# Coeficientes de Determinação, Predição Intrinsicamente Multivariada e Genética

Carlos Henrique Aguena Higa

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO DE MESTRE
EM
CIÊNCIAS

Área de Concentração: Ciência da Computação
Orientador: Prof. Dr. Ronaldo Fumio Hashimoto

Durante a elaboração deste trabalho o autor recebeu auxílio financeiro da FAPESP

São Paulo, 17 de novembro de 2006.

# Coeficientes de Determinação, Predição Intrinsicamente Multivariada e Genética

Este exemplar corresponde à redação final da dissertação devidamente corrigida e defendida por Carlos Henrique Aguena Higa e aprovada pela Comissão Julgadora.

São Paulo, 21 de dezembro de 2006.

Banca Examinadora:

Prof. Dr. Ronaldo Fumio Hashimoto (orientador) - IME/USP

Prof. Dr. Roberto Marcondes Cesar Jr. - IME/USP

Prof. Dr. Hernando Antonio del Portillo - ICB/USP

*Aos meus pais.*

# Agradecimentos

Em primeiro lugar agradeço ao meu orientador, Prof. Dr. Ronaldo Fumio Hashimoto, por sua dedicação e orientação que tornou possível a realização deste trabalho.

Agradeço também ao Prof. Dr. Hugo Aguirre Armelin, juntamente com o Dr. Fábio Nakano e a Drª Paula F. Asprino, que colaboraram para a realização da pesquisa.

Agradeço especialmente a minha família pelo apoio e compreensão durante todo o tempo em que me dediquei a este trabalho. Aos meus pais por me ensinarem o significado da palavra "sacrifício"; palavra que torna nosso trabalho, de certa forma, mais valioso. Aos familiares: vovó Aguena, tios e tias, primos e primas, e pirralhos (Diguinho, Renan, Natty Patty, Vivi Patty e Let Patty). Muito obrigado pelo apoio.

Aos meus amigos, em especial os amigos do DCT/UFMS pelo companheirismo desde os tempos de graduação.

E agradeço a FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo, por ter concedido o apoio financeiro ao nosso trabalho.

# Conteúdo

**Agradecimentos** **ii**

**Conteúdo** **i**

**Lista de Figuras** **ii**

**Lista de Tabelas** **iii**

**Resumo** **1**

**Abstract** **3**

**1 Introdução** **5**

1.1 Motivação . . . . . 5

1.2 Objetivos . . . . . 6

1.3 Organização do Trabalho . . . . . 6

**2 A Tecnologia de Microarrays** **8**

2.1 Os genes e a sua importância . . . . . 8

2.2 Microarray . . . . . 10

2.2.1 Análise e processamento da imagem . . . . . 11

2.2.2 Razão de expressão . . . . . 13

2.2.3 Transformações da razão de expressão . . . . . 13

2.2.4 Normalização dos dados . . . . . 15

**3 Análise binária de expressão gênica** **16**

3.1 Binarização dos dados . . . . . 17

**4 Fundamentação Teórica** **20**

4.1 Redes Booleanas . . . . . 20

4.1.1 Atratores e Bacias de Atração . . . . . 24

4.1.2 Redes Booleanas com perturbação . . . . . 25

4.2 Cadeias de Markov . . . . . 26

4.2.1 Cadeias de Markov Ergódicas . . . . . 27

4.3 Redes Booleanas com Perturbação e Cadeias de Markov . . . . . 29

4.4 Distribuição de Probabilidade Estacionária . . . . . 30

4.5 Classificação . . . . . 31

4.5.1 Classificador de Bayes . . . . . 31

4.5.2 Erro de Bayes . . . . . 32

4.6 Seleção de Características . . . . . 33

4.7 Coeficiente de Determinação . . . . . 34

4.8 Genes de Predição Intrinsicamente Multivariada . . . . . 35

**5 Modelando Dados de Expressão Gênica usando Redes Booleanas com Perturbação** **37**

5.1 Dados de Expressão Gênica . . . . . 38

5.1.1 Sistema em Equilíbrio . . . . . 38

5.2 Microarrays e Redes Booleanas com Perturbação . . . . . 40

**6 Estimação de Erro** **43**

6.1 Construindo um classificador a partir de amostras de microarray . . . . . 44

6.2 Estimadores de erro . . . . . 45

6.2.1 Holdout . . . . . 46

6.2.2 Resubstituição . . . . . 46

6.2.3 Cross-validation . . . . . 46

6.2.4 Bootstrap . . . . . 47
6.2.5 .632 Bootstrap . . . . . 48

**7 Programas Desenvolvidos 49**
7.1 CoD Estimator . . . . . 49
7.2 CoD Cleaner . . . . . 53
7.3 IMP Genes Finder . . . . . 54

**8 Busca por genes envolvidos na morte de células tumorigênicas disparada por FGF2 56**
8.1 Dados e Trabalho realizado . . . . . 57
8.2 Resultados . . . . . 58

**9 Conclusão 60**
9.1 Trabalhos futuros . . . . . 61

**Referências Bibliográficas 62**

# Lista de Figuras


2.1 Figura ilustrando um gene em uma cadeia de DNA. . . . . 9

2.2 Um microarray pode conter milhares de spots. . . . . 10

2.3 Esquema ilustrando os passos de um experimento com microarray. . . . . 11

2.4 Aproximação em um *spot* de uma lâmina de microarray. . . . . 13

3.1 *Threshold* selecionado para binarização. . . . . 18

3.2 Alternativa para selecionar um *threshold*. . . . . 18

4.1 Diagrama ilustrando um exemplo de regulação celular. Flechas representam ativação e linhas com barras no final indicam inibição. . . . . 22

4.2 Diagrama de transição de estados de uma BN com 5 genes. . . . . 24

4.3 Um atrator e sua bacia de atração. . . . . 25

4.4 Rede Booleana com perturbação. . . . . 26

4.5 Distribuição de probabilidade estacionária de uma BN com 5 genes. . . . . 30

4.6 Transcrição do gene C. . . . . 36

5.1 Atrator de uma rede Booleana. . . . . 38

5.2 Atrator de uma rede Booleana com um único estado. Neste caso, o padrão dos genes se mantém fixo no estado 0110. . . . . 39

5.3 Digrama de transição de uma rede Booleana com 6 árvores. . . . . 39

5.4 Tirando uma fotografia do sistema. . . . . 40

5.5 Microarray correspondente à foto. . . . . 40

5.6 Microarrays de vários pacientes. . . . . 41

8.1 Relação entre $\mathbf{Z} = \{x_i, x_j\}$ e $Y$ na rede de CoDs. . . . . 58
8.2 Sub-rede de CoDs. . . . . 59

# Lista de Tabelas

4.1 Tabelas verdades das funções Booleanas com cinco genes. . . . . 23
4.2 Configuração de CoDs. . . . . 35
4.3 Segunda configuração de CoDs. . . . . 36

6.1 Conjunto de amostras. . . . . 44
6.2 Tabela de freqüência. . . . . 44

# Resumo

Esta dissertação de mestrado tem como finalidade descrever o trabalho realizado em uma pesquisa que envolve a análise de expressões gênicas provenientes de *microarrays* com o objetivo de encontrar genes importantes em um organismo ou em uma determinada doença, como o câncer. Acreditamos que a descoberta desses genes, que chamamos aqui de *genes de predição intrinsicamente multivariada* (genes IMP), possa levar a descobertas de importantes processos biológicos ainda não conhecidos na literatura.

A busca por genes IMP foi realizada em conjunto com estudos de modelos e conceitos matemáticos e estatísticos como *redes Booleanas*, *cadeias de Markov*, *Coeficiente de Determinação* (CoD), *Classificação* em análise de expressões gênicas e *métodos de estimação de erro*.

No modelo de redes Booleanas, introduzido na Biologia por Kauffman, as expressões gênicas são quantizadas em apenas dois níveis: "ligado" ou "desligado". O nível de expressão (estado) de cada gene, está relacionado com o estado de alguns outros genes através de uma função lógica. Adicionando uma *perturbação* aleatória a este modelo, temos um modelo mais geral conhecido como *redes Booleanas com perturbação*. O sistema dinâmico representado pela rede é uma *cadeia de Markov ergódica* e existe então uma distribuição de probabilidade estacionária. Temos a hipótese de que os experimentos de microarray seguem esta distribuição estacionária.

O CoD é uma medida normalizada de quanto a expressão de um gene alvo pode ser melhor predita observando-se a expressão de um conjunto de genes preditores. Uma determinada configuração de CoDs caracteriza um gene alvo como sendo um gene IMP. Podemos trabalhar não somente com genes alvo, mas também com fenótipos alvo, onde o fenótipo de um sistema biológico poderia ser representado por uma variável aleatória binária. Por exemplo, podemos estar interessados em saber quais genes estão relacionados ao fenótipo de vida/morte de uma célula.

Como a distribuição de probabilidade das amostras de microarray é desconhecida, o estudo dos CoDs é feito através de estimativas. Entre os métodos de estimação de erro estudados para este propósito podemos citar: *Holdout*, Resubstituição, *Cross-validation*, *Bootstrap* e *.632 Bootstrap*. Os métodos foram implementados para calcular os CoDs, permitindo então a busca por genes IMP.

Os programas implementados na pesquisa foram usados em conjunto com uma pesquisa

realizada pelo Prof. Dr. Hugo A. Armelin do Instituto de Química da USP. Este estudo em particular envolve a busca de genes importantes relacionados à morte de células tumorigênicas de camundongo disparada por FGF2 (*Fibroblast Growth Factor 2*). Nesta pesquisa observamos sub-redes de genes envolvidos no processo biológico em questão e também encontramos genes que podem estar relacionados ao fenômeno de morte das células de camundongo ou que estão, de fato, participando de alguma via disparada pelo FGF2.

Esta abordagem de análise de expressões gênicas, juntamente com a pesquisa realizada pelo Prof. Armelin, resulta em uma metodologia para buscas de genes envolvidos em novos mecanismos de células tumorigênicas, ativados pelo FGF2. Na realidade esta metodologia pode ser aplicada em qualquer processo biológico de interesse científico, desde que seja possível modelar o problema proposto no contexto de redes Booleanas, coeficientes de determinação e genes IMP.

# Abstract

This Master's degree dissertation describes a research that involves an analysis of gene expression data from *microarray* experiments with the purpose to find important genes in certain organisms or diseases such as cancer. We believe that these type of genes, called *intrinsically multivariately predictive genes* (IMP genes), can lead to the discovery of important biological process that are unknown in the literature.

The search for IMP genes was done with the study of mathematical and statistical models such as *Boolean Networks*, *Markov Chains*, *Coefficient of Determination* (CoD), *Classification* and *Error Estimation Methods*.

In the Boolean network model, introduced in Biology by Kauffman, the gene expression is quantized in only two levels: ON and OFF. The expression level (state) of each gene is related with the state of some other genes through a logical function. Adding a random *perturbation* to this model, we have a more general Boolean-type model called *Boolean network with perturbation*. The dynamical system represented by this network is an *ergodic Markov chain* and thereby it possesses a steady-state distribution. We have the hypothesis that the microarray experiments follow this steady-state distribution.

The CoD is a normalized measure of how much a gene expression of a *target gene* can be better predicted observing the expression of a set of *predictor genes*. A certain configuration of CoDs characterizes a target gene as an IMP gene. We can deal not only with target genes, but also with target phenotypes, where the phenotype of a biological system could be represented by a binary random variable. For example, we could be interested in knowing which genes are related to a life/death cell phenotype.

Since the joint probability distribution of the gene expressions is unknown, the CoDs must be computed through estimated values. Among the error estimation methods studied we can cite: Holdout, Resubstitution, Cross-validation, Bootstrap and .632 Bootstrap. Those methods were implemented as a software in order to compute the CoDs and thereby allowing us to search for IMP genes.

The software we implemented in this research was used within a research developed by Professor Dr. Hugo A. Armelin from the Instituto de Química - University of Sao Paulo. This particular research involves the search for important genes related to the death of tumorigenic mouse cells triggered by FGF2 (Fibroblast Growth Factor 2). From this research cooperation, we built some gene subnetworks involved in the target biological

process and we found some genes that could be related to the death phenotype of mouse cells.

This approach of gene expression analysis, together with the research developed by Professor Armelin, results in a methodology to search for important genes that could be involved in new mechanisms of tumorigenic cells triggered by FGF2. Actually, this methodology can be applied to any biological process of scientific interest, if one can model the proposed problem in the context of Boolean Networks, Coefficient of Determination and IMP genes.

# Capítulo 1

# Introdução

Neste capítulo iremos apresentar a motivação que nos levou a realizar este trabalho assim como os objetivos da pesquisa e tema desta dissertação de mestrado. Por último apresentamos como o trabalho está organizado para auxiliar o leitor na busca de um tópico específico da pesquisa.

## 1.1 Motivação

A análise de expressões gênicas é muito importante em diversas pesquisas biológicas e da Bioinformática, uma vez que mudanças na fisiologia de um organismo ou de uma célula é acompanhada de mudanças nos padrões das expressões gênicas. Uma importante fonte de dados que temos hoje é a tecnologia de *microarrays*. Estes dados consistem de expressões gênicas de uma grande quantidade de genes, sendo que alguns desses genes podem ser importantes no estudo de determinados organismos ou fenômenos biológicos.

Problemas que envolvem a análise de expressões gênicas têm ganhado uma forte atenção nos últimos anos. Um problema chave que envolve a análise de expressões gênicas provenientes de *microarrays* consiste na predição da expresssão de um *gene alvo* em termos da expressão de outros genes, chamados *genes preditores*. Para medir a qualidade de tais predições usamos o Coeficiente de Determinação, ou simplesmente CoD (do inglês *Coefficient of Determination*) [Dougherty 00]. Dado um conjunto de amostras de microarray, os CoDs devem ser estimados, uma vez que a distribuição de probabilidade das variáveis que representam as expressões gênicas não é conhecida. Assim, o estudo de métodos de estimação de erro foi indispensável para nossa pesquisa. Como veremos, uma certa configuração de CoDs relacionada com um gene alvo define o que chamamos de *genes de predição intrinsicamente multivariada*.

Um modelo para representar a interação entre os genes, conhecido como Redes Booleanas, ou BN (*Boolean Networks*), tem a intenção de fornecer uma visão matemática de como essa interação ocorre. Este modelo foi proposto por Kauffman [Kauffman 69] e ele

é completamente determinístico. Se adicionamos uma perturbação a este modelo, modelo conhecido como BN com perturbação, a célula ou organismo podem ser vistos dentro de um modelo estocástico no contexto de cadeias de Markov. Esse modelo, entre outros, tem um potencial grande para nos ajudar a explicar a interação entre os genes dentro de uma célula [Shmulevich 02c, Shmulevich 02d, Brun 05]. Dessa forma, o seu estudo se torna de fundamental importância para que possamos entender resultados experimentais e biológicos já conhecidos e/ou fornecer novas hipóteses para verificação experimental.

## 1.2 Objetivos

Entre os objetivos principais da pesquisa podemos citar:

- Estudar modelos que representem a interação entre genes em células/organismos;
- Implementação de algoritmos para encontrar genes de predição intrinsicamente multivariada a partir de dados de microarray;
- Encontrar alguns genes que poderiam levar a descobertas de processos biológicos em determinados tipos de organismos ou câncer.

## 1.3 Organização do Trabalho

Nesta dissertação de mestrado iremos apresentar o trabalho realizado na busca de genes importantes em um sistema biológico utilizando dados de expressões gênicas. No Capítulo 2 iremos descrever uma técnica de análise de expressão gênica, conhecida como *microarray*, que está sendo muito utilizada em pesquisas da Bioinformática. Iremos descrever brevemente como esta técnica é aplicada.

O Capítulo 3 também é relacionado aos dados de microarray, porém é focado na análise binária desse tipo de dado. Iremos discutir sobre vantagens e desvantagens em utilizar uma representação binária para expressões gênicas de microarrays. Apresentamos um algoritmo usado para binarizar dados de expressões gênicas, uma vez que dados de microarray não são de natureza binária.

No Capítulo 4 introduzimos os conceitos teóricos necessários para entender como os objetivos da pesquisa foram alcançados. Iremos apresentar o modelo de *redes Booleanas com perturbação e cadeias de Markov*, utilizado para modelar a interação entre genes de um sistema biológico. Outro tópico importante visto neste capítulo é o de *Classificação* na análise de expressões gênicas. Veremos como os *classificadores* são utilizados no cálculo de *coeficientes de determinação*, cuja definição formal também é apresentada neste capítulo. Apresentamos também o conteito de *genes de predição intrinsicamente*

*multivariada*, que juntamente com os coeficientes de determinação formam o foco central da pesquisa apresentada nesta dissertação.

No Capítulo 5 apresentamos o modelo proposto aplicado às expressões gênicas para estudarmos os coeficientes de determinação. O modelo utiliza o conceito de redes Booleanas com perturbação, introduzido no Capítulo 4.

O Capítulo 6 é necessário para compreender como é feito o cálculo dos coeficientes de determinação a partir de dados de microarray. Veremos que a distribuição de probabilidade dos dados de microarray é desconhecida, e por isso o cálculo dos CoDs é feito através de estimativas. Neste capítulo apresentamos os métodos de estimação de erro implementados na pesquisa.

Os programas desenvolvidos na pesquisa são apresentados no Capítulo 7. Descrevemos o objetivo de cada programa, como ele foi implementado e alguns exemplos de execução.

O Capítulo 8 tem como finalidade apresentar um trabalho realizado juntamente com alguns colaboradores: o Prof. Dr. Hugo Aguirre Armelin do Instituto de Química da USP, Fábio Nakano do BIOINFO-USP e a Dr$^{a}$ Paula F. Asprino do IQ-USP. O trabalho envolve a busca por genes envolvidos na morte de células tumorigênicas disparada por FGF2. Apresentaremos alguns resultados obtidos e perspectivas futuras para este tema.

E por último, no Capítulo 9 faremos a conclusão do trabalho realizado para fins desta dissertação de mestrado.

# Capítulo 2

# A Tecnologia de Microarrays

É importante saber como são obtidos os principais dados que são utilizados em nosso trabalho, por isso este capítulo tem como finalidade apresentar brevemente a tecnologia de microarrays. O objetivo do capítulo não é detalhar por completo o assunto, mas sim fazer com que o leitor tenha em mente como os dados usados no trabalho são obtidos. Primeiro vamos fazer uma introdução ao que chamamos de genes. Depois iremos apresentar a tecnologia de microarrays como uma forma de capturar o nível de expressão de uma grande quantidade de genes. E por fim, falaremos de como são representados os dados de microarray utilizados em nossa pesquisa.

## 2.1 Os genes e a sua importância

Na genética clássica, um gene era um conceito abstrato, uma unidade de hereditariedade que carregava uma característica de pai para filho [Pearson 06]. Os genes estão codificados no material genético do organismo (DNA - ácido desoxirribonucleico), e controlam de certa forma o comportamento do organismo. O DNA é formado por pares de moléculas ligadas por pontes de hidrogênio e é organizado como duas fitas complementares, com as pontes de hidrogênio entre elas. Cada fita é constituída de *bases nitrogenadas* ou nucleotídeos, que podem ser de quatro tipos: adenina (A), timina (T), citosina (C) ou guanina (G). Estes quatro tipos de base formam o *alfabeto genético*. No entanto, no RNA a timina é substituída pela uracila (U).

Entre as duas fitas, cada base pode "parear" com apenas outra base determinada. Os possíveis pareamentos são A+T, T+A, C+G ou G+C. Sendo assim, um A encontrado em uma das fitas será pareado com um T na fita complementar.

Um gene é uma seqüência específica do DNA, e a seqüência de três nucleotídeos consecutivos no gene é chamada de *códon*. Cada códon codifica um tipo de aminoácido. Como temos 20 aminoácidos e $4^3 = 64$ possíveis códons, um aminoácido pode ser determinado por mais de um códon. Por exemplo, o códon CAA e CAG codificam o aminoácido glu-

tamina. A seqüência de códons em um gene especifica a seqüência de aminoácidos da proteína que o gene codifica.

Existem regiões do DNA que não codificam proteínas, dando origem ao termo *DNA não-codificante*. Em células eucariontes os genes são geralmente fragmentados internamente por seqüências não-codificantes chamadas *íntrons*, que aparecem entre as seqüências codificantes, chamadas *exons* (Figura 2.1).

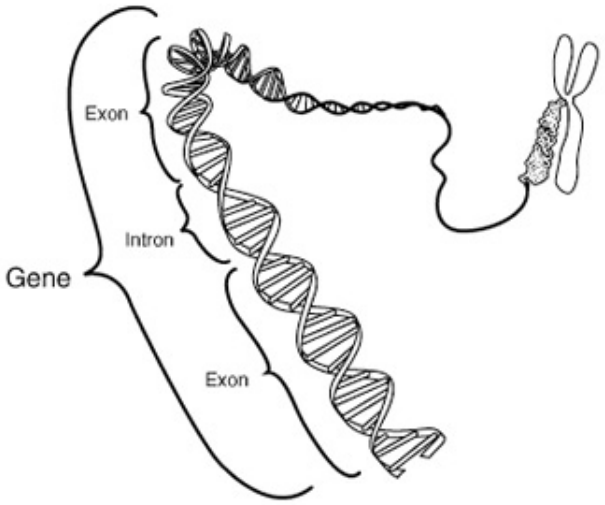


Figura 2.1: Figura ilustrando um gene em uma cadeia de DNA.

Dizemos que um gene está expresso quando a sua seqüência correspondente no DNA é transcrita para o RNA. Os cientistas podem avaliar o estado de uma célula baseado em quais genes estão expressos nela. Uma "fotografia" da expressão de uma grande quantidade de genes poderia nos ajudar a observar a atividade desses genes. Por exemplo, uma foto de "pessoas" pode mostrar quem está vestindo diferente tipos de "roupas" ou "sapatos" (DNA normal ou mutações) e capturar a atividade delas, tais como "dormindo" ou "correndo" (nível de atividade dos genes, onde alguns estão não-ativos enquanto outros possuem um nível alto/baixo de atividade).

Observando os níveis de atividades de diferentes genes, os cientistas podem restringir suas pesquisas em alto/baixo nível de atividade. Um exemplo seria uma comparação de genes de um tecido cancerígeno com um tecido normal, não-cancerígeno. Se existe um alto/baixo nível de atividade nos genes da célula no tumor quando comparados aos genes da célula normal, então estes genes poderiam ser objetos de pesquisa para um estudo mais aprofundado (descobrir a função de tais genes, por exemplo).

## 2.2 Microarray

O objetivo da maioria dos experimentos de microarray é examinar padrões de expressões gênicas determinando o nível de expressão de milhares de genes simultaneamente [Quackenbush 02].

Não existe apenas um único tipo de microarray. Podemos ter microarrays baseados em membrana de nylon ou lâmina de vidro, por exemplo. Iremos descrever a tecnologia de microarray de acordo com o segundo exemplo. Sendo assim, um microarray é basicamente uma lâmina de vidro na qual moléculas de DNA são fixadas em lugares específicos denominados *spots*. Um microarray pode conter milhares de spots e cada spot pode conter milhões de cópias de moléculas de DNA idênticas que correspondem unicamente a um gene (Figura 2.2) [Bradley 03]. O DNA no spot pode ser um DNA genômico ou uma curta fita de oligonucleotídeos correspondente a um gene [Grant 04]. Esse trabalho de fixar as moléculas nos spots da lâmina de vidro é feita por um robô.

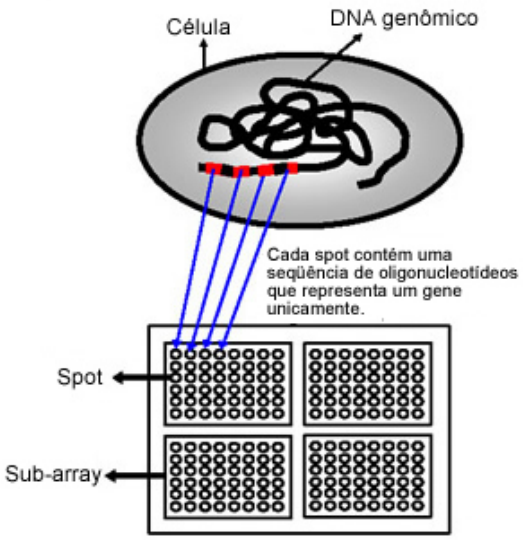


Figura 2.2: Um microarray pode conter milhares de spots.

Microarrays podem ser usados para medir a expressão gênica de várias maneiras, mas a aplicação mais comum é comparar a expressão de um conjunto de genes de uma célula mantida em uma determinada condição (condição A ou câncer, por exemplo) de um mesmo conjunto de genes de uma célula mantida em condições normais (condição B).

Mas como podemos observar as diferenças entre os dois conjuntos de genes? A Figura 2.3 ilustra os passos experimentais envolvidos.

Primeiro, o mRNA é extraído das células. A seguir, moléculas de mRNA no extrato são transcritos em cDNA usando uma enzima trancriptase reversa e nucleotídeos de coloração fluorescente. Por exemplo, o cDNA das células cancerígenas podem ser rotuladas com a cor vermelha, e o das células normais com a cor verde. O próximo passo é misturar estas duas

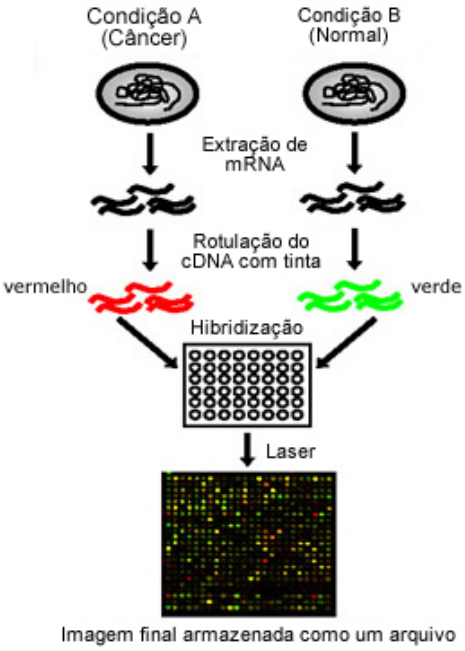


Figura 2.3: Esquema ilustrando os passos de um experimento com microarray.

populações de cDNAs marcadas à lâmina de vidro, em condições propícias a hibridização de seqüências complementares. Após a hibridização, os spots são estimulados por um laser para detectar as colorações vermelhas e verdes. A quantidade de ácido nucléico nos spots é proporcional ao brilho fluorescente. Sendo assim, se o cDNA da condição A (câncer) para um gene em particular estiver em maior abundância do que da condição B (normal), o spot será encontrado na cor vermelha. Caso contrário o spot estará verde. Se o gene estiver expresso igualmente nas duas condições, o spot irá tomar uma coloração amarela, e se o gene não estiver expresso em nenhuma das condições o spot estará preto. Portanto, o que temos ao final do experimento é uma imagem do microarray, em que cada spot correspondente a um gene tem um valor associado ao brilho fluorescente relativo ao nível de expressão daquele gene [Grant 04].

### 2.2.1 Análise e processamento da imagem

Um experimento de microarray produz uma imagem, como foi mencionado anteriormente. Nesta imagem temos o nível de expressão relacionado a cada gene (população de RNA nas duas amostras). O primeiro passo na análise de um dado de microarray é o processamento desta imagem. A maioria dos fabricantes de *scanners* para microarray oferecem seus próprios programas, no entanto, é importante entender como os dados são realmente extraídos a partir da imagem, pois este é o primeiro passo para a coleta de dados e forma a base para análises futuras.

O processamento da imagem envolve os seguintes passos:

1. *Identificar os spots e distingui-los de ruídos.*

   O microarray é processado após a hibridização e um arquivo de imagem é gerado. Uma vez que a geração da imagem esteja completa, é feita a identificação dos spots. No caso de microarrays, os spots são organizados em sub-arrays (Figura 2.2). A maioria dos programas de processamento de imagens exige que o usuário especifique aproximadamente onde cada sub-array se encontra e alguns parâmetros adicionais relacionados ao sub-array. Estas informações são utilizadas para identificar regiões que correspondem aos spots.

2. *Determinação da área do spot a ser examidada, determinação da região local para estimar a hibridização do fundo (background).*

   Após identificar as regiões correspondentes aos sub-arrays, uma área dentro do sub-array deve ser selecionada para se obter uma medida do sinal do spot e uma estimativa para a intensidade do fundo (Figura 2.4). Existem dois métodos para definir o sinal do spot. O primeiro é utilizar uma área de tamanho fixo que é centralizada no centro de massa do spot. Este método tem a vantagem de ser menos caro computacionalmente, mas possui a desvantagem de ser mais provável de erros na estimação da intensidade do spot e do fundo. Um método alternativo é definir precisamente os limites para um spot e incluir apenas pixels dentro do limite. Este método tem a vantagem de obter uma estimativa melhor da intensidade do spot, mas tem a desvantagem de ser mais caro computacionalmente.

3. *Relatar estatísticas e atribuir intensidade aos spots após a subtração da intensidade do fundo.*

   Uma vez que foram definidas as áreas dos spots e do fundo, uma variedade de estatísticas para cada spot em cada canal (vermelho e verde) são relatadas. Geralmente, cada pixel (Figura 2.4) dentro da área é levado em consideração, e a média, mediana e o valor total da intensidade considerando todos os pixels na área definida são relatados para ambos, o spot e o fundo. A maioria das abordagens usam a mediana do spot subtraindo-se a mediana do fundo como uma medida para representar a intensidade do spot. A mediana é um valor onde metade dos pixels medidos possuem intensidades maiores do que este valor, e a outra metade medida possuem valores menores do que este valor.

Outra consideração a ser feita no processamento da imagem é o número de pixels a serem incluídos para a análise do spot na imagem. Para muitos scanners, o tamanho padrão do pixel é 10$\mu$m. Isso significa que um spot de 200$\mu$m de diâmetro terá em média 314 pixels. No entando, para um spot de tamanho menor é melhor utilizar um tamanho de pixel menor para garantir que uma quantidade suficiente de pixels seja amostrada. Apesar de usar um tamanho de pixel menor melhorar a confiança na medida, a desvantagem é que o tamanho do arquivo de imagem tende a aumentar se comparado com imagens criadas utilizando-se um tamanho de pixel maior [Grant 04].

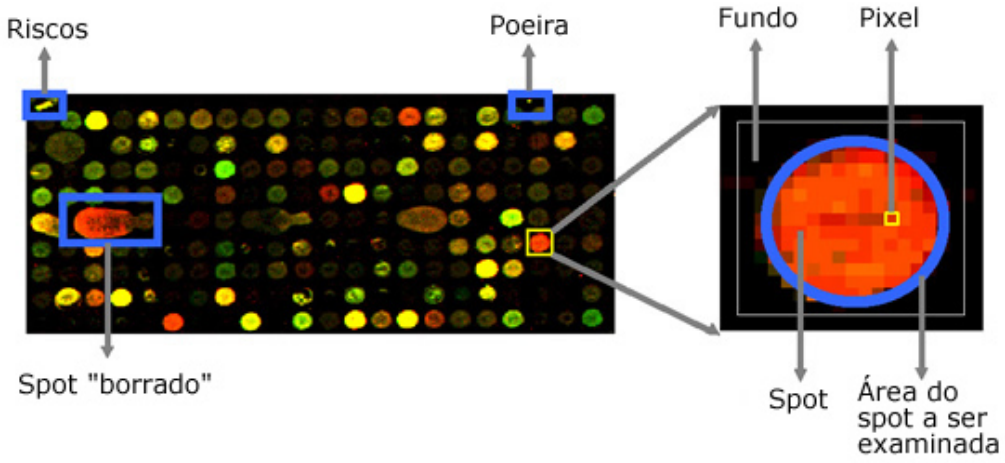


Figura 2.4: Aproximação em um *spot* de uma lâmina de microarray.

### 2.2.2 Razão de expressão

Vimos que o nível de expressão relativo a um gene pode ser medido de acordo com o brilho fluorescente vermelho ou verde emitido após o estímulo do laser. Uma medida utilizada para relatar esta informação é chamada *razão de expressão*, denotada por $T_i$ e definida como:

$$T_i = \frac{R_i}{G_i} \tag{2.1}$$

para cada gene $i$, onde $R_i$ representa a intensidade do spot para a amostra teste (condição A) e $G_i$ representa a intensidade do spot para a amostra de referência (condição B). Como foi mencionado, a intensidade do spot para cada gene pode ser representado pelo valor da intensidade total ou pelo valor da subtração da mediana pelo fundo. Se escolhermos o valor da mediana, então a razão de expressão mediana é dada por:

$$T_{\text{mediana}} = \frac{R_{\text{mediana spot}} - R_{\text{mediana fundo}}}{G_{\text{mediana spot}} - G_{\text{mediana fundo}}} \tag{2.2}$$

onde $R_{\text{mediana spot}}$ e $R_{\text{mediana fundo}}$ são as medianas para os valores das intensidades do spot e do fundo respectivamente, para a amostra teste.

### 2.2.3 Transformações da razão de expressão

A razão de expressão é uma forma de representar as diferenças de expressão de maneira intuitiva. Por exemplo, genes que não diferem no seu nível de expressão pussuem razão de expressão igual a 1. No entanto, esta representação pode não ajudar muito quando necessitamos representar genes como sendo expresso ou não-expresso. Por exemplo, um

gene que está expresso por um fator de 4 possui uma razão de expressão igual a 4 ($R/G = 4G/G = 4$). No entanto, para o caso em que o gene está não-expresso por um fator de 4, a razão de expressão se torna 0.25 ($R/G = R/4R = 1/4$). Dessa maneira, um gene expresso é mapeado entre 1 e infinito, enquanto um gene não-expresso é mapeado entre 0 e 1 apenas.

$$\begin{aligned} \text{expresso} &\longrightarrow [1, \infty] \\ \text{não-expresso} &\longrightarrow [0, 1] \end{aligned} \tag{2.3}$$

Para eliminar esta inconsistência no intervalo de mapeamento, podemos realizar dois tipos de transformações na razão de expressão: *transformação inversa* ou *transformação logarítmica*.

### Transformação inversa

A transformação inversa converte a razão de expressão em um outro valor, onde para genes com razão de expressão menor do que 1 a inversa da razão de expressão é multiplicada por $-1$. Se a razão de expressão é maior ou igual a 1 então o valor não é mudado. A vantagem de tal transformação é que podemos representar um gene como expresso ou não-expresso em um intervalo de mapeamento similar.

$$\text{expressão} = \begin{cases} T_i & \text{se } T_i \geq 1 \\ \dfrac{-1}{T_i} & \text{se } T_i < 1 \end{cases} \tag{2.4}$$

No entanto, este método também possui o problema em que o espaço de mapeamento é descontínuo entre $-1$ e $+1$, o que pode ser um problema para análises matemáticas futuras.

### Transformação logarítmica

Um procedimento de transformação melhor é tomar o valor do logarítmo na base 2 da razão de expressão, ou seja, $\log_2$(razão de expressão). A maior vantagem é que assim tratamos genes expressos e não-expressos igualmente, e também temos um espaço de mapeamento contínuo. Por exemplo, se a razão de expressão é 1, então $\log_2(1) = 0$ representa nenhuma diferença na expressão. Se a razão de expressão é 4, então $\log_2(4) = 2$ e, se a razão de expressão é 1/4 temos $\log_2(1/4) = -2$. Assim, nesta transformação o espaço de mapeamento é contínuo e genes expressos e não-expressos são comparáveis.

Tendo explicado as vantagens de se utilizar a razão de expressão como medida de expressão gênica, também devemos entender que existem desvantagens ao usar razões de expressões ou transformações das razões para a análise dos dados. Apesar da razão de expressão revelar padrões relacionados aos dados, ela remove toda a informação absoluta

a respeito dos níveis de expressão dos genes. Por exemplo, genes que possuem razões $R/G$ de 400/100 e 4/1 terão a mesma razão de expressão igual a 4.

### 2.2.4 Normalização dos dados

Na última seção vimos que a razão de expressão e suas transformações são medidas razoáveis para detectar genes diferentemente expressos. No entanto, quando alguém tenta comparar o nível de expressão de genes que não deveriam mudar nas duas condições (*housekeeping genes*) o que pode ser encontrado é uma variação nas razões de expressão. Isso pode acontecer devido a vários motivos, por exemplo, uma variação causada por uma ineficiência no momento de rotular as amostras com os corantes fluorescentes, ou uma diferença na quantidade inicial de mRNA nas duas amostras. Por isso, no caso de experimentos de microarray existem muitas fontes de variação que afetam as medidas dos níveis de expressão gênica [Yang 02].

Normalização é um termo usado para descrever o processo de eliminação de tais variações para que seja possível realizar uma comparação entre os dados obtidos a partir das duas amostras. Existem muitos métodos de normalização, e a discussão de cada um deles está fora do escopo deste capítulo.

# Capítulo 3

# Análise binária de expressão gênica

No Capítulo 2 introduzimos a tecnologia de microarrays como fonte de dados para o nosso trabalho. Vimos que estes dados representam expressões gênicas de uma grande quantidade de genes. A forma mais usual de representar estes dados é através da transformação logarítmica nas razões de expressão dos genes. Esta transformação faz com que os dados obtidos sejam representados por números reais ($\mathbb{R}$). Neste capítulo veremos que em algumas situações é mais vantajoso trabalhar com dados representados no domínio binário ($\{0, 1\}$), e por isso precisamos transformar os dados que estão no domínio dos reais para o domínio binário.

A questão da complexidade do modelo é importante no desenvolvimento de modelos de predição e redes de regulação gênica, especialmente em situações nas quais o número de genes é muito grande e o número de amostras é muito pequeno. Modelos menos complexos exigem menos computação e são compreendidos de forma mais fácil. Um sistema deve ser modelado em um nível baixo de complexidade para que seja possível a realização dos objetivos para o qual o modelo foi desenvolvido [Zhou 03a], como por exemplo, modelar a interação entre os genes dentro de um sistema biológico.

Como veremos adiante, a quantização de dados em valores binários pode proporcionar um certo nível de robustez em relação aos ruídos nos dados observados e pode até mesmo melhorar a precisão em pesquisas de classificação/predição. Além disso, representação de dados de microarray no domínio binário nos permite utilizar modelos mais "grossos" para modelar a interação entre os genes de uma célula ou sistema biológico. Modelos "finos" com muitos parâmetros podem ser capazes de capturar fenômenos em um nível mais baixo, como concentração de proteínas e reações químicas, mas uma grande quantidade de dados pode ser exigida. Por outro lado, modelos grossos com poucos parâmetros e menos complexos são melhores para capturar fenômenos em um nível mais alto, como analisar quando um gene está expresso ou não em um determinado momento, exigindo uma quantidade menor de dados [Shmulevich 02d]. Um exemplo de modelo que utiliza análise binária de expressão de genes é o modelo de *redes Booleanas* que será descrito no próximo capítulo.

Matematicamente, podemos descrever a binarização dos dados da seguinte maneira. Para cada gene, um *threshold* apropriado é escolhido de forma que todos os valores de expressões acima desse *threshold* terão seu valor alterado para '1', enquanto os outros valores serão rotulados com '0'. Na próxima seção apresentaremos uma técnica de binarização de dados, lembrando que existem outras técnicas para realizar este processo.

## 3.1 Binarização dos dados

Nesta seção mostraremos como podemos fazer a binarização dos dados. Quando nos referimos a "dados" estamos falando de expressões gênicas que, como foi visto, podem ser representadas por razões de expressão e/ou transformações logarítmicas. Antes vamos introduzir algumas definições necessárias para o compreendimento do processo de binarização.

Seja $n$ o número de genes e $k$ o número de amostras de microarray. Seja $G_{i,j}$ a expressão gênica do gene $i$ na amostra $j$. Então, o *perfil gênico* do gene $i$ ($1 \leq i \leq n$) é o vetor $G_i = (G_{i,1}, G_{i,2}, \ldots, G_{i,k})$. Dessa maneira, $G_{n \times k}$ é uma matriz onde as linhas são perfis gênicos e as colunas são as amostras de microarray.

Com o propósito de binarizar os dados, necessitamos de um procedimento para selecionar um *threshold*, que discutiremos a seguir. Genes diferentes apresentam diferentes intervalos de níveis de expressão gênica. Assim, selecionar um *threshold* global para aplicar em todas as amostras de microarray não é apropriado [Shmulevich 02b]. Temos então que selecionar um *threshold* individual para cada perfil gênico. Existem várias maneiras para selecionar o *threshold*, mas a idéia básica que apresentamos é que devemos selecionar o *threshold* onde a separação entre os altos e baixos níveis de expressão gênica seja maior. Em outras palavras, se ordenarmos o perfil gênico, o *threshold* corresponderá ao primeiro "salto" que aparecer, ou seja, onde a diferença entre valores sucessivos exceder um valor predefinido no perfil ordenado.

A Figura 3.1 ilustra esta idéia graficamente. A figura contém um perfil gênico e o correspondente perfil ordenado. Podemos ver pelo perfil ordenado que o primeiro "salto" ocorre entre as posições 15 e 16, que correspondem respectivamente às amostras #21 e #10. Assim, a amostra #10 e todas as amostras em que o valor da expressão for maior do que o valor da amostra #10 terão seus valores alterados para 1, enquanto os outros valores serão alterados para 0. A questão é como definir um "salto" no perfil gênico ordenado. O caso mais difícil é quando todos os valores de expressão gênica se encontram igualmente espaçados, e assim o perfil gênico ordenado seria uma reta e a diferença entre os valores sucessivos seriam iguais.

Uma alternativa é considerar o perfil ordenado em ordem decrescente, ou seja, selecionar o *threshold* que corresponde ao último "salto" que aparecer no perfil gênico ordenado. Como esta alternativa não resolve o problema em que os valores estão igualmente espaçados podemos combinar as duas abordagens para selecionar o *threshold*. Uma outra

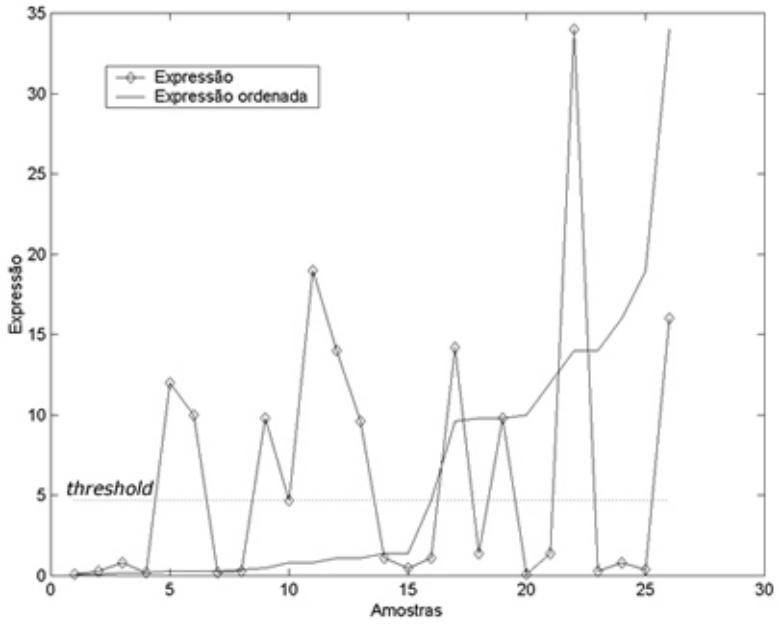


Figura 3.1: *Threshold* selecionado para binarização.

alternativa seria então encontrar dois *thresholds*, o primeiro considerando o perfil ordenado em ordem crescente e o segundo em ordem decrescente, e então assumir a média entre os dois *thresholds* como sendo o *threshold* a ser utilizado na binarização. A Figura 3.2 ilustra esta idéia, onde *threshold 1* corresponde ao primeiro salto, e *threshold 2* ao último.

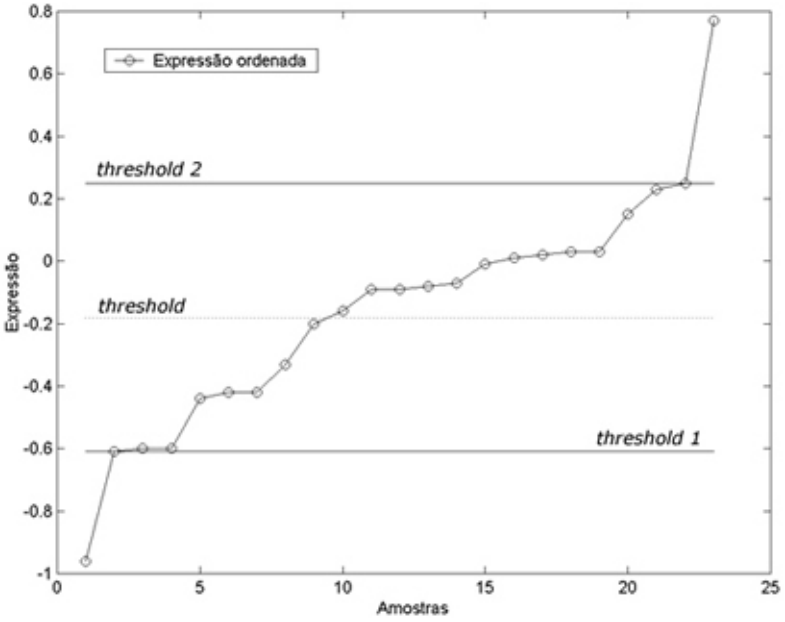


Figura 3.2: Alternativa para selecionar um *threshold*.

Temos então o seguinte algoritmo para converter um perfil gênico $G_i = (G_{i,1}, G_{i,2}, \ldots, G_{i,k})$ em um perfil binarizado $B_i = (B_{i,1}, B_{i,2}, \ldots, B_{i,k})$.

**Algorithm 1** Binarize: ENTRADA $G_i$, SAIDA $B_i$

1: $S_i \leftarrow \text{sort}(G_{i,1}, G_{i,2}, \ldots, G_{i,k})$
2: **for** $j = 1$ to $k - 1$ **do**
3: $\quad D_{i,j} \leftarrow (S_{i,j+1} - S_{i,j})$
4: **end for**
5: $jump \leftarrow (S_{i,k} - S_{i,1})/(k-1)$
6: $l \leftarrow \min\{j : D_{i,j} \geq jump\}$
7: $m \leftarrow \max\{j : D_{i,j} \geq jump\}$
8: $threshold \leftarrow (S_{i,l} + S_{i,m})/2$
9: **for** $j = 1$ to $k$ **do**
10: $\quad$ **if** $G_{i,j} \geq threshold$ **then**
11: $\quad\quad B_{i,j} \leftarrow 1$
12: $\quad$ **else**
13: $\quad\quad B_{i,j} \leftarrow 0$
14: $\quad$ **end if**
15: **end for**

O primeiro passo consiste em ordenar o perfil gênico $G_i$ e armazená-lo em $S_i$ (linha 1). Em seguida é calculado a distância entre as expressões ordenadas em $S_i$. As distâncias são armazenadas em $D_i$. Sendo assim, $D_{i,j}$ é a distância entre o valor da expressão ordenada $S_{i,j}$ e $S_{i,j+1}$ (linhas 2-4). O próximo passo é computar o valor de um "salto". Este valor pode ser definido de várias maneiras, por exemplo, calculando-se a média de todas as distâncias em $D_i$. No algoritmo apresentado, o "salto" é definido a partir do perfil gênico ordenado e corresponde à distribuição do maior salto possível $(S_{i,k} - S_{i,1})$ entre o total de saltos existentes $(k-1)$ (linha 5). Em seguida definimos onde ocorre o primeiro e o último "salto". Um salto ocorre quando a distância entre as expressões de $S_{i,j}$ e $S_{i,j+1}$ excede o valor definido para "salto" no passo anterior. Sendo assim, o primeiro (último) salto corresponde ao menor (maior) índice $j$ tal que $D_{i,j}$ excede o valor definido para "salto" (linhas 6 e 7). O primeiro e o último salto correspondem ao *threshold 1* e ao *threshold 2* da Figura 3.2, respectivamente. O *threshold* usado para binarizar os dados é obtido pela média aritmética entre *threshold 1* e *threshold 2* (linha 8). Com o *threshold* obtido basta realizar a binarização do perfil gênico $G_i$ (linhas 9-15).

Após a binarização as expressões gênicas estarão sendo representadas por apenas dois valores: 0 e 1. No próximo capítulo apresentaremos o modelo de *redes Booleanas* que nos permite modelar a interação entre os genes de um sistema biológico. Neste modelo, a expressão gênica de cada gene pode assumir apenas dois valores: 0 (desligado) ou 1 (ligado), possibilitando a modelagem dos dados binários de microarrays como veremos no Capítulo 5.

# Capítulo 4

# Fundamentação Teórica

Neste capítulo apresentamos alguns conceitos necessários para a compreensão da pesquisa feita neste mestrado. Primeiro vamos apresentar o modelo de redes Booleanas com perturbação. Em seguida introduzimos os conceitos relacionados ao que conhecemos como cadeias de Markov. Veremos como o modelo de redes Booleanas com perturbação está relacionado com o conceito de cadeia de Markov ergódica. Este relacionamento resulta na hipótese de que os dados de microarray seguem uma distribuição de probabilidade estacionária.

Nas seções seguintes falaremos a respeito de *classificação* em análise de expressões gênicas, onde assumimos que existe a distribuição de probabilidade estacionária citada acima, permitindo o projeto do *classificador de Bayes* e o cálculo do *erro de Bayes*. Nesta pesquisa, a classificação é, na verdade, uma predição da expressão gênica de um gene alvo a partir de outros genes. Surge então um outro problema conhecido como *seleção de características* onde o objetivo é encontrar os genes que fazem tal predição da melhor maneira possível. A qualidade desta predição é medida através do *coeficiente de determinação*, ou CoDs, que iremos introduzir neste capítulo. Em seguida, veremos que uma determinada configuração de CoDs define o que chamamos de *Genes de Predição Intrinsicamente Multivariada*, ou genes IMP. Os CoDs e os genes IMP formam o foco central da pesquisa apresentada nesta dissertação.

## 4.1 Redes Booleanas

As reações químicas que resultam na expressão de genes no nível molecular são complexas e ainda não são totalmente compreendidas. Um aspecto importante deste fenômeno é a interação entre os genes. Sabe-se que os genes enviam, recebem e processam informações formando uma complexa rede de comunicação [Bhalla 99], mas a arquitetura e a dinâmica destas redes não são completamente conhecidas, mesmo para organismos mais simples.

Modelos matemáticos e computacionais estão sendo desenvolvidos para explicar interações gênicas [Jong 02] e existe um número considerável de tentativas de modelar redes de expressões de genes incluindo modelos lineares [D'Haeseller 99], redes Bayesianas [Friedman 00], equações diferenciais [Mestl 95] e modelos incluindo componentes estocásticos em nível molecular [Arkin 98]. A escolha de um modelo matemático apropriado para descrever o comportamento dinâmico de uma rede de regulação gênica possui um papel importante na descoberta de mecanismos regulatórios [Ivanov 06]. O modelo que tem recebido muita atenção é o modelo de *redes Booleanas*, originalmente introduzido na biologia por Kauffman [Kauffman 69] trinta anos atrás. Neste modelo, a expressão de um gene pode ter somente dois valores discretos: "ligado" ou "desligado".

O modelo de redes Booleanas pode nos ajudar a entender o comportamento dinâmico de sistemas biológicos, como uma célula tumorigênica, proporcionando assim novos alvos para o desenvolvimento de drogas. Pesquisas realizadas mostram que muitas questões biológicas podem ser respondidas com o formalismo Booleano [Shmulevich 02d].

Vamos apresentar um exemplo extraído de [Shmulevich 02a], que mostra a representação lógica de um ciclo de regulação celular. Este processo de crescimento e divisão celular é altamente regulado. Um desbalanço neste processo resulta em um crescimento desregulado da célula em doenças como o câncer. Quando o material genético (DNA) é replicado para as células filhas, uma série de moléculas, tais como a cyclin E e a cyclin-dependent-kinase 2 (cdk2) trabalham juntas para fosforilar a proteína retinoblastoma (Rb) e inativá-la. cdk2/cyclin E é regulada por duas chaves: uma chave positiva chamada cdk activating kinase (CAK) e uma chave negativa p21/WAF1. O complexo CAK pode ser composto por dois produtos: cyclin H e cdk7. Quando cyclin H e cdk7 estão presentes, o complexo pode ativar cdk2/cyclin E. Um regulador negativo de cdk2/cyclin E é o p21/WAF1, que pode ser ativado pelo p53. Quando o p21/WAF1 se liga ao cdk2/cyclin E, o complexo kinase é desativado [Gartel 99]. Esta regulação negativa é um importante sistema defensivo das células. Por exemplo, quando células são expostas a mutação, o DNA pode ser danificado. Para o bem da célula é melhor que seja feito um reparo no dano antes da replicação do DNA, para que o material genético danificado não passe para a próxima geração. Trabalhos extensos têm mostrado que quando o DNA é danificado chaves são ativadas, que por sua vez ativam o p53, que então ativa o complexo p21/WAF1. O p21/WAF1 inibe o cdk2/cyclin E, e assim o Rb se torna ativo e a síntese de DNA é abortada. p53 também inibe cyclin H, desligando assim a chave que ativa o cdk2/cyclin E.

Para ilustrar vamos considerar o diagrama simplificado da Figura 4.1. O p53 e outros fatores regulatórios não são considerados na figura.

O exemplo acima mostra que alguns processos biológicos podem ser modelados de acordo com o formalismo Booleano em que genes ou proteínas se interagem ativando/desativando umas às outras. As redes Booleanas modelam estas interações como veremos a seguir.

Uma rede Booleana $R(V_R, F_R)$ é definida por um conjunto de variáveis $V_R = \{x_1, x_2, \ldots, x_n\}$ e uma lista de funções Booleanas $F_R = \{f_1, f_2, \ldots, f_n\}$. Cada $x_i \in \{0, 1\}$, $i =$

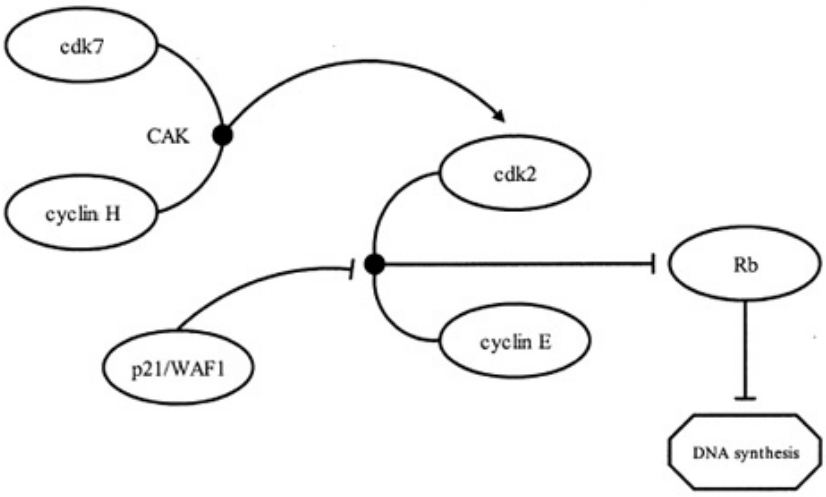


Figura 4.1: Diagrama ilustrando um exemplo de regulação celular. Flechas representam ativação e linhas com barras no final indicam inibição.

$1, 2, \ldots, n$, representa a expressão do gene $i$, onde $x_i = 1$ representa o fato que o gene $i$ está ligado e $x_i = 0$ significa que o gene $i$ está desligado. A lista de funções Booleanas $F_R$ representa as regras das interações regulatórias entre os genes.

O valor de cada variável $x_i$ é completamente determinado no tempo $t + 1$ pelos valores de algumas outras variáveis $x_{j_1(i)}, x_{j_2(i)}, \ldots, x_{j_{k_i}(i)}$ no tempo $t$ usando a função Booleana $f_i \in F_R$. Ou seja, existem $k_i$ genes associados ao gene $i$ e o mapeamento $j_k : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$, $k = 1, 2, \ldots, k_i$, determina as dependências do nível de expressão $x_i$. Assim, podemos escrever

$$x_i(t + 1) = f_i(x_{j_1(i)}(t), x_{j_2(i)}(t), \ldots, x_{j_{k_i}(i)}(t)). \tag{4.1}$$

Podemos definir um vetor $w_i \in \{1, 2, \ldots, n\}^{k_i}$ tal que $w_i = (j_1(i), j_2(i), \ldots, j_{k_i}(i))$. O vetor $w_i$ é chamado *vetor de predição* para o gene $i$ se a restrição $f_i|w_i$ se iguala a $f_i$. Em outras palavras, a expressão gênica do gene $i$ no tempo $t + 1$ é determinada pela função $f_i$ aplicada aos genes mapeados por $w_i$. Dessa maneira, temos que

$$f_i|w_i(x_1, x_2, \ldots, x_n) = f_i(x_{j_1(i)}, x_{j_2(i)}, \ldots, x_{j_{k_i}(i)}). \tag{4.2}$$

**Exemplo 4.1.1** *Este exemplo, retirado do artigo [Shmulevich 02d], mostra uma rede Booleana consistindo de cinco genes $\{x_1, x_2, \ldots, x_5\}$ com as correspondentes funções Booleanas dadas pelas tabelas verdades mostradas na Tabela 4.1.*

A *conectividade máxima* de uma rede Booleana é definida como $K = \max_i\{k_i\}$. No Exemplo 4.1.1, a conectividade máxima é $K = 3$, embora a repetição de algumas variáveis seja permitida. Por exemplo, considere a função $f_4$ que é a tabela verdade para a função

| $x_{j_1(i)}\ x_{j_2(i)}\ x_{j_3(i)}$ | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ |
|---|---|---|---|---|---|
| 000 | 0 | 0 | 0 | 0 | 0 |
| 001 | 1 | 1 | 1 | 0 | 0 |
| 010 | 1 | 1 | 1 | 0 | 0 |
| 011 | 1 | 0 | 0 | 1 | 0 |
| 100 | 0 | 0 | 1 | 0 | 0 |
| 101 | 1 | 1 | 1 | 1 | 0 |
| 110 | 1 | 1 | 0 | 1 | 0 |
| 111 | 1 | 1 | 1 | 1 | 1 |
| $j_1$ | 5 | 3 | 3 | 3 | 5 |
| $j_2$ | 2 | 5 | 1 | 4 | 4 |
| $j_3$ | 4 | 4 | 5 | 4 | 1 |

Tabela 4.1: Tabelas verdades das funções Booleanas com cinco genes.

Booleana majoritária[1]. Uma vez que $j_1(4) = 3$ e $j_2(4) = j_3(4) = 4$, $f_4(x_3, x_4, x_4) = x_4$. Dessa forma a função $f_4$ é uma função de uma única variável[2].

Um *estado* ou *configuração* de uma rede Booleana com $n$ variáveis é uma $n$-upla de valores binários correspondente a uma possível instância do vetor $(x_1, \ldots, x_n)$. Dessa forma, um estado de uma rede Booleana é um elemento do conjunto $\{0,1\}^n$. Assim, uma rede Booleana que modela um sistema com $n$ variáveis possui $2^n$ estados possíveis.

As funções restritas $f_i|w_i$ de cada gene $i$ podem ser representadas por um vetor de funções $\mathbf{f} = (f_1|w_1, \ldots, f_n|w_n)$. Seja $s \in \{0,1\}^n$ um estado da rede Booleana, podemos definir $\mathbf{f}(s)$ como sendo

$$\mathbf{f}(s) = (f_1|w_1(s), f_2|w_2(s), \ldots, f_n|w_n(s)). \tag{4.3}$$

**Definição 4.1.1** *Um **diagrama de transição de estados** de uma rede Booleana $R$ com $n$ genes é um grafo dirigido $G(V_G, A_G)$, onde os vértices são os todos estados da rede, isto é, $V_G = \{0,1\}^n$. Existe um arco de $s_k \in V_G$ para $s_l \in V_G$ (isto é, $(s_k, s_l) \in A_G$) se, e somente se, $s_l = \mathbf{f}(s_k)$, onde $\mathbf{f} = (f_1|w_1, \ldots, f_n|w_n)$ é o vetor de funções Booleanas de transição em $F_R$ que serão aplicadas em cada variável de $V_R$.*

Podemos representar uma rede Booleana graficamente através do seu diagrama de transição de estados. Vale notar que um diagrama de transição de estados pode representar várias redes Booleanas. A Figura 4.2 [Shmulevich 02d] ilustra o diagrama de transição de estados da rede Booleana do Exemplo 4.1.1 que é composta por 5 genes. Sendo assim, temos os $2^5 = 32$ estados possíveis, onde cada estado é representado por um círculo e as flechas indicam as transições de acordo com as funções em $F_R = \{f_1, \ldots, f_5\}$. Veremos na

[1] majority function.

[2] Note que a maioria entre $x_3$, $x_4$ e $x_4$ é sempre $x_4$.

próxima seção que alguns estados podem ser revisitados de maneira cíclica, constituindo o que chamamos de *atratores*.

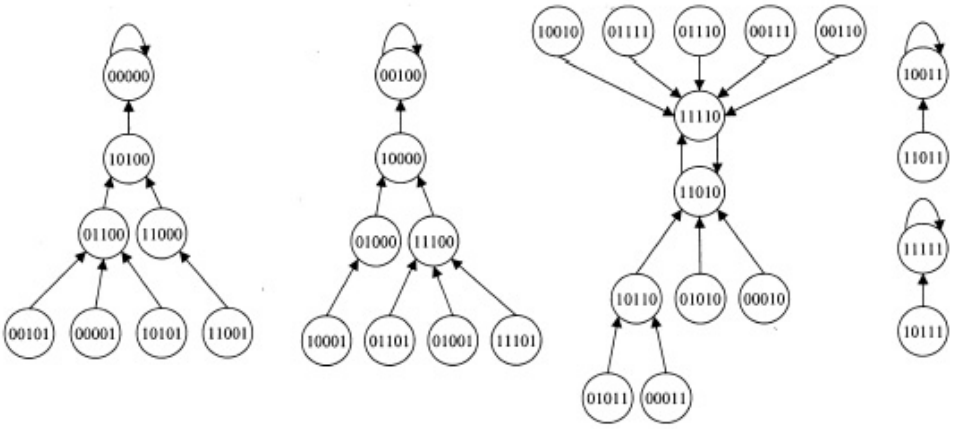


Figura 4.2: Diagrama de transição de estados de uma BN com 5 genes.

### 4.1.1 Atratores e Bacias de Atração

Uma vez que o número de estados de uma BN é finito e que a rede sempre transita de um estado para outro, é fácil ver que depois do sistema iterar por um determinado tempo, certos estados serão revisitados de uma maneira cíclica. Tais estados formam o que chamamos de *atratores*. Um *atrator* é um ciclo dirigido no diagrama de transição estados da BN. Dado um atrator, o conjunto de todos os estados que conduzem a este atrator é chamado de *bacia de atração*. As bacias formam uma partição no diagrama de estados de uma BN. Os estados que não estão nos atratores são chamados de *estados transientes* e eles são visitados somente uma única vez em qualquer trajetória de transição de estados de uma BN (Figura 4.3 [Wuensche 98]).

No diagrama da Figura 4.2, temos cinco atratores: $A_1 = \{00000\}$, $A_2 = \{00100\}$, $A_3 = \{11010, 11110\}$, $A_4 = \{10011\}$ e $A_5 = \{11111\}$. Note que em vez de representar cada estado do diagrama como uma seqüência de números binários (números na base 2), podemos representar cada estado deste diagrama pelos respectivos números inteiros decimais (números na base 10). Assim, poderíamos dizer que os atratores deste diagrama correspondem a $A_1 = \{0\}$, $A_2 = \{4\}$, $A_3 = \{26, 30\}$, $A_4 = \{19\}$ e $A_5 = \{31\}$.

A intuição de Kauffman [Kauffman 93] era de que os atratores nas BNs poderiam corresponder a tipos de células. Esta interpretação é bastante razoável se os tipos de células forem caracterizados por padrões recorrentes de expressão de genes. Assim, a partir de uma BN podemos modelar um sistema no qual todos os estados que conduzem a um atrator são associados a um correspondente tipo de célula.

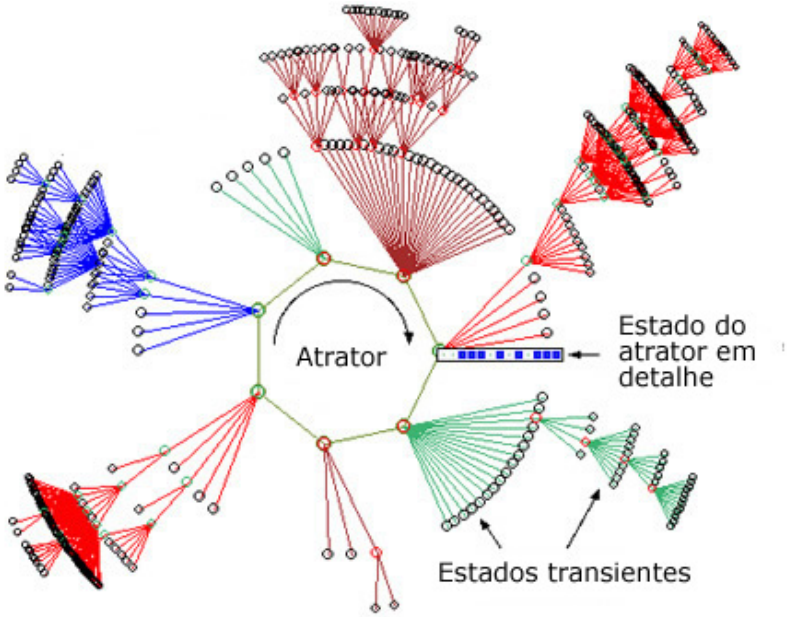


Figura 4.3: Um atrator e sua bacia de atração.

### 4.1.2 Redes Booleanas com perturbação

Adicionando uma perturbação no modelo de redes Booleanas obtemos o que é conhecido como *redes Booleanas com perturbação*. Nesta rede, existe uma probabilidade $p \approx 0$ de que a mudança da expressão de um gene não seja realizada através de uma função Booleana, mas sim por uma perturbação. Como exemplo, suponhamos que nosso sistema possua 3 genes e que o sistema se encontre no estado $s_0 = 000$. Suponha que as funções Booleanas levem ao estado $s_1 = 001$ no próximo passo. Se ocorrer uma perturbação, o próximo estado da rede pode não ser necessariamente o estado $s_1$. A perturbação pode agir em um determinado gene, como por exemplo, o primeiro gene (da esquerda para a direita), modificando sua expressão de 0 para 1. Isso levaria o sistema ao estado $s_4 = 100$ (Figura 4.4). Este tipo de aleatoriedade tem um significado biológico, uma vez que o nível de expressão dos genes pode sofrer alterações devido a estímulos externos tais como o calor ou radiação. Note ainda que neste modelo, a célula ou organismo são vistos dentro de um modelo estocástico.

Podemos formalizar esta perturbação da seguinte maneira. Suponhamos que a cada passo da rede exista um *vetor de perturbação* aleatório $\gamma \in \{0,1\}^n$. Se o $i$-ésimo componente de $\gamma$ é igual a 1, então a expressão do $i$-ésimo gene é mudada trocando-se 0 por 1 e vice-versa, caso contrário não. Em geral, $\gamma$ não precisa ser independente e identicamente distribuído (i.i.d.), mas iremos assumir isso por simplicidade [Shmulevich 02c]. Assim, suponhamos que $\Pr\{\gamma_i = 1\} = \mathrm{E}[\gamma_i] = p$ para todo $i = 1, 2, \ldots, n$. Claramente,

$$\Pr\{\gamma = (0, \ldots, 0)\} = (1-p)^n. \tag{4.4}$$

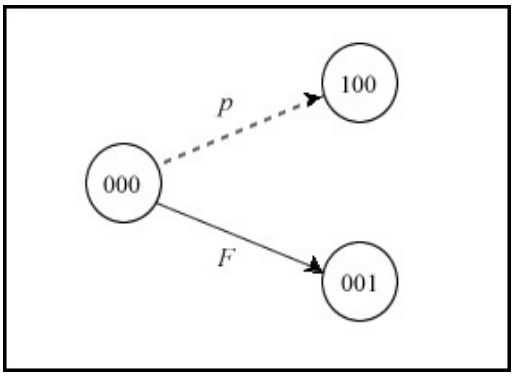


Figura 4.4: Rede Booleana com perturbação.

Seja $s_k = (x_1, \ldots, x_n)$ o estado atual da rede (ou seja, a expressão de todos os genes) em um dado momento. Então, o próximo estado $s_l$ é dado por

$$s_l = \begin{cases} s_k \oplus \gamma, & \text{com probabilidade } 1-(1-p)^n \\ \mathbf{f}(s_k), & \text{com probabilidade } (1-p)^n \end{cases}, \tag{4.5}$$

onde $\oplus$ é a operação XOR (OU exclusivo) bit a bit e $\mathbf{f} = (f_1|w_1, \ldots, f_n|w_n)$ é o vetor de funções Booleanas de transição que serão aplicadas em cada gene.

Um aspecto importante no estudo de BNs com perturbação é seu comportamento dinâmico, em particular a determinação do comportamento depois de deixar o sistema iterar por um longo período de tempo. No contexto de redes gênicas, um biólogo poderia estar interessado no comportamento conjunto de um certo grupo de genes depois do sistema atingir seu equilíbrio homeostático.

Uma propriedade interessante é que o comportamento dinâmico das BNs com perturbação pode ser analisado no contexto de cadeias de Markov. Dessa forma, na próxima seção introduziremos alguns conceitos de Cadeias de Markov que serão utilizados nesta dissertação.

## 4.2 Cadeias de Markov

Em princípio, quando observamos uma seqüência de experimentos, todos os resultados podem influenciar na predição do próximo experimento. Em 1906, Andrei A. Markov começou a estudar um novo tipo de processo, onde o resultado de um dado experimento em um tempo $t$ pode afetar o resultado do próximo experimento no tempo $t+1$ [Basharin 04]. Este tipo de processo é chamado de cadeia de Markov.

Uma cadeia de Markov pode ser definida da seguinte forma: seja $S = \{s_1, s_2, \ldots, s_r\}$ um conjunto finito de *estados*. Considere um processo que começa em um desses estados e se move sucessivamente de um estado para outro. Cada movimento é chamado de *passo*.

Se a cadeia está atualmente no estado $s_i$, então ela se move para o estado $s_j$ no próximo passo com uma probabilidade denotada por $p_{ij}$, e esta probabilidade não depende de quais estados a cadeia ocupou anteriormente antes do estado atual [Grinstead 97]. Mais formalmente, dizemos que uma seqüência de variáveis aleatórias $S^{(0)}, S^{(1)}, \ldots, S^{(t)}, S^{(t+1)}$ em $S$ é uma cadeia de Markov se a seqüência for Markoviana no seguinte sentido, para todo $t$ temos

$$p_{ij} = \Pr(S^{(t+1)} = s_j | S^{(0)} = s_{k_0}, \ldots, S^{(t-1)} = s_{k_{t-1}}, S^{(t)} = s_i) = \Pr(S^{(t+1)} = s_j | S^{(t)} = s_i), \tag{4.6}$$

onde $k_l \in \{0, 1, \ldots, r\}$, $l = 0, 1, \ldots, t-1$ e $s_i, s_j \in S$.

As probabilidades $p_{ij}$ são chamadas *probabilidades de transição*. O processo pode continuar no estado em que se encontra, e isto ocorre com probabilidade $p_{ii}$. A matriz quadrada $\mathbf{P}$ contendo os elementos $p_{ij}$ é chamada *matriz de transição*. Adicionalmente, se o número de estados é finito temos uma *cadeia de Markov de estados finitos*.

Considere a questão de determinar qual é a probabilidade de uma cadeia de Markov ir de um estado $s_i$ para um estado $s_j$ em dois passos. Denotamos esta probabilidade por $p_{ij}^{(2)}$. De uma forma geral, se uma cadeia de Markov possui $r$ estados, então

$$p_{ij}^{(2)} = \sum_{k=1}^{r} p_{ik} p_{kj}. \tag{4.7}$$

**Teorema 4.1** *Seja* $\mathbf{P}$ *a matriz de transição de uma cadeia de Markov. A* $ij$*-ésima entrada* $p_{ij}^{(n)}$ *da matriz* $\mathbf{P}^n$ *é a probabilidade de que a cadeia de Markov, iniciando no estado* $s_i$*, esteja no estado* $s_j$ *após* $n$ *passos (Prova do Teorema em [Grinstead 97]).*

### 4.2.1 Cadeias de Markov Ergódicas

Nesta seção iremos definir o conceito de *cadeias de Markov ergódicas*. Antes precisamos introduzir conceitos que classificam os estados de uma cadeia de Markov.

Vamos considerar a variável aleatória $T_i$ como sendo o próximo passo em que o estado $s_i$ é visitado (*hittint time*) dado que a cadeia se encontra no estado $s_i$:

$$T_i = \min\{n \geq 1 : S^{(n)} = s_i\}. \tag{4.8}$$

Um estado $s_i$ é *transiente* se existe uma probabilidade não-nula de que a cadeia nunca retorne para o estado $s_i$. Sendo assim, o estado $s_i$ é transiente se $1 - \Pr(T_i < \infty) > 0$. Se um estado $s_i$ não é transiente dizemos que ele é *recorrente*, ou seja, se $s_i$ é recorrente então $\Pr(T_i < \infty) = 1$.

Agora vamos discutir duas condições relacionadas às cadeias de Markov: *irredutibilidade* e *aperiodicidade*. A irredutibilidade é a propriedade em que todos os estados de uma cadeia de Markov podem ser alcançados por todos os outros estados [Häggström 02]. Considere uma cadeia de Markov $(S^{(0)}, S^{(1)}, \ldots)$ com um conjunto de espaços $S = \{s_1, s_2, \ldots, s_r\}$ e matriz de transição $\mathbf{P}$. Dizemos que um estado $s_i$ se "comunica" com um estado $s_j$, denotado por $s_i \to s_j$, se a cadeia possui uma probabilidade positiva de alcançar $s_j$ a partir de $s_i$. Em outras palavras, $s_i$ comunica-se com $s_j$ se existe um $n$ tal que

$$p_{ij}^{(n)} > 0. \tag{4.9}$$

Se $s_i \to s_j$ e $s_j \to s_i$ dizemos que os estados $s_i$ e $s_j$ se "intercomunicam", e escrevemos $s_i \leftrightarrow s_j$. A partir disso segue a definição de irredutibilidade.

**Definição 4.2.1** *Uma cadeia de Markov $(S^{(0)}, S^{(1)}, \ldots)$ com um conjunto de espaços $S = \{s_1, s_2, \ldots, s_r\}$ e matriz de transição $\mathbf{P}$ é **irredutível** se para todo $s_i, s_j \in S$ tivermos que $s_i \leftrightarrow s_j$.*

Agora vamos considerar o conceito de aperiodicidade.

**Definição 4.2.2** *Um estado recorrente $s_i$ é **periódico** de período $d$, se $d \geq 2$ é o maior divisor comum de todos os inteiros $n \geq 1$ para o qual $p_{ii}^{(n)} \geq 0$. Se não existe tal $d \geq 2$, então $s_i$ é **aperiódico**.*

**Definição 4.2.3** *Uma cadeia de Markov é **aperiódica** se todos os seus estados são aperiódicos. Caso contrário a cadeia é **periódica**.*

**Definição 4.2.4** *Uma cadeia de Markov é **ergódica** se a cadeia é irredutível e aperiódica.*

**Teorema 4.2** *Seja $\mathbf{P}$ a matriz de transição de uma cadeia de Markov ergódica. Então, quando $n \to \infty$, a potência $\mathbf{P}^n$ se aproxima de uma matriz $\mathbf{W}$ com todas as linhas sendo o mesmo vetor $\mathbf{v}$. O vetor $\mathbf{v}$ é a distribuição de probabilidade estácionária $F$ dos estados da cadeia (Prova do Teorema em [Grinstead 97]).*

Uma conseqüência do Teorema 4.2 é que a probabilidade de que um estado $s_j$ seja visitado é independente do estado inicial da cadeia de Markov ergódica quando o número de passos $n \to \infty$. Podemos ver isso através da matriz $\mathbf{W}$. Se todas as linhas de $\mathbf{W}$ são iguais a um vetor $\mathbf{v} = (v_1, v_2, \ldots, v_r)$, então uma coluna $j$ fixa de $\mathbf{W}$ contém sempre um mesmo valor. Sendo assim, para uma cadeia de Markov ergódica que possui $r$ estados temos $w_{1j} = w_{2j} = \ldots = w_{rj} = v_j$, para $j = 1, 2, \ldots, r$ e $w_{ij} \in \mathbf{W}$. Como $\mathbf{P}^{(n)}$ se aproxima de $\mathbf{W}$ quando $n \to \infty$, temos que $p_{ij}^{(n)} = w_{ij} = v_j$. Isso significa que

$p_{1j}^{(n)} = p_{2j}^{(n)} = \ldots = p_{rj}^{(n)} = w_{ij} = v_j$ para um estado fixo $s_j$. Assim, a matriz $\mathbf{W}$ teria a forma:

$$W = \begin{array}{c} \begin{matrix} s_1 & s_2 & \cdots & s_r \end{matrix} \\ \begin{bmatrix} v_1 & v_2 & \cdots & v_r \\ v_1 & v_2 & \cdots & v_r \\ \vdots & \vdots & \cdots & \vdots \\ v_1 & v_2 & \cdots & v_r \end{bmatrix} \end{array} \begin{matrix} s_1 \\ s_2 \\ \vdots \\ s_r \end{matrix}$$

Como vemos, a probabilidade $p_{ij}^{(n)}$ de ir de um estado $s_i$ para $s_j$ em um número grande de passos ($n \to \infty$) é $w_{ij} = v_j$, para $i = 1, 2, \ldots, r$. Ou seja, após o sistema iterar por um longo período de tempo, a probabilidade do sistema visitar o estado $s_j$ é $v_j$, independente do estado inicial $s_i$. Como dito anteriormente, o vetor $\mathbf{v}$ é a distribuição de probabilidade estácionária $F$ dos estados da cadeia.

## 4.3 Redes Booleanas com Perturbação e Cadeias de Markov

Uma propriedade interessante das redes Booleanas com pertubação é que seu comportamento dinâmico pode ser analisado no contexto de cadeias de Markov. Em [Brun 05] é fornecido uma fórmula explícita para a matriz de transição em termos das funções Booleanas de uma BN e da probabilidade de perturbação $p$, como veremos a seguir.

Considere o diagrama de estados de uma BN com perturbação $p$. A probabilidade do sistema transitar de um estado $s_i$ para um estado $s_j$ é dado por

$$p_{ij} = \mathbf{1}_{[s_j = \mathbf{f}(s_i)]}(1-p)^n + (1 - \mathbf{1}_{[s_i = s_j]})p^{\eta(s_i,s_j)} \times (1-p)^{n-\eta(s_i,s_j)}, \tag{4.10}$$

onde (a função indicadora) $\mathbf{1}_{[P]} = 1$ se a proposição $P$ é verdadeira; e $\mathbf{1}_{[P]} = 0$ se a proposição $P$ é falsa; $\eta(s_i, s_j)$ é a distância de Hamming (números de bits diferentes) entre $s_i$ e $s_j$; e $\mathbf{f} = (f_1|w_1, \ldots, f_n|w_n)$ é o vetor de funções Booleanas da BN.

Ainda em [Brun 05] é mostrado que, se $p > 0$, a cadeia de Markov correspondente à BN com perturbação é ergódica. A partir desta constatação, pelo Teorema 4.2, o sistema tem uma distribuição de probabilidade estacionária para os estados do sistema.

**Exemplo 4.3.1** *O gráfico apresentado na Figura 4.5 é a distribuição de probabilidade estacionária $F$ para a BN do Exemplo 4.1.1 com probabilidade de perbutação $p = 0,0001$. Observe que os estados que estão nos atratores (ver diagrama de transição de estados na Figura 4.2) têm mais probabilidade de ocorrer (estados* 0*,* 4*,* 19*,* 26*,* 30 *e* 31*). Observe*

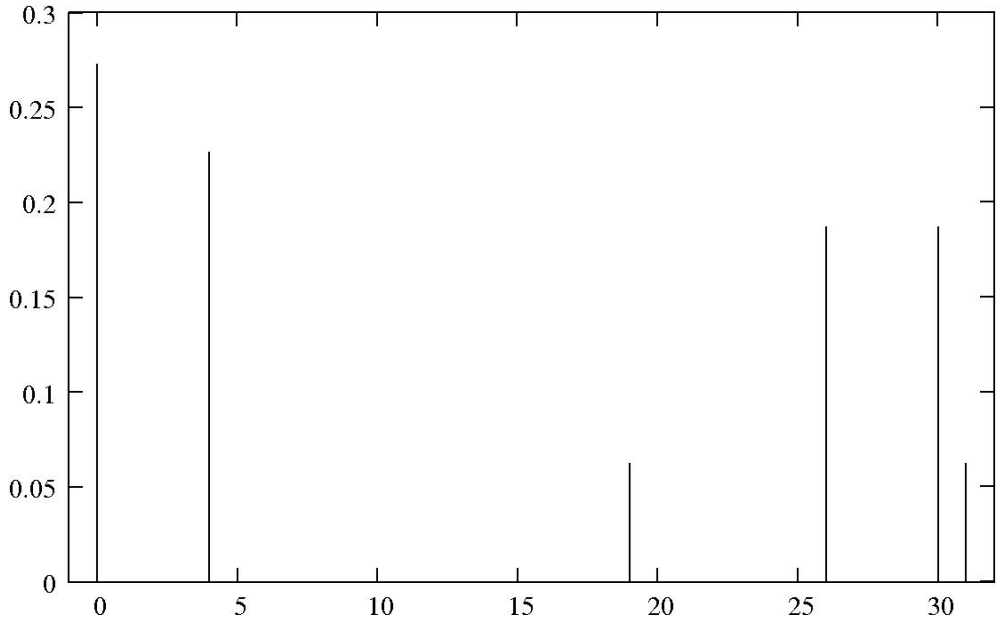


Figura 4.5: Distribuição de probabilidade estacionária de uma BN com 5 genes.

*ainda que a probabilidade do estado* 26 *é muito parecida com a probabilidade do estado* 30*, ou seja,* $F(26) \approx F(30)$*. Isto pode ter ocorrido pelo fato destes dois estados estarem em um mesmo atrator.*

## 4.4 Distribuição de Probabilidade Estacionária

Vimos na seção anterior que uma BN com perturbação está relacionada com uma cadeia de Markov ergódica. Esta cadeia possui uma matriz de transição $\mathbf{P}$ que pode ser calculada utilizando a fórmula apresentada na Seção 4.3.

Suponhamos agora que a cadeia de Markov percorra de estado em estado por um longo período de tempo. Assim, a matriz de transição $\mathbf{P}^n$, quando $n \to \infty$ irá representar a distribuição de probabilidade estacionária $F$ dos estados da BN com perturbação.

Neste sentido, quando modelamos um sistema biológico através de uma BN com pertubação, em uma situação de equilíbrio (isto é, depois do sistema iterar por um longo período de tempo), os estados seguem esta distribuição de probabilidade estacionária $F$. Assim, cada estado $s_i$ tem uma probabilidade $F(s_i)$ de ser observado, independente do estado inicial do sistema.

Considerando a distribuição de probabilidade estacionária $F$ do Exemplo 4.3.1 (Figura 4.5), podemos ver que $F$ é uma distribuição de probabilidade conjunta dos estados dos cinco genes que constituem o sistema. Dessa forma, as cinco variáveis da rede Booleana com pertubação podem ser consideradas como variáveis aleatórias $X_1, X_2, \ldots, X_5$. Assim,

$F(00000) \approx 0,275$ é a probabilidade conjunta $P(X_1 = 0, X_2 = 0, X_3 = 0, X_4 = 0, X_5 = 0)$.

Neste sentido, podemos dizer que a variáveis da rede Booleana com pertubação após iterar por um longo período de tempo podem ser consideradas como variáveis binárias aleatórias com distribuição de probabilidade conjunta dada pela distribuição de probabilidade estacionária da correspondente cadeia de Markov.

Neste contexto, seria muito pertinente perguntar qual é a dependência destas variáveis aleatórias uma com as outras dentro de uma rede Booleana com pertubação passado um longo período de tempo. Assim, poderíamos nos perguntar qual o valor do gene $X_1$ dados que os genes $X_2 = 1$ e $X_3 = 0$?[3] Como o valor de $X_1$ pode ter somente dois valores (0 ou 1), o problema de encontrar uma solução para esta pergunta significa encontrar um classificador que atribua um rótulo 0 ou 1 a $X_1$ com menor probabilidade de erro. Este problema é conhecido como problema de classificação e é muito importante dentro da área de Reconhecimento de Padrões.

## 4.5 Classificação

O problema de classificação, no contexto desta dissertação, envolve um vetor de variáveis aleatórias $\mathbf{X} = (X_1, X_2, \ldots, X_{n-1})$ (conhecido também como *vetor de características*) no espaço $\{0,1\}^{n-1}$ e uma variável binária aleatória $Y = X_n$ (*variável alvo*). O problema consiste em encontrar um *classificador* $\psi : \{0,1\}^{n-1} \to \{0,1\}$ que atua como um preditor de $Y = X_n$. Note que poderíamos ter escolhido qualquer uma das $n$ variáveis $\{X_1, X_2, \ldots, X_n\}$ como sendo a variável $Y$. Escolhemos a variável $X_n$ apenas por simplicidade. Se assumimos que existe uma distribuição de probabilidade conjunta $F$ para o par $(\mathbf{X}, Y)$, então o problema de classificação está completamente caracterizado e este consiste em encontrar um classificador que tenha a menor probabilidade de erro. Este classifcador é conhecido como classificador de Bayes.

### 4.5.1 Classificador de Bayes

O espaço de todos os classificadores, que no nosso caso é o espaço de todas as funções binárias em $\{0,1\}^{n-1}$ é denotado por $\mathcal{F}$. O erro $\varepsilon[\psi]$ de $\psi \in \mathcal{F}$ é a probabilidade de que a classificação esteja errada, ou seja, $\varepsilon[\psi] = P(\psi(\mathbf{X}) \neq Y)$. Podemos escrever isso como

$$\varepsilon[\psi] = E_F[|Y - \psi(\mathbf{X})|], \tag{4.11}$$

onde a esperança é tomada em relação a distribuição de probabilidade conjunta $F$ para o par $(\mathbf{X}, Y)$ (indicado pela notação $E_F$). Em outras palavras, $\varepsilon[\psi]$ é igual ao erro médio

[3]Note que, para este exemplo, esta dependência é diferente da dependência dada pela definição da função Booleana $f_1$ (veja Tabela 4.1) onde o gene $x_1$ depende dos genes $x_5$, $x_2$ e $x_4$. Aqui estamos considerando dependência das variáveis aleatórias após o sistema iterar por um longo período de tempo.

absoluto (MAE) entre o rótulo e a classificação. Devido à natureza binária de $\psi(\mathbf{X})$ e $Y$, $\varepsilon[\psi]$ também se iguala ao erro quadrado médio (MSE) entre o rótulo e a classificação:

$$\varepsilon[\psi] = E_F[|Y - \psi(\mathbf{X})|^2]. \tag{4.12}$$

Um *classificador ótimo* $\psi_{\mathbf{X}}$, é aquele que possui o menor erro $\varepsilon_{\mathbf{X}}$ entre todos os $\psi \in \mathcal{F}$. O classificador ótimo $\psi_{\mathbf{X}}$ é chamado de *classificador de Bayes* e seu erro $\varepsilon_{\mathbf{X}}$ é chamado de *erro de Bayes*. O classificador de Bayes e seu erro dependem da distribuição de probabilidade conjunta $F(\mathbf{X}, Y)$, ou seja, como os rótulos estão distribuídos entre as variáveis usadas para discriminá-los, e como as variáveis estão distribuídas no espaço $\{0,1\}^{n-1}$.

### 4.5.2 Erro de Bayes

Nesta subseção, iremos mostrar como se calcula o erro de Bayes para classificar (ou predizer) o valor da variável aleatória $Y = X_n$ usando como variáveis preditoras um subconjunto das variáveis aleatórias $\{X_1, \ldots, X_{n-1}\}$.

Seja $\mathbf{X}^i$ a *configuração* do vetor aleatório $\mathbf{X} = (X_1, X_2, \ldots, X_{n-1})$ cuja expressão binária é igual ao inteiro $i$. Por exemplo, para $n-1 = 3$, $\mathbf{X}^0 = 000$, $\mathbf{X}^1 = 001$,..., $\mathbf{X}^7 = 111$. Para $n-1$ variáveis, existem $m = 2^{n-1}$ possíveis configurações. Assumindo que as variáveis aleatórias binárias $(X_1, X_2, \ldots, X_{n-1}, Y = X_n)$ possuem uma distribuição de probabilidade estacionária dos estados $F$ obtida a partir do comportamento dinâmico da correspondente BN com perturbação, para cada configuração $\mathbf{X}^i$, podemos calcular as probabilidades $r_i = \Pi(\mathbf{X} = \mathbf{X}^i)$ e $p_i = \Pi(Y = 1|\mathbf{X}^i)$.

#### Erro de Bayes na Ausência de Observações

O classificador de Bayes para $Y = X_n$ em termos do erro absoluto médio (MAE), na ausência de observações dos valores de outras variáveis, é a sua esperança $E[Y] = \mu = \sum_{i=0}^{m-1} p_i\, r_i$ seguida por uma função de *threshold* binária $T : [\,0,1] \to \{0,1\}$ definida como $T(x) = 0$ se, e somente se, $x \leq 0,5$. O erro de Bayes para este classificador é dado por

$$\varepsilon_\bullet = \begin{cases} \mu & \text{se } \mu \leq 0,5 \\ 1-\mu & \text{se } \mu > 0,5. \end{cases} = \min\{\mu, 1-\mu\} \tag{4.13}$$

#### Erro de Bayes Considerando Observações de Outras Variáveis

Agora, se levarmos em conta as observações das outras variáveis aleatórias, então podemos projetar um outro classificador para $Y$ e diminuir o erro $\varepsilon_\bullet$. Seja $\mathbf{Z} = \{X_{j_1}, X_{j_2}, \ldots, X_{j_\ell}\}$ um subconjunto das variáveis $\{X_1, X_2, \ldots, X_{n-1}\}$ tal que $1 \leq j_1 < j_2 < \ldots < j_\ell \leq n-1$.

Para se obter uma fórmula analítica para o erro de Bayes em termos do MAE baseado nas observações das variáveis $X_{j_1}, X_{j_2}, \ldots, X_{j_\ell}$, considere $\mathbf{Z}^k$ como a configuração do vetor aleatório $\mathbf{Z} = (X_{j_1}, X_{j_2}, \ldots, X_{j_\ell})$ cuja expressão binária é igual ao inteiro $k$. Para uma configuração fixa $\mathbf{Z}^k$ de $\mathbf{Z}$, existem algumas configurações $\mathbf{X}^i$ do vetor $\mathbf{X}$ que se emparelham com $\mathbf{Z}^k$. Por exemplo, suponha que $\mathbf{X} = (X_1, X_2, X_3, X_4, X_5)$ e que desejamos predizer $Y = X_6$ usando somente as variáveis $X_2$ e $X_4$. Assim, $\mathbf{Z} = (X_2, X_4)$ e, para $k = 1$, temos $\mathbf{Z}^1 = 01$. As configurações de $\mathbf{X}^i$ que emparelham com $\mathbf{Z}^1$ são dadas na seguinte tabela:

| $i$ | $\mathbf{X}^i$ | $i$ | $\mathbf{X}^i$ | $i$ | $\mathbf{X}^i$ | $i$ | $\mathbf{X}^i$ |
|---|---|---|---|---|---|---|---|
| 2 | 0$\underline{0}$1$\underline{0}$ | 3 | 0$\underline{0}$1$\underline{1}$ | 6 | 0$\underline{1}$1$\underline{0}$ | 7 | 0$\underline{1}$1$\underline{1}$ |
| 18 | 1$\underline{0}$0$\underline{1}$0 | 19 | 1$\underline{0}$0$\underline{1}$1 | 22 | 1$\underline{0}$1$\underline{1}$0 | 23 | 1$\underline{0}$1$\underline{1}$1 |

(4.14)

Se considerarmos as configurações para $\mathbf{X}^i$ como $(X^i_1, \ldots, X^i_{j_1}, \ldots, X^i_{j_2}, \ldots, X^i_{j_\ell}, \ldots, X^i_n)$ e para $\mathbf{Z}^k$ como $(X^k_{j_1}, X^k_{j_2}, \ldots, X^k_{j_\ell})$, então, para um $k$ fixo, podemos definir o conjunto $\{i : \mathbf{X}^i \sim \mathbf{Z}^k\}$ como $\{i : \mathbf{X}^i \sim \mathbf{Z}^k\} = \{i : X^i_{j_1} = X^k_{j_1}, X^i_{j_2} = X^k_{j_2}, \ldots, X^i_{j_\ell} = X^k_{j_\ell}\}$. Por exemplo, se $k = 1$, o conjunto $\{i : \mathbf{X}^i \sim \mathbf{Z}^1\}$ para o exemplo anterior é $\{2, 3, 6, 7, 18, 19, 22, 23\}$.

Agora, apresentamos duas probabilidades:

$$A_k(\mathbf{Z}) = P(\mathbf{Z} = \mathbf{Z}^k) = \sum_{\{i:\mathbf{X}^i \sim \mathbf{Z}^k\}} r_i$$

e

$$B_k(\mathbf{Z}) = P(Y = 1 \text{ e } \mathbf{Z} = \mathbf{Z}^k) = \sum_{\{i:\mathbf{X}^i \sim \mathbf{Z}^k\}} p_i\, r_i.$$

Em termos dessas probabilidades, o erro do classificador de Bayes para predizer o valor de $Y$ usando as variáveis do vetor $\mathbf{Z}$ é

$$\varepsilon_{\mathbf{Z}} = \sum_{k=0}^{2^\ell - 1} \min\{B_k(\mathbf{Z}), A_k(\mathbf{Z}) - B_k(\mathbf{Z})\}. \tag{4.15}$$

## 4.6 Seleção de Características

Vimos na seção anterior que dada a distribuição de probabilidade conjunta $F$ para o par de variáveis aleatórias $(\mathbf{X} = \{X_1, X_2, \ldots, X_{n-1}\}, Y = X_n)$, podemos obter a fórmula analítica para calcular o erro de Bayes para quaisquer subconjuntos de $\mathbf{X} = \{X_1, X_2, \ldots, X_{n-1}\}$. Dessa forma, podemos selecionar, entre todos os subconjuntos possíveis de $\mathbf{X}$, aqueles que tenham o menor erro de Bayes para predizer (classificar) a variável aleatória $Y = X_n$, ou seja, escolher aqueles subconjuntos que nos permitam fazer a melhor predição da variável $Y$.

O problema de predição/classificação descrito acima pode ser visto como um problema de seleção de características em Reconhecimento de Padrões: selecionar um subconjunto de $k$ variáveis aleatórias (genes) de um conjunto de $n-1$ variáveis que fornece um classificador ótimo com erro mínimo entre todos os classificadores para subconjuntos de tamanho $k$.

Suponhamos que a distribuição de probabilidade de onde as expressões gênicas são obtidas seja conhecida. Se $\{\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_r\}$ é o conjunto dos possíveis subconjuntos formados a partir do conjunto $\{X_1, X_2, \ldots, X_{n-1}\}$ de variáveis aleatórias binárias assumindo que $i < j$ se $\mathbf{Z}_i \subset \mathbf{Z}_j$, e se $Y = X_n$ é uma variável aleatória binária para ser classificada, então existe uma distribuição de probabilidade conjunta $P(X_1, X_2, \ldots, X_{n-1}, Y = X_n)$ tal que $\varepsilon_{\mathbf{Z}_1} > \varepsilon_{\mathbf{Z}_2} > \cdots > \varepsilon_{\mathbf{Z}_r}$, onde $\varepsilon_{\mathbf{Z}_j}$ é o erro de Bayes do classificador de Bayes $\psi_{\mathbf{Z}_j}$ aplicado ao subconjunto $\mathbf{Z}_j$, dado por $\varepsilon_{\mathbf{Z}_j} = P(Y \neq \psi_{\mathbf{Z}_j}(\mathbf{Z}_j))$. A condição $i < j$ se $\mathbf{Z}_i \subset \mathbf{Z}_j$ é necessária porque esta condição implica que $\varepsilon_{\mathbf{Z}_i} \geq \varepsilon_{\mathbf{Z}_j}$. Uma conseqüência disso é que para se encontrar os melhores subconjuntos de tamanho $k$ (seleção de características), todos os subconjuntos de $k$ elementos devem ser verificados. Várias heurísticas têm sido desenvolvidas para contornar o problema combinatorial de calcular o erro de Bayes de todos os $\binom{n-1}{k}$ subconjuntos.

Podemos abordar o problema de seleção de características em termos de coeficiente de determinação, que nos fornece uma medida normalizada de quanto a variável aleatória $Y$ pode ser melhor predita ao observarmos subconjuntos de outras variáveis aleatórias do que com ausência de observação.

## 4.7 Coeficiente de Determinação

Em termos gerais, o *coeficiente de determinação* (CoD) mede o quanto a predição de um gene alvo $Y$ (variável aleatória alvo) é melhorada pela informação obtida a partir de um subconjunto de genes preditores (variáveis aleatórias) $\mathbf{Z} \subseteq \{X_1, X_2, \ldots, X_{n-1}\}$ em relação à própria capacidade preditiva de $Y$. Mais formalmente, o *CoD* do gene alvo $Y$ relativo ao subconjunto de genes preditores $\mathbf{Z}$ é definido como

$$\theta_{\mathbf{Z}}(Y) = \frac{\varepsilon_\bullet - \varepsilon_{\mathbf{Z}}}{\varepsilon_\bullet}, \tag{4.16}$$

onde $\varepsilon_\bullet$ é o erro de Bayes para predizer $Y$ na ausência de qualquer observação de variáveis preditores e $\varepsilon_{\mathbf{Z}}$ é o erro de Bayes de $\mathbf{Z}$ para predizer $Y$.

Observe que, conhecida a distribuição de probabilidade conjunta das variáveis aleatórias, podemos calcular o CoD calculando o erro de Bayes para $\varepsilon_\bullet$ e $\varepsilon_{\mathbf{Z}}$ utilizando diretamente as Equações (4.13) e (4.15).

Pela definição de CoD, é evidente que $0 \leq \theta_{\mathbf{Z}}(Y) \leq 1$. Observe que, por um lado, um CoD igual a 0 (zero) significa que não existe nenhum acréscimo no poder de predição

da expressão do gene alvo a partir dos genes preditores. Por outro lado, um CoD igual a 1 (um) significa que o conhecimento da expressão dos genes preditores permite obter completamente a expressão do gene alvo. Dessa forma, em termos de coeficiente de determinação, o nosso problema de seleção de características passa a ser o de encontrar subconjuntos de variáveis aleatórias de tamanho $k$ com CoDs altos.

## 4.8 Genes de Predição Intrinsicamente Multivariada

Uma propriedade importante do CoD é que dado um gene alvo fixo, o CoD aumenta na medida em que novos genes preditores são adicionados. Sendo assim, para um gene alvo $Y$ temos que $\theta_{\{X_1,\ldots,X_{k-1}\}}(Y) \leq \theta_{\{X_1,\ldots,X_k\}}(Y)$. Embora ambos os CoDs possam ter o mesmo valor é muito comum obter uma desigualdade restrita. Dessa maneira, obtemos incrementos de CoD da forma $\theta_{\{X_1\}}(Y) < \theta_{\{X_1,X_2\}}(Y) < \theta_{\{X_1,X_2,X_3\}}(Y)$. Se os genes preditores fornecem informação suficiente, o CoD atinge valores altos após um certo número de genes adicionados.

Considere a seguinte configuração de CoDs apresentada na Tabela 4.2. De acordo com a tabela o CoD é alto quando os três genes são usados simultaneamente como preditores, mas se cada gene for usado individualmente como preditor os seus CoDs são muito baixos, ou seja, nenhuma relação preditiva é obtida utilizando-se apenas um gene de cada vez. Para entender melhor a situação exposta, um gene alvo que satisfaz esta relação poderia ter um papel importante e vital dentro da célula, pois necessita de todos os três genes para ser controlado. Assim, genes alvos que satisfazem esta propriedade devem ser genes envolvidos em processos biológicos na qual somente podem ser ativados ou desativados por um determinado subconjunto de genes de forma simultânea. Genes alvos que se comportam desta maneira são chamados de *genes de predição intrinsicamente multivariada*, ou simplesmente *genes IMP* (do inglês *Intrinsically Multivariately Predictive genes*).

| $\mathbf{Z}$ | $\theta_{\mathbf{Z}}(Y)$ |
|---|---|
| $\{X_1\}$ | $\approx 0$ |
| $\{X_2\}$ | $\approx 0$ |
| $\{X_3\}$ | $\approx 0$ |
| $\{X_1, X_2, X_3\}$ | $\approx 1$ |

Tabela 4.2: Configuração de CoDs.

Quando temos a configuração da Tabela 4.2 dizemos que $Y$ é um gene IMP de grau 1. Este conceito pode ser estendido para subconjuntos maiores de genes preditores. Por exemplo, considere a segunda configuração de CoDs na Tabela 4.3 abaixo:

Neste caso, os genes preditores não ajudam a predizer o estado do gene alvo $Y$ quando tomados de dois em dois. Porém, quando tomamos os três genes predidores simultaneamente temos uma forte relação preditiva. Sendo assim, $Y$ é um gene IMP de grau 2.

| $\mathbf{Z}$ | $\theta_{\mathbf{Z}}(Y)$ |
|---|---|
| $\{X_1, X_2\}$ | $\approx 0$ |
| $\{X_1, X_3\}$ | $\approx 0$ |
| $\{X_2, X_3\}$ | $\approx 0$ |
| $\{X_1, X_2, X_3\}$ | $\approx 1$ |

Tabela 4.3: Segunda configuração de CoDs.

Para entender melhor o interesse em genes que se comportam como um gene IMP vamos considerar um exemplo simplificado de transcrição, onde a informação genética é transferida do DNA para o RNA. Na Figura 4.6 temos três genes: A, B e C. Para que o gene C seja transcrito ele necessita dos fatores de transcrição A e B. Se apenas o fator de transcrição A estiver presente o gene C não é transcrito. O mesmo ocorre se tivermos apenas o fator de transcrição B presente. Sendo assim, o gene C necessita de ambos os genes, A e B, e seus respectivos fatores de transcrição. Neste exemplo o gene C está fazendo o papel de gene IMP, onde os genes A e B atuam como seus preditores.

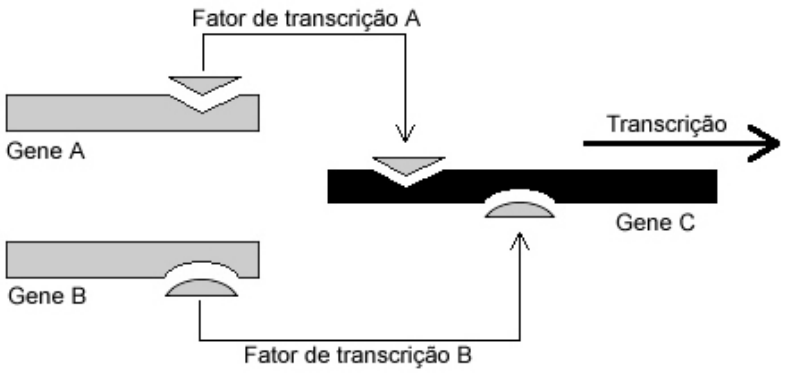


Figura 4.6: Transcrição do gene C.

Finalizando, vimos que a interação entre os genes de um sistema biológico pode ser modelada por uma rede Booleana com perturbação e cadeias de Markov, onde podemos obter a distribuição estacionária dos estados da rede. Assumindo que tal distribuição é conhecida, podemos utilizar o erro de Bayes para calcular os coeficientes de determinação. A partir dos CoDs, podemos encontrar então os genes IMP. Uma questão que não foi focada em nossa pesquisa mas que pode ser estudada futuramente é a de encontrar características em redes Booleanas com pertubação que seriam necessárias e suficienes para a existência de genes IMP.

# Capítulo 5

# Modelando Dados de Expressão Gênica usando Redes Booleanas com Perturbação

Vimos no capítulo anterior que se um sistema biológico é modelado usando redes Booleanas com pertubação e cadeias de Markov, então podemos obter a distribuição estacionária dos estados da rede. Esta distribuição corresponde à distribuição de probabilidade conjunta dos genes que compõem a rede. Neste sentido, podemos dizer que as variáveis da rede Booleana com pertubação após iterar por um longo período de tempo podem ser consideradas como variáveis aleatórias com distribuição de probabilidade conjunta dada pela distribuição de probabilidade estacionária da correspondente cadeia de Markov. Vimos que conhecendo tal distribuição, podemos calcular o erro de Bayes e conseqüentemente obter os CoDs. Posteriormente, a partir dos CoDs, podemos encontrar os genes IMP.

Neste capítulo, iremos considerar um sistema biológico modelado por uma rede Booleana com pertubação em que as funções Booleanas e a probabilidade de pertubação não são conhecidas. Conseqüentemente, a distribuição de probabilidade conjunta não é conhecida (como a maioria dos problemas em Reconhecimento de Padrões). Assim, os CoDs devem ser estimados a partir de observações (amostras) do sistema biológico. No nosso caso, estas observações são dados de expressão gênica provenientes de microarrays. Desta forma, para calcular os CoDs e realizar a busca por genes IMP, devemos entender como fazer a análise destes dados. Com este intuito, vamos modelar os dados de expressão gênica provenientes de microarrays usando o modelo de redes Booleanas com perturbação.

## 5.1 Dados de Expressão Gênica

É possível encontrar na literatura várias pesquisas em Bioinformática que usam dados temporais de expressão gênica provenientes de microarrays. Estes dados são obtidos através de observações (microarrays) do sistema biológico em certos intervalos de tempo. No entanto, a maioria dos dados de microarray são expressões gênicas (não temporais) que foram resultados de amostras obtidas observando o sistema em estado de equilíbrio.

### 5.1.1 Sistema em Equilíbrio

Um sistema dinâmico entra em estado de equilíbrio após ele iterar por um longo período de tempo. Sob o modelo de redes Booleanas, isto significa que o sistema está em um atrator (Figura 5.1), ou seja, o sistema em equilíbrio muda de estado em estado formando um ciclo. No modelo de redes Booleanas com pertubação, a dinâmica do sistema é representada por uma cadeia de Markov ergódica que após iterar por um longo período de tempo entra em regime estacionário, ou seja, existe uma distribuição estacionária dos estados da rede. Esta distribuição corresponde à distribuição de probabilidade conjunta dos genes que compõem a rede. Como vimos, a maioria da massa de probabilidade desta distribuição está concentrada nos estados que estão nos atratores (Figura 4.5). Dessa forma, dentro do modelo de redes Booleanas (com ou sem pertubação), podemos considerar que o sistema está em equilíbrio quando ele se encontra em um atrator. Assim, é esperado que a maioria dos dados obtidos a partir de sistemas em equilíbrio sejam provenientes de estados que estão em atratores [Brun 05, Pal 05].

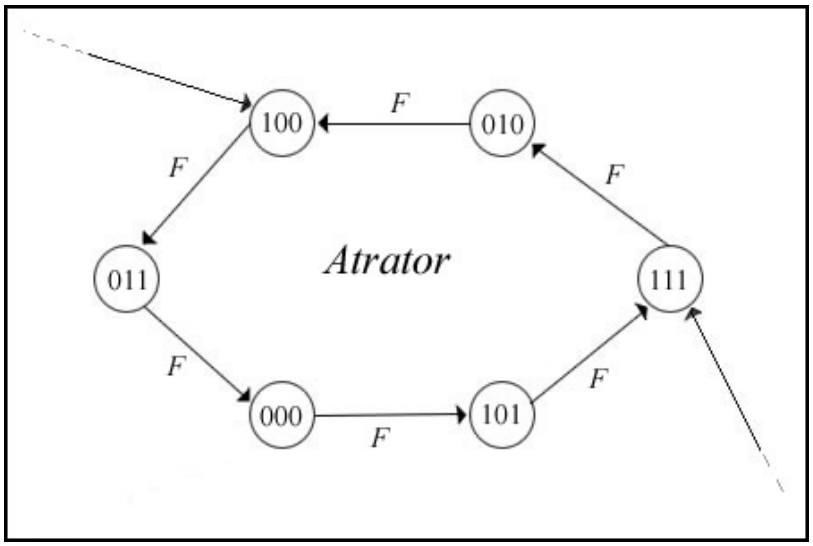


Figura 5.1: Atrator de uma rede Booleana.

Sob a hipótese que estamos em um sistema em equilíbrio, uma estabilidade biológica do estado de uma célula poderia significar que alguns genes mantêm fixo (por certo período de

tempo) seu padrão de expressão. Neste caso, dentro do modelo Booleano, se consideramos as configurações dos estados possíveis da célula observando os valores de expressão gênica somente deste grupo de genes, o sistema estaria em um ciclo consistindo somente de um único estado (veja Figura 5.2). Assim, se o diagrama de transição de estados possuísse somente um único atrator, este consistiria de uma árvore cuja raiz seria este atrator. No caso de se ter mais atratores, este sistema geraria um diagrama de estados consistindo de várias árvores (ou seja, o diagrama seria uma floresta). Para fins de nossa pesquisa, considerando sistemas em equilíbrio em estabilidade biológica, vamos modelar os sistemas biológicos usando redes Booleanas com pertubação em que o correspondente diagrama de transição de estados da rede Booleana é uma floresta (ou seja, todos os atratores consistem somente de um único estado) como descrito anteriormente (veja Figura 5.3). No contexto de redes Booleanas com pertubação, após o sistema iterar por um longo período de tempo, ele estaria em um atrator de um único estado (ou seja, em estabilidade biológica) por um certo tempo até que ocorresse uma pertubação que levaria o sistema para a mesma (ou a uma outra) bacia de atração e conseqüentemente ao mesmo (ou a um outro) atrator.

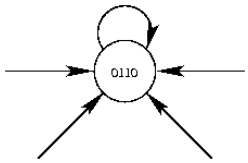


Figura 5.2: Atrator de uma rede Booleana com um único estado. Neste caso, o padrão dos genes se mantém fixo no estado 0110.

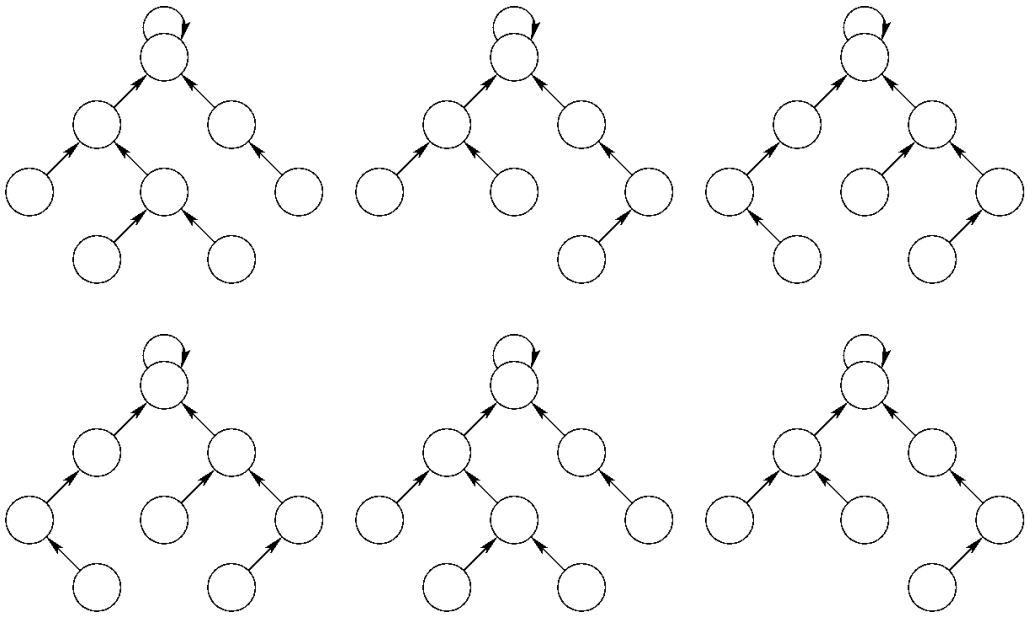

Figura 5.3: Digrama de transição de uma rede Booleana com 6 árvores.

## 5.2 Microarrays e Redes Booleanas com Perturbação

Vamos supor, para efeitos de explicação, que estamos interessados em estudar a interação entre os genes relacionados a alguma doença, por exemplo, a leucemia mielóide aguda (LMA). No modelo proposto estamos supondo que exista uma rede biológica para a LMA, e que esta rede pode ser modelada por uma rede Booleana com perturbação como foi descrito acima.

Dado que este sistema esteja em equilíbrio, esperamos que ele esteja em um atrator (que contém um único estado). Considerando esta situação, desejamos analisar o sistema tirando uma "fotografia" para saber em qual estado da rede o sistema se encontra. Tal fotografia corresponde a um microarray que nos permite analisar a expressão dos genes naquele instante (Figura 5.4). Considerando, por exemplo, que este sistema possui $n = 7$ genes, teremos um total de $2^7 = 128$ estados possíveis. Se o estado do atrator é o $s_{25}$, microarray obtido do sistema biológico possui a expressão dos genes correspondente a este estado (Figura 5.5).

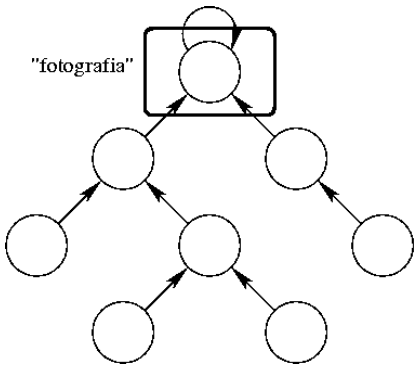


Figura 5.4: Tirando uma fotografia do sistema.

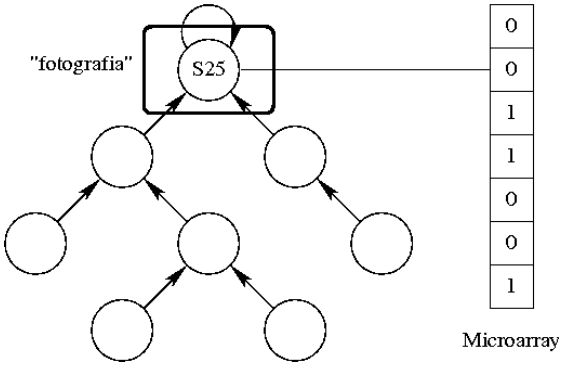


Figura 5.5: Microarray correspondente à foto.

Neste modelo, analisar amostras de microarrays de vários pacientes com LMA, digamos $P_A, P_B, \ldots, P_N$, seria como se tivéssemos várias fotografias do mesmo sistema. Existe uma probabilidade de que cada estado que representa um microarray esteja em um atrator

diferente quando as amostras forem retiradas. Eventualmente, podemos ter um mesmo microarray para pacientes diferentes. A Figura 5.6 ilustra a situação.

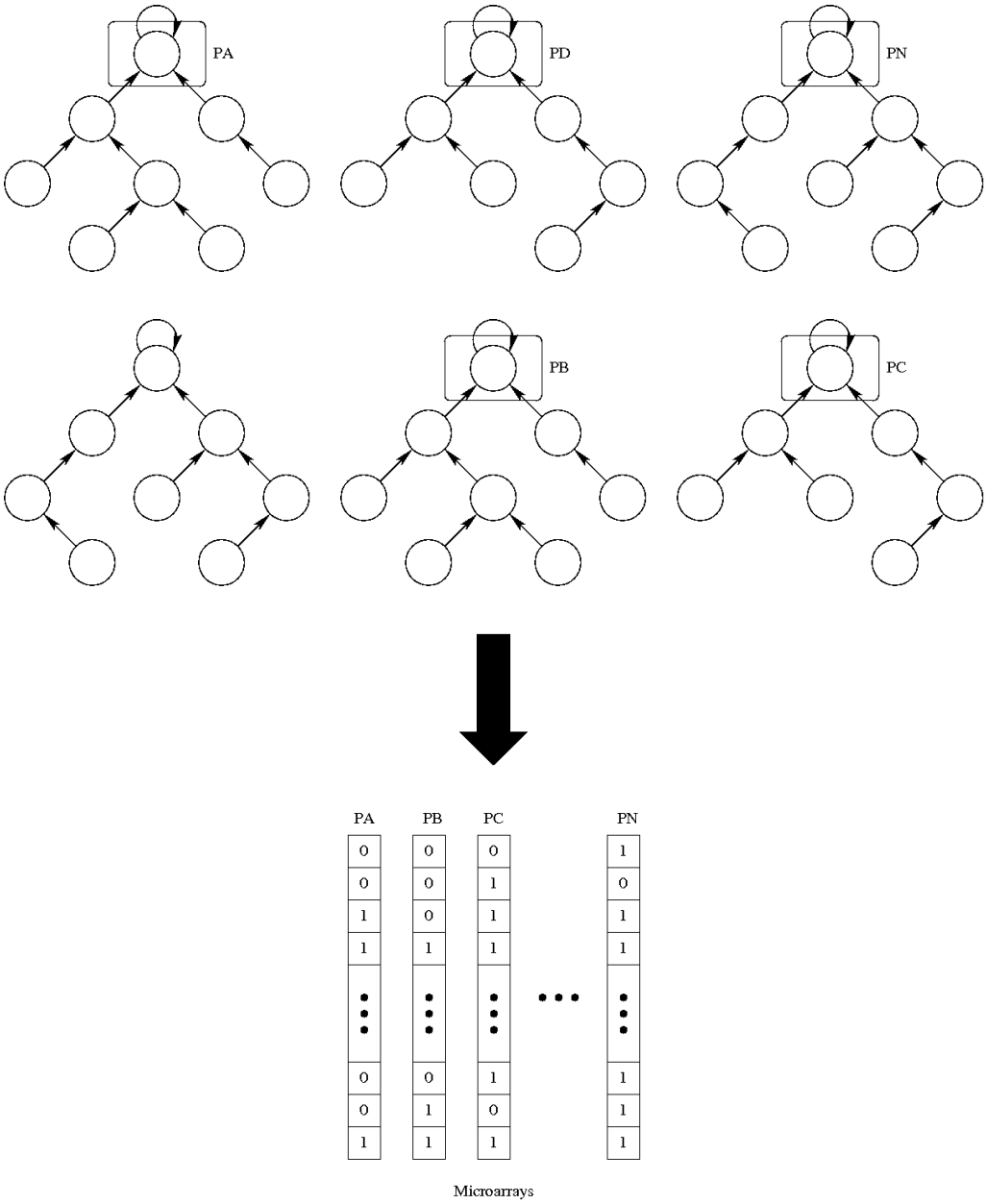


Figura 5.6: Microarrays de vários pacientes.

Neste caso, como cada paciente é uma pessoa distinta, cada amostra é independente e identicamente distribuída (i.i.d.) uma da outra. Além disso, as amostras são retiradas após um período de tempo em que supomos o equilíbrio do sistema, ou seja, cada amostra corresponde a um estado dentro de um atrator. No modelo de redes Booleanas com pertubação, isto significa que os microarrays são amostras i.i.d. que são obtidas aleatoriamente seguindo a distribuição estacionária de probabilidades (veja um exemplo

na Figura 4.5).

Dessa forma, dentro deste modelo, como cada microarray é um estado do sistema que está em um atrator de tamanho 1 (um), temos que para predizer a expressão gênica de um gene alvo $Y$ usando um conjunto $\mathbf{Z}$ de genes preditores, podemos considerar que os valores de $Y$ e $\mathbf{Z}$ estão na mesma amostra (microarray). No próximo capítulo veremos como um classificador é construído, a partir das amostras de microarray, para predizer a expressão do gene alvo $Y$.

# Capítulo 6

# Estimação de Erro

O cálculo do erro do classificador de Bayes está ligado diretamente com o cálculo dos coeficientes de determinação, uma vez que, como vimos na Equação 4.16, os CoDs são calculados através dos erros $\varepsilon_\bullet$ e $\varepsilon_{\mathbf{Z}}$. No entanto, a expressão de cada gene é representada por uma variável aleatória cuja distribuição de probabilidade é desconhecida. Dessa forma, os erros devem ser obtidos através da análise da expressão dos genes fornecidos pelas amostras de microarray. Neste capítulo, vamos considerar um conjunto de $N$ amostras (microarrays) $S_N = \{s_1, s_2, \ldots, s_N\}$. Cada $s_i$ é uma configuração binária de $n$ variáveis aleatórias (genes), ou seja, $s_i = (x_1, x_2, \ldots, x_n)$.

Para calcular os CoDs referentes a um gene alvo $Y$ analisamos conjuntos $\mathbf{Z}$ de $k$ genes preditores. Além disso, todas as $\binom{n-1}{k}$ combinações de genes são consideradas para calcular os CoDs. Sendo assim, o vetor $\mathbf{z}_i$ representa a configuração dos genes preditores na amostra $s_i$ e $y_i$ é a expressão gênica do gene alvo $Y$ na amostra $s_i$.

Como a distribuição de probabilidade das variáveis aleatórias que representam as expressões gênicas é desconhecida, o cálculo dos CoDs é feito com erros estimados e portanto, o valor do CoD também é uma estimativa:

$$\hat{\theta}_{\mathbf{Z}}(Y) = \frac{\hat{\varepsilon}_\bullet - \hat{\varepsilon}_{\mathbf{Z}}}{\hat{\varepsilon}_\bullet}. \tag{6.1}$$

O erro $\hat{\varepsilon}_{\mathbf{Z}}$ é, na verdade, o erro do classificador $\psi_N$ ao utilizar o conjunto de genes preditores $\mathbf{Z}$ para predizer a expressão de $Y$. O classificador $\psi_N$ é construído a partir do conjunto de amostras $S_N$. Na próxima seção veremos um exemplo de como construir um classificador $\psi_N$. E na Seção 6.2 apresentaremos alguns estimadores de erro estudados e implementados durante a pesquisa.

## 6.1 Construindo um classificador a partir de amostras de microarray

Uma regra que podemos utilizar para projetar um classificador $\psi_N$ no caso de variáveis aleatórias binárias é a *discriminação multinomial*. Nesta regra, um número finito de padrões observados são possíveis, digamos $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{m-1}$. Se estamos analisando $k$ genes, então $m = 2^k$. Atribuímos então a cada padrão $\mathbf{x}_i$ o rótulo que possui a freqüência máxima relativa as amostras correspondentes a $\mathbf{x}_i$. Em outras palavras, podemos dizer que é gerada uma tabela de freqüência para contar cada padrão observado. O rótulo que aparecer mais vezes para o padrão $\mathbf{x}_i$ é a classe escolhida para o padrão $\mathbf{x}_i$ [Dougherty 05].

Para exemplificar a construção de um classificador $\psi_N$ utilizando-se a discriminação multinomial vamos considerar $\mathbf{Z} = \{x_2, x_5\}$ o conjunto de genes preditores e $Y = x_1$ o gene alvo. Temos então um total de $m = 2^2$ padrões possíveis. Abaixo temos os perfis gênicos de $x_1$, $x_2$ e $x_5$ em um conjunto de $N = 17$ amostras:

| | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ | $s_7$ | $s_8$ | $s_9$ | $s_{10}$ | $s_{11}$ | $s_{12}$ | $s_{13}$ | $s_{14}$ | $s_{15}$ | $s_{16}$ | $s_{17}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| $x_2$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| $x_5$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

Tabela 6.1: Conjunto de amostras.

A partir do conjunto de amostras acima montamos uma tabela de freqüência para contar cada um dos $m = 4$ padrões possíveis para $x_2$ e $x_5$:

| $\mathbf{x}_i = (x_2, x_5)$ | $x_1 = 0$ | $x_1 = 1$ | $\psi_N(\mathbf{x}_i)$ |
|---|---|---|---|
| $\mathbf{x}_0 = (0,0)$ | 6 | 4 | 0 |
| $\mathbf{x}_1 = (0,1)$ | 0 | 1 | 1 |
| $\mathbf{x}_2 = (1,0)$ | 0 | 3 | 1 |
| $\mathbf{x}_3 = (1,1)$ | 0 | 3 | 1 |

Tabela 6.2: Tabela de freqüência.

A partir da Tabela 6.2 podemos observar que o classe 0 é atribuída ao gene $x_1$ sempre que o padrão $\mathbf{x}_0 = (0,0)$ é observado. Em outras palavras, dizemos que o gene $x_1$ está "desligado" quando $x_2$ e $x_5$ estão "desligados". Qualquer outra configuração de $x_2$ e $x_5$ classifica o gene $x_1$ como "ligado". Neste exemplo utilizamos todas as $N = 17$ amostras para construir o classificador. Na próxima seção veremos que alguns métodos de estimação de erro podem usar apenas uma parte da amostra para construir $\psi_N$ e a outra parte é utilizada para testar o classificador e assim estimar o seu erro.

## 6.2 Estimadores de erro

Um estimador de erro pode ser *não-aleatório* ou *aleatório*. Entre os estimadores não-aleatórios temos o método da resubstituição, *leave-one-out* e *fixed-fold cross validation*. Já os estimadores de erro aleatórios possuem algum aspecto aleatório interno que influenciam em sua saída. Os mais populares são o *random-fold cross validation* e todos os estimadores de erro do tipo *bootstrap*. Discutiremos alguns estimadores de erro mais adiante.

Um fator que diz respeito ao desempenho de um estimador de erro é a sua variância, especialmente quando temos poucas amostras [Dougherty 05]. A *variância interna* é a variância devida apenas aos seus fatores internos. Nos estimadores não-aleatórios essa variância é zero. A *variância total* de um estimador de erro leva em consideração as incertezas introduzidas pelas amostras. Para os estimadores aleatórios devemos fazer com que a variância interna seja a menor possível com a intenção de diminuir a variância total do estimador. Devido a esse fator os estimadores aleatórios podem ser ineficientes computacionalmente.

Antes de apresentarmos os métodos de estimação de erro vamos analisar um caso particular que pode acontecer quando trabalhamos com erros estimados. Quando trabalhamos com estimação de CoD devemos considerar o caso em que o erro estimado $\hat{\varepsilon}_\bullet$ é 0 (zero). Quando isso ocorre, devemos redefinir o CoD [Zhou 03b]. Para isso, vamos considerar duas relações que conhecemos a priori em teoria que devem ser satisfeitas:

$$\hat{\varepsilon}_{\mathbf{Z}} \leq \hat{\varepsilon}_\bullet, \tag{6.2}$$

e

$$\hat{\varepsilon}_\bullet > 0. \tag{6.3}$$

Vamos analisar as quatro possibilidades que podem ocorrer:

(1) $0 < \hat{\varepsilon}_{\mathbf{Z}} \leq \hat{\varepsilon}_\bullet$: Neste caso a definição de CoD é aplicada diretamente.

(2) $0 < \hat{\varepsilon}_\bullet < \hat{\varepsilon}_{\mathbf{Z}}$: Nós utilizamos a condição em 6.2 e ajustamos $\hat{\varepsilon}_{\mathbf{Z}} = \hat{\varepsilon}_\bullet$. Logo, $\hat{\theta}_{\mathbf{Z}}(Y) = 0$ de acordo com a definição original.

(3) $\hat{\varepsilon}_\bullet = \hat{\varepsilon}_{\mathbf{Z}} = 0$: Utilizamos a condição em 6.3 e ajustamos $\hat{\varepsilon}_\bullet = z$, onde $z$ é um número desconhecido positivo muito pequeno. Logo, $\hat{\theta}_{\mathbf{Z}}(Y) = 1$ de acordo com a definição original.

(4) $0 = \hat{\varepsilon}_\bullet < \hat{\varepsilon}_{\mathbf{Z}}$: Pela condição em 6.3, nós ajustamos $\hat{\varepsilon}_\bullet = z$, como no caso anterior, tal que $\hat{\theta}_{\mathbf{Z}}(Y) = \dfrac{z - \hat{\varepsilon}_{\mathbf{Z}}}{z}$. Agora fazemos $z \to 0$ (que poderia ter sido feito no caso anterior mas não foi necessário porque o quociente se reduz ao caso 1). Quando

$z \to 0$, $z$ vai se tornando cada vez menor do que $\hat{\varepsilon}_{\mathbf{Z}}$, e pela condição 6.2 ajustamos $\hat{\varepsilon}_{\mathbf{Z}} = z$, tal que $\hat{\theta}_{\mathbf{Z}}(Y) = 0$ para $z$ muito pequeno.

Nas subseções a seguir apresentamos os estimadores de erro estudados e implementados na pesquisa: *Holdout*, Resubstituição, *Cross-validation*, *Bootstrap* e *.632 Bootstrap*.

### 6.2.1 Holdout

Quando temos um grande número de amostras podemos separá-las em *dados de treinamento* e *dados de teste*. Um classificador é construído com os dados de treinamento e seu erro é estimado de acordo com a proporção de erros cometidos ao processar os dados de teste. Nós denotamos esse erro por $\hat{\varepsilon}_{N,m}$, onde $m$ é o número de amostras separadas para teste. Como os dados de teste são escolhidos ao acaso e de forma independente dos dados de treinamento, este é um estimador de erro aleatório.

Um problema ao utilizar dados de treinamento e teste é que na prática não temos dados o suficiente para fazermos $N$ e $m$ grandes. Desta forma, devemos utilizar o mesmo conjunto de dados para o treinamento e para o teste como no método da *resubstituição* descrito a seguir.

### 6.2.2 Resubstituição

Esta abordagem consiste em utilizar toda a amostra para construir um classificador $\psi_N$ e estimar $\varepsilon_N$ aplicando $\psi_N$ nos mesmos dados [Braga-Neto 04]. Temos então que o erro de resubstituição $\hat{\varepsilon}_N^R$ é a fração dos erros cometidos por $\psi_N$:

$$\hat{\varepsilon}_N^R = \frac{1}{N} \sum_{i=1}^{N} |y_i - \psi_N(\mathbf{z}_i)| \tag{6.4}$$

O método da resubstituição é otimista. Isto quer dizer que geralmente o erro estimado é menor do que o erro verdadeiro: $E[\hat{\varepsilon}_N^R] \leq E[\varepsilon_N]$, mas nem sempre isto acontece [Dougherty 05].

### 6.2.3 Cross-validation

No *k-fold cross validation*, a amostra $S_N$ é particionada em $k$ blocos $S_{(i)}$, para $i = 1, 2, \ldots, k$; onde assumimos que $k$ divide $N$ por simplicidade [Braga-Neto 04]. Cada bloco é deixado de fora na construção do classificador e é utilizado para teste. Estimamos o erro a partir dos erros cometidos em todos os blocos:

$$\hat{\varepsilon}_{N,k}^{CV} = \frac{1}{N}\sum_{i=1}^{k}\sum_{j=1}^{N/k} |y_j^{(i)} - \psi_{N,i}(\mathbf{z}_j^{(i)})| \tag{6.5}$$

onde $\psi_{N,i}$ é construído com o conjunto de amostras $S_N - S_{(i)}$ e cada $(\mathbf{z}_j^{(i)}, y_j^{(i)})$ é uma amostra em $S_{(i)}$. Se escolhermos os blocos aleatoriamente temos uma variação do método que chamamos de *random-fold cross validation*. De outra forma, se selecionarmos antes quais partes da amostra irão para cada bloco temos o *fixed-fold cross validation*, que é um estimador não-aleatório. O processo pode ser repetido onde vários erros são estimados e o resultado final é a média das estimativas. O estimador *k-fold cross validation* não é otimista nem pessimista como estimador de $E[\varepsilon_{N-N/k}]$, ou seja, $E[\hat{\varepsilon}_{N,k}^{CV}] = E[\varepsilon_{N-N/k}]$ [Sima 05].

Um estimador *leave-one-out* é um *N-fold cross validation*. Para cada uma das $N$ amostras deixada de fora contruímos um classificador que é aplicado nesta amostra e o erro $\hat{\varepsilon}_N^{CV}$ é $1/N$ vezes o total de erros cometidos pelos $N$ classificadores.

### 6.2.4 Bootstrap

O *bootstrap* é uma técnica de reamostragem que pode ser aplicada na estimação de erros. Ela é baseada na noção de uma *distribuição empírica* $F^*$, que serve como substituta para a distribuição original desconhecida $F$. Uma *amostra bootstrap* $S_N^*$ de $F^*$ consiste de $N$ sorteios com reposição dos dados a partir da amostra original $S_N$. Cada amostra possui a mesma probabilidade de ser sorteada. Assim, algumas amostras podem aparecer várias vezes em $S_N^*$, enquanto outras podem não aparecer. A probabilidade de uma amostra não aparecer em $S_N^*$ é $(1 - 1/N)^N \approx e^{-1}$. Assim, uma amostra bootstrap de tamanho $N$ contém na média $(1 - e^{-1})N \approx (0,632)N$ das amostras originais [Braga-Neto 04].

O estimador de erro bootstrap básico é o *estimador bootstrap zero*, definido por

$$\hat{\varepsilon}_N^{BZ} = E_{F^*}[|y - \psi_N^*(\mathbf{z})| : (\mathbf{z}, y) \in S_N - S_N^*]. \tag{6.6}$$

O classificador $\psi_N^*$ é construído com a amostra bootstrap e é testado com os dados que foram deixados de fora. Na prática aplicamos o método de Monte Carlo para obter o erro estimado, ou seja, realizamos o experimento $B$ vezes de forma independente. Assim, para $S_{N,b}^*, b = 1, 2, \ldots, B$ temos:

$$\hat{\varepsilon}_N^{BZ} = \frac{\displaystyle\sum_{b=1}^{B}\sum_{i=1}^{N} |y_i - \psi_{N,b}^*(\mathbf{z}_i)| \cdot \mathbf{1}_{[(\mathbf{z}_i, y_i) \in S_N - S_{N,b}^*]}}{\displaystyle\sum_{b=1}^{B}\sum_{i=1}^{N} \mathbf{1}_{[(\mathbf{z}_i, y_i) \in S_N - S_{N,b}^*]}}, \tag{6.7}$$

onde (a função indicadora) $\mathbf{1}_{[P]} = 1$ se a proposição $P$ é verdadeira; e $\mathbf{1}_{[P]} = 0$ se a proposição $P$ é falsa.

O estimador bootstrap zero é claramente um estimador aleatório. Com a intenção de manter a variância interna baixa, $B$ deve ser um número grande. Na literatura, $B$ entre 25 e 200 tem sido recomendado [Braga-Neto 04].

### 6.2.5 .632 Bootstrap

O estimador bootstrap zero é considerado pessimista, ou seja, o erro obtido é geralmente mais alto do que o erro verdadeiro, uma vez que o número de amostras disponíveis na construção do classificador é apenas $0,632N$ na média. O *.632 bootstrap* tenta corrigir isso aplicando pesos ao erro do estimador bootstrap zero e do estimador via resubstituição:

$$\hat{\varepsilon}_N^{B632} = (1 - 0,632)\ \hat{\varepsilon}_N^R\ +\ 0,632\ \hat{\varepsilon}_N^{BZ} \tag{6.8}$$

O estimador .632 bootstrap possui uma baixa variância, mas pode ser inadequado se o erro de resubstituição for muito otimista [Braga-Neto 04].

No próximo capítulo apresentamos os programas implementados durante a pesquisa. Entre eles, o CoD Estimator, é o programa utilizado para estimar os coeficientes de determinação. Para isso, os métodos de estimação de erro citados neste capítulo foram implementados.

# Capítulo 7

# Programas Desenvolvidos

Neste capítulo apresentamos os programas que foram desenvolvidos na pesquisa. Os programas estão implementados em linguagem C++ devida a sua portabilidade e desempenho. Os programas têm como foco principal o estudo dos coeficientes de determinação e genes IMP. Podemos citar três programas aplicados nesse estudo: CoD Estimator, CoD Cleaner e IMP Genes Finder.

O programa CoD Estimator estima valores de CoDs para um gene alvo baseado em dados de microarray binarizados. Veremos que os CoDs estimados podem apresentar valores inadequados para a busca de genes IMP, e por isso precisam ser tratados de alguma maneira. Este tratamento é feito pelo programa CoD Cleaner que será apresentado na Seção 7.2. O terceiro programa, IMP Genes Finder, faz a busca de genes IMP de acordo com os CoDs estimados por CoD Estimator e tratados pelo CoD Cleaner. Nas próximas seções explicaremos o funcionamento de cada programa.

## 7.1 CoD Estimator

O CoD Estimator é um programa desenvolvido na pesquisa para estimar os coeficientes de determinação relativos a um determinado gene alvo.

Como as amostras de microarray são limitadas e a distribuição dos dados é desconhecida, os CoDs devem ser estimados. Sendo assim, o programa fornece vários métodos para se estimar os erros da Equação 6.1 e então calcular os coeficientes. As amostras de microarray utilizadas são representadas por matrizes binárias $n \times N$, onde $n$ é o número de genes e $N$ o número de microarrays.

Os parâmetros para a execução do programa são definidos em um arquivo de configuração `config.cfg`:

- `TARGET`: Gene alvo que será levado em consideração no cálculo do erro $\hat{\varepsilon}_\bullet$ na Equação

6.1;

- `NGEN`: Número de genes no conjunto $\mathbf{Z}$ a ser considerado na Equação 6.1;
- `MODE`: Este parâmetro define o método de estimação de erro utilizado para calcular o erro $\hat{\varepsilon}_{\mathbf{Z}}$ na Equação 6.1. Os possíveis valores são:

  0 - Resubstituição

  1 - Holdout

  2 - Leave One Out

  3 - K-fold Cross Validation

  4 - Bootstrap

  5 - .632 Bootstrap

- `HOLD`: Quando o modo Holdout é selecionado, este parâmetro indica quantas amostras devem ser separadas aleatoriamente para o teste do classificador;
- `KFOLD`: Para o K-fold Cross Validation, este parâmetro defino o número $K$ de blocos (folds) a ser utilizado para separar as amostras;
- `BSIM`: Indica quantas vezes o experimento deve ser realizado no caso do Bootstrap. Como o .632 Bootstrap utiliza o erro $\hat{\varepsilon}_n^{BZ}$ este parâmetro deve ser considerado em ambos os métodos. Ver Equações 6.7 e 6.8;
- `NCODS`: Número de CoDs a serem armazenados no número de saída. Se for definido como 0 (zero) todos os $\binom{n-1}{\texttt{NGEN}}$ CoDs serão armazenados;

O entrada do programa é um arquivo texto contendo a matriz que representa as amostras de microarray. A primeira linha do arquivo indica quantas linhas (genes) a matriz possui e a segunda linha quantas colunas (amostras)[1]. Por exemplo:

```
25
7

1 0 0 0 0 1 1
0 1 1 1 1 0 0
1 1 0 0 1 0 1
0 1 1 1 0 0 0
1 1 0 0 1 0 1
0 1 0 1 0 1 1
0 1 0 1 0 1 0
1 0 1 1 0 0 1
```

[1] A enumeração dos genes e das amostras inicia-se em 0 (zero).

```
1 0 0 1 1 0 1
1 0 0 1 0 0 0
0 0 1 1 0 0 0
0 0 1 0 1 1 0
0 0 1 1 1 1 1
1 1 0 1 0 1 1
1 0 0 0 0 0 0
1 0 0 0 1 0 0
0 1 0 0 0 1 1
0 0 1 1 1 1 0
1 0 0 1 0 1 0
0 1 0 1 0 0 0
1 1 0 0 0 1 0
0 1 0 1 1 0 1
0 1 1 0 0 1 0
1 1 1 1 0 0 1
0 1 0 1 0 1 1
```

Este exemplo pode ser considerado pequeno pois na prática o número de genes excede o valor de 500 enquanto que o número de mostras não chega a ser tão grande, às vezes temos no máximo 30 microarrays disponíveis.

A saída gerada pelo programa pode ser vista abaixo. Mostramos apenas alguns valores pois para este exemplo teríamos um total de 2.024 CoDs calculados, isto porque estamos considerando conjuntos $\mathbf{Z}$ de $k = 3$ genes (`NGEN` $= 3$) para predizer o gene alvo $x_5$ (`TARGET` $= 5$). As primeiras linhas do arquivo contém informações sobre a execução do programa, como por exemplo o número de genes e amostras considerados, qual o gene alvo escolhido, o estimador de erro utilizado, entre outros. Esta informação é seguida dos valores dos CoDs ordenados decrescentemente.

```
Genes = 25
Amostras = 7

========== Configuracoes ==========
Gene alvo: 5
Modo: Bootstrap com 100 simulacoes.
Analisar todas as combinacoes de 25 genes 3 a 3.
Armazenar TODOS os 2024 CoDs no arquivo de saida.

Total CODs: 2024
> 5 3 2024
```

```
  Rank          CODs     Genes

1            0.987552   :16:19:24:
2            0.974684   :6:16:19:
3            0.974249   :6:19:24:
4            0.972727   :6:16:24:
5            0.860870   :1:18:24:
6            0.842105   :0:14:19:
7            0.840708   :0:12:18:
8            0.834821   :16:19:22:
9            0.833333   :1:6:24:
10           0.832636   :1:11:24:
11           0.832599   :6:11:24:
12           0.831169   :3:8:16:
13           0.826271   :9:13:17:
14           0.825000   :6:16:23:
15           0.822581   :13:14:17:

.
.
.

2020         0.241803   :7:10:17:
2021         0.240816   :10:12:20:
2022         0.235808   :0:4:7:
2023         0.232068   :2:10:20:
2024         0.217742   :2:10:12:
----------------------------
```

Como pode ser observado, o maior CoD é o $\hat{\theta}_{\{x_{16},x_{19},x_{24}\}}(x_5) \approx 0,98$. Isto significa que os genes $x_{16}$, $x_{19}$ e $x_{24}$ podem possuir uma forte influência na predição do gene alvo $Y = x_5$.

O programa CoD Estimator foi desenvolvido utilizando-se os conceitos de orientação a objetos suportados pela linguagem C++. Os arquivos de definição de classes são:

- `config.h`: Define a classe responsável por armazenar os parâmetros pra a execução do programa;
- `sample.h`: Define a classe que irá representar a amostra do experimento;
- `cod.h`: Define a classe que representa um CoD;
- `errorestimator.h`: Este é o arquivo contendo as classes que implementam os métodos de estimação de erro e cálculo dos CoDs.

A classe `tConfig` em `config.h` possui métodos para ler o arquivo de configuração `config.cfg`. `tConfig` é responsável por fornecer os parâmetros de configuração às outras classes quando necessário, além de fazer a validação da configuração.

No arquivo `sample.h` temos a definição da classe `tSample`. Esta classe é responsável por ler o arquivo de entrada, ou seja, as amostras a serem utilizadas no experimento. Quando o programa necessita de dados da amostra, os métodos da classe `tSample` são invocados.

Um CoD é representado por um objeto da classe `tCOD` definida em `cod.h`. Esta classe armazena o valor de um CoD e o conjunto $\mathbf{Z}$ de genes preditores correspondente.

As principais classes do programa estão definidas em `errorestimator.h`. A classe abstrata `tErrorEstimator` é definida com os atributos e métodos comuns a todos os estimadores de erro implementados ou a serem implementados no programa. As classes que implementam os métodos de estimação de erro específicos devem ser derivadas da classe `tErrorEstimator`. As classes derivadas de `tErrorEstimator` são: `tResubstitution`, `tHoldout`, `tLeaveOneOut`, `tKFoldCV`, `tBootstrap` e `t632Bootstrap`. Novas classes de estimadores de erro podem ser adicionadas em `errorestimator.h` caso exista esta necessidade. Para isso basta derivar uma nova classe a partir de `tErrorEstimator` e implementar o método de estimação de erro correspondente para calcular os CoDs.

Como vimos, a saída do programa consiste em um arquivo texto contendo informações sobre a configuração usada no cálculo dos CoDs seguida de uma lista com os CoDs ordenados decrescentemente (CoDs altos aparecem no início do arquivo). Ao lado dos valores dos coeficientes podem ser encontrados os genes preditores correspondentes.

## 7.2 CoD Cleaner

Como foi visto na Seção 4.8, uma propriedade importante dos CoDs é que dado um gene alvo fixo, o CoD aumenta na medida em que novos genes preditores são adicionados. Sendo assim, temos que $\theta_{\{x_1,\ldots,x_{k-1}\}}(Y) \leq \theta_{\{x_1,\ldots,x_k\}}(Y)$. Porém, devido aos erros de estimação podem aparecer casos em que $\hat{\theta}_{\{x_1,\ldots,x_{k-1}\}}(Y) > \hat{\theta}_{\{x_1,\ldots,x_k\}}(Y)$. Estes casos devem ser tratados para evitar conclusões incorretas.

O CoD Cleaner é um programa que foi implementado para tratar essas inconsistências nos valores dos CoDs. Como entrada, o programa recebe dois arquivos contendo os CoDs estimados. Estes arquivos são os arquivos de saída do programa CoD Estimator da seção anterior. O primeiro arquivo deve conter os CoDs relativos a um gene alvo $Y$ para conjuntos de genes $\mathbf{Z}$ de tamanho $k$. O segundo arquivo deve conter os CoDs relativos ao mesmo gene alvo $Y$ do primeiro arquivo, mas para conjuntos de genes $\mathbf{Z}'$ de tamanho $k' > k$.

O programa possui duas formas de tratamento. A primeira simplesmente descarta o coeficiente de menor valor. A segunda substitui o CoD do segundo arquivo pelo CoD do

primeiro arquivo. Neste caso teremos $\hat{\theta}_{\{x_1,\ldots,x_k\}}(Y) \leftarrow \hat{\theta}_{\{x_1,\ldots,x_{k-1}\}}(Y)$. Após o tratamento dos CoDs com o programa CoD Cleaner podemos verificar se o gene alvo em questão é um gene IMP. Para isso utilizamos o programa IMP Genes Finder, descrito na próxima seção.

## 7.3 IMP Genes Finder

Para verificar se um gene alvo $Y$ é um gene de predição intrinsicamente multivariada basta olhar os valores dos CoDs estimados. Se ocorrer algo como $\hat{\theta}_{\{x_1\}}(Y) \approx 0$, $\hat{\theta}_{\{x_2\}}(Y) \approx 0$ e $\hat{\theta}_{\{x_3\}}(Y) \approx 0$, mas $\hat{\theta}_{\{x_1,x_2,x_3\}}(Y) \approx 1$ dizemos que $Y$ é um IMP gene de grau 1. Um caso semelhante pode ocorrer, como por exemplo, se tivermos $\hat{\theta}_{\{x_1,x_2\}}(Y) \approx 0$, $\hat{\theta}_{\{x_1,x_3\}}(Y) \approx 0$ e $\hat{\theta}_{\{x_2,x_3\}}(Y) \approx 0$, mas $\hat{\theta}_{\{x_1,x_2,x_3\}}(Y) \approx 1$ dizemos que $Y$ é um IMP gene de grau 2.

O programa IMP Genes Finder faz a busca por genes de predição intrinsicamente multivariada baseada nas condições citadas no parágrafo anterior. Para isso o programa recebe como entrada dois arquivos de CoDs (referentes ao mesmo gene alvo $Y$) do mesmo formato que o programa CoD Cleaner recebe. Por exemplo, se o primeiro arquivo de entrada possui CoDs de conjuntos $\mathbf{Z}$ de $k$ genes preditores e o segundo arquivo possui CoDs onde o tamanho dos conjuntos de genes preditores é $k'$, onde $k' > k$, então o programa IMP Genes Finder irá encontrar os conjuntos de genes preditores que caracterizam o gene alvo $Y$ como um gene IMP de grau $k$.

Também fazem parte da entrada dois parâmetros $\lambda$ e $\delta$, pois o IMP Genes Finder irá buscar IMP genes satisfazendo condições como, por exemplo, $\theta_{\{x_1\}}(Y) \leq \lambda$, $\theta_{\{x_2\}}(Y) \leq \lambda$ e $\theta_{\{x_3\}}(Y) \leq \lambda$, mas $\theta_{\{x_1,x_2,x_3\}}(Y) \geq \delta$. Sendo assim devemos ter $0 \leq \lambda < \delta \leq 1$. Tipicamente, $\lambda \leq 0,2$ e $\delta \geq 0,8$.

A saída do programa é um arquivo texto contendo os CoDs e os genes preditores caso o gene alvo em questão seja, de fato, um gene IMP. Por exemplo, se um dado gene alvo $Y = x_5$ for um gene IMP de grau 2 teremos na saída um arquivo texto semelhante a este:

```
21
0.2 0.9

0.00   :2::8:
0.00   :2::13:
0.00   :8::13:
0.97   :2:8:13:

0.00   :3::8:
0.00   :3::11:
0.00   :8::11:
0.95   :3:8:11:
```

```
.
.
.

0.10    :20::23:
0.00    :20::24:
0.13    :23::24:
0.91    :20:23:24:
```

No exemplo acima, a primeira entrada indica que foram encontradas 21 configurações de CoDs que caracterizam o gene alvo $Y = x_5$ como gene IMP. Em seguida são listados os valores de $\lambda = 0,2$ e $\delta = 0,9$ utilizados na busca. Por último temos a lista das 21 configurações de CoDs encontradas (no exemplo acima a lista está resumida). Na primeira configuração temos $\hat{\theta}_{\{x_2,x_8\}}(x_5) = 0,00$; $\hat{\theta}_{\{x_2,x_{13}\}}(x_5) = 0,00$; $\hat{\theta}_{\{x_8,x_{13}\}}(x_5) = 0,00$ e $\hat{\theta}_{\{x_2,x_8,x_{13}\}}(x_5) = 0,97$. É fácil notar que quanto maior o valor de $\lambda$ e menor o valor de $\delta$, mais configurações de CoDs irão aparecer na lista.

No próximo capítulo iremos apresentar um caso prático em que os programas desenvolvidos foram utilizados na busca de genes importantes em um processo biológico envolvendo células tumorigênicas de camundongo.

# Capítulo 8

# Busca por genes envolvidos na morte de células tumorigênicas disparada por FGF2

Este capítulo tem a finalidade de apresentar parte de um trabalho realizado em conjunto com o Prof. Dr. Hugo Aguirre Armelin do Instituto de Química da USP e seus alunos Fábio Nakano (BIOINFO-USP) e Paula Fontes Asprino (IQ-USP). Apresentaremos o problema biológico proposto, e qual foi a nossa contribuição até o momento para com a pesquisa. Como resultado positivo desta parceria, um pôster foi aceito e apresentado no ISMB 2006[1].

A proteína-alvo no processo de interesse é o FGF2 (*Fibroblast Growth Factor 2*). A proteína codificada pelo gene FGF2 é membro da família FGF - Fator de crescimento de fibroblastos. *Fibroblast Growth Factors* (FGFs) e suas vias de sinalização parecem ter um papel importante não apenas no desenvolvimento normal, mas também na progressão e desenvolvimento de tumores [Powers 00]. Esta proteína tem sido estudada em diversos processos biológicos como desenvolvimento do sistema nervoso, cicatrização de feridas e crescimento de tumores. A célula utilizada como modelo biológico é a Y1 - célula tumoral de adrenocortical de camundongo [Yasumura 66], que são altamente responsivas ao FGF2 [Forti 06].

A pesquisa do Prof. Dr. Hugo Armelin é sobre ciclo celular e fatores peptídicos e utiliza a célula Y1 como modelo. Sabe-se que o fator de crescimento FGF2 leva a célula a morte, independente da atividade mitogênica bem conhecida do FGF2 [Costa 04]. A descoberta de genes importantes envolvidos neste processo pode auxiliar o desenvolvimento de terapias e de novas drogas no tratamento de tumores.

[1] 14th Annual International Conference On Intelligent Systems For Molecular Biology, Fortaleza - Brasil, de 06 a 10 de agosto de 2006.

## 8.1 Dados e Trabalho realizado

Nesta pesquisa de células Y1 e FGF2 utilizamos dados de microarray. Estes experimentos de microarray foram obtidos durante a pesquisa de doutorado da Drª Paula F. Asprino [Asprino 05] que envolve a análise de expressões gênicas de células adrenais tumorigênicas a partir de microarrays.

No total trabalhamos com uma lista de 115 genes e 23 amostras de microarray. Temos duas representações possíveis para os dados. A primeira consiste da representação da expressão gênica de cada gene. Uma matriz $A_{115\times23}$ contendo os valores das expressões foi utilizada para fazer os cálculos dos CoDs. Na segunda representação temos a probabilidade de indução de cada gene. Neste caso, uma matriz $P_{115\times23}$ com as probabilidades foi usada.

Como os dados de microarray não são de natureza binária, o primeiro passo foi aplicar o algoritmo de binarização descrito na Seção 3.1 para os dados que representam expressões gênicas, obtendo-se assim uma matriz $A'_{115\times123}$. Para binarizar os dados baseados em probabilidades de indução foi utilizado um *threshold* $T = 0,5$. Sendo assim obtemos a matriz $P'_{115\times23}$ da seguinte forma:

$$P'_{i,j} = \begin{cases} 1 & \text{se } P_{i,j} \geq 0,5 \\ 0 & \text{caso contrário} \end{cases} \tag{8.1}$$

Após a binarização, duas linhas foram adicionadas em cada uma das matrizes, aumentando suas dimensões para $117 \times 23$. A primeira linha adicionada corresponde ao fenótipo de vida/morte da célula Y1. Seja $i$ o índice da linha adicionada. Se a célula apresenta um fenótipo de morte na amostra $j$, $A'_{ij}$ e $P'_{ij}$ recebem o valor 0 (zero) indicando a morte. Caso contrário temos $A'_{ij} = 1$ e $P'_{ij} = 1$, indicando sobrevivência da célula. Analogamente, a segunda linha foi adicionada para indicar a presença de FGF2 na amostra, onde o valor 1 indica a presença de FGF2 e 0 a ausência de FGF2.

Com os dados de microarray binarizados é possível calcular os coeficientes de determinação através de métodos de estimação de erro apresentados no Capítulo 6. O propósito principal é analisar os dados para encontrar genes envolvidos no processo de morte por FGF2. Para isso, todos os 115 genes foram tomados como gene alvo para o cálculo dos CoDs. Além disso, o fenótipo de vida/morte da célula também foi tomado como "gene" alvo, aqui chamado de fenótipo alvo.

Os CoDs foram calculados, ou melhor, estimados, usando $k$ genes preditores para cada gene alvo, onde $1 \leq k \leq 3$. Dessa forma, determinamos quais conjuntos de $k$ genes estão mais relacionados com cada gene alvo e também com o fenótipo de vida/morte da célula. O método utilizado para estimar os CoDs foi o *Bootstrap*, pois apresenta uma variância baixa quando aplicado juntamente com o método de Monte Carlo, apesar de ser mais lento [Braga-Neto 04][Dougherty 05]. Com os CoDs estimados podemos então realizar a busca por genes/fenótipo IMP utilizando os programas implementados que foram descritos no Capítulo 7.

## 8.2 Resultados

Como parte da contribuição no trabalho que foi publicado em um pôster no ISMB 2006, os coeficientes de determinação foram utilizados para se obter uma rede de CoDs modelada posteriormente como uma rede Bayesiana composta por genes que supostamente estariam envolvidos no fenômeno de morte de células Y1 por FGF2. O título do pôster é *Searching for Genes Involved in Novel Mechanisms of Tumor Cell Death Triggered by FGF2: Introducing New Methodological Approaches* e envolve uma continuação de um trabalho apresentado no X-Meeting 2005 [Nakano 05]. Nesta subseção apresentamos uma nova metodologia, proposta pelo Fábio Nakano, para encontrar sub-redes de genes que estão envolvidos no processo de morte por FGF2.

A rede de CoDs foi obtida tomando-se cada gene como alvo, inclusive o fenótipo vida/morte. Além disso, um conjunto $\mathbf{Z}$ de $k = 2$ genes preditores foi utilizado nesta etapa. Como trabalhamos com 115 genes, mais o fenótipo de vida/morte e o *fator FGF* (presença de FGF2 na amostra, como foi descrito na seção anterior), temos um total de $n = 117$ genes. Para cada gene alvo $Y$ o maior CoD $\hat{\theta}_{\mathbf{Z}}(Y)$ foi utilizado para compor a rede. Por exemplo, seja $\mathbf{Z} = \{x_i, x_j\}$ tal que $\hat{\theta}_{\mathbf{Z}}(Y)$ seja o maior CoD para o gene alvo $Y$. Esta relação entre $\mathbf{Z}$ e $Y$ é representada na rede de CoDs como na Figura 8.1.

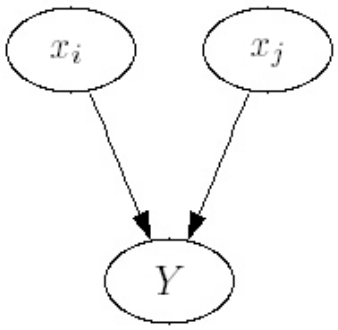


Figura 8.1: Relação entre $\mathbf{Z} = \{x_i, x_j\}$ e $Y$ na rede de CoDs.

A partir da rede de CoDs completa, isto é, após considerar cada gene como alvo e seu respectivo conjunto de genes preditores cujo CoD tenha o maior valor, sub-redes de genes foram identificadas com a intenção de relacionar o fator FGF com o fenótipo de vida/morte da célula.

Quando analisamos o gene Arhgap5 como alvo, notamos que FGF_factor (fator FGF) e Pdgfra (*Platelet-derived Growth Factor Receptor* $\alpha$) são os predidores cujo CoD é máximo para Arhgap5. Além disso, Arhgap5 é um gene IMP pois temos a seguinte configuração de CoDs:

- $\hat{\theta}_{\{\text{FGF\_factor}\}}(\text{Arhgap5}) = 0,00$;
- $\hat{\theta}_{\{\text{Pdgfra}\}}(\text{Arhgap5}) = 0,00$ e

- $\hat{\theta}_{\{\text{FGF\_factor, Pdgfra}\}}(\text{Arhgap5}) = 0,98.$

Os CoDs acima indicam que a expressão do gene Arhgap5 está de alguma forma associada ao estímulo com FGF2 e ao Pdgfra simultaneamente, mas FGF2 e Pdgfra não afetam o estado de Arhgap5 quando agem individualmente.

O gene Pdgfra é um gene preditor quando tomamos Arhgap5 como alvo mas também é considerado um gene IMP. Vários genes preditores aparecem quando consideramos Pdgfra como gene alvo: RhoA, Akt, Itga7, Smad3, entre outros.

Outra possibilidade é tomar o fenótipo de vida/morte como alvo. Neste caso os genes Itga7 e Akt formam o melhor conjunto de genes preditores. Seguindo esta metodologia, a sub-rede da Figura 8.2 foi obtida, onde as interações entre os genes nos levam a hipótese de que os genes presentes na sub-rede estão relacionados com o fenômeno de morte de células Y1 disparadas por FGF2.

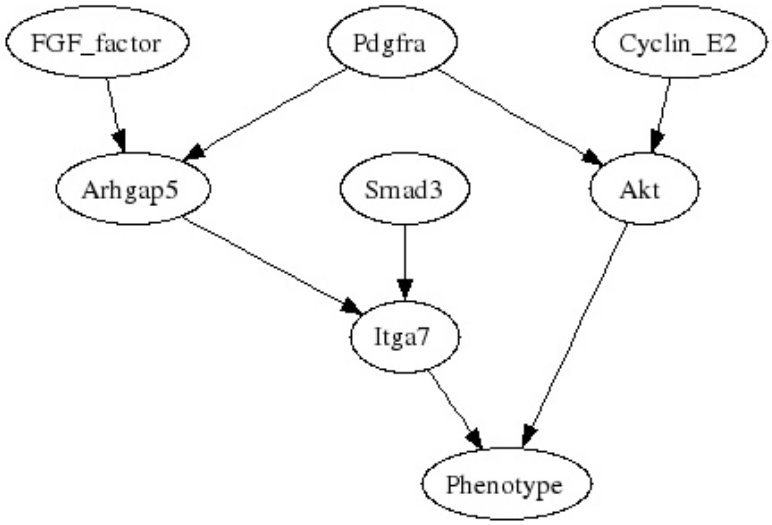


Figura 8.2: Sub-rede de CoDs.

Uma hipótese gerada a partir da análise da sub-rede da Figura 8.2 é que a proteína produzida pelo gene Arhgap5 está altamente relacionada com o fenótipo de vida/morte em células Y1 tratadas com FGF2. Arhgap5 é uma GTPase conhecida como Rho-GAP, que têm a capacidade de inibir proteínas da família RAS, como por exemplo, RhoA. Como foi apresentado no pôster, é conhecido que a célula sobrevive se RhoA é desativada por uma proteína dominante negativa [Forti]. Esta hipótese de que Arhgap5 inibe RhoA está confirmada em [Burridge 04], validando a nossa metodologia na busca de genes envolvidos em processos biológicos.

De maneira similar, outras sub-redes podem ser obtidas. Por exemplo, podemos utilizar conjuntos $\mathbf{Z}$ de tamanho $k = 3$ genes preditores. Também podemos gerar uma sub-rede de CoDs de acordo com os maiores valores de CoDs para cada gene alvo, e não apenas com o maior CoD.

# Capítulo 9

# Conclusão

O trabalho apresentado nesta dissertação de mestrado consiste de uma abordagem para a busca de genes importantes em processos biológicos. Esta busca tem início na análise de expressões gênicas realizada através de experimentos de microarray, onde são obtidos os dados utilizados em nossa abordagem. A contribuição principal do trabalho consiste da modelagem dos dados utilizando os conceitos de redes Booleanas com perturbação, cadeias de Markov e coeficientes de determinação. Esta modelagem resulta na metodologia para a busca de genes IMP.

Os objetivos iniciais da pesquisa foram cumpridos e novas pesquisas podem surgir a partir dos resultados obtidos, como será mencionado na próxima seção. Estudamos modelos que representam a interação entre genes dentro de um sistema biológico. Em particular, estudamos o modelo de redes Booleanas com perturbação. Este modelo foi estudado dentro do contexto de cadeias de Markov ergódicas, e foi usado na modelagem de dados de expressão gênica.

Também foram estudados métodos de estimação de erro para o cálculo dos CoDs e os algoritmos para a busca de genes IMP foram implementados. Os programas desenvolvidos foram utilizados em uma pesquisa cujo tema era a busca por genes importantes no processo de morte de células tumorigênicas disparada por FGF2. Os resultados apresentados na Seção 8.2 sugerem que as hipóteses geradas na pesquisa sejam verificadas experimentalmente.

Com os objetivos iniciais da pesquisa cumpridos, concluímos que a contribuição deste trabalho consiste de uma nova metodologia para a busca de genes importantes em processos biológicos de interesse científico. Tal metodologia envolve temas que vão desde a análise binária de expressões gênicas até o levantamento de novas hipóteses a respeito do processo biológico estudado.

## 9.1 Trabalhos futuros

Esta pesquisa teve como resultado importante a abertura de novos caminhos para a realização de trabalhos relacionados ao tema. Uma próxima etapa seria, talvez, a verificação experimental de hipóteses que surgiram durante a pesquisa. Por exemplo, verificar se os genes citados na seção anterior estão relacionados, de fato, com a morte de células Y1 por FGF2.

Caso as verificações resultem em descobertas significantes, cabe a nós o trabalho de publicar um artigo científico juntamente com todos os colaboradores envolvidos na pesquisa.

A principal contribuição deste trabalho é a nova metodologia para a busca de genes importantes em determinados processos biológicos, não necessariamente o de morte de células adrenocorticais de camundongo por FGF2, mas qualquer outro processo biológico de interesse científico, desde que seja possível modelar o problema proposto no contexto de redes Booleanas, coeficiente de determinação e genes IMP.

# Bibliografia


[Arkin 98] A. Arkin, J. Ross and H. H. McAdams; *Stochastic Kinetic Analysis of Developmental Pathway Bifurcation in Phage-Infected Escherichia Coli Cells*, Genetics 149, 1633-1648, 1998.

[Asprino 05] P. F. Asprino; *Análise de expressão gênica por microarrays de cDNA em linhagens de células adrenais tumorigênicas tratadas com FGF2 e ACTH*, Tese de Doutorado, IQ-USP, 2005.

[Basharin 04] G. P. Basharin, A. N. Langville, V. A. Naumov; *The life and work of A.A. Markov*, Linear Algebra and its Applications 386, 3-26, 2004.

[Bhalla 99] U. S. Bhalla and R. Iyengar; *Emergent Properties of Networks of Biological Signalising Pathways*, Science 283, 381-387, 1999.

[Bradley 03] C. Bradley; *You want Ketchup with your DNA Chips? An Overview of Expression Microarrays*, BioTeach Journal, Vol. 1, 89-94, 2003.

[Braga-Neto 04] U. Braga-Neto, E. R. Dougherty; *Bolstered Error Estimator*, Pattern Recognition 37, 1267-1281, 2004.

[Brun 05] M. Brun, E. R. Dougherty; *Steady-state probabilities for attractors in probabilistic Boolean networks*, Signal Processing, Vol. 85, 1993-2013, 2005.

[Burridge 04] K. Burridge, K. Wennerberg; *Rho and Rac Take Center Stage*, Cell, Vol. 116, 167-179, 2004.

[Costa 04] E. T. Costa, F. L. Forti, K. M. Rocha, M. S. Moraes, H. A. Armelin; *Molecular mechanisms of cell cycle control in the mouse Y1 adrenal cell line*, Endocr Res. 2004 Nov; 30(4):503-509.

[D'Haeseller 99] P. D'Haeseller, X. Wen and R. Somogyi; *Linear Modeling of mRNA Expression Levels During CNS Development and Injury*, Pacific Symposium on Biocomputing, Vol. 4, 41-52, 1999.

[Dougherty 00] E. R. Dougherty, S. Kim, Y. Chen; *Coefficient of determination in nonlinear signal processing*, Signal Processing 80, 2219-2235, 2000.

[Dougherty 05] E. R. Dougherty, I. Shmulevich, J. Chen, Z. J. Wang; *Genomic Signal Processing and Statistics*, EURASIP Book Series on SP&C, Volume 2, 2005.

[Forti 06] F. L. Forti, E. T. Costa, K. M. Rocha, M. S. Moraes, H. A. Armelin; *c-Ki-ras oncogene amplification and FGF2 signaling pathways in the mouse Y1 adrenocortical cell line*, Annals of the Brazilian Academy of Sciences, 78(2): 231-239, 2006.

[Forti] F. L. Forti, H. A. Armelin, Unpublished Experimental Results.

[Friedman 00] N. Friedman, M. Linial, I. Nachmam and D. Pe'er; *Using Bayesian Network to Analyse Expression Data*, Journal of Computational Biology 7, 601-620, 2000.

[Gartel 99] A. L. Gartel, A. L. Tyner; *Transcriptional regulation of the p21(WAF1/CIP1) gene*, Exp. Cell Res., Vol. 246, no. 2, 280-289, 1999.

[Grant 04] R. P. Grant; *Computational Genomics: Theory and Application*, Horizon Bioscience, 2004.

[Grinstead 97] C. M. Grinstead and J. L. Snell; *Introduction to Probability*, American Mathematical Society, 2 ed., 1997.

[Häggström 02] O. Häggström; *Finite Markov Chains and Algorithmic Applications*, Cambridge University Press, 2002.

[Hashimoto 03] R. F. Hashimoto, E. R. Dougherty, M. Brun, Z. Zhou, M. L. Bittner, J. M. Trent; *Efficient selection of feature sets possessing high coefficients of determination based on incremental determinations*, Signal Processing 83, 695-712, 2003.

[Ivanov 06] I. Ivanov, E. R. Dougherty; *Modeling Genetic Regulatory Networks: Continuous or Discrete?*, Journal of Biological Systems, Vol. 14, No. 2, 219-229, 2006.

[Jong 02] H. D. Jong; *Modeling and Simulation of Genetic Regulatory Systems: A Literature Review*, Journal of Computational Biology, 9(1): 67-103, 2002.

[Kauffman 69] S. A. Kauffman; *Metabolic stability and epigenesis in randomly constructed genetic nets*, J. Theoret. Biol., Vol. 22, 437-69, 1969.

[Kauffman 93] S. A. Kauffman. *The Origins of Order. Self-Organization and Selection in Evolution.* Oxford University Press, 1993.

[Kijima 97] M. Kijima; *Markov Process For Stochastic Modeling*, Cambridge University Press, 1997.

[Mestl 95] T. Mestl, E. Plahte and S. W. Omholt; *A Mathematical Framework for Describing and Analysing Gene Regulatory Networks*, Journal of Theoretical Biology, 176(2): 291-300, 1995.

[Nakano 05] F. Nakano, P. F. Asprino, E. T. Costa, C. A. B. Pereira, H. A. Armelin; *Searching for Genes Involved in Novel Mechanisms of Tumor Cell Death Triggered by FGF2*, In Proceedings of X-Meeting First International Conference of the AB3C, October, 2005, p. 126.

[Pal 05] R. Pal, A. Datta, M. L. Bittner, E. R. Dougherty; *Generating Boolean Networks with a Prescribed Attractor Structure*. Bioinformatics 21(21), 4021-4025, 2005.

[Pearson 06] H. Pearson; *What Is A Gene?*, Nature Vol. 441, 398-401, 2006.

[Powers 00] C. J. Powers, S. W. McLeskey, A. Wellstein; *Fibroblast growth factors, their receptors and signaling*, Endocrine-Related Cancer 7, 165-197, 2000.

[Quackenbush 02] J. Quackenbush; *Microarray data normalization and transformation*, Nature Genetics, 32, 496-501, 2002.

[Shmulevich 02a] I. Shmulevich, E. R. Dougherty, S. Kim, W. Zhang; *Probabilistic Boolean Networks: A rule-based uncertainty model for gene regulatory networks*, Bioinformatics Vol. 18, no. 2, 261-274, 2002.

[Shmulevich 02b] I. Shmulevich and W. Zhang; *Binary analysis and optimization-based normalization of gene expression data*, Bioinformatics Vol. 18, no. 4, 555-565, 2002.

[Shmulevich 02c] I. Shmulevich, E. R. Dougherty and W. Zhang; *Gene Perturbation and intervention in probabilistic Boolean networks*, Bioinformatics Vol. 18, no. 10, 1319-1331, 2002.

[Shmulevich 02d] I. Shmulevich, E. R. Dougherty and W. Zhang; *From Boolean to Probabilistic Boolean Networks as Models of Genetic Regulatory Networks*, Proceedings of the IEEE, Vol. 90, no. 11, 1777-1792, 2002.

[Sima 05] C. Sima, U. Braga-Neto, E. R. Dougherty; *Superior feature-set ranking for small samples using bolstered error estimator*, Bioinformatics, Vol. 21, no. 7, 1046-1054, 2005.

[Wuensche 98] A. Wuensche; *Genomic Regulation Modeled as a Network with Basins of Attraction*, Pacific Symposium on Biocomputing '98 eds. R.B.Altman, A.K.Dunker, L.Hunter, T.E.Klien. World Scientific, Singapore, 1998.

[Yang 02] Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai and T. P. Speed; *Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation*, Nucleic Acids Research, Vol. 30, No. 4, ed. 15, 2002.

[Yasumura 66] Y. Yasumura, Y. Buonassissi, G. Sato; *Clonal analysis of differentiated function in animal cell cultures. I. Possible correlated maintenance of differentiated function and the diploid karyotype*, Cancer Research 26, 529-535, 1966.

[Zhou 03a] X. Zhou, X. Wang and E. R. Dougherty; *Binarization of Microarray Data on the Basis of a Mixture Model*, Molecular Cancer Terapeutics, Vol. 2, 679-684, 2003.

[Zhou 03b] X. Zhou, X. Wang, E. R. Dougherty; *Construction of genomic networks using mutual-information clustering and reversible-jump Markov-chain-Monte-Carlo predictor design*, Signal Processing 83, 745-761, 2003.