# Uma técnica automática
## baseada em morfologia matemática
## para medida de sinal
## em imagens de cDNA

**Daniel Oliveira Dantas**

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO

PARA OBTENÇÃO DO GRAU DE MESTRE
EM
CIÊNCIA DA COMPUTAÇÃO

Área de Concentração : **Ciência da Computação**
Orientador : **Prof. Dr. Junior Barrera**

– São Paulo, janeiro de 2004 –

*aos meus pais Rubem e Eliana*

*e à minha tia Silvia*

# Agradecimentos

Agradeço a Deus por me criar e a Jesus por me salvar. A toda minha família por estar sempre presente. Aos meus professores, colegas e amigos do CCPA, que é muito mais que um colégio, por seu companheirismo na conquista do Vestibular. Aos professores do IME por seu profissionalismo e competência. Aos colegas e amigos do BIOINFO e do Instituto Ludwig, que me apoiaram durante o programa de mestrado.

Em particular,

Aos meus pais Rubem e Eliana, por me darem à luz e pela melhor educação e instrução possível, e à minha tia Silvia, que teve grande influência na minha educação, e que me inspirou a querer estudar na USP.

Aos meus primos e primas, Ana Luiza, Anizio, Carol, Diogo, Fabiane, Fábio, Fernanda, Fernando, Ivana, Iza, Julia, Lucas, Marta Carine, Paula, Thiago etc pela amizade incondicional durante toda minha vida. Aos meus padrinhos Ana Luzia e Rivaldo e a todos os outros tios, tias, e avós pelos bons conselhos e amizade.

Aos amigos e colegas do colégio, Anelar Nunes de Carvalho Filho, Gilberto Monte Lima, Alexandre Dantas Pereira, André Gustavo Andrade Monteiro, Daniel Moura de Figueiredo, Décio Fragata da Silva, Jefferson Andrade Almeida, Rodolpho Rabelo Pedrosa Costa etc, que me ajudaram a tornar a luta por uma vaga na universidade uma coisa leve e divertida, e que me ensinaram a ver o companheiro não como um concorrente, mas como um membro de uma mesma equipe.

À Universidade de Soka e à CCINT-USP pela oportunidade de realizar meu sonho de fazer um intercâmbio no exerior.

Aos amigos Edward Autoexec Iamamoto, Ailton A. de Oliveira, Cléber da C. Oliveira, Henry K. Oyagawa, Pratip Roy Chowdhury e Silvio R. de Faria Jr., com participação especial de Edson, que me ajudaram a derrotar Masaishi.

Ao professor Adilson Simonis por todas as oportunidades, e à Luciana Delfini de Campos, que, mais do que uma companheira de trabalho, acabou se tornando uma grande amiga.

Aos membros do BIOINFO, Eduardo Jordão Neves, João Eduardo Ferreira, Marcel Brun, Nina Sumiko Tomita Hirata, Roberto Hirata, Roberto Marcondes Cesar Junior, Nestor Walter Trepode etc, pelas discussões, sugestões e apoio que tornaram possível a realização deste trabalho.

Às pesquisadoras Beatriz Simonsen Stolf e Helena Paula Brentani do Instituto Ludwig, e Maria Aparecida Nagai da FM-USP, pelas imagens que aparecem neste trabalho.

Aos pesquisadores Hernando A. del Portillo do ICB-USP e Luiz Fernando Lima Reis do Institudo Ludwig pelas imagens, sugestões e participação do decorrer do meu mestrado.

Ao pesquisador Gustavo Henrique Esteves pela participação e colaboração direta com imagens, sugestões, dados e idéias que foram de grande valor na elaboração desta dissertação.

E finalmente ao meu professor e orientador Junior Barrera, que há muitos anos me adotou como parte da família BIOINFO, antigo Laboratório de Processamento de Imagens, e que teve papel decisivo na elaboração deste trabalho.

A todos vocês, muito obrigado!

# Resumo

O objetivo deste trabalho é apresentar uma *técnica automática baseada em morfologia matemática para medida de sinal em imagens de cDNA* desenvolvida no BIOINFO, em parceria com o Instituto Ludwig de Pesquisa contra o Câncer.

A tecnologia de lâminas de cDNA é um processo baseado em hibridização que possibilita observar a concentração relativa de mRNA de amostras de tecidos analisando a luminosidade de sinais fluorescentes ou radioativos. Hibridização é o processo bioquímico onde duas fitas de ácido nucleico com seqüências complementares se combinam.

A técnica apresentada permite o cálculo da expressão gênica com alto grau de automação, podendo o usuário corrigir com facilidade eventuais erros de segmentação. O usuário interage com o programa apenas para selecionar as imagens e inserir os dados de geometria da lâmina.

A estratégia de solução usada tem três fases: gradeamento dos blocos, gradeamento dos *spots* e segmentação dos *spots*. Todas as fases utilizam filtros morfológicos e as fases de gradeamento possuem um passo final de correção baseado nos dados de geometria da lâmina o que aumenta a robustez do processo, que funciona bem mesmo em imagens ruidosas.

# Abstract

The objective of this work is to present the *automated technique for measuring signal from cDNA images* developed in BIOINFO, associated with the Ludwig Institute for Cancer Research.

Microarray technology is a hybridization based process that makes possible to quantify the relative abundance of mRNA in two tissue samples analysing the luminosity of fluorescent or radioactive signals. Hybridization is a biochemical process where a strand of nucleic acid matches up its counterpart.

The developed technique permits the calculation of gene expression with a high level of automation. Besides that, the user can easily correct eventual segmentation mistakes. The user interacts with the program only to select the images and to set the slide geometry parameters.

The solution strategy has three main steps: subarray griding, spots gridding and spots detection. All the steps use morphological filters, and the two gridding steps have a final correction substep based on the slide geometry, increasing the process robustness, that works well even in noisy images.

# Sumário

# Lista de Figuras

# Lista de Tabelas

# Capítulo 1

# Introdução

O objetivo da tecnologia de lâminas de cDNA é medir o nível de expressão dos genes. Baseia-se no fato de que quando um fragmento de RNA encontra outro de cDNA com seqüência complementar, ambos tendem a se unir. Um experimento é feito em três etapas: primeiro fragmentos de cDNA de diversos genes são dispostos sobre algum substrato, cada gene em um pequeno círculo; a segunda etapa consiste em extrair RNA dos tecidos a serem analisados e marcá-lo com corantes fluorescentes. Finalmente a lâmina é mergulhada na sopa de RNA, cujos framentos se ligarão ao substrato nas regiões que contêm o cDNA correspondente. Quanto mais RNA de um certo gene na sopa, mais os pontos com genes correspondentes fluorescerá. Devido à dificuldade de se obter os níveis absolutos de expressão [3], os experimentos costumam ser feitos usando-se duas amostras de RNA, e o que se calcula são suas expressões relativas.

Existem diversos tipos de substrato, corantes e técnicas de extração de RNA. A síntese de cDNA pode ser feita por transcrição reversa ou pela tecnologia Affymetrix® GeneChip® [4, 5]. Nesse trabalho, ao nos referirmos a lâminas de cDNA com genes impressos em pequenos círculos dispostos de forma matricial, usaremos a palavra *microarray*.

Para analisar computacionalmente o experimento bioquímico, a lâmina de *microarray* é digitalizada por um *scanner* de alta resolução gerando uma imagem para cada canal. Cada *pixel* da imagem é a média do sinal luminoso emitido por moléculas marcadas com corante que aderiram à lâmina numa determinada área. A esse valor são adicionados ruídos estocásticos provenien-

tes, tanto do experimento bioquímico, quanto do procedimento de digitalização, amplificação eletrônica etc. Nas câmaras que funcionam com luz não polarizada, pontos com alta intensidade influenciam pontos vizinhos causando um espalhamento do sinal e gerando imagens sem muito contraste.

Além do ruído, outra dificuldade encontrada é que as moléculas da amostra, os *targets*, numa proporção pequena mas não desprezível, aderem à lâmina sem encontrar moléculas de seqüência complementar, os *probes* correspondentes (seguindo a definição *probe* e *target* adotada no suplemento da Nature Genetics [6]), e até mesmo no próprio substrato. Normalmente se subtrai a intensidade encontrada nas regiões sem *probe*, o chamado *background*, da intensidade do sinal como forma de eliminar a hibridização não específica [2], ou seja, a contribuição que não é devida à hibridização do mRNA da amostra com o DNA da lâmina. No entanto experimentos com controles negativos mostram *spots* com intensidade inferior ao *background*, dando evidências de que a hibridização não específica não deveria ser medida pelo valor do *background*, e sim pelo valor dos controles negativos [7].

As moléculas marcadas tendem a hibridizar com as moléculas da lâmina com seqüência complementar, assim, os *spots* da lâmina que contêm genes mais expressos na amostra marcada com o corante cy3 devem aparecer na imagem como círculos verdes intensos; caso contenham genes mais expressos na amostra com cy5, parecerão vermelhos; se a expressão for a mesma, devem parecer amarelos.

Para fazer a medida, analisa-se a imagem da lâmina usando algum *software* que separe o sinal, no interior dos *spots*, do ruído fora dos mesmos, chamado *background*. Separados os *pixels* do sinal, estima-se, para cada *spot*, a razão entre seus dois canais. Além disso, o *software* deve ser capaz de identificar qual a posição do *spot* no respectivo *subarray* ou bloco, e a posição do bloco na lâmina por meio de índices que relacionem o *spot* com os dados de uma tabela de genes usados no experimento.

A tarefa de segmentação de imagens de *microarray* envolve vários problemas que dificultam sua automatização. As imagens são ruidosas, os *spots* e *subarrays*, também chamados blocos, são espaçados irregularmente e os *spots* apresentam formas de crescente e de rosca, além do formato esperado, aproximadamente circular.

Cada *software* apresenta uma abordagem do problema, mas muitas vezes o usuário precisa interagir com a imagem por algumas horas até obter uma segmentação de qualidade que gere bons dados. Geralmente o usuário escolhe alguns *spots* nos extremos da imagem e o *software* os interpola assumindo que os demais blocos e *spots* são uniformemente distribuídos. Porém, a distribuição dos *spots* quase nunca é perfeitamente regular, e, para obter bons dados de expressão, o usuário deve ajustar a grade manualmente para os *spots* desalinhados.

Alguns programas podem opcionalmente ajustar a grade para *spots* desalinhados, tentando encontrar uma posição melhor, próxima à encontrada pela interpolação. Mas isso geralmente não funciona e é facilmente afetado pelo ruído.

Além disso, os programas que assumem que *spots* têm o formato perfeitamente circular podem gerar dados de má qualidade, já que, nesses casos, serão identificados *pixels* do *background* como sendo parte do sinal.

O objetivo do nosso trabalho é automatizar tal tarefa, tornando os dados de segmentação reprodutíveis, uma vez que, na segmentação manual, mesmo que o operador do *software* e a imagem sejam os mesmos, o resultado de duas segmentações deverá ser diferente. Além disso, queremos garantir a precisão dos resultados, encontrando corretamente a posição dos *spots* sem que haja a necessidade de interferência do usuário. Um método totalmente automático também economiza mão-de-obra e oferece maior rapidez no processamento das imagens.

Dividimos a segmentação em: gradeamento dos blocos, gradeamento dos *spots* e segmentação dos *spots*.

O gradeamento dos blocos e dos *spots* é importante para fazer o chamado endereçamento, que é associar a cada *spot* um índice, para que se saiba qual gene foi nele colocado, e sua expressão, que é calculada no final do processo.

Para realizar todo o processo, a informação necessária são nove valores numéricos referentes à geometria da lâmina: o número de linhas de blocos, número de colunas de blocos, distância horizontal entre blocos, distância vertical entre blocos, número de linhas e colunas de *spots* por bloco, distância vertical e horizontal entre os centros dos *spots*, e diâmetro do *spot*.

## 1.1 Contribuições desta dissertação

O objetivo da ferramenta proposta nesse trabalho é gerar a saída de forma automática como o UCSF Spot, permitindo, porém, que o usuário corrija com facilidade eventuais erros de segmentação. Recebe como entrada os dados da geometria da lâmina, ou seja, número de blocos, de *spots* em cada bloco na vertical e na horizontal, respectivas distâncias e diâmetro do *spot*.

Como o processo é completamente automático, a saída sempre será a mesma, independente do usuário.

Recentemente, um método similar ao nosso desenvolvido independentemente foi publicado na *Bioinformatics* [8]. Sua técnica de segmentação dos blocos é um pouco diferente da nossa: enquanto analisamos os perfis horizontal e vertical, sua abordagem analisa a própria imagem reduzida em algumas vezes. Ambas as técnicas usam *watershed* para segmentar os *spots*.

## 1.2 Organização do trabalho

Neste primeiro capítulo foram apresentados os objetivos do trabalho, assim como as motivações para a criação de um método completamente automático de segmentação de imagens de cDNA. No Capítulo 2 serão apresentados os fundamentos da expressão gênica e como a tecnologia de *microarrays* é usada para medi-la. Também são apresentadas de forma sucinta algumas das soluções disponíveis no mercado. No Capítulo 3 é apresentado em detalhes o algoritmo de segmentação desenvolvido, bem como uma introdução à morfologia matemática, necessária para o seu entendimento. No Capítulo 4, modelos de expressão que mostram a relação entre seu valor real e o valor observado na lâmina são apresentados. Também são mostrados os métodos de estimação do sinal mais usados e a influência do *background* no sinal é discutida. No Capítulo 5 há uma descrição detalhada do programa desenvolvido, sua interface com o usuário, dados de entrada e o formato do arquivo de saída. No Capítulo 6 são comparados os diversos métodos de estimação do sinal apresentados previamente e os respectivos erros são medidos, mostrando que o programa desenvolvido se comporta da forma esperada. As conclusões e comentários finais são apresentados no Capítulo 7 .

No Apêndice A há uma listagem das publicações associadas a esta dissertação. No Apêndice B são descritas algumas técnicas de normalização, que devem ser aplicadas aos dados de saída antes de continuar a sua análise.

# Capítulo 2

# Descrição do problema

## 2.1 Expressão gênica

Genes são importantes para transferir, de pai para filho, ou, dentro de um mesmo organismo, às células resultantes de uma divisão celular, as informações necessárias para regular a atividade metabólica das células. São seqüências ordenadas de nucleotídeos agrupadas em estruturas auto-replicáveis de ácido desoxirribonucléico (DNA) nos núcleos das células eucariotas, denominadas cromossomos.

A molécula de DNA localizada no núcleo das células normalmente se encontra enrolada sobre si mesma, desenrolando-se somente durante a divisão celular, ou em alguns trechos para a síntese das moléculas de ácido ribonucléico (RNA) que irão para o citoplasma. É formada por uma hélice dupla de DNA que se mantém unida através de pontes de hidrogênio entre suas bases nitrogenadas.

Nucleotídeos são a unidade fundamental dos cromossomos e das moléculas de RNA, sendo compostos por uma base nitrogenada, uma molécula de fosfato e uma molécula de açúcar. As bases nitrogenadas encontradas no DNA são a adenina, guanina, citosina e timina, e no RNA são a adenina, guanina, citosina e uracila. A molécula de açúcar do DNA é a desoxirribose e a do RNA é a ribose.

Como a guanina se liga apenas à citosina e a adenina, apenas à timina ou uracila, pode-se

deduzir a seqüência original a partir de uma cópia.

Expressão gênica é o processo que envolve a conversão da informação contida nos genes em proteínas. Esse processo ocorre em dois passos (Figura 2.1): i - *transcrição* da informação codificada em um gene, mediante a síntese de moléculas de RNA mensageiro (mRNA), isto é, cópias do gene que são enviadas para fora do núcleo da célula, e ii - *tradução* da informação codificada nos nucleotídeos do mRNA mediante a síntese de uma proteína; cada seqüência de mRNA define uma única proteína.

Saber os níveis de transcrição em vários tecidos é importante para responder perguntas como qual o papel de diferentes genes e em que processos atuam; como os genes e seus produtos interagem; como variam os níveis de expressão gênica em vários tipos de células e estados, ou com doenças e tratamentos. Apesar do mRNA não ser o último produto de um gene, transcrição é o primeiro passo da expressão, e informação sobre os níveis de transcrição é necessária para entender as redes de regulação gênica [9].

No caso da síntese de RNA ribossomal (rRNA), encontrado nos ribossomos, e RNA transportador (tRNA), que se liga a aminoácidos e participa da síntese de proteínas, apenas a transcrição é necessária.

Proteínas são macromoléculas compostas por uma ou mais cadeias de aminoácidos em uma certa ordem, dada pela seqüência de bases nitrogenadas do gene que a codifica. Proteínas sao necessárias para a estrutura, funcionamento e regulação das células, tecidos e órgãos. A quantidade de cada proteína constitui uma parte importante do funcionamento interno das células e dos tecidos já que muitas reações ocorrem através delas. Cada proteína tem uma única função. Exemplos de proteínas são as enzimas, hormônios e anticorpos.

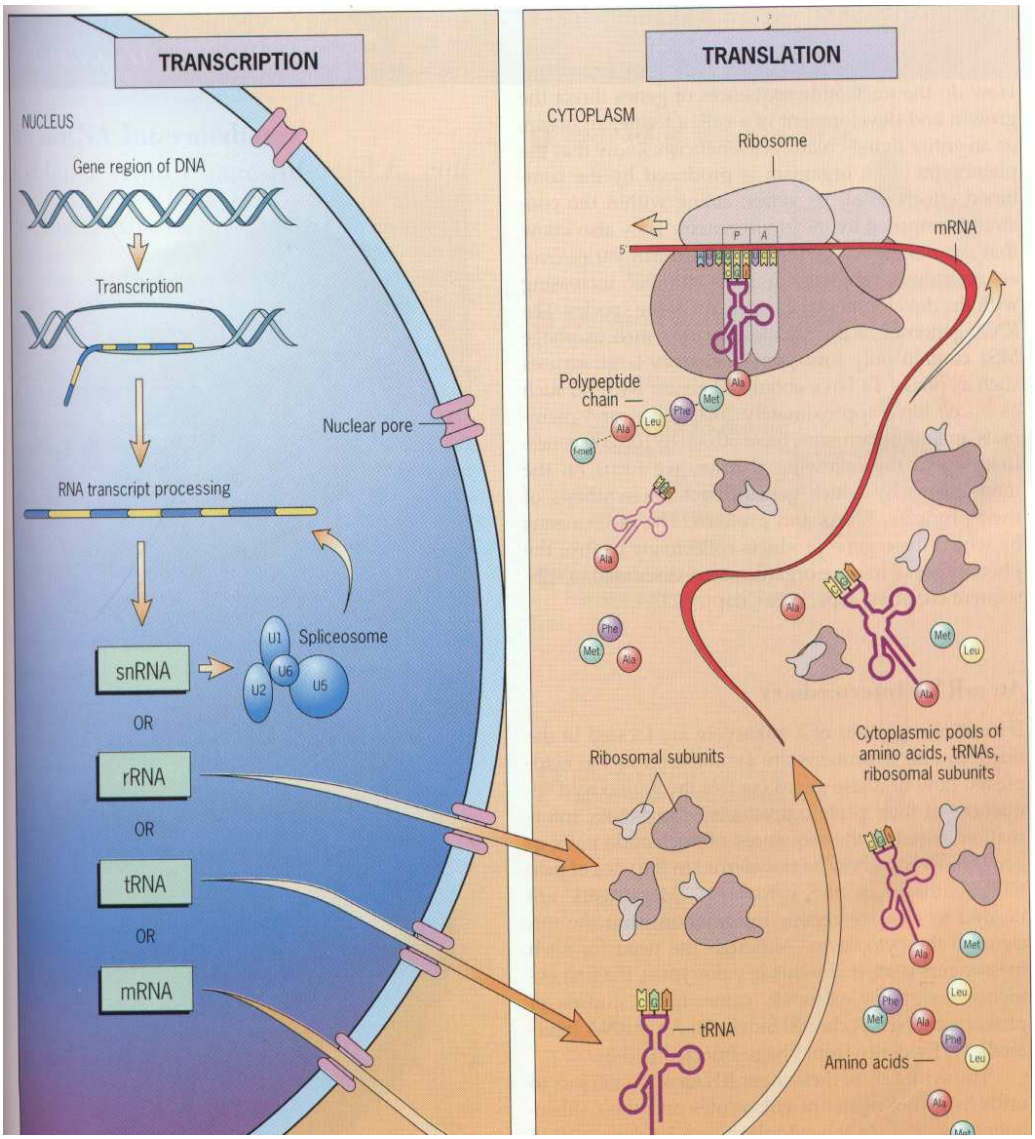



Figura 2.1: Expressão gênica: i - Transcrição, ii - Tradução.
*Ref: Biology Pages*

Podemos medir o nível de expressão de cada gene medindo quantas cópias de mRNA estão presentes na célula. Dentre as técnicas mais usadas de medida dos níveis de mRNA estão [10]:

- Microarrays de cDNA
- Chips de oligonucleotídeos Affymetrix® GeneChip®
- RT-PCR (*Reverse Transcriptase Polymerase Chain Reaction*)
- Análise Serial da Expressão Gênica (SAGE: *Serial Analysis of Gene Expression*)

## 2.2 Tecnologia de *microarrays*

A tecnologia de *microarrays* é um processo baseado em hibridização que possibilita observar a concentração de mRNA de uma amostra de células analisando a luminosidade de sinais fluorescentes. Hibridização é o processo bioquímico onde duas fitas de ácido nucleico com seqüências complementares se combinam.

Uma lâmina de *microarray* tem, em cada *spot*, pedaços de cDNA de um gene que se quer estudar. Uma única lâmina pode conter milhares de *spots*, permitindo que se analisem milhares de genes ao mesmo tempo. Essa lâmina de vidro é mergulhada em uma solução com mRNA extraído da amostra de células, rotulado com corante fluorescente.

Normalmente os experimentos de *microarray* são feitos com duas amostras de mRNA, cada uma rotulada com um corante fluorescente diferente, em geral cy3 e cy5. Assim, o mRNA hibridiza com o cDNA correspontende, fazendo com que o *spot* acenda mais na cor correspondente à amostra que contém mais mRNA proveniente do seu gene.

Após esse processo a lâmina é digitalizada em alta resolução gerando uma imagem que deve ser analisada em vários passos. Primeiro se faz o endereçamento, que é associar a cada *spot* o gene correspondente. Em seguida, deve-se localizar exatamente a região que corresponde ao sinal do *spot*, a chamada segmentação do *spot*. Por fim estimar o valor do sinal luminoso de cada *spot*. Em experimentos com duas amostras de mRNA é medida a expressão relativa do gene nas duas amostras, valor proporcional à intensidade relativa dos dois corantes em cada *spot*.

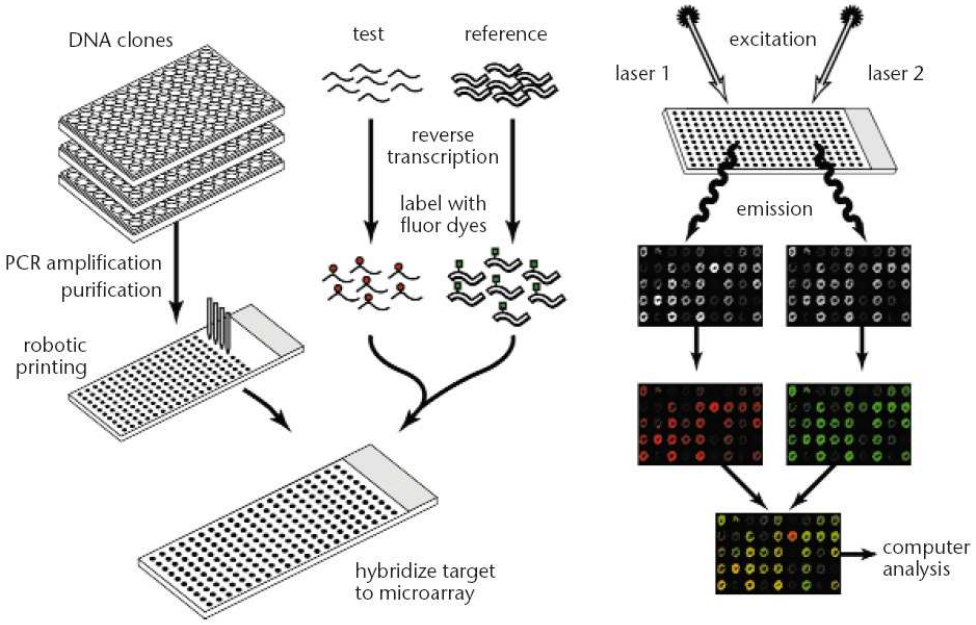



Figura 2.2: Tecnologia de *microarray* [1].

A região útil, que contém *spots*, de uma lâmina de *microarray* típica possui dimensões em torno de 2x4cm. Tal lâmina, se digitalizada a laser com uma resolução de 10$\mu$m por *pixel*, terá 2000x4000 *pixels*. Uma boa digitalização deve ter mais de 256 tons de cinza, assim, costuma-se digitalizar para 65536 ($2^{16}$) tons de cinza, onde cada *pixel* ocupa dois *bytes* de informação. Se tal lâmina for hibridizada com dois corantes, deve ser digitalizada para uma imagem com dois canais. Essa imagem ocupará cerca de 32 *megabytes* de informação.

As Figuras 2.3 e 2.4 são exemplos de imagens digitais de lâminas de *microarray*. A Figura 2.5 mostra o resultado do gradeamento dos blocos.

A tecnologia de *microarrays* visa analisar a expressão gênica de milhares de genes simultaneamente, objetivando alto *throughput*. No entanto, encontra um gargalo na fase de segmentação da imagem da lâmina para geração dos dados de expressão gênica. Analisamos diversos programas de análise de imagens de *microarray* e todos exigem um penoso processo manual, são intensamente interativos. Na Seção 2.5 descrevemos algumas das ferramentas mais populares.

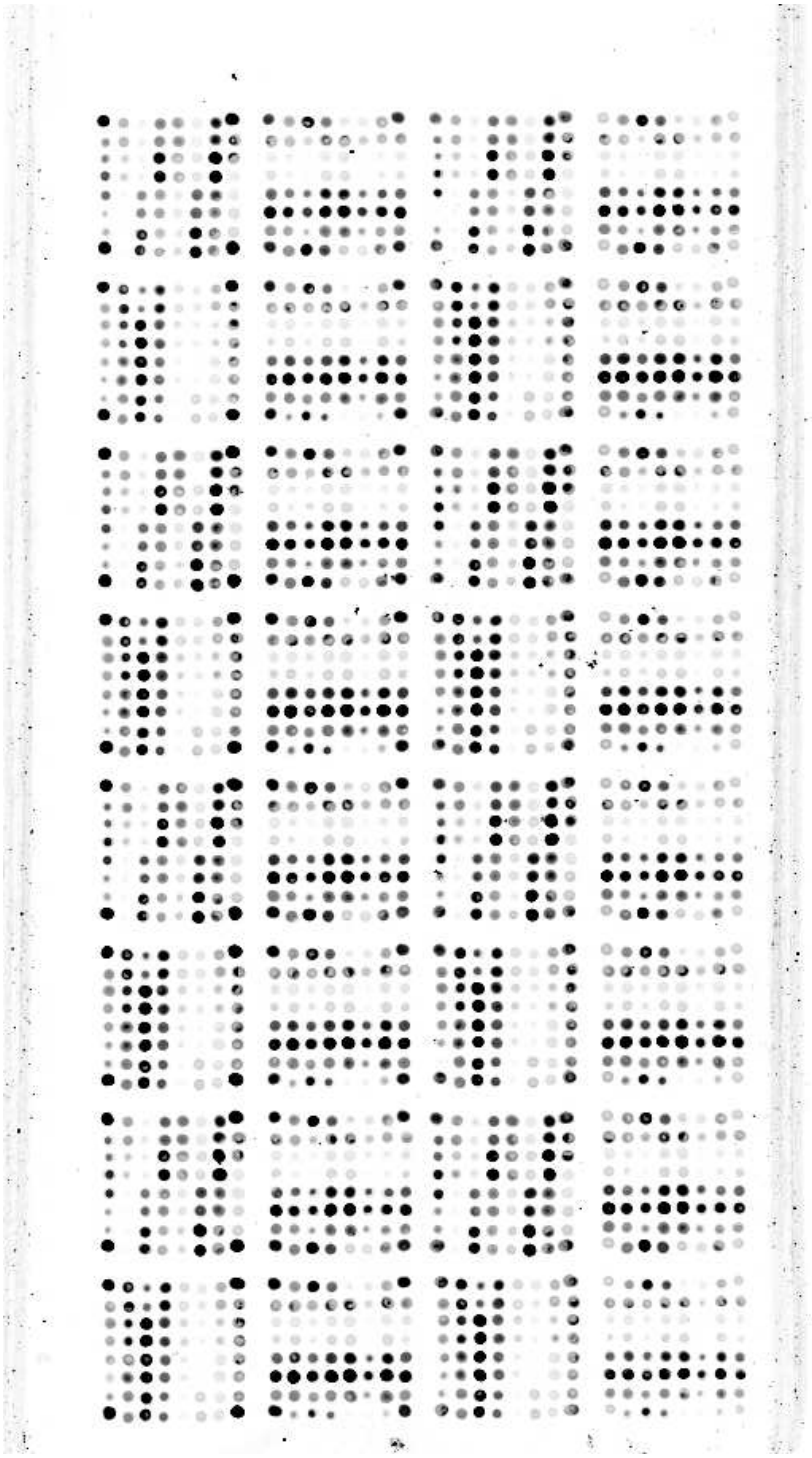


Figura 2.3: Exemplo de imagem de *microarray* digitalizada a laser (GHE037, produzida com genes sintéticos pelo pesquisador Gustavo Henrique Esteves, do Instituto Ludwig).

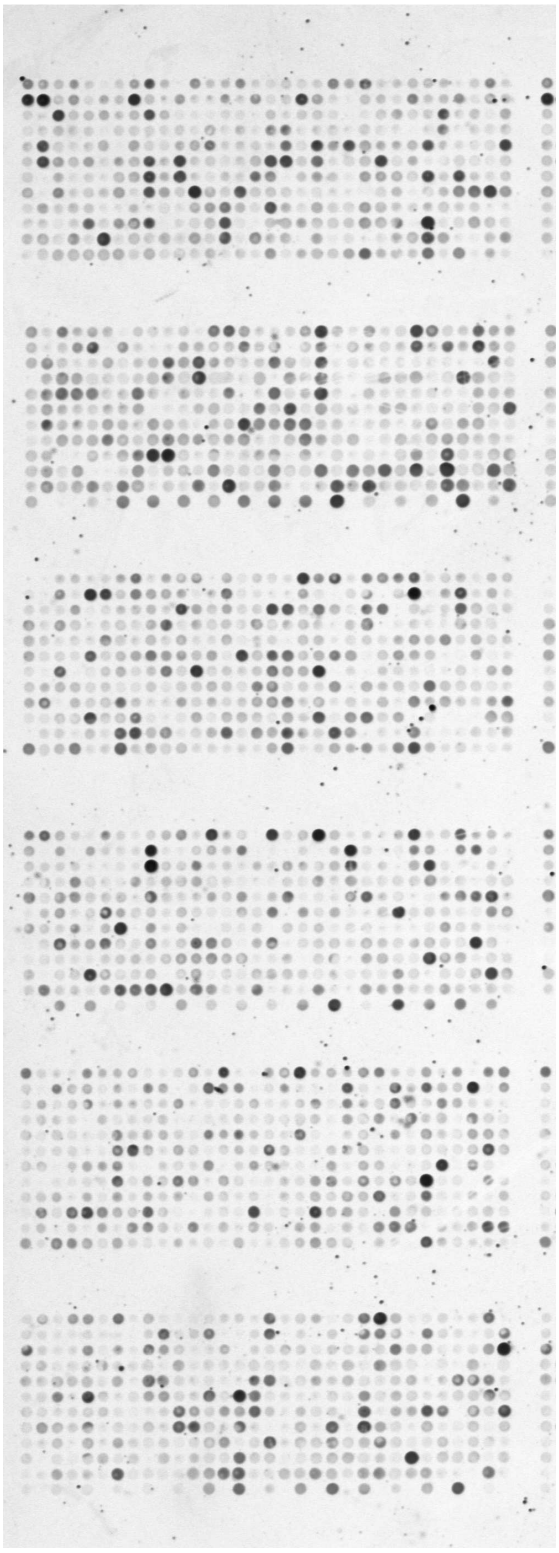


Figura 2.4: Exemplo de imagem de *microarray*.

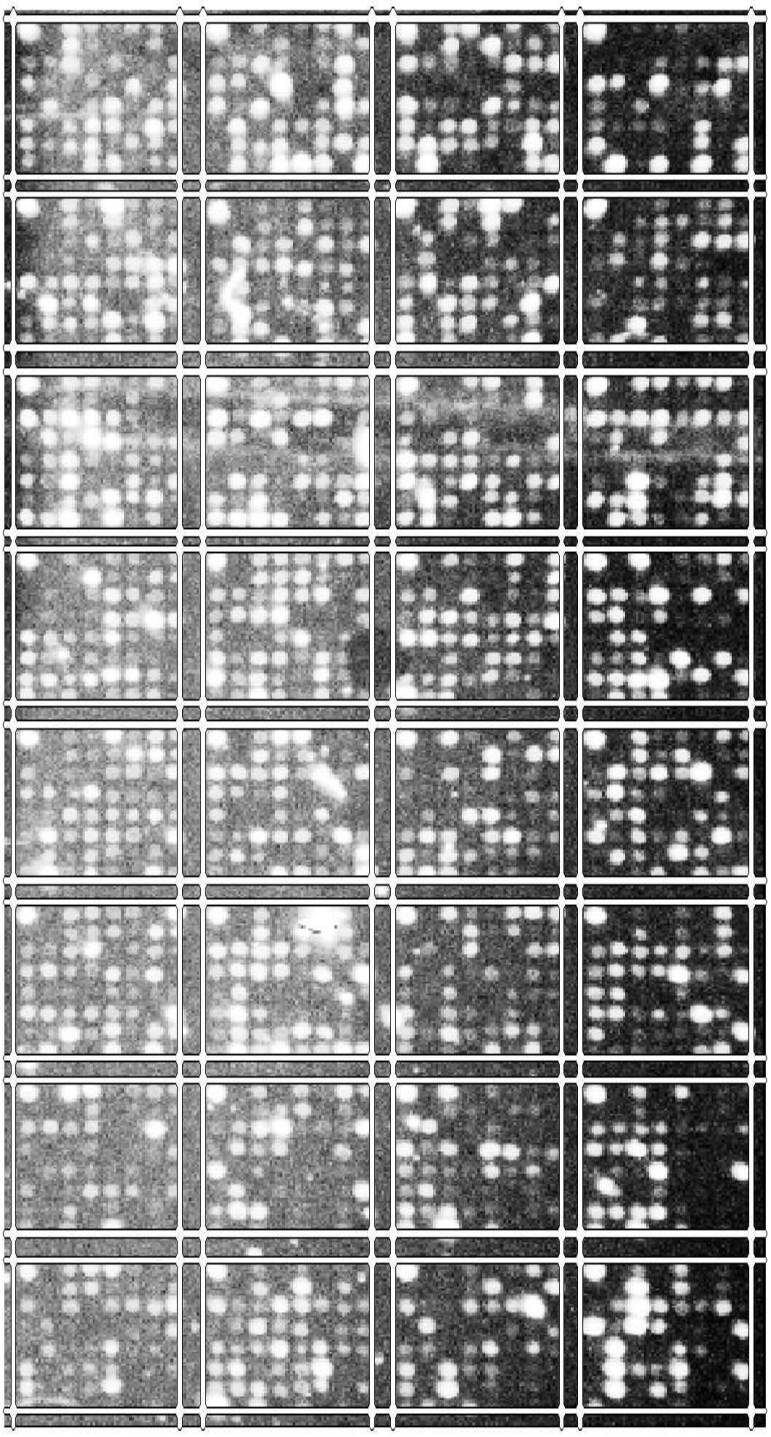


Figura 2.5: Exemplo de segmentação de blocos de *microarray* (lâmina produzida pela pesquisadora Beatriz Simonsen Stolf, do Instituto Ludwig).

A técnica desenvolvida elimina tal gargalo, gerando, com mínima interação com o usuário, os dados de expressão gênica.

## 2.3 Variantes da tecnologia de *microarrays*

A tecnologia de *microarray* tem algumas variantes que, para efeito de análise de imagens, são equivalentes. Uma delas é a lâmina de oligonucleotídeos [11], que usa *probes* sintetizados e posteriormente depositados na lâmina. O uso de oligonucleotídeos muito curtos resulta em hibridizações menos específicas e menor sensibilidade, assim oligonucleotídeos mais longos (50-100 bases) são usados. Oferecem a vantagem de que a informação sobre seqüencia é suficiente para sintetizar os *probes*, evitando o manuseio de cDNA [12].

Outra tecnologia é a lâmina de *nylon*, em geral usada com detecção radioativa, ou seja, os *targets* são marcados radioativamente. Essas lâminas permitem a análise de apenas uma amostra de cada vez. Os chamados *macroarrays* são similares, usam o mesmo substrato e marcação, diferindo apenas na escala. Tais lâminas têm em média 10 centímetros de lado [13].

## 2.4 Problema de medida

A partir a imagem, precisa-se extrair uma tabela com o valor de expressão gênica de cada *spot*. Os dados dessa tabela, se cruzados com os nomes dos respectivos genes, indicam quais genes são mais ou menos expressos na célula sob a condição do experimento, e podem ser usados para responder às perguntas apresentadas na seção 2.1.

A intensidade do sinal fluorescente em cada *spot* guarda uma relação linear com a concentração de mRNA na amostra num intervalo de pelo menos cinco ordens de grandeza de diluições [3]. Assim, assumimos que a intensidade da fluorescência ($Y$) no interior de cada *spot* deve ser

$$Y_{ik} = a_i + b_i X_{ik}$$

onde $i = 1, 2$ denota os canais; $k = 1, \ldots, K$, os genes; $a$, a intensidade no *background*; $X_{ik}$ é a intensidade do gene $k$ no canal $i$; e $b_i$ é o fator de ganho do fluorcromo usado no canal $i$ [14].

Devido a diferenças físicas dos corantes e à configuração do *scanner*, os valores de $a_i$ e $b_i$ são diferentes nos dois canais. Neste modelo, as variações de $a_i$ e $b_i$ dentro de um mesmo canal são ignoradas.

Deve-se observar ainda que a relação acima, em dados reais de *microarray* sofre influência de ruídos estocásticos de diversas fontes, como amplificação eletrônica na fase da digitalização, hibridização não específica etc.

Na fronteira do *spot*, o sinal passa a não obedecer a função linear e o sinal cai até se confundir com o *background*. Assim, a fase de segmentação deve identificar tais *pixels* como não sendo parte do sinal, ou o resultado será de baixa qualidade.

## 2.5 Soluções disponíveis no mercado

Esta Seção descreve de forma sucinta as principais características, como facilidade de uso, técnicas de segmentação e de cálculo do sinal, das soluções mais populares do mercado.

### 2.5.1 ScanAlyze

*Software* elaborado pelo *Eisen Lab*[1] da Universidade da Califórnia em Berkeley. A licença é gratuita para uso não comercial.

Requer que o usuário identifique alguns *spots* e encontra os demais por interpolação. Tem uma função para encontrar *spots* desalinhados, mas não funciona bem. A segmentação dos *spots* é sempre circular ou elíptica, o que na realidade nem sempre ocorre, ou seja, o círculo identifica como sinal uma região de *background*, diminuindo a qualidade dos valores de expressão encontrados, o que aumenta a dispersão.

[1] rana.lbl.gov/EisenSoftware.htm

### 2.5.2 Spotfinder (TIGR)

Solução criada por *The Institute for Genomic Research*[2], sua licença também é gratuita para uso não comercial.

A segmentação dos blocos é feita manualmente, mas a segmentação do sinal dos *spots* é automática e parametrizada. O usuário escolhe um valor de raio e a área do sinal muda proporcionalmente. Além disso, identifica bem a área do sinal do *spot* mesmo quando tem forma de meia lua ou de rosca.

### 2.5.3 Arrayvision

Solução comercial da *Imaging Research Inc*[3].

Esta ferramenta oferece muitas opções de uso, o que pode confundir o usuário iniciante, mas por outro lado, é uma vantagem para quem já está habituado com o *software*. É capaz de encontrar *spots* mal alinhados.

### 2.5.4 Quantarray

Solução da *Packard Bioscience*[4]. Pode analisar imagens com até cinco cores. Oferece três formas de calcular a expressão: método do histograma, círculo fixo e adaptativo.

### 2.5.5 UCSF Spot

O UCSF Spot é um projeto do *Jain Lab*[5], da Universidade da Califórnia em São Francisco.

Sua intenção é fazer todo o processo de segmentação e geração dos dados de forma automática. No entanto, quando a imagem é ruidosa, o resultado pode sair errado e a correção manual é bastante trabalhosa.

[2] www.tigr.org/software
[3] www.imagingresearch.com
[4] www.packardbioscience.com
[5] jainlab.ucsf.edu/Projects.html

# Capítulo 3

# Segmentação desenvolvida

## 3.1 Introdução

Neste Capítulo daremos algumas definições necessárias para o entendimento de nossa abordagem dos problemas de gradeamento dos *subarrays*, gradeamento dos *spots* e detecção dos *spots*, que será descrita em detalhes em seguida. As definições foram extraídas de [15] e de [16].

## 3.2 Morfologia matemática

Seja $\mathbf{Z}$ o conjunto dos números inteiros; a origem de $\mathbf{Z}^2$ é denotada por $o = (0, 0)$. Seja $E$ um retângulo finito e não vazio em $\mathbf{Z}^2$, e $K$ um intervalo $[0, k]$ de $\mathbf{Z}$, com $k > 0$.

Uma função $f$ de $E$ em $K$, $f \in K^E$, representa uma *imagem em tons de cinza*, doravante denominada apenas *imagem*. Cada ponto de $E$ representa um *pixel* ou ponto da imagem, ou seja, $p \in E$ é um ponto da imagem $f$, cujo tom de cinza vale $f(p)$.

Um subconjunto $B$ de $E$ também é chamado *elemento estruturante*.

O operador *identidade* é denotado por $\iota$, e é tal que $\iota(f) = f$.

A *união* de duas imagens $f_1$ e $f_2$, denotada $f_1 \vee f_2$, é a função em $K^E$ dada por, para todo $x \in E$, $(f_1 \vee f_2)(x) = \max\{f_1(x), f_2(x)\}$.

A *interseção* de duas imagens $f_1$ e $f_2$, denotada $f_1 \wedge f_2$, é a função em $K^E$ dada por, para todo $x \in E$, $(f_1 \wedge f_2)(x) = \min\{f_1(x), f_2(x)\}$.

A *adição* de duas imagens $f_1$ e $f_2$, denotada $f_1 + f_2$, é a função em $K^E$ dada por, para todo $x \in E$,

$$(f_1 + f_2)(x) = \begin{cases} f_1(x) + f_2(x) & \text{se } f_1(x) + f_2(x) \leq k \\ k & \text{caso contrário} \end{cases}$$

A *reflexão* de um subconjunto $X \subseteq E$ é o subconjunto $\check{X} = \{r \in E : r = -x, x \in X\}$.

Para todo $X \subseteq E$ e $y \in E$, $X_y$ denota $X$ transladado de $y$, ou seja, $X_y = \{x \in E : x - y \in X\}$.

A *dilatação* e *erosão* de uma imagem $f$ por um elemento estruturante $B$ são, respectivamente, as funções $\delta_B(f)$ e $\varepsilon_B(f)$ em $K^E$ dadas por, para todo $x \in E$,

$$\delta_B(f) = \max\{f(y) : y \in \check{B}_x \cap E\}$$

e

$$\varepsilon_B(f) = \min\{f(y) : y \in B_x \cap E\}$$

Os operadores $\delta_B^n$ e $\varepsilon_B^n$ dados, para $n > 0$ pelas $n-1$ composições sucessivas $\delta_B^n = (\delta_B)^n$ e $\varepsilon_B^n = (\varepsilon_B)^n$, e para $n = 0$, $\delta_B^0 = \iota$ e $\varepsilon_B^0 = \iota$, são chamados, respectivamente, de *n-dilatação* e *n-erosão* por B.

Seja $f$ um elemento de $K^E$. Os operadores $\delta_{B,f}$ e $\varepsilon_{B,f}$, de $K^E$ em $K^E$, dados por $\delta_{B,f} = \delta_B \wedge f$ e $\varepsilon_{B,f} = \varepsilon_B \vee f$, são chamados *dilatação condicional* e *erosão condicional* por $B$ dado $f$.

Seja $n$ um inteiro positivo. A sucessão de $n$ dilatações condicionais $\delta_{B,f}$ (respectivamente erosões condicionais $\varepsilon_{B,f}$), denotada por $\delta_{B,f}^n = \delta_{B,f}\delta_{B,f}\ldots\delta_{B,f}$ ($\varepsilon_{B,f}^n = \varepsilon_{B,f}\varepsilon_{B,f}\ldots\varepsilon_{B,f}$) é chamada *n-dilatação condicional* (*n-erosão condicional*).

Seja $g$ um elemento de $K^E$. Os operadores $\gamma_{B,g}$ e $\phi_{B,g}$, de $K^E$ em $K^E$, dados por, para todo $f \in K^E$, $\gamma_{B,g} = \delta_{B,f}^\infty(g)$ e $\phi_{B,g} = \varepsilon_{B,f}^\infty(g)$ são chamados, respectivamente, de *inf-* e *sup-reconstrução* do marcador $g$.

Para toda $f \in K^E$, a *erosão última*, $\varrho_B(f)$, de uma função $f$ por um elemento estruturante

$B$ é dada por $\varrho_B(f) = \bigvee\{\varepsilon_B^i(f) - \gamma_{B,\varepsilon_B^{i+1}(f)}(\varepsilon_B^i(f))\}$ para todo $i \geq 0$.

Os operadores $\gamma_B$ e $\phi_B$, de $K^E$ em $K^E$, dados por $\gamma_B = \delta_B\varepsilon_B$ e $\phi_B = \varepsilon_B\delta_B$, são chamados, respectivamente, de *abertura* e *fechamento* pelo elemento estruturante $B$.

O operador $\iota - \gamma_B$ é chamado *top-hat* pelo elemento estruturante $B$.

Sejam $A$ e $B$ subconjuntos do quadrado 3 x 3. O operador $\nabla_{A,B}$ de $K^E$ em $K^E$, dado por $\nabla_{A,B} = \delta_A - \varepsilon_B$ é chamado *gradiente morfológico*. Esse operador destaca as bordas da imagem.

Seja $t \geq |E|$, e seja $i \mapsto x_i$ um processo de numeração dos elementos de $E$, ou seja, uma bijeção de $[1, \ldots, E] \subset \mathbf{N}$ em $E$, e seja $f$ um elemento de $K^E$ tal que $f(x_i) = i$ para $x \in E$. O operador $\Lambda_B$ de $\{0,t\}^E$ em $K^E$, dado por, para todo $g \in \{0,t\}^E, \Lambda_B(g) = \gamma_{B,g\wedge f}(g)$ é chamado *rotulação* de $g$. Note que, em $\Lambda_B(g)$, cada ponto de uma *componente conexa* [17] de $g$ é associado ao mesmo valor.

Dado um ponto $(a,b) \in E$, definimos $E_{x=a} \subseteq E$ como a linha que corta $E$ na direção vertical passando pela coordenada $(a,b)$, ou seja, $E_{x=a} = \{(x,y) \in E : x = a\}$. De modo similar, podemos definir $E_{y=b} = \{(x,y) \in E : y = b\}$ como a linha que corta $E$ na direção horizontal passando pela coordenada $(a,b)$.

Dada uma imagem $f : E \to K$, sua *projeção horizontal*, denotada por $P_h(f)$, é a função de $E_{x=0}$ em $\mathbf{Z}$, tal que, para todo $(0,i) \in E_{x=0}$,

$$P_h(f)(0,i) = \sum_{p \in E_{y=i}} f(p)$$

De forma análoga, podemos definir a *projeção vertical* de $f$, denotada por $P_v(f)$, como a função de $E_{y=0}$ em $\mathbf{Z}$, tal que, para todo $(i,0) \in E_{y=0}$,

$$P_v(f)(i,0) = \sum_{p \in E_{x=i}} f(p)$$

Um *máximo* (ou *mínimo*) *regional* $M \subset E$ de uma função $f \in K^E$ é uma componente conexa com um dado valor $f(p) = h, \forall p \in M$ (platô no nível de $h$) tal que todo ponto na vizinhança de $M$ tem um valor estritamente menor (maior) que $h$. Os máximos e mínimos regionais são dados

pelos operadores morfológicos $\varrho_B^{\max}$ e $\varrho_B^{\min}$. A vizinhança de um ponto é definida pelo elemento estruturante $B$.

Dada uma imagem $f$, o *threshold* de $f$ por $c$, denotado por $\tau_c(f)$, é a função de $E$ em $\{0,1\}$ dada por, para todo $x \in E$,

$$\tau_c(f)(x) = \begin{cases} 1 & \text{se } f(x) \geq (c) \\ 0 & \text{caso contrário} \end{cases}$$

Seja $x$ um valor real ou inteiro qualquer, seu módulo, denotado por $|x|$ é igual a $x$ se $x$ for maior ou igual a zero e $-x$ se $x$ for menor que zero. A distância *city-block* entre dois pontos $(x_0, y_0)$ e $(x_1, y_1)$ é dada por $|x_0 - x_1| + |y_0 - y_1|$.

### 3.2.1 Paradigma de Beucher-Meyer

Esse operador é também conhecido como *watershed* [18, 19]. É um operador que identifica os contornos da imagem, ou seja, regiões de mudança de intensidade, onde o gradiente é maior. Este operador é muito robusto pois seu resultado não depende de parâmetros além das imagens de entrada, e é capaz de identificar contornos mesmo que o gradiente seja próximo de zero.

Dado o gradiente $f$ de uma imagem que contém o assunto a ser segmentado, e uma imagem $g$ binária, com marcadores que identifiquem as regiões que devem ser separadas, o algoritmo encontra os pontos de maior intensidade em $f$ que separam as duas regiões. Este operador é denotado por $\Omega(\nabla(f), g)$.

## 3.3 Gradeamento automático dos blocos

A segmentação dos blocos consiste em encontrar a posição correta e os limites de cada bloco. É realizada aplicando-se vários filtros aos perfis horizontal e vertical da imagem da lâmina de *microarray*. O perfil vertical (horizontal) é calculado somando-se os valores de cada coluna (linha) de pontos da imagem original (Figura 3.1). Caso haja mais de um canal, usa-se a média entre eles.

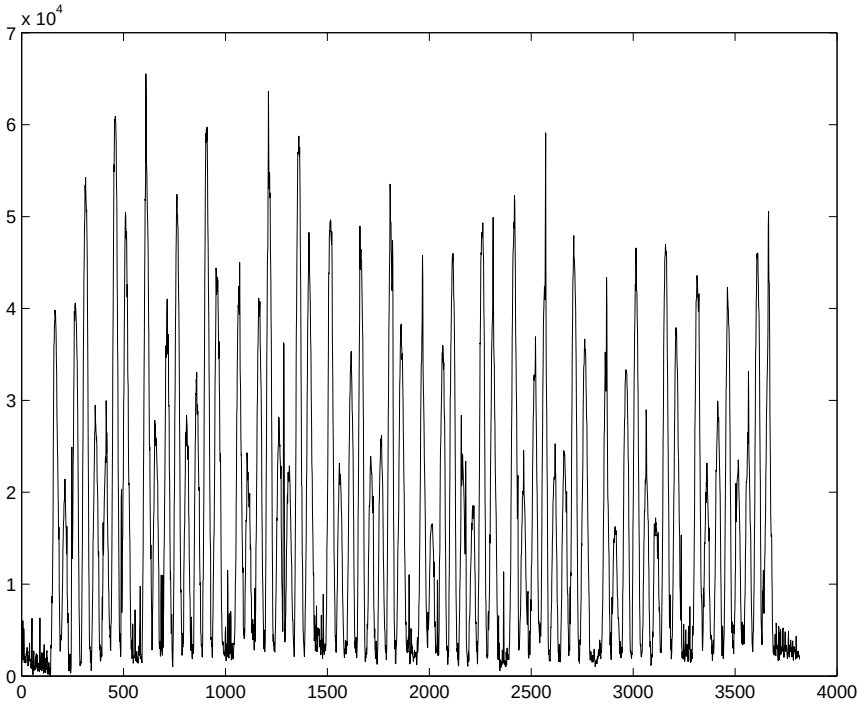



Figura 3.1: Exemplo de perfil vertical.

Se a lâmina for impressa e digitalizada com o mesmo ângulo de inclinação, será possível ver que os *spots* estarão alinhados paralelamente às bordas da imagem. Nesse caso, nos perfis são facilmente visíveis grupos de picos estreitos nas regiões onde os blocos se encontram.

O perfil vertical dá as linhas verticais da grade de blocos, e o horizontal, as linhas horizontais.

O primeiro passo da filtragem (Figura 3.2) é aplicar um fechamento morfológico por um elemento estruturante linear simétrico de tamanho maior que a distância entre os *spots* mas menor que a distância entre os blocos aos perfis da imagem. Este operador agrupa perfis de *spots* pertencentes ao mesmo bloco.

O segundo passo (Figura 3.3) é uma abertura morfológica por um elemento estruturante linear simétrico de tamanho igual a algumas distâncias entre *spots*. Esta operação apaga picos estreitos, e deixa os picos mais largos que correspondem às regiões dos blocos.

Em seguida, calcula-se a negação dos mínimos locais. É um gráfico binário que vale zero nas regiões entre os blocos. O passo seguinte é eliminar os componentes conexos que tocam a borda

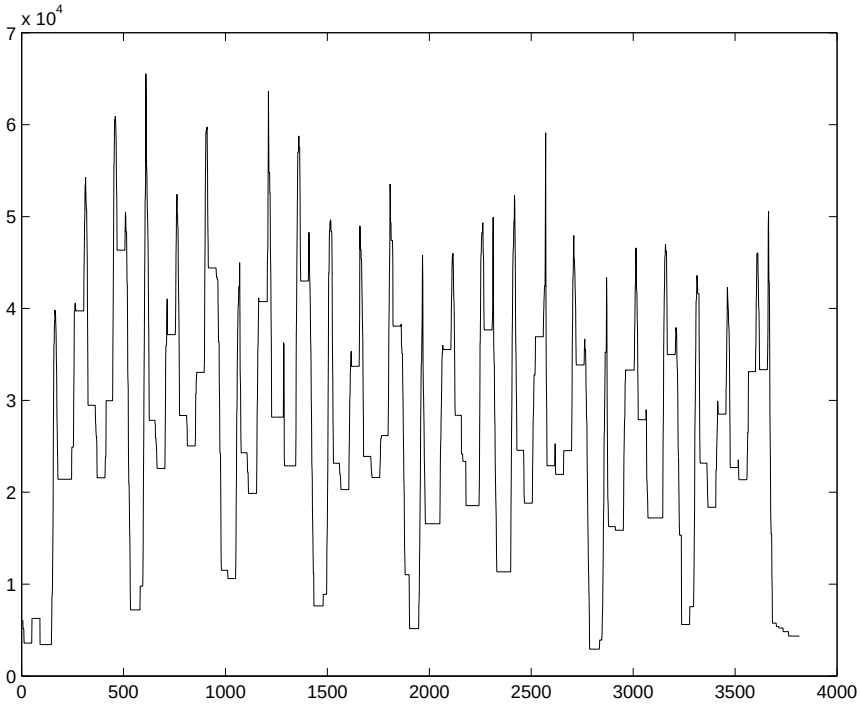



Figura 3.2: Resultado do primeiro filtro.

do gráfico. Esse passo é importante especialmente em imagens com borda ruidosa, já que nesses casos é comum que tal ruído seja identificado como região pertencente a um bloco (Figura 3.4).

Em geral, nesse ponto do processo, boas imagens já permitem obter uma boa aproximação da segmentação dos blocos. No entanto, imagens ruidosas são muito comuns e adicionamos um passo de correção baseado na geometria da lâmina. Esse passo é acionado quando a diferença entre o tamanho estimado de algum bloco ou distância entre blocos e o respectivo tamanho teórico, fornecido pelo usuário no início do processo, é maior que uma certa tolerância. A tolerância usada é igual a meia distância entre centros de *spots* adjacentes.

A posição final do bloco é tal que satisfaça a tolerância, maximizando o valor médio do perfil dentro de uma região igual ao tamanho teórico do bloco. É de se esperar que, se o bloco for mal centralizado, o valor médio do perfil será baixo já que está incluída uma região do perfil entre blocos.

A Figura 3.5 mostra quatro gráficos superpostos, em unidades arbitrárias. De cima para

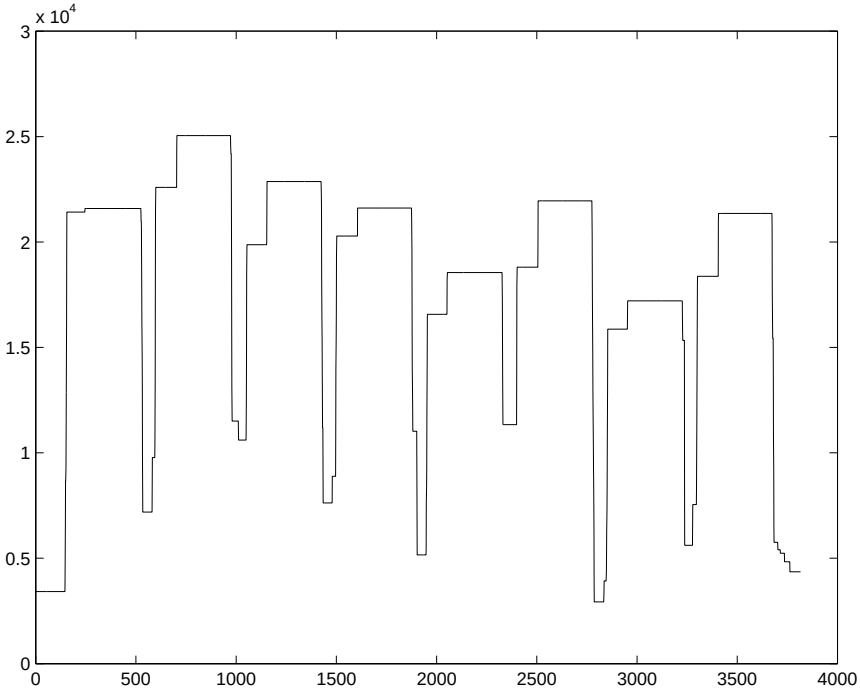



Figura 3.3: Resultado do segundo filtro.

baixo, perfil médio na região igual ao tamanho teórico do bloco começando naquele ponto; perfil da imagem; posições dos blocos calculadas pelos filtros morfológicos; e posições dos blocos calculadas por esse passo de correção.

O algoritmo se baseia nas posições dos blocos que satisfazem a tolerância para calcular as novas posições dos que foram considerados errados. Além disso, usa informações do perfil, o que o torna mais robusto. Quando há vários blocos errados em seqüência, o algoritmo se baseia no bloco mais próximo do centro, que é a região menos sujeita a ruído.

Quando todos os blocos são considerados errados, o algoritmo calcula a curva do valor médio do perfil no interior dos blocos segundo dois parâmetros: i - início do primeiro bloco; e ii - espaçamento entre os blocos que obedecem a tolerância (teórico mais ou menos meia distância entre *spots*). O tamanho do bloco usado é o teórico, fornecido pelo usuário. Os parâmetros do ponto máximo da curva são os escolhidos como a segmentação correta dos blocos.

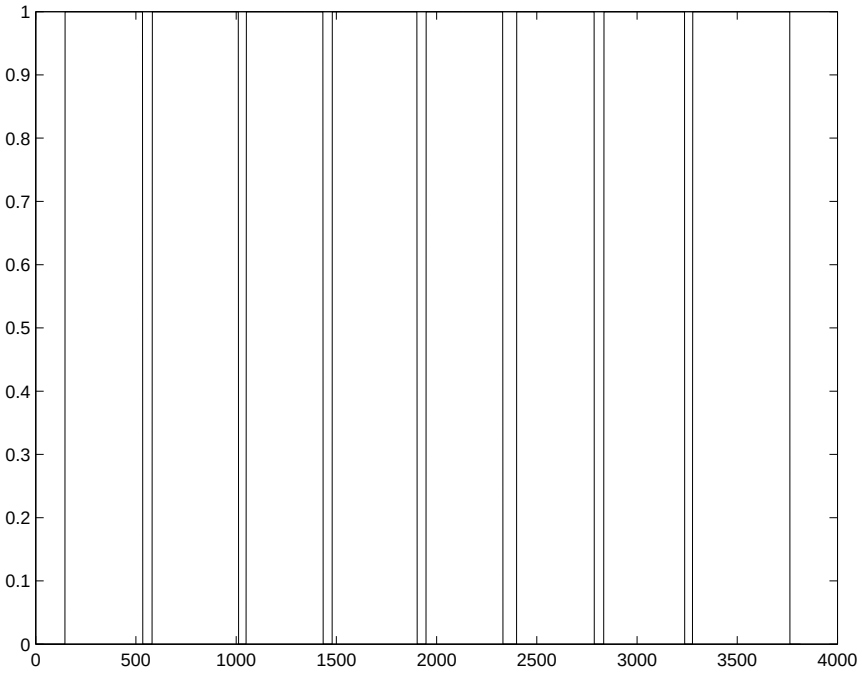



Figura 3.4: Fronteiras dos blocos.

## 3.4 Gradeamento automático dos *spots*

Para calcular o gradeamento dos *spots*, o algoritmo se baseia na segmentação dos blocos. O cálculo também é feito baseado nos perfis, mas agora de cada bloco, individualmente. O programa calcula os perfis $f_i$, onde $i$ é o índice do bloco, os filtra e toma os mínimos regionais como as linhas de grade. A diferença é que a filtragem é muito mais simples que a da segmentação dos blocos. Os *spots* são responsáveis pelos valores mais altos do perfil, e o *background*, pelos valores mais baixos.

As figuras 3.6 (a) e 3.7 (a) mostram os perfis do bloco na Figura 3.8 sendo segmentado. Embora simples, a filtragem é necessária para eliminar mínimos locais que podem aparecer entre os mínimos realmente interessantes que indicarão a posição da grade. A filtragem elimina mínimos locais causados por ruído que podem aparecer, tanto nos picos do perfil, quanto próximos ao fundo dos vales entre fileiras de *spots*.

A filtragem é simplesmente uma abertura por um elemento estruturante $B_n$ linear, onde $n$

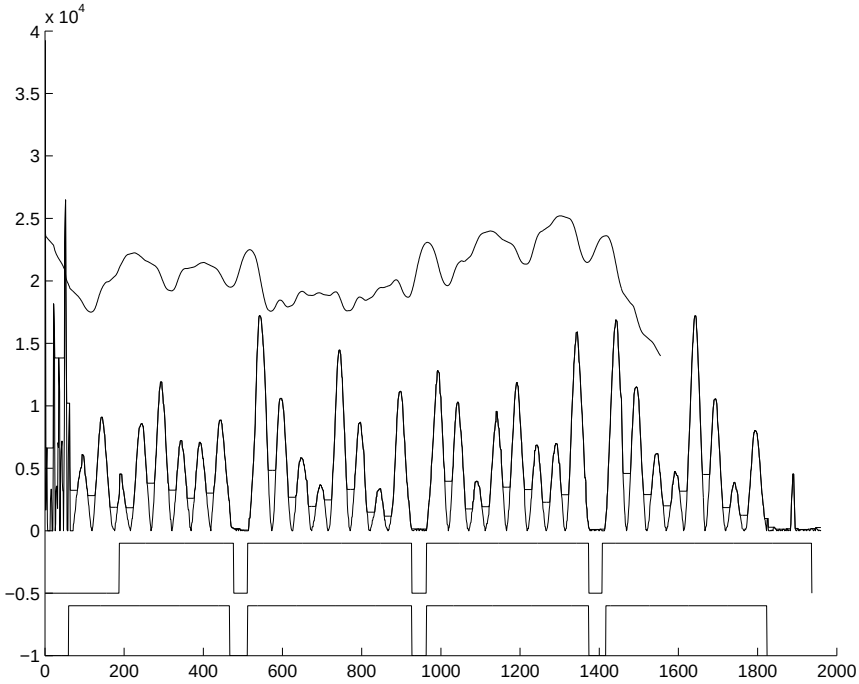



Figura 3.5: Correção da segmentação morfológica.

é aproximadamente igual ao diâmetro médio do *spot*. As figuras 3.6 (b) e 3.7 (b) mostram o resultado da filtragem aplicada aos perfis do bloco.

O passo seguinte é calcular os mínimos regionais do perfil filtrado. Essa operação dará um conjunto de pontos não nulos que indicam a posição da grade. As figuras 3.6 (c) e 3.7 (c) mostram o resultado do cálculo dos mínimos regionais do passo anterior. Em 3.6 (d) e 3.7 (d) temos as grades resultantes e finalmente em 3.9, a composição das grades com a imagem original do bloco.

A equação abaixo mostra todos os passos do processo para o perfil horizontal de bloco $f$, indicado por $P_h(f)$.

$$M_h = \varrho_B^{min}(\gamma_{B_n}(P_h(f)))$$

A grade vertical, indicada por $M_v$, é calculada por um processo similar.

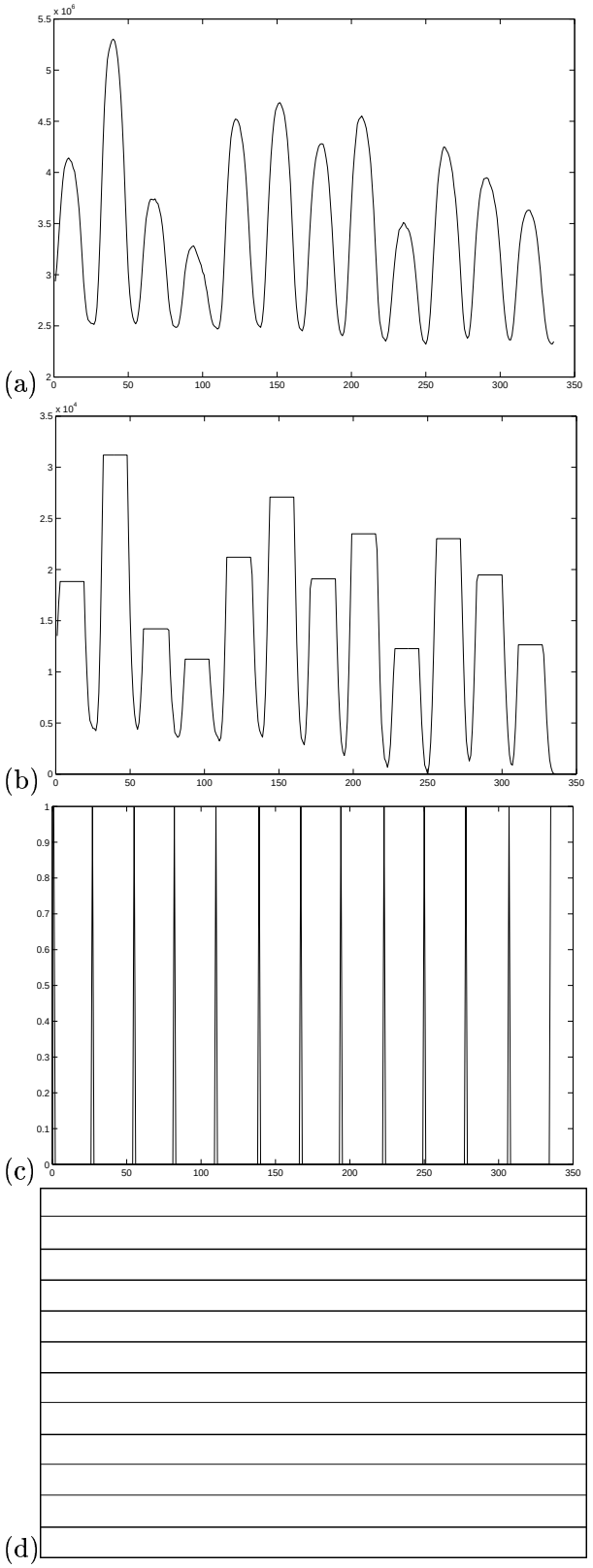



Figura 3.6: Gradeamento horizontal.

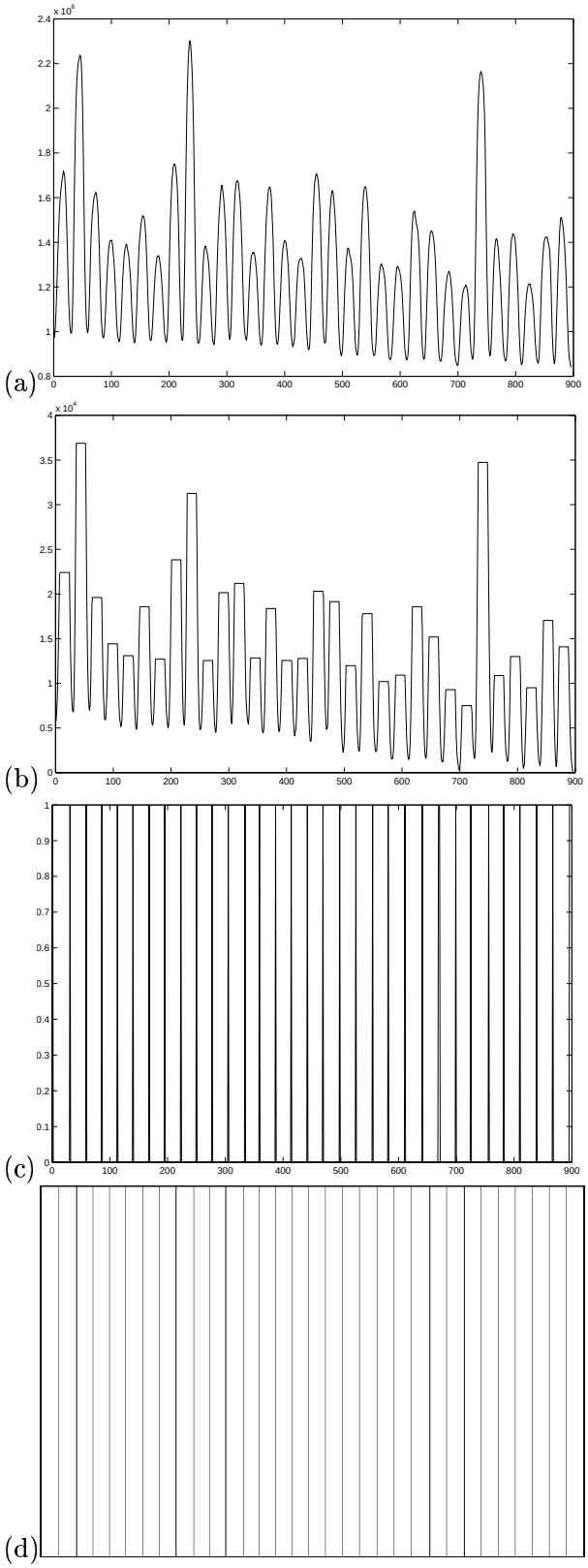



Figura 3.7: Gradeamento vertical.

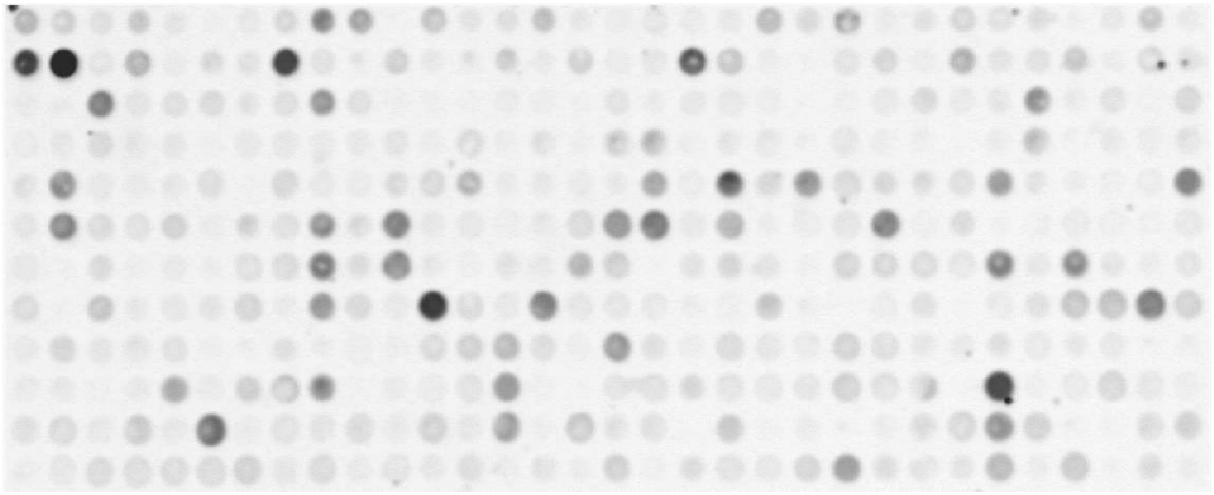


Figura 3.8: Primeiro bloco do microarray da Fig. 2.4.

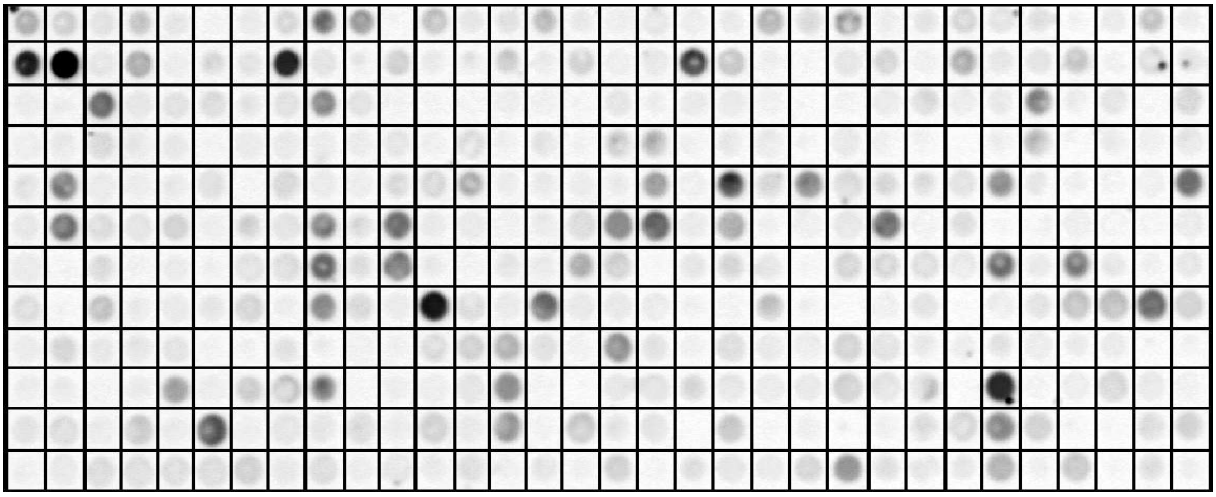


Figura 3.9: Composição das linhas de grade com o bloco da Fig. 3.8.

A posição dos elementos não nulos encontrados em $M_h$ e $M_v$, se armazenadas em um vetor, podem ser usadas facilmente para indicar a região de interesse de qualquer *spot* daquele bloco. Seja $J_h$ a seqüência de índices tais que $j \in J_h$ se $M_h(j) \neq 0$. O $J_h$ é uma seqüência de índices que indicam onde as linhas horizontais da grade estão localizadas. As linhas verticais, indicadas por $J_v$ são calculadas de forma equivalente.

Os *pixels* da linha $m$ de *spots* são todos os que têm coordenadas maiores que $J_h(m)$ e menores

que $J_h(m+1)$. Similarmente, os pixels da coluna $n$ de *spots* são todos os que têm coordenadas maiores que $J_v(n)$ e menores que $J_v(n+1)$. Os valores de $m$ e $n$ variam, respectivamente, de 1 até o número de linhas e colunas de *spots* por bloco.

Para encontrar em um bloco a região de interesse de um único *spot*, localizado na linha $m$ e na coluna $n$, basta escolher a interseção dos *pixels* da linha e coluna respectivas, ou seja, os pixels do conjunto $\{(x,y) : J_v(n) < x < J_v(n+1)$ e $J_h(m) < x < J_h(m+1)\}$.

### 3.4.1 Passo de correção

O método morfológico mostrado é robusto e produz resultados corretos para diversas das imagens testadas. No entanto, devido a defeitos nos contornos dos *spots* (contaminações por partículas de poeira, hibridização irregular, erro na manipilação etc) o método pode não funcionar corretamente.

A Figura 3.10(a) mostra a imagem de um bloco que contém um defeito bastante grande, provavelmente causado por alguma partícula de poeira depois da impressão ou da hibridização. Esse defeito causou dois tipos diferentes de problema de gradeamento: há uma linha horizontal extra particionando uma linha de *spots*, apontada por um triângulo grande, e uma linha vertical não foi encontrada na posição apontada por um triângulo pequeno.

Uma forma de corrigir esses erros seria filtrando os resultados do gradeamento usando conhecimento prévio a respeito da geometria da lâmina, como o número de *spots* em cada linha ou coluna, diâmetro e espaçamento entre eles. Como essa informação já foi fornecida pelo usuário, o processo pode continuar sem sua interferência.

O algoritmo de correção primeiro apaga as linhas muito próximas entre si, ou que aparentam estar mal posicionadas, e em seguida adiciona novas linhas entre as que estiverem muito distantes. Sejam $E_h$ e $E_v$ as dimensões horizontal e vertical em *pixels* da imagem do bloco. Seja $n_h$ o número de *spots* em cada linha do bloco, e $d_h$ a distância média entre cada linha da grade naquele bloco, ou seja, $d_h = E_h/n_h$. A distância média entre as linhas de grade que segmentam o bloco na vertical é $d_v = E_v/n_v$, onde $n_v$ é o número de *spots* em cada coluna do bloco.

Sejam $J_v$ e $J_h$ seqüências de índices que identificam onde as linhas horizontais e verticais da

grade estão localizadas, conforme definido no final da seção 3.4. Definimos $\Delta J_h$ como a seqüência das diferenças entre a posição de uma linha e a seguinte, ou seja, $\Delta J_h[i] = |J_h[i+1] - J_h[i]|$, $i \in [1, |J_h| - 1]$. A Figura 3.11 mostra $J_h$ (seqüência de cima) e respectivo $\Delta J_h$ (seqüência de baixo) para as linhas horizontais da imagem mostrada na Figura 3.10(a).

Se a diferença $\Delta J_h[i]$ for menor que $d_h/2$, então uma das linhas, $J_h[i]$ ou $J_h[i+1]$, está mal posicionada e deve ser apagada. O mesmo vale para as linhas verticais. Para decidir qual linha deve ser apagada, usamos uma função de custo que avalia as distâncias entre duas linhas sucessivas. Seja $l$ a função de custo definida em $\mathbf{R}$ por

$$l_z(t) = \begin{cases} 0.5 & \text{se } t \leq \frac{d_z}{2} \\ |\text{round}(\frac{t}{d_z}) - \frac{t}{d_z}| & \text{caso contrário} \end{cases}$$

onde round$(\cdot)$ é o arredondamento para o inteiro mais próximo de $(\cdot)$, $z$ é igual a $h$ ou $v$ se referente às linhas horizontais ou verticais da grade respectivamente. A Figura 3.12 é um exemplo de gráfico da função de custo para $d_z = 10$ e $t \in [1, 100]$.

Para definir a função de custo, supomos que é mais provável que uma linha esteja separada de outra por $kd_z$ pixels, $k$ inteiro. Tais linhas devem ter um custo próximo de zero. Por outro lado, é menos provável que uma linha esteja separada de outra por $k + \frac{1}{2}d_z$, portanto o seu custo é mais alto.

O algoritmo que apaga linhas é executado para todas as linhas $i$ para as quais $\Delta J_z[i-1] < d_z/2$, exceto para a primeira e duas últimas linhas. Supomos que a primeira e a última linhas estão corretas. A penúltima linha, de índice $i-1$ é avaliada em separado: se $\Delta J_z[i-1] < d_z/2$, a tal linha é apagada. O algoritmo recebe os seguintes parâmetros:

- $n_z$ — número de *spots* por fileira.
- $d_z$ — distância média entre linhas.
- $J_z$ — seqüência de posições iniciais das linhas.
- $\Delta J_z$ — seqüência de distâncias iniciais entre as linhas de $J_z$.

No final, temos um novo $J_z$ com o resultado corrigido. O símbolo "\" nas linhas 6 e 8 indica que o elemento $J_z[i]$ da seqüência $J_z[]$ está sendo eliminado.

**Algoritmo 1.** Algoritmo para apagar linhas.

1: $cost_1 \leftarrow \ell(\triangle J_z[i-1])$ ;
2: $cost_3 \leftarrow \ell(\triangle J_z[i+1])$ ;
3: $cost_{12} \leftarrow \ell(\triangle J_z[i-1] + \triangle J_z[i])$ ;
4: $cost_{23} \leftarrow \ell(\triangle J_z[i] + \triangle J_z[i+1])$ ;
5: **if** $(cost_{12} + cost_3) > (cost_1 + cost_{23})$ **then**
6: $\quad J_z[] \leftarrow J_z[] \setminus J_z[i]$
7: **else**
8: $\quad J_z[] \leftarrow J_z[] \setminus J_z[i+1]$
9: **end if**
10: calcule novo $\triangle J_z[]$

O algoritmo verifica se a diferença entre as linhas $J_z[i]$ e $J_z[i-1]$, onde $2 \leq i \leq |J_z| - 2$ é muito pequena, ou seja, se é menor que metade da distância teórica $d_z$ entre as linhas. Em caso afirmativo, ou a linha $i$ ou a linha $i+1$ é eliminada. A escolha é baseada na função de custo $\ell$ aplicada às diferenças $\Delta J_z$. Por exemplo, na Figura 3.11 temos, para $i = 2$, $\Delta J_h[1] = 2$, $\Delta J_h[2] = 54$ e $\Delta J_h[3] = 1$. O valor de $d_h$ é 48,75. Assim, temos $cost_1 = 0,5$; $cost_3 = 0,5$; $cost_{12} = 0,1487$ e $cost_{23} = 0,1282$. Como $(cost_{12} + cost_3) > (cost_1 + cost_{23})$, o algoritmo elimina a linha $J_h[2]$.

Antes de adicionar linhas, o algoritmo verifica se o número de linhas já está certo, ou seja, se já há $n_z + 1$ linhas. O algoritmo insere linhas iterativamente enquanto seu número for menor que $n_z + 1$.

Para inserir uma linha, procuramos o índice $i$ com o maior valor de $\Delta J_z[i]$. Encontrado o índice, consideramos duas formas: uma é inserir algumas linhas entre $J_z[i]$ e $J_z[i+1]$, e a outra é excluir $J_z[i+1]$ e inserir linhas entre $J_z[i]$ e $J_z[i+2]$. Escolhemos a alternativa que produza a menor soma de custos.

As linhas são inseridas sempre uniformemente espaçadas entre si, e o número de linhas é calculado de forma que tais espaços sejam próximos de $d_z$. Ou seja, o algoritmo percebe se entre duas linhas é melhor inserir uma ou mais. O número de linhas a serem inseridas entre:

- $J_z[i]$ e $J_z[i+1]$ é $\text{round}(\Delta_z[i]/d_z) - 1$.
- $J_z[i]$ e $J_z[i+2]$ é $\text{round}((\Delta_z[i] + \Delta_z[i+1])/d_z) - 1$.

O algoritmo também pára quando tenta todos os $i$ sem inserir nenhuma linha.

A Figura 3.10(b) mostra o resultado do procedimento que apaga linhas (a linha apagada aparece em preto e as demais em branco), e a Figura 3.10(c) mostra a correção final após o procedimento que adiciona linhas. Observe que o procedimento apagou uma linha vertical e inseriu duas outras.

As figuras 3.13(a), 3.13(b), 3.14(a) e 3.13(b) mostram o gradeamento morfológico e subseqüente correção de dois outros blocos. Na Figura 3.13(a) estão faltando a segunda linha horizontal e a quarta linha vertical. Na Figura 3.14(a), a terceira linha horizontal está mal posicionada.

Um defeito desse algoritmo é que ele, ao contrário do algoritmo de correção do gradeamento dos blocos, não aproveita a informação do perfil, o que pode ocasionar erros nas grades, especialmente em blocos com muitas fileiras de *spots*.

## 3.5 Segmentação dos *spots*

Este é o último passo da segmentação. Os pinos do robô que imprime a lâmina são cilíndricos, assim, espera-se que as regiões que contêm o sinal de cada *spot* sejam aproximadamente circulares. Porém, devido a diferentes condições físicas e químicas, seja na impressão e secagem da lâmina, seja na hibridização, tais regiões podem acabar não tão regulares. Irregularidades nos contornos e na intensidade dentro da região do *spot* são comuns. Geralmente os programas comerciais assumem que as regiões do sinal são sempre círculos ou elipses perfeitos e o ajuste do raio e forma é feito com muita interação do usuário.

Uma abordagem simples para a segmentação dos *spots* é o operador *top-hat*. Dada a imagem $f$ a segmentação seria dada pela expressão $\tau_c(\iota - \gamma_{B_s})(f)$, onde $\gamma_{B_s}$ é uma abertura por $B_s$, um elemento estruturante em forma de disco de raio aproximadamente igual ao raio do *spot*. E $\tau_c$ é o operador *threshold* pelo nível de cinza $c$.

No entanto, tal abordagem não dá bons resultados. Não é fácil ajustar o parâmetro $c$, e a informação da posição da grade não é aproveitada. Para melhorar a segmentação, usamos um

método que não requeira parâmetros absolutos como o do *threshold*, e que aproveite a informação da grade. Por apresentar tais características, o método escolhido foi o paradigma de Beucher-Meyer, ou *watershed* [18, 19].

Seja a imagem $f$ a composição dos canais da imagem da lâmina a ser segmentada. O procedimento de segmentação recebe como entrada duas imagens: a primeira é o gradiente morfológico da imagem $f$ após alguma filtragem; e a segunda é uma imagem marcadora $m$, binária, composta pelos centros aproximados dos *spots* mais a grade. Assumimos que um *spot* está sempre completamente incluído em um retângulo da grade.

A máscara $m$ é calculada pela seguinte expressão:

$$m = g \vee \varrho_{B_s}(\varrho^{max}_{B_4}(\gamma_{B_n}((\iota - \gamma_{B_s})(f))))$$

onde $g$ é uma imagem com a grade, calculada conforme descrito na última seção. É uma imagem binária com linhas verticais e horizontais com um único *pixel* de espessura, que particiona o bloco em fileiras verticais e horizontais de *spots*. O lado direito da união calcula o centro aproximado de cada *spot*. O elemento estruturante $B_s$ é um círculo com raio aproximadamente igual ao raio médio dos *spots*, e $B_n$ é também um disco de raio igual a um terço do raio do *spot*, usado para filtrar o ruído.

A imagem $f$ é filtrada com uma abertura morfológica que elimina pequenas irregularidades devidas ao ruído. Seja $h = \gamma_{B_n}(f)$ o resultado de tal filtragem, a segmentação dos *spots* é dada por

$$s = \Omega(\nabla_{B_4}(h), m)$$

onde $\Omega$ é o operador *watershed* [20, 21, 22]. e $B_4$ é uma cruz elementar, ou seja, uma cruz $3 \times 3$ centrada na origem.

O resultado desta segmentação é uma imagem com uma máscara que indica quais são os *pixels* do sinal, e é usada na estimação do nível de hibridização. A Figura 3.15 mostra o resultado da

segmentação da Figura 3.9.

O operador *watershed* gerou resultados visualmente muito bons, separando bem o sinal dos *spots*, mesmo os irregulares ou pouco intensos, do *background*.

A Figura 3.16 mostra o resultado final do nosso *software*, com o gradeamento e segmentação de uma lâmina de oligonucleotídeos [11].

A Figura 3.17 mostra uma lâmina de membrana e o resultado de seu gradeamento. A Figura 3.18 mostra um exemplo de segmentação de seus *spots*.

A Figura 3.19 mostra uma lâmina de cDNA onde, apesar de tanto os blocos quanto os *spots* se apresentarem bastante próximos entre si, o gradeamento foi bem sucedido. A Figura 3.20 mostra um exemplo de segmentação de seus *spots*.

A Figura 3.21 mostra a segmentação dos *spots* de um bloco de uma lâmina de genes sintéticos digitalizada a laser. A Figura 3.22 mostra a segmentação dos *spots* de um bloco de uma lâmina digidalizada por CCD (Charge-Coupled Device). Observe a diferença de contraste entre as duas figuras.

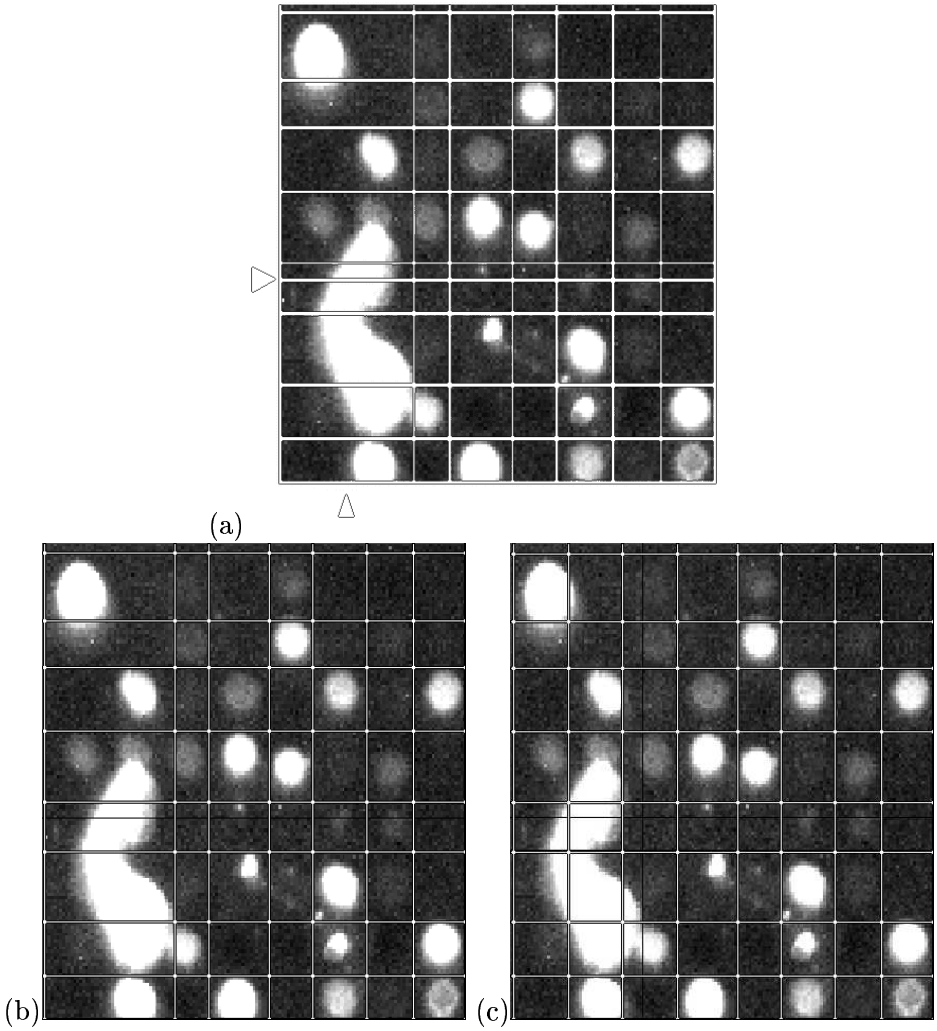



Figura 3.10: (a) Gradeamento morfológico do bloco 6 da lâmina da Figura 2.5. (b) Apagando linhas da grade. (c) Adicionando linhas à grade.

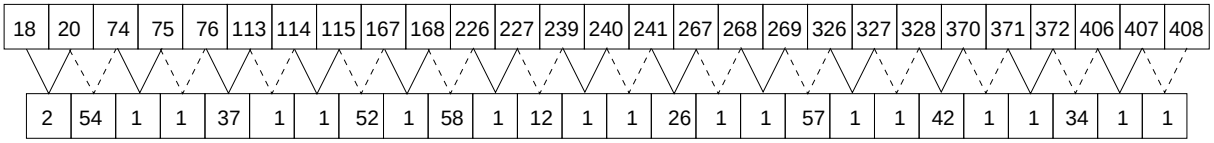



Figura 3.11: $J_h$ e $\Delta J_h$ da Figura 3.10(a)

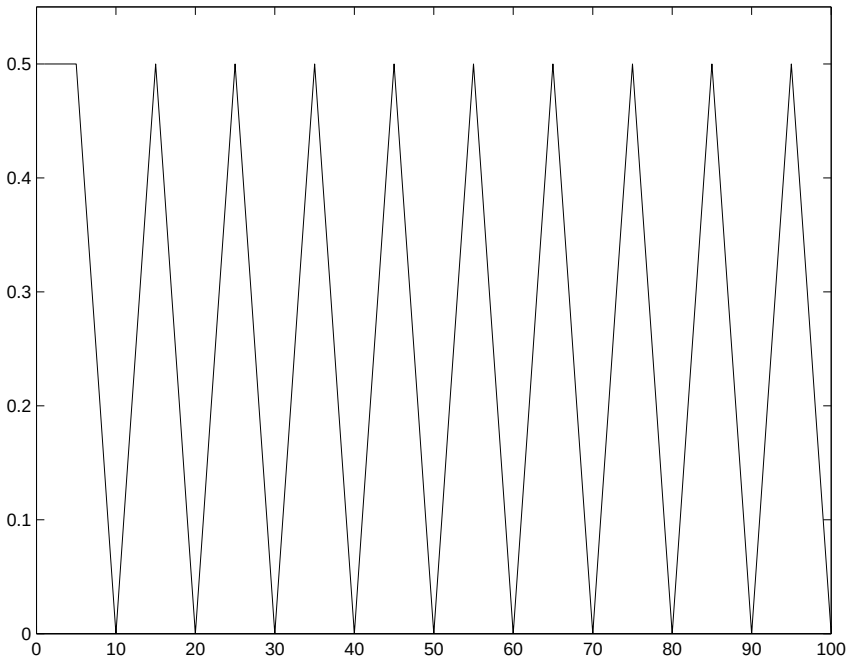
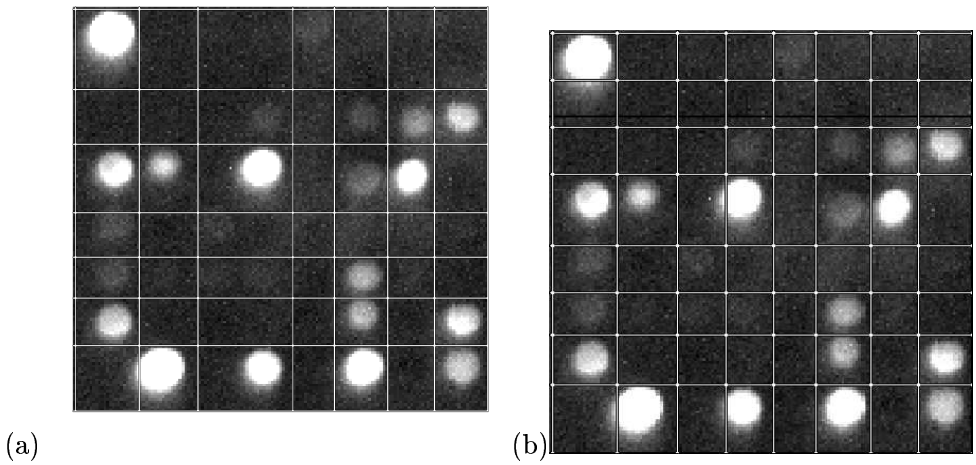
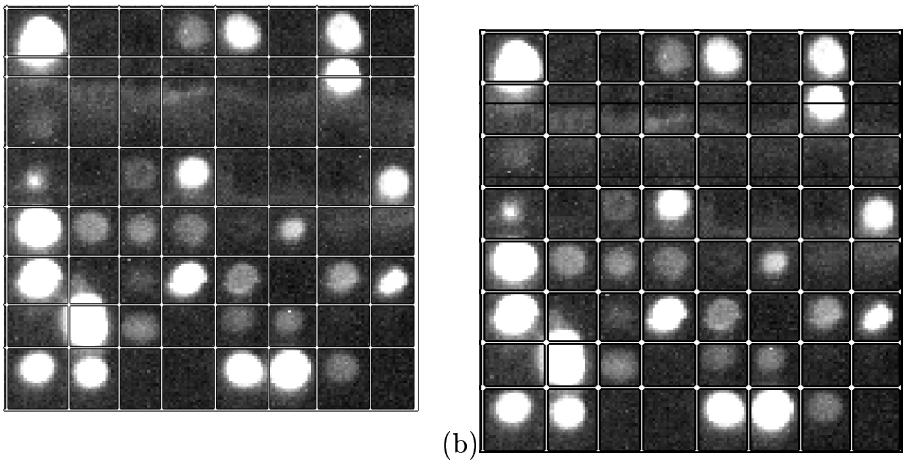



Figura 3.12: Gráfico da função de custo para $d_z = 10$ e $t \in [1, 100]$



(a) (b)

Figura 3.13: (a) Gradeamento morfológico do bloco 4 (linha 1, coluna 4) da lâmina da Figura 2.5. (b) Correção final.



(a) (b)

Figura 3.14: (a) Gradeamento morfológico do bloco 11 (linha 3, coluna 3) da lâmina da Figura 2.5. (b) Correção final.

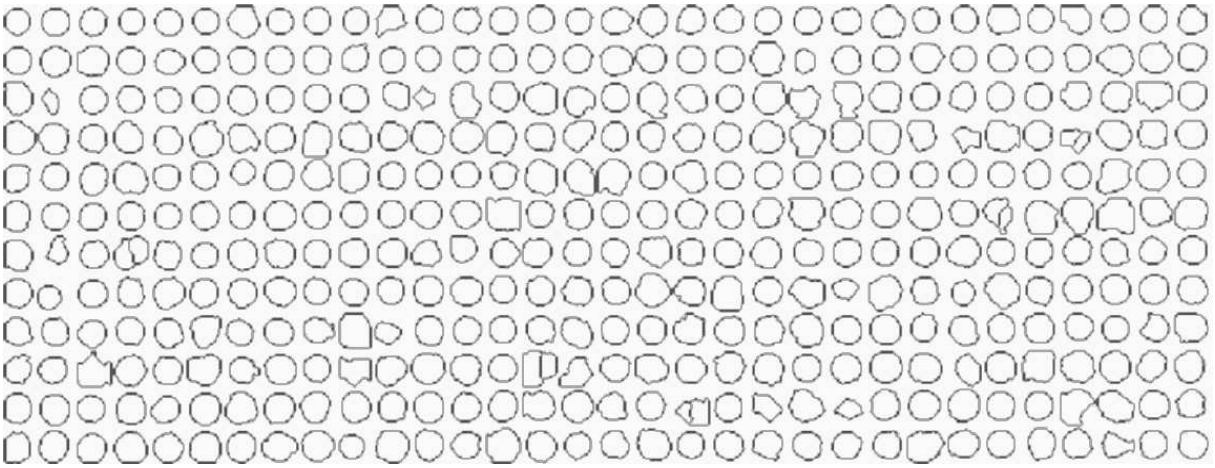


Figura 3.15: Contorno dos *spots* da Figura 3.9.

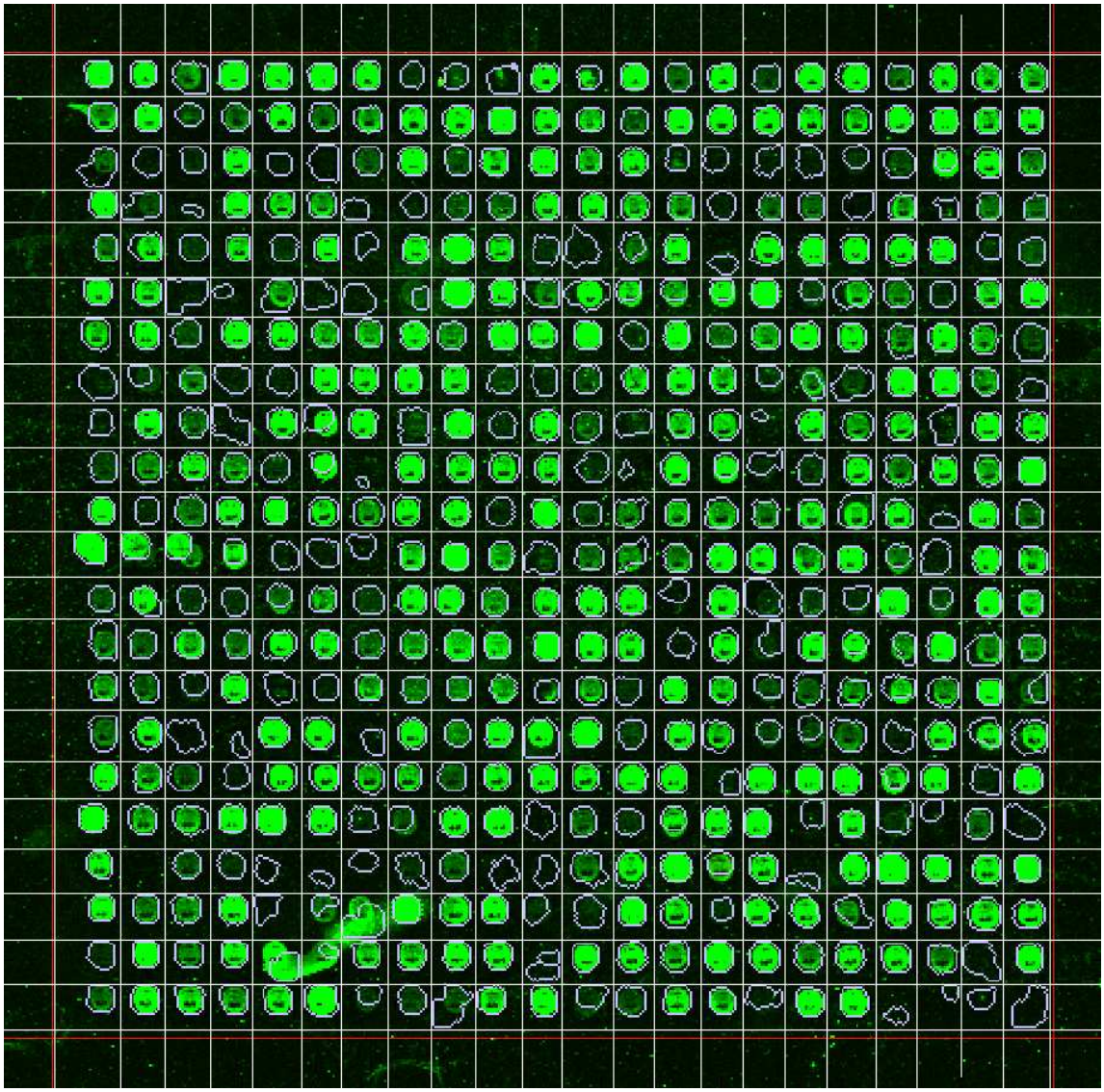


Figura 3.16: Exemplo de segmentação de *spots* de lâmina de oligonucleotídeos (mos13-083 extraída da página `http://derisilab.ucsf.edu/falciparum/`).

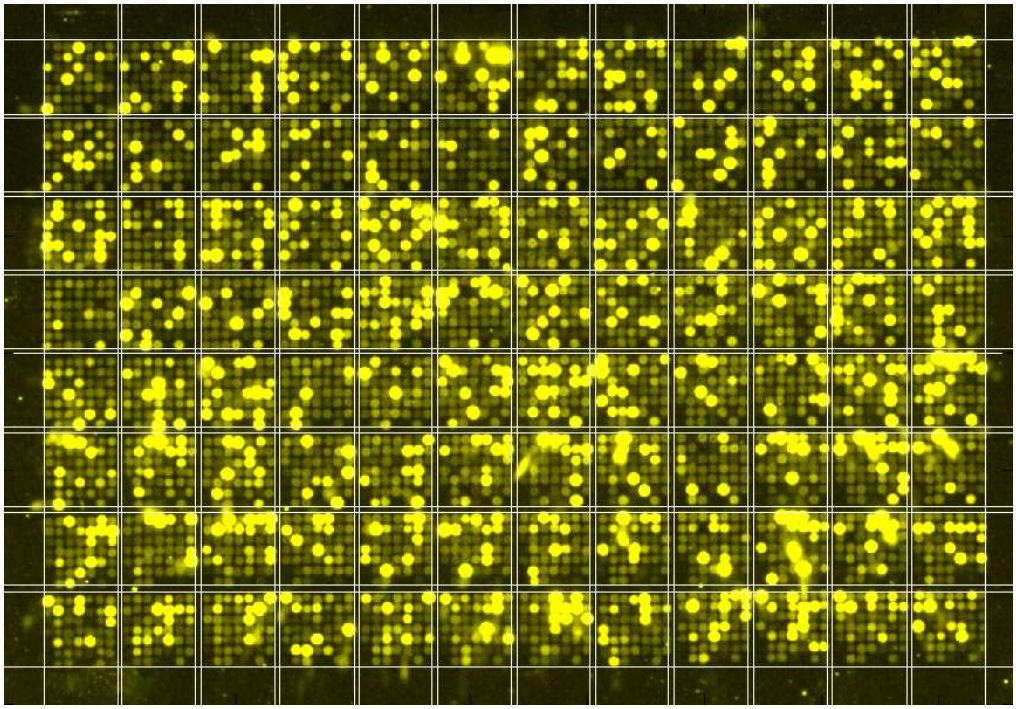


Figura 3.17: Exemplo de gradeamento de membrana (lâmina produzida pela pesquisadora Maria Aparecida Nagai, Departamento de Radiologia, FM-USP).

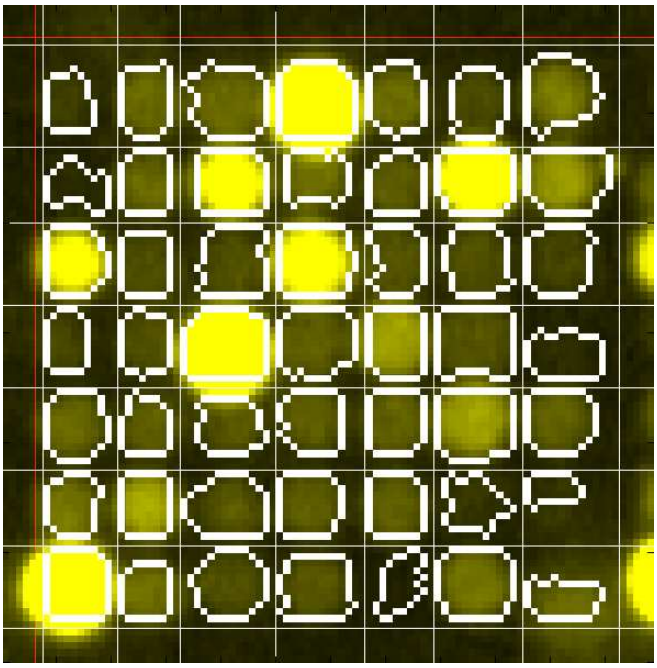


Figura 3.18: Segmentação do primeiro bloco da Figura 3.17

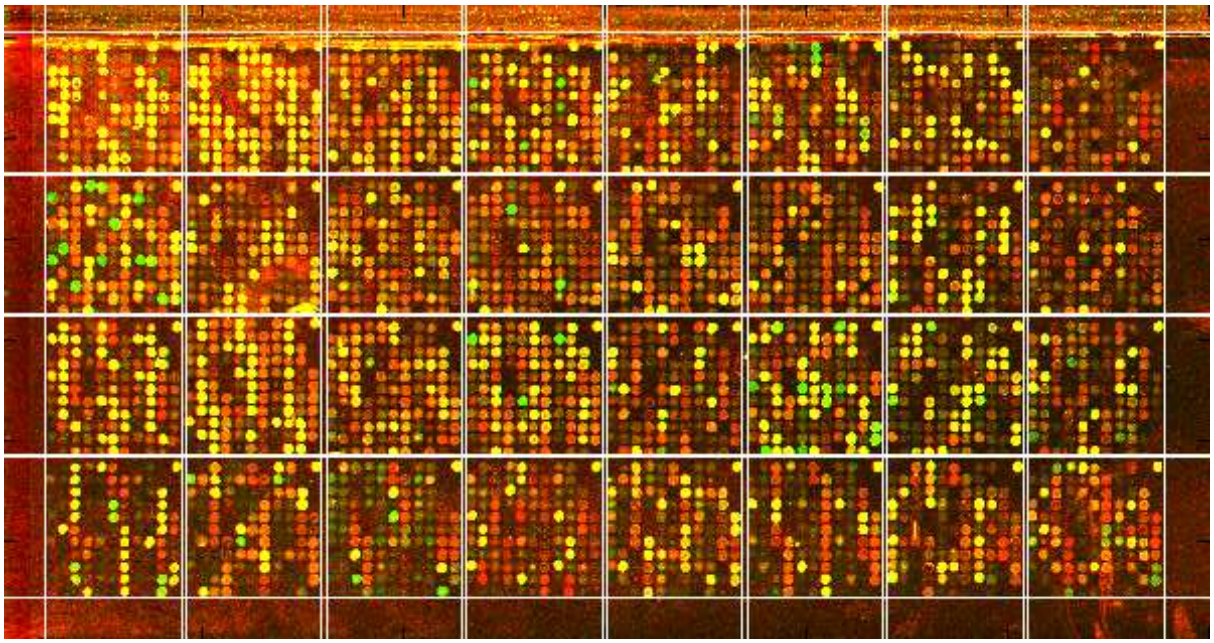


Figura 3.19: Exemplo de gradeamento de lâmina de cDNA de experimento de câncer digitalizada a laser (lâmina produzida pela pesquisadora Helena Paula Brentani, Instituto Ludwig).

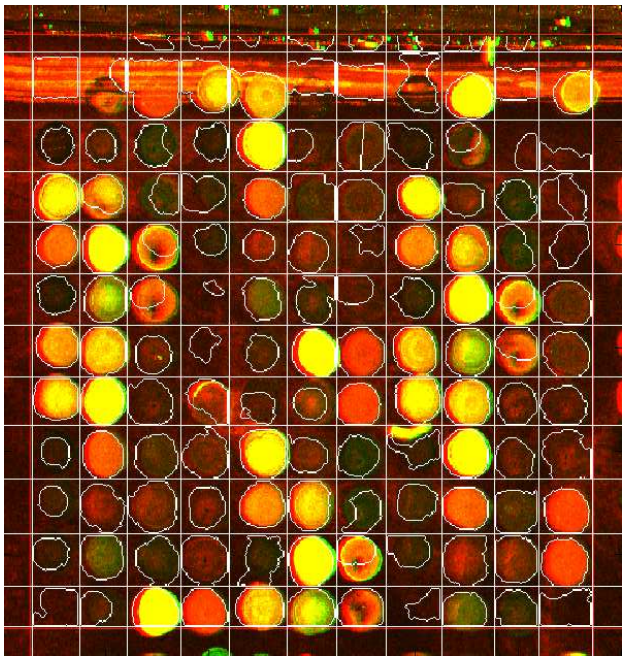


Figura 3.20: Segmentação do primeiro bloco da Figura 3.19

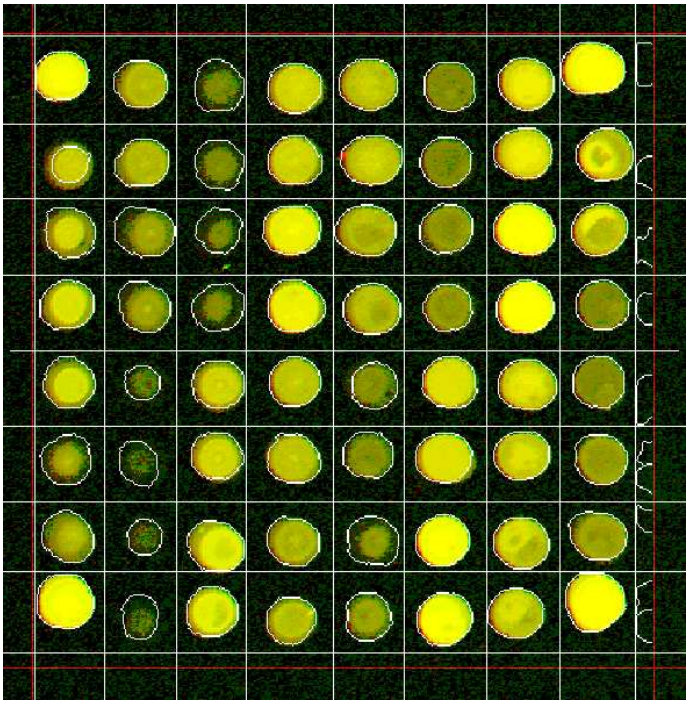


Figura 3.21: Segmentação do primeiro bloco da lâmina de genes sintéticos da Figura 2.3

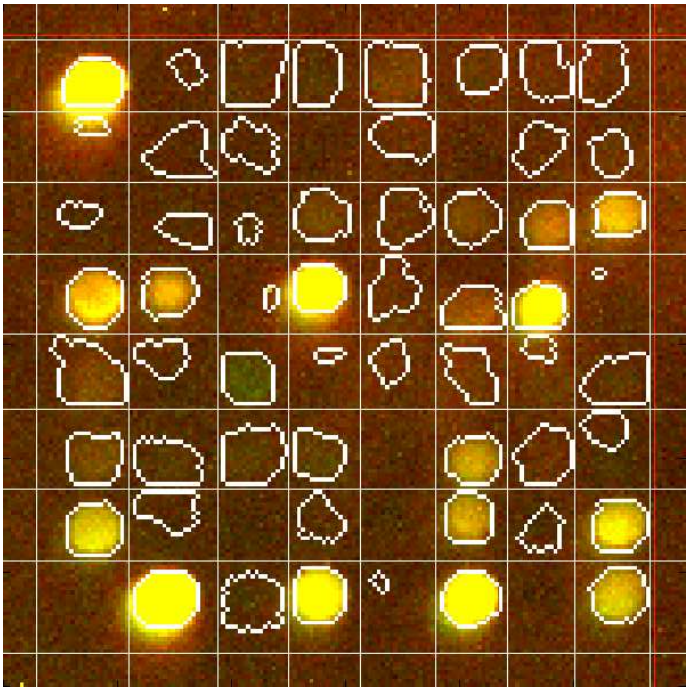


Figura 3.22: Segmentação do bloco da Figura 3.13

# Capítulo 4

# Modelos e medida de expressão gênica

## 4.1 Introdução

Neste Capítulo alguns modelos de expressão gênica, que mostram a relação entre seu valor real e o valor observado na lâmina, são apresentados. Em seguida são mostradas algumas das formas de medida do sinal encontradas na literatura e nas soluções comerciais, e a influência do *background* no valor do sinal é discutida.

## 4.2 Modelos de expressão gênica

### 4.2.1 Modelo linear

O modelo linear já foi apresentado na Seção 2.4. É um modelo simplificado cuja fórmula, $Y_{ik} = a_i + b_i X_{ik}$, não apresenta nenhuma variável aleatória que represente o ruído. Na Seção 4.3 são apresentadas várias formas de estimativa de valor do sinal e do *background* que seguem esse modelo.

### 4.2.2 Modelo de ruído aditivo

A intensidade média do $k$-ésimo *spot* ($\bar{r}_k^m$ no canal vermelho e $\bar{g}_k^m$ no canal verde) consiste do sinal fluorescente do cDNA hibridizado na lâmina ($\bar{r}_k$ e $\bar{g}_k$), ruído proveniente de hibridização não específica ($\bar{r}_k^b$ e $\bar{g}_k^b$) e variações de intensidade ($r_k^e$ e $g_k^e$), causadas por ruído da etapa de hibridização, digitalização etc:

$$\bar{r}_k^m = \bar{r}_k + \bar{r}_k^b \pm r_k^e$$

$$\bar{g}_k^m = \bar{g}_k + \bar{g}_k^b \pm r_k^e$$

A expressão relativa do $k$-ésimo *spot* é então:

$$Z_k = c\frac{\bar{r}_k + \bar{r}_k^b \pm r_k^e}{\bar{g}_k + \bar{g}_k^b \pm r_k^e}$$

Na maioria dos experimentos de *microarray*, por exemplo nos que contêm milhares de *spots* e onde se espera que haja expressão diferencial em poucos genes, podemos assumir que a expressão relativa média $\bar{Z} = 1/n \sum_{k=1}^{n} Z_k \approx 1$, onde $n$ é o número de *spots* na lâmina. Numa lâmina pequena, porém, cujos genes são escolhidos justamente por serem diferencialmente expressos, tal hipótese não vale e genes de controle devem ser usados. Com isso, podemos estimar a variável $c$ usando um método de regressão robusta [7].

### 4.2.3 Modelo de ruído exponencial

Esse modelo permite que intervalos de confiança para a expressão gênica sejam calculados, além de prever dois tipos de ruído. O ruído aditivo, sempre presente e notado especialmente nos *spots* com valores próximos a zero, é representado por $\epsilon$; e um ruído proporcional ao nível do sinal, representado por $e^{\eta}$, observável especialmente nos *spots* mais intensos [23].

O modelo é semelhante ao linear, com as componentes do erro inseridas na fórmula:

$$Y_{ik} = a_i + X_{ik}e^{\eta_i} + \epsilon_i$$

onde o índice $k$ representa cada gene e $i$, cada canal. A variável $Y$ é a intensidade observada, $X$ é o nível de expressão em unidades arbitrárias, e $a$ é o *background* médio, ou seja, média dos genes pouco ou não expressos. Nosso melhor estimador de $X$ é $Y - \hat{a}$, ou seja, a intensidade observada corrigida pela subtração do *background*.

Note que, nesse modelo, apenas as regiões com *spots* são consideradas e o *background* é calculado usando controles negativos ou genes pouco expressos. As regiões externas aos *spots* são ignoradas.

Assume-se a normalidade dos termos $\epsilon$ e $\eta$ por conveniência, o que é na prática uma suposição razoável. Assim, o ruído aditivo é $\epsilon \sim N(0, \sigma_\epsilon)$, que representa o ruído observado nos genes pouco expressos. O outro termo é $\eta \sim N(0, \sigma_\eta)$, que representa o ruído mais facilmente observável nos *spots* mais intensos.

Em [14], um modelo equivalente é apresentado, e dados reais são comparados com dados simulados.

Três formas de estimação do *background* são consideradas e explicadas nas subseções seguintes.

### Estimação do *background* usando controles negativos

A forma mais fácil de estimar o valor de $a$ médio e de $\sigma_\epsilon$ é através de réplicas de controles negativos. O desvio padrão dos controles negativos pode ser usado como estimador de $\sigma_\epsilon$, e sua intensidade média como estimador do *background* médio $a$.

### Estimação do *background* com genes replicados

Mesmo se tivermos medidas replicadas, sem ter necessariamente controles negativos, podemos estimar a média e desvio padrão do *background* para cada canal. O algoritmo é iterativo e

compreende os seguintes passos:

- (i) Comece com um conjunto de genes com baixa intensidade, por exemplo, os 10% dos genes menos intensos. Calcule a média $\bar{x}_B$ de todas as réplicas dos genes escolhidos, e o seu desvio padrão amostral $s_B$. Para cada gene no grupo, compute o desvio padrão das réplicas $s_j$. Se o número de genes replicados é $m$, calcule

$$s_B = \sqrt{\frac{1}{n-m}\sum_{j=1}^{m} s_j^2(n_j - 1)}$$

  onde $n_j$ é o número de réplicas do gene $j$ e $n = \sum_{j=1}^{m} n_j$.
- (ii) Defina um novo subconjunto como os genes cujos valores de intensidade estejam no intervalo $[\bar{x}_B - 2s_B, \bar{x}_B + 2s_B]$ e recalcule $\bar{x}_B$ e $2s_B$.
- (iii) Repita o passo anterior até que o conjunto de genes não mude.

**Estimação do *background* sem genes replicados**

Caso não haja réplicas dos genes, podemos usar o procedimento abaixo para estimar a média e desvio padrão do *background*.

- (i) Comece com um conjunto de genes com baixa intensidade, por exemplo, os 10% dos genes menos intensos. Calcule a média $\bar{x}_B$ de suas intensidades, e o seu desvio padrão $s_B$.
- (ii) Defina um novo subconjunto como os genes cujos valores de intensidade estejam no intervalo $[\bar{x}_B - 2s_B, \bar{x}_B + 2s_B]$ e recalcule $\bar{x}_B$ e $2s_B$.
- (iii) Repita o passo anterior até que o conjunto de genes não mude.

## 4.3 Medida da expressão gênica

Nesta Seção descreveremos algumas das técnicas de estimação da expressão gênica mais usadas pelas soluções disponíveis no mercado.

A fase de segmentação retorna, para cada *spot*, uma partição da sua região de interesse, que se divide em sinal e *background*. O resultado, no entanto, pode variar enormemente dependendo da forma que usamos tal informação, se usamos todos ou apenas alguns dos *pixels* do sinal, ou se calculamos a expressão relativa pela inclinação da nuvem de pontos formada pelo gráfico dos valores dos *pixels* em um canal versus o outro canal.

### 4.3.1 Histograma

Este método calcula o histograma dos valores dos *pixels* na região de interesse do *spot* e usa quatro parâmetros para quantificar o valor do *spot*:

- Mínimo do sinal: Menor percentil dos valores dos *pixels* a ser usado para o cálculo do valor do sinal.
- Máximo do sinal: Maior percentil a ser usado para o cálculo do valor do sinal.
- Mínimo do *background*: Menor percentil a ser usado para o cálculo do valor do *background*.
- Máximo do *background*: Maior percentil a ser usado para o cálculo do valor do *background*.

Esses dois intervalos são calculados nos dois canais. Os valores do sinal e do *background* podem ser obtidos pela média, mediana ou moda dos valores dos *pixels* no intervalo. Como esse método usa a mesma máscara para o sinal e o *background*, a estimativa pode ser afetada por *pixels* que tenham intensidade próxima ao do sinal e que não estejam agrupados com este. Esse método é implementado no Quantarray [24].

Observe que, para esse método funcionar, o *background* deve ser menor que o sinal em todos os *spots*, o que normalmente não ocorre no caso de genes usados como controles negativos.

### 4.3.2 Círculo fixo

Usa como máscara do sinal um círculo que tem o mesmo tamanho em todos os *spots*. A máscara do *background* é escolhida de várias formas. O Quantarray [24] usa uma coroa de círculo circunscrita ao círculo do sinal. O ScanAlyze [25] usa um retângulo circunscrito ao círculo do sinal, sendo excluídos os *pixels* que fazem parte do sinal dos *spots*. O CSIRO Spot [26] usa quadrados

rotacionados de quarenta e cinco graus posicionados no centro de cada quadrado formado pelo centro de quatro *spots* adjacentes. A Figura 4.1 ilustra as diferentes formas.

Os valores do sinal e *background* são então obtidos pela média, moda ou mediana dos valores dos *pixels* na região obtida. Uma variação desse método é a média aparada, que ignora percentis extremos na estimativa do sinal e do *background* visando eliminar a influência de *outlyers*.

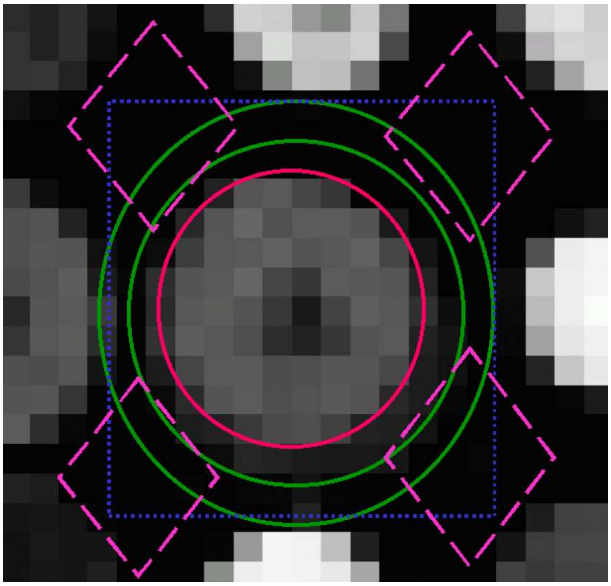


Figura 4.1: Diferentes formas de segmentação do *background*. A região delimitada pelo círculo mais interno representa o sinal. As outras regiões representam as diferentes formas de delimitar o *background*: os outros dois círculos em linha cheia são usados pelo Quantarray, o quadrado pontilhado pelo ScanAlyze, e os quatro quadrados tracejados pelo CSIRO Spot [2].

### 4.3.3 Adaptativo

Método implementado no Quantarray, usa a mesma máscara descrita em 4.3.2. Oito *pixels* do sinal e oito do *background* são escolhidos e é feito um teste estatístico (Mann-Whitney) que retorna um p-valor que representa a probabilidade de que os *pixels* das duas amostras pertençam a populações diferentes. O teste é iterado com *pixels* do sinal cada vez mais brilhantes até que o teste encontre uma diferença significativa entre os *pixels* do *background* e do sinal, ou seja, até que um p-valor menor que o definido pelo usuário seja encontrado [27].

### 4.3.4 Regressão

Esse método tem a vantagem de não necessitar de uma estimativa do *background* para encontrar a expressão relativa. Consiste em calcular a reta que melhor ajusta os valores dos *pixels* de um canal pelo outro. Deve ser usada a regressão robusta, que minimiza a distância entre os pontos e a reta e dá o mesmo valor independentemente da escolha do canal para ser a abcissa ou ordenada. Esse método é implementado no UCSF Spot [28].

### 4.3.5 Segmentação morfológica

Podem-se usar técnicas morfológicas de segmentação para separar o sinal do *background*. A vantagem sobre o círculo fixo é que este considera *pixels* do *background* como sendo do sinal e vice-versa, o que pode adicionar ruído ao valor da expressão.

A abordagem do CSIRO Spot [26] é por crescimento de regiões (*seeded region growing*). A nossa abordagem é pelo paradigma de Beucher-Meyer, conhecido por *watershed*.

Em ambos os casos, o sinal e o *background* podem ser estimados pela média ou mediana. O nosso programa permite que o usuário escolha os percentis de onde será tirada a estatística.

## 4.4 Correção do *background*

Uma vez estimado o *background*, pode-se ou não fazer o ajuste. A forma mais simples e também a mais usada consiste em subtrair o valor do *background* de cada *spot*, $b_k$, do seu valor de sinal, $f_k$. Seja $c_k$ o valