1.2.
3. Atualize a matriz de dissimilaridade, calculando as dissimilaridades entre o novo grupo e os demais grupos.
4. Repita os passos 2 e 3 até que todos os objetos estejam em um único grupo.

A utilização de diferentes definições da função dissimilaridade utilizada para combinar grupos resultam em diferentes variantes de algoritmos hierárquicos aglomerativos: *single linkage*, *complete linkage*, *average linkage*, *median linkage*, etc.

2.3.2 Algoritmo *k*-means

O mais simples e mais utilizado algoritmo de agrupamento particional é o *k*-means, que emprega o critério de minimização do erro quadrático:

$$e_s = \sum_{i=1}^{k} \sum_{xj \in Si} \lVert x_j - \mu_i \rVert$$

onde k é o número de grupos, S = {S_1, …, S_k} e μ_i é a média dos pontos, x_j, pertencentes ao grupo Si. O algoritmo é composto por 3 passos [Jain et al.]:

1. escolha k centros para os agrupamentos, que podem coincidir com k pontos escolhidos aleatoriamente ou k pontos definidos aleatoriamente dentro do hipervolume que contém o conjunto de pontos;
2. atribua cada ponto ao centro de agrupamento mais próximo;
3. recalcule o centro dos agrupamentos usando a atribuição atual;
4. se um critério de convergência pré-definido não é satisfeito, vá para o passo 2; Critérios de convergência comumente usados incluem nenhuma ou um número mínimo de re-atribuição de pontos a agrupamentos, decréscimo mínimo no erro quadrático.

É possível obter variações do algoritmo *k*-means utilizando diferentes definições para os centroides (ex: um objeto representativo ou *medoid*) e/ou estratégias de re-atribuição dos objetos (ex: síncrona e assíncrona).

2.3.3 Algoritmo SOM

Self Organizing Maps (SOM) é um tipo de rede neural (HAYKIN, 1999) utilizada para representar dados em alta dimensão por meio de protótipos (pontos representativos dos dados) e visualizá-los em um estrutura de *lattice* de baixa dimensão, usualmente 2D ou 3D. Sua estrutura define a topologia da rede neural SOM, com cada unidade correspondendo a um neurônio.

O processo de aprendizado da rede neural SOM tem como objetivo fazer com que partes diferentes da rede sejam ativadas em resposta a diferentes padrões de entrada. Este processo de especialização é denominado auto-organização, em referência ao fato de que diferentes partes do córtex do cérebro humano são responsáveis pelo tratamento de diferentes tipos dos sinais sensoriais, visual, auditivo, olfativo, etc.

O processo de treinamento de uma rede neural SOM consiste de um processo iterativo, composto pelos seguintes passos (XU; WUNSCH, 2005):

1. Faça t = 0; Defina a topologia da rede e inicialize os protótipos $m_i(0)$, i = 1, …, K, aleatoriamente.
2. Apresente o padrão de entrada **x** para a rede e escolha o neurônio vencedor J mais similar a **x**, J = $\arg_j \min\{||\mathbf{x}-m_j||\}$.
3. Atualize os vetores protótipos na vizinhança de J, $m_i(t+1)=m_i(t) + h_{ci}(t)\{\mathbf{x}-m_i(t)\}$, onde $h_{ci}(t)$ é a função vizinhança do neurônio vencedor (definida pelo *lattice*), usualmente, dada por:

$$h_{ci}(t) = \alpha(t) \exp \frac{- ||r_c - r_i||^2}{2\sigma^2(t)}$$

   onde α(t) é uma função monotonicamente decrescente, denominada função de aprendizado, r_i é a posição do correspondente neurônio i e σ(t) é igual a α(t) se c pertence à vizinhança de J e 0 em caso contrário.
4. Repetir os passos 2 e 3 até que a atualização na posição de nenhum nerônio seja maior que um limiar pré especificado.

Uma rede neural SOM pode ser interpretada de duas formas:

- Uma vez que na fase de treinamento todos os pesos na vizinhança do neurônio vencedor são movidos na mesma direção, neurônios adjacentes tendem a ser excitados por padrões de dados similares. Desta forma, diz-se que SOM forma um mapa semântico, onde amostras similares são mapeadas para a mesma região da rede neural, enquanto padrões dissimilares tendem a ser mapeados para regiões diferentes.
- Os vetores de pesos dos neurônios também podem ser vistos como protótipos no espaço dos dados de entrada, formando uma aproximação discreta da distribuição dos dados utilizados no treinamento da rede SOM. Desta forma, mais protótipos serão utilizados para representar regiões onde a densidade de amostras é alta, enquanto menos protótipos serão utilizados para representar regiões onde as amostras são esparsas.

Note que no contexto de agrupamento os protótipos podem ser interpretados como os centros dos agrupamentos, de forma muito similar aos centroides calculados pelo algoritmo *k*-means.

2.4 Visualização dos dados

A visualização de dados multivariados é um aspecto muito importante da análise de dados, em especial da análise de agrupamento. Ela pode prover pistas sobre a estrutura dos dados, em particular sugerindo que os dados podem ser agrupados e, por vezes, o número aproximado de grupos a ser considerado. Uma técnica simples para visualização de dados, mas amplamente utilizada na análise de agrupamentos, é o *heatmap*, que consiste da apresentação da matriz de dados com uma escala de cores representando os valores dos dados. As linhas (ou colunas) são ordenadas, tal que linhas (colunas) com valores similares aparecem próximas. Os respectivos dendrogramas, correspondentes a agrupamentos hierárquicos das linhas e colunas são apresentados nas margens do gráfico, revelando, simultaneamente, a estrutura hierárquica das linhas e colunas da matriz de dados. Além disso, os *scripts* apresentados na seção 3 também utilizam a Análise de Componentes Principais (PCA) e a Multidimensional Scaling (MDS) para reduzir a dimensionalidade dos dados e visualizá-los em 2D. As definições básicas dessas técnicas são apresentadas a seguir. Ao leitor interessado, sugere-se os trabalhos de (JOLLIFFE, 2002; MARDIA et al., 1979; SAMMON JUNIOR, 1969) para uma descrição detalhada.

2.4.1 Análise de componentes principais (PCA)

A análise de componentes principais permite que se reduza a dimensionalidade de um conjunto de dados que contém um grande número de variáveis inter-relacionadas, retendo o máximo da variância contida nos dados (JOLLIFFE, 2002). Para isso, utiliza-se uma transformação linear para expressar os dados por meio de um novo conjunto de variáveis não correlacionadas, com as novas variáveis ordenadas de forma que as primeiras contém a maior parte da variação presente no conjunto de variáveis originais. Especificamente, se **x** é um vetor aleatório n-dimensional com média μ e matriz de covariância **Σ**, os componentes principais são

obtidos pela transformação $\mathbf{y} = T^t(\mathbf{x}-\boldsymbol{\mu})$, onde **T** é ortogonal, $\boldsymbol{\Lambda} = \mathbf{T}^t\boldsymbol{\Sigma}\mathbf{T}$ é diagonal, tal que $\boldsymbol{\lambda}_1 >= \lambda_1 >= \ldots = \lambda_1$ (MARDIA et al., 1979). As variáveis que resultam desta transformação linear são conhecidas como componentes principais e a redução de dimensionalidade se dá pela retenção dos poucos componentes principais que explicam a maior parte da variância. Usualmente, para visualização de um conjunto de dados, são retidas 2 ou 3 componentes principais.

2.4.2 Multidimensional scaling (MDS)

Um algoritmo de MDS determina uma configuração no R^n para um conjunto de pontos no R^m, n << m, de forma a preservar o máximo possível as distâncias entre os objetos. Em geral, utiliza-se n = 2 ou 3, tal a coleção de objetos pode ser inspecionada visualmente. Dada uma coleção I de objetos, com a distância entre cada par de objetos sendo denotada por δ_{ij}, i,j = 1,..., I, o objetivo de MDS é encontrar I vetores, $x_1, \ldots, x_I$, pertencentes ao Rn , tal que $||x_i - x_j||$ seja aproximadamente igual a δ_{ij}. Existem diversas abordagens para MDS, em geral, formulados como um problema de otimização, sendo a solução obtida por meio de métodos numéricos. Neste sentido, um dos métodos mais simples é o proposto por Samon Junior (1969). Já a denominada solução clássica para o problema de MDS baseia-se na determinação dos autovalores e autovetores da matriz **HAH**, onde **A** é a matriz de distâncias e **H** é a matriz centrar (centring matrix) (MARDIA et al., 1979).

2.5 Medidas para validação de agrupamentos

Procedimentos para avaliação dos resultados obtidos ao se utilizar um algoritmo de agrupamento são conhecidos como *cluster validity*. Em geral, essas abordagens são divididas em 3 classes (HANDL et al., 2005):

- métodos externos, que avaliam o agrupamento obtido em uma estrutura pré-especificada, que reflete um conhecimento "externo" sobre a estrutura de grupos existente nos dados,, em geral, significando que os rótulos dos dados são conhecidos a priori;
- métodos internos, que avaliam o agrupamento utilizando apenas os próprios dados e a matriz de distância entre eles, em geral, privilegiando grupos compactos e separados uns dos outros; e
- métodos relativos, que avaliam o agrupamento comparando-o com outros agrupamentos que resultam da utilização do mesmo algoritmo, mas com diferentes parâmetros de entrada.

As medidas de validade de agrupamentos utilizadas nos scripts apresentados na seção 3 pertencem ao grupo de medidas internas: índice de Dunn, a Silhouette width e conectividade (HANDL et al., 2005). No caso do agrupamento hierárquico, utiliza-se a correlação cofenética (EVERITT et al., 2001) para avaliar a distorção entre as relações de dissimilaridade original e a imposta pela hierarquia obtida. A seguir, são apresentadas as definições básicas dessas medidas. Para maiores detalhes, sugere-se ao leitor interessado os estudos de (EVERITT et al., 2001; HALKIDI et al., 2002a, 2002b; HANDL et al., 2005).

O índice de Dunn é definido como a razão entre a mínima distância entre dois grupos e o tamanho do maior grupo:

$$V(C) = \frac{\min_{h,k=1,\ldots,K;\, h\neq k} d(C_k, C_h)}{\max_{k=1,\ldots,K} \Delta(C_{kl})}$$

onde $C = \{C_1, \ldots, C_K\}$, $d(C_k, C_h)$ é a distância entre dois grupos e $\Delta(C_k)$ é o tamanho do grupo C_k. O valor de V(C) depende da medida de distância entre grupos e o tamanho dos grupos, compreendendo uma faixa de valores entre 0 e ∞. Quando maior o valor de V(C), melhor a qualidade do agrupamento.

A *silhouette width* é definida pela média dos valores de *silhouette value*, S(i), para cada objeto i, dada por:

$$S(i) = \frac{b(i) - \mathrm{a}(i)}{\max\{a(i), \mathrm{b}(i)\}}$$

onde ai é distância média entre o objeto i e todos os objetos no mesmo grupo e bi é o mínimo da distância média entre i e os objetos dos demais grupos ou, equivalentemente, a distância média entre i e os objetos do grupo vizinho mais próximo. A silhouette width assume valores no intervalo [-1,1], sendo que valores próximos de 1 indicam menor distorção das relações de dissimilaridade original e a imposta pela hierarquia obtida.

A conectividade para um conjunto de dados com N objetos de dimensionalidade M, agrupados em K grupos, $C = \{C_1, \ldots, C_K\}$, é definida por:

$$\mathrm{Con}(C) = \sum_{i=1}^{N} \sum_{j=1}^{M} x_{i,m_{i(j)}}$$

onde $nn_{i(j)}$ é o j-ésimo padrão mais próximo do objeto i, $x_{i,nni(j)}$ é igual a zero se i e j pertencem ao mesmo grupo e 1/j em caso contrário. A conectividade assume valores entre 0 e ∞, tal que quanto menor seu valor, melhor é a qualidade do agrupamento.

Finalmente, coeficiente de correlação cofenética é definido pelo coeficiente de correlação entre a matriz de distâncias dos dados originais e a matriz formada pelas correspondentes distâncias cofenéticas, onde a distância cofenética entre dois objetos que foram agrupados hierarquicamente é definida pela altura no dendrogra-

ma em que os dois objetos passam a fazer parte do mesmo grupo.

3 Exemplo de análise de agrupamento utilizando R

Para ilustrar a análise de agrupamentos será utilizado um conjunto de dados de um experimento de extremos de resistência a carrapatos, analisados no âmbito da Rede Genômica Animal (IBELLI et al., 2010). Foram utilizadas amostras de linfonodo (LN) de bovinos de 4 grupos genéticos (Nelore, Angus x Nelore, Simental x Nelore e Canchim x Nelore) previamente classificados em resistentes (R) e sensíveis (S). Os dados de expressão gênica foram obtidos utilizando-se a plataforma Affymetrix. Os pacotes R Affy e Maanova foram utilizados para realização dos processos de controle de qualidade, correção de background, normalização e pacote análise de expressão diferencial (CARDOSO, 2009). Utilizando-se 36 arranjos que satisfizeram aos critérios de controle de qualidade, a final da análise de expressão diferencial identificou 393 genes diferencialmente expressos (q-valor < 0,05 e fold change > 0,5 para mais ou para menos). Desta forma, os dados considerados na análise de agrupamento podem ser representados por uma matriz com 393 linhas, correspondentes aos genes, e 36 colunas, correspondentes às amostras utilizadas no experimento. O script R completo utilizado nesta análise encontra-se em anexo.

Supondo que esses dados encontram-se em um arquivo em formato csv, o seguinte comando R carrega os dados em uma estrutura denominada data.frame.

```
>dts <- data.frame(read.csv(file = dados.csv, header
= TRUE, row.names=1))
>
```

Inicialmente, pode-se utilizar a técnica de *multimensional scaling* para visualizar as amostras dados em 2 dimensões. Note que cada amostra é representada por um vetor 393-dimensional, onde cada dimensão refere-se ao nível de expressão de um gene.

```
> ndts <- scale(t(dts))
> library("MASS")
> distm <- dist(ndts, method = "euclidian")
> iso <- isoMDS(distm, tol=1e-5)
> plot(iso$points, xlab = "dim 1", ylab = "dim 2",
main = "Amostras")
> text(iso$points, label = rownames(ndts))
>
```

A primeira linha de comando padroniza os valores de expressões dos genes do conjunto de dados, tal que cada gene possui média igual zero e variância igual a 1 e armazena a matriz transposta de dts em uma variável denominada ndts. As próximas duas linhas executam a preparação para execução do procedimento de MDS, carregando as bibliotecas necessárias e calculando a matriz de distâncias entre cada amostra do conjunto de dados. A distância utilizada é a euclidiana, mas outras como a manhathan ou minkowski também poderiam ser utilizadas. Por fim, as duas últimas linhas executam o procedimento de MDS, guardando o resultado numa variável denominada iso, que é plotada pela linha seguinte.

A Figura 1 apresenta o gráfico 2D obtido pela utilização de MDS. As amostras de animais sensíveis (S) aparecem preferencialmente na parte inferior esquerda do gráfico enquanto os animais resistentes (R) ocupam toda a parte direita. Ela sugere a existência de dois grupos, estando em acordo com o desenho experimental, mas também mostra que algumas amostras como S_20_NILN e S_256_TALN, estão mais próximas de amostras do grupo R que do grupo S e, portanto, merecem uma atenção especial. Uma análise de componentes principais (PCA) produz uma Figura com as mesmas informações. De fato, utilizando MDS com distância euclidiana é equivalente a PCA.

Para gerar o agrupamento hierárquico das amostras utilizando o comando hclust. Note que a mesma matriz de distâncias (euclidiana) foi utilizada, em conjunto com o método de combinação de grupos *average linkage*.

```
> hcl <- hclust(distm, method = "average")
> disth <- cophenetic(hcl)
> cor(distm, disth)
[1] 0.6984863
> plot(hcl, main = "Agrupamento das amostras", xlab
= "Amostras", ylab = "Distância")
>
```

A Figura 2 apresenta o dendrograma correspondente às amostras. Ela corrobora o resultado observado na análise de multimensional scaling, indicando a presença de dois grupos (S e R), com a presença de algumas amostras rotuladas com prefixo S entre as amostras rotuladas com o prefixo R. Note também que o valor de correlação cofenético de aproximadamente 0,7 indica um nível de distorção entre as matrizes de distância original e obtida a partir do dendrograma.

Para gerar uma agrupamento do tipo particional utilizando k-means é preciso especificar o número de grupos desejado. Assim, embora as duas análises precedentes indiquem a existência de dois grupos, pode-se utilizar as medidas de validação apresentadas na seção 2.5 para avaliar o número de grupos mais apropriado.

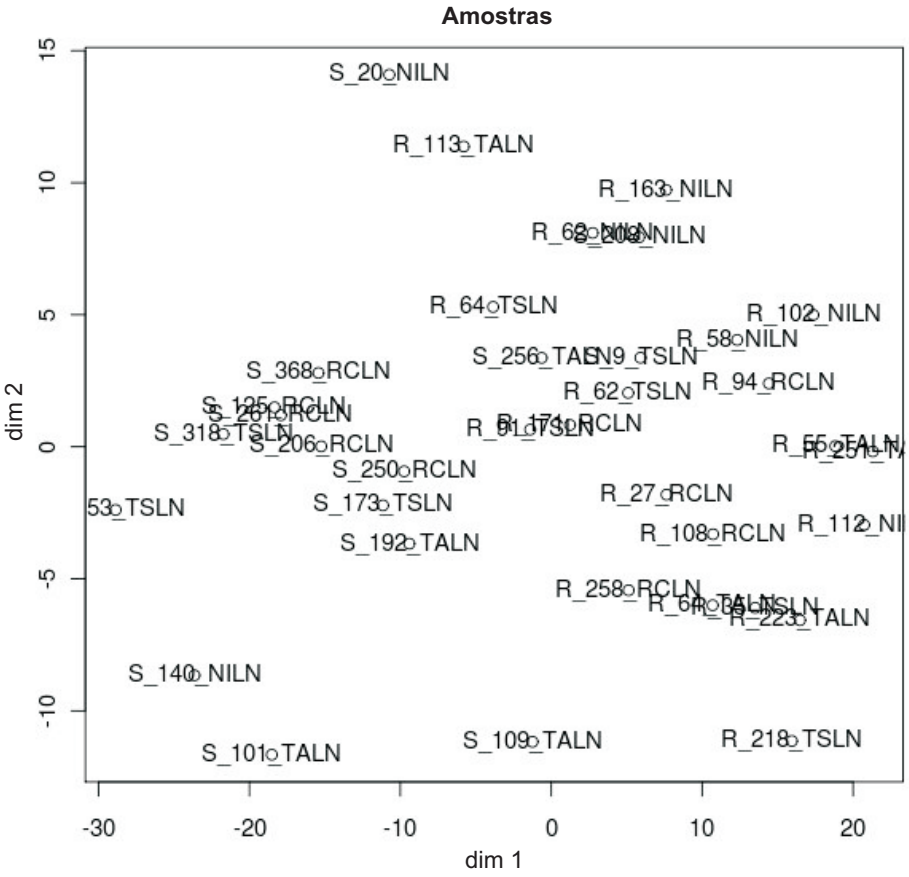

Figura 1. Multidimensional scaling de amostras.

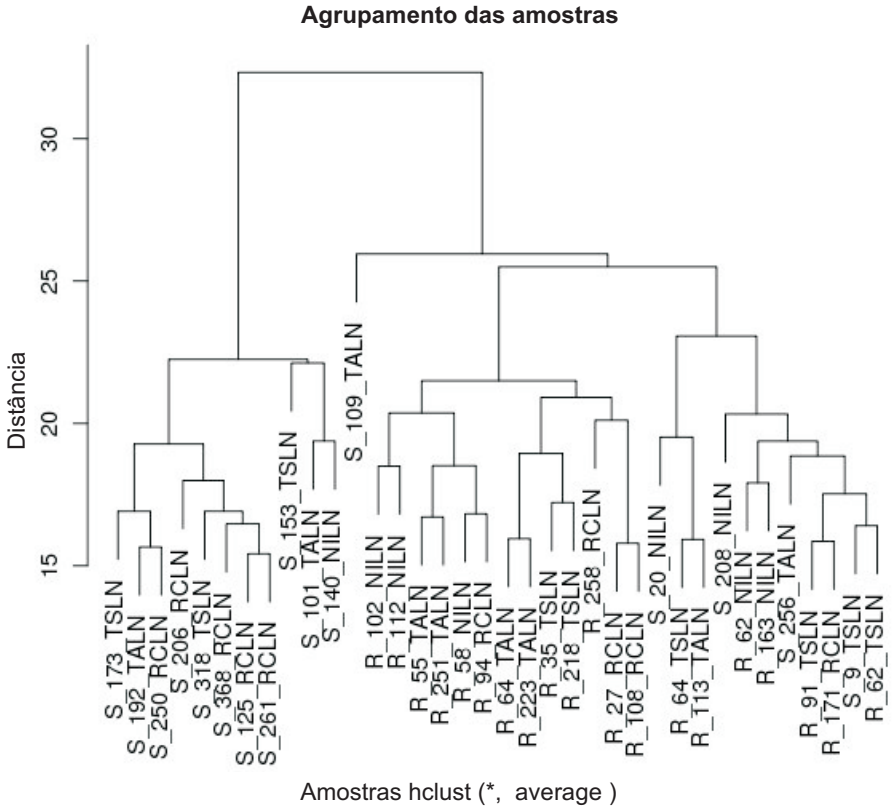

Figura 2. Agrupamento hierárquico das amostras.

```
> library(clValid)
> intern <- clValid(ndts, 2:5, clMethods =
"kmeans", validation = "internal")
> summary(intern)

Clustering Methods:
 kmeans

Cluster sizes:
 2 3 4 5

Validation Measures:
                     2       3       4       5

kmeans Connectivity 8.1865 13.7845 21.9246 29.8790
       Dunn         0.5311  0.6082  0.6170  0.6265
       Silhouette   0.2925  0.1864  0.1463  0.1335

Optimal Scores:

                Score     Method    Clusters
Connectivity   8.1865     kmeans           2
Dunn           0.6265     kmeans           5
Silhouette     0.2925     kmeans           2
```

Os comando R acima utilizam a biblioteca clValid para testar o número de grupos mais apropriado para utilização com k-means. Pode-se observar que as medidas conectividade e silhouette concordam com as análises precedentes, enquanto a medida de Dunn aumenta com o número de grupos considerados. Neste caso, opta-se por utilizar k-means para separar as amostras em dois grupos. Além disso, note que o bloco de comandos abaixo também lista as amostras pertencentes a cada um dos grupos encontrados por k-means.

```
> ncluster <- 2
> kcl <- kmeans(ndts, ncluster, nstart = 30)
> cl <- sort(kcl$cluster)
> for (i in 1:ncluster) {
+    idx <- cl == i
+    gi <- names(cl[idx])
+    msg <- paste("grupo", as.character(i), ":")
+    print(msg)
+    print(gi)
+ }
[1] "grupo 1 :"
 [1] "S_20_NILN"  "R_64_TSLN"  "S_101_TALN"
"R_113_TALN" "S_125_RCLN"
 [6] "S_140_NILN" "S_153_TSLN" "S_173_TSLN"
"S_192_TALN" "S_206_RCLN"
[11] "S_250_RCLN" "S_261_RCLN" "S_318_TSLN"
"S_368_RCLN"
[1] "grupo 2 :"
 [1] "S_9_TSLN"   "R_27_RCLN"  "R_35_TSLN"  "R_55_
TALN"  "R_58_NILN"
 [6] "R_62_TSLN"  "R_62_NILN"  "R_64_TALN"  "R_91_
TSLN"  "R_94_RCLN"
[11] "R_102_NILN" "R_108_RCLN" "S_109_TALN"
"R_112_NILN" "R_163_NILN"
[16] "R_171_RCLN" "S_208_NILN" "R_218_TSLN"
"R_223_TALN" "R_251_TALN"
[21] "S_256_TALN" "R_258_RCLN"
>
```

Para realizar a análise de agrupamentos dos genes, utiliza-se praticamente o mesmo conjunto de comandos considerando a matriz de dados transposta.

```
>ndts <- t(ndts)
```

Deve-se ressaltar que, devido ao grande número de genes analisados, nem sempre é produtivo produzir os mesmos gráficos apresentados acima. Outro aspecto a ser ressaltado é que uma medida de dissimilaridade mais apropriada para analisar genes é baseada no conceito de correlação (Tabela 1). Por fim, deve-se ressaltar que o gráfico de heatmap, que apresenta simultaneamente a matriz de dados e os gráficos hierárquicos de genes e amostras pode fornecer informações adicionais muito interessantes.

```
> library(Heatplus)
> dcor <- cor(ndts)
> distm <- as.dist((1 - dcor)/2)
> hcl1 <- hclust(distm, method = "average")
> hcl1$dist.method <- "correlation"
> distm <- dist(ndts, method = "manhattan")
> hcl2 <- hclust(distm, method = "euclidian")
> heatmap_2(as.matrix(ndts), col=RGBColVec(64),
legend=1, legfrac=10, scale = "row", Rowv =
as.dendrogram(hcl2), Colv = as.dendrogram(hcl1))
>
```

A Figura 3 apresenta o heatmap para o conjunto de dados. Observando, simultaneamente, a matriz de dados e os dendrogramas de genes e amostras tornam-se aparentes os padrões de expressões que definem os grupos obtidas anteriormente tanto para os genes quanto para as amostras.

4 Discussão

A análise de agrupamento constitui uma etapa importante do processo de análise de expressão gênica utilizando a tecnologia de microarranjos. Contudo, sua realização pode tornar-se um problema para o pesquisador não familiarizado com as diferentes técnicas envolvidas e ferramentas computacionais potencialmente úteis.

Este trabalho abordou a análise de agrupamento de dados de expressão gênica, conforme realizado no escopo da "Rede Genômica Animal". Foram apresentados os conceitos básicos das técnicas envolvidas nesta análise e um exemplo de análise em que a utilização do script desenvolvido para a plataforma de análise estatística R foi ilustrado. O *script* completo é apresentado em anexo.

Assim, com a disseminação do script apresentado neste trabalho, espera-se outros pesquisadores interessados em realizar análises de agrupamento em dados de expressão gênica, mas não familiarizados com este tipo de analise estejam aptos a realizar estas análises

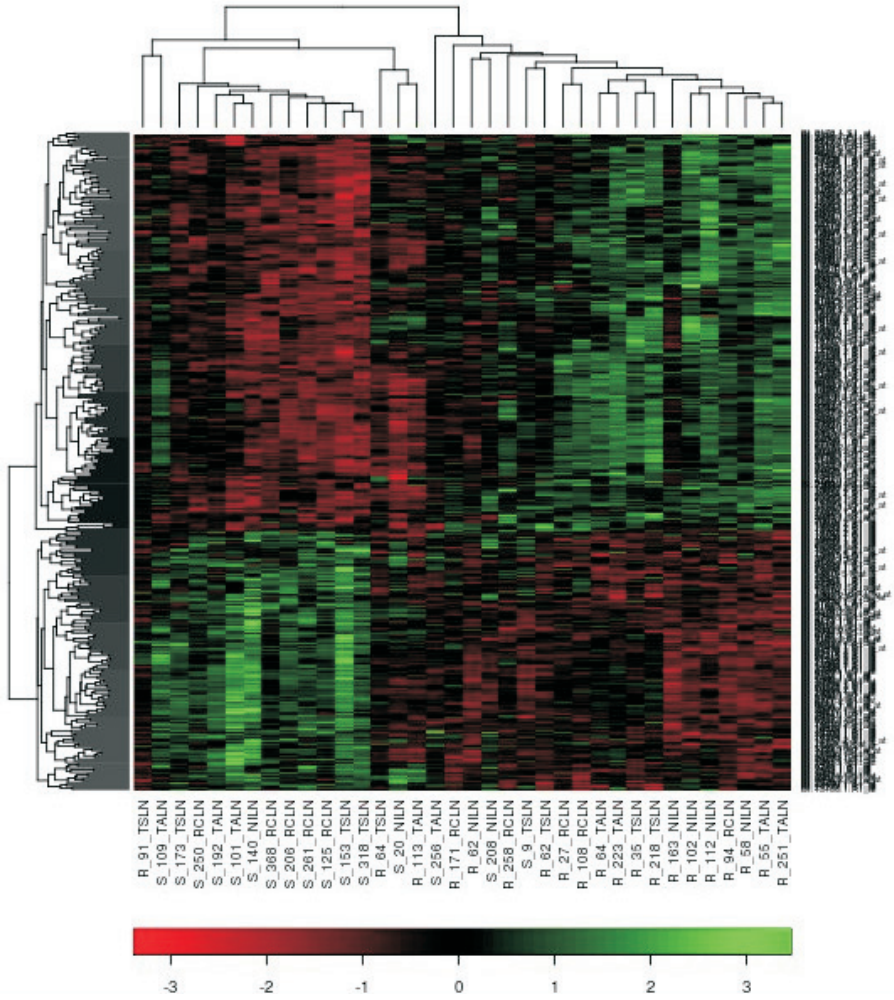

Figura 3. Heatmap apresentando simultaneamente agrupamentos de amostras e genes.

re-utilizando o script apresentado, bem como adaptando-o às suas necessidades específicas.

5 Anexo: *Script* completo para análise de agrupamento

```
# Suposicao: arquivo "dados.csv" contém dados de
experssão gênica em formato csv,
# onde cada linha representa um gene e cada coluna
uma amostra.

dts <- data.frame(read.csv(file = dados.csv, header
= TRUE, row.names=1))
probes_id <- rownames(dts)

# Obtem informacao  sobre a identidade dos genes
DE.
library("annotate")
library("bovine.db")

# Gene symbols.
gene_symbol_list <- mget(probes_id, bovineSYMBOL)
gene_symbol_list[is.na(probes_id)] = "Nao anotado"

# Nova escala dos dados (opcional) e obtem a matriz
transposta.
ndts <- scale(t(dts))
# Usa gene symbols como nome de colunas.
colnames(ndts) <- gene_symbol_list

###################### Amostras ##################

# Visualiza das Amostras - MDS

library("MASS")

# Distancia: manhattan.
distm <- dist(ndts, method = "euclidian")
# isoMDS
iso <- isoMDS(distm, tol=1e-5)
# Plota grafico.
plot(iso$points, xlab = "dim 1", ylab = "dim 2",
main = "Amostras")
text(iso$points, label = rownames(ndts))

# Visualiza das Amostras - PCA

pcs <- prcomp(ndts, retx = TRUE, scale = TRUE)
summary(pcs)
plot(pcs, main = "PCs")
# Plota grafico.
plot(pcs$x[,1], pcs$x[,2], xlab = "PC1", ylab =
"PC2",  main = "Amostras")
text(pcs$x[,1], pcs$x[,2], rownames(pcs$x))

# Cluster hierarquico das Amostras
```

```
distm <- dist(ndts, method = "euclidian")
hcl <- hclust(distm, method = "complete")
# Correlacao cophenetica
disth <- cophenetic(hcl)
corrcoph <- cor(distm, disth)
corrcoph
# Plota gráfico.
plot(hcl, main = "Agrupamento das amostras", xlab
= "Amostras", ylab = "Distância")

# k-means

library(clValid)

# Verifica numero de clusters.
intern <- clValid(ndts, 2:5, clMethods = "kmeans",
validation = "internal")
summary(intern)

# Executa k-means com numero de cluster
determinado.
kcl <- kmeans(ndts, 3, nstart = 30)
cl <- sort(kcl$cluster)
for (i in 1:3) {
 idx <- cl == i
 gi <- names(cl[idx])
 msg <- paste("grupo", as.character(i), ":")
 print(msg)
 print(gi)
}

###################### genes ##################

ndts <- t(ndts)     # transposta.

#  MDS

library("MASS")

# Distancia: manhattan.
distm <- dist(ndts, method = "manhattan")
# MDS (isoMDS)
iso <- isoMDS(distm, tol=1e-5)
plot(iso$points, xlab = "dim 1", ylab = "dim 2",
main = "Genes DE")
text(iso$points, label = rownames(ndts))

#  PCA

pcs <- prcomp(ndts, retx = TRUE, scale = TRUE)
summary(pcs)
plot(pcs, main = "PCs")
plot(pcs$x[,1], pcs$x[,2], xlab = "PC1", ylab =
"PC2",  main = "Genes DE")
text(pcs$x[,1], pcs$x[,2], rownames(pcs$x))

# Cluster hierarquico dos genes.

#(1 - Correlacao de Pearson)/2.
dcor <- cor(t(ndts))
distm <- as.dist((1 - dcor)/2)
hcl <- hclust(distm, method = "complete")
# Correlacao cophenetica
disth <- cophenetic(hcl)
corrcoph <- cor(distm, disth)
corrcoph
# Plota gráfico.
plot(hcl, main = "Agrupamento dos genes DE", xlab
= "Genes DE", ylab = "Distância")

# SOM

library(kohonen)

m <- as.matrix(ndts)
som.genes <- som(m,rlen = 20000,  alpha = c(0.1,
0.001), keep.data = TRUE, grid = somgrid(2,2,
"rectangular"))
summary(som.genes)
plot(som.genes, main = "SOM para genes DE", type =
"dist.neighbours")

winning <- som.genes$unit.classif
som_lab <- rep("", 4)
n <- dim(ndts)[1]
for (i in 1:n) {
 som_lab[winning[i]] <- paste(som_lab[winning[i]],
rownames(ndts)[i])
}
som_lab

# Dendrograma dos protótipos.
dprot <- dist(som.genes$codes)
hcl <- hclust(dprot, method = "average")
plot(hcl, main = "Dendrograma de SOM prototypes", ,
xlab = "SOM prototypes", ylab = "Distância")

# k-means.

library(clValid)

# Verifica número de clusters.
intern <- clValid(ndts, 2:5, clMethods = "kmeans",
validation = "internal")
summary(intern)
# Executa k-means com número de clusters
determinado.
kcl <- kmeans(ndts, 2, nstart = 30)
cl <- sort(kcl$cluster)
for (i in 1:2) {
 idx <- cl == i
 gi <- names(cl[idx])
 msg <- paste("grupo", as.character(i), ":")
 print(msg)
 print(gi)
}

# Heatmap

library(Heatplus)

# Genes - colunas.
#(1 - Correlacao de Pearson)/2.
dcor <- cor(ndts)
distm <- as.dist((1 - dcor)/2)
hcl1 <- hclust(distm, method = "average")
hcl1$dist.method <- "correlation"

# Amostras - linhas.
distm <- dist(ndts, method = "euclidian")
hcl2 <- hclust(distm, method = "complete")
heatmap_2(as.matrix(ndts), col=RGBColVec(64),
legend=1, legfrac=10, scale = "row", Rowv =
as.dendrogram(hcl2), Colv = as.dendrogram(hcl1))
```

6 Referências

ANDERBERG, M. R. **Cluster analysis for applications**. New York: Academic Press,1973. 353 p. (Probability and Mathematical Statistics, 19).

BRUN, M.; JOHNSON, C. D.; RAMOS, K. S. Clustering: revealing intrinsic dependencies in microarray data. In: DOUGHERTY, E. R.; SHMULEVICH, I.; CHEN, J.; WANG, J. (Ed.). **Genomic signal processing and statistics**. New York: Hindawi Publishing Corporation, 2005. p. 129-162.

CARDOSO, F. F. **Métodos para análise de micro-arranjos de oligonucleotídeos em estudos de expressão gênica**. In: REUNIÃO ANUAL DA REGIÃO BRASILEIRA DA SOCIEDADE INTERNACIONAL DE BIOMETRIA, 54.; SIMPÓSIO DE ESTATÍSTICA APLICADA À EXPERIMENTAÇÃO AGRONÔMICA, 13., 2009. **Programa e resumos**... São Carlos, SP: UFSCar; Embrapa Pecuária Sudeste, 2009. p. 28.

EVERITT, B. S.; LANDAU, S.; LEESE, M. **Cluster analysis**. 4th ed. London: Arnold; New York: Oxford University, 2001. 237 p.

GENTLEMAN, R. C.; CAREY, V. J.; BATES, D. M.; BOLSTAD, B.; .DETTLING, M.; DUDOIT, S.; ELLIS, B.; GAUTIER, L.; GE, Y.; GENTRY, J.; HORNIK, K.; HOTHORN, T.; HUBER, W.; IACUS, S.; IRIZARRY, R.; LEISCH, F.; LI, C.; MAECHLER, M.; ROSSINI, A. J.; SAWITZKI, G.; SMITH, C.; SMYTH, G.; TIERNEY, L.; YANG, J. Y.; ZHANG, J. Bioconductor: open software development for computational biology and bioinformatics. **Genome Biology**, v. 5, n. 10, 2004. R80.

HALKIDI, M.; BATISTAKIS, Y.; VAZIRGIANNIS, M. Cluster validity methods: part I. **SIBMOD Record**, v. 31, n. 2. p. 40-45, 2002a.

HALKIDI, M.; BATISTAKIS, Y.; VAZIRGIANNIS, M. Cluster validity methods: part II. **SIBMOD Record**, v. 31, n. 3. p. 19-27, 2002b.

HANDL, J.; KNOWLES, J.; KELL, D. B. Computational cluster validation in post-genomic data analysis. **Bioinformatics**, v. 21, n. 15. p. 3201-3212, 2005.

HAYKIN, S.**Neural networks**: a comprehensive foundation. 2nd. ed. Upper Saddle River: Prentice Hall [1999]. 842 p.

IBELLI, A. M. G.; HIGA, R. H.; GIACHETTO, P. F.; YAMAGISHI, M. E. B.; OLIVEIRA, M. C. S.; CARDOSO, F. F.; ALENCAR, M. M.; REGITANO, L. C. A. Genes e vias metabólicas envolvidas nos mecanismos de resistência e susceptibilidade de bovinos infestados com carrapato Rhipicephalus microplus. In: CONGRESSO BRASILEIRO DE GENÉTICA, 56., 2010, Guarujá. **Resumos**... Ribeirão Preto: Sociedade Brasileira de Genética, 2010. p. 74.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. **ACM Computing Surveys**, v. 31, n. 3. p. 264-323, 1999.

JOLLIFFE, I. T. **Principal component analysis**. 2nd ed. New York: Springer, 2002. 487 p. (Springer series in statistics).

MARDIA, K. V.; KENT, J. T.; BIBBY, J. M. **Multivariate analysis**. London: Academic Press, 1979. 521 p. (Probability and Mathematical statistics: a Series of Monographs and textbooks).

R DEVELOPMENT CORE TEAM. **R**: a language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing, 2010. Disponível em: <http://www.R-project.org/>. Acesso em: 20 ago. 2010.

SAMMON JUNIOR, J. W. A nonlinear mapping for data structure analysis. **IEEE Transactions on Computers**, v. C18, n. 5, p. 401-409, 1969.

WIT, E.; MCCLURE, J. Statistics for microarrays; design, analysis and inference. Chichester, England; Hoboken: John Wiley, 2004. 265 p.

XU, R.; WUNSCH, D. Survey of clustering algorithms. **IEEE Transactions on Neural Networks**, v. 16, n. 3. p. 645-678, 2005.

Comunicado Técnico, 101

Embrapa Informática Agropecuária
Endereço: Caixa Postal 6041 - Barão Geraldo
13083-886 - Campinas, SP
Fone: (19) 3211-5700
Fax: (19) 3211-5754
http://www.cnptia.embrapa.br
e-mail: sac@cnptia.embrapa.com.br

Ministério da Agricultura, Pecuária e Abastecimento

1ª edição on-line - 2010

CGPE 9086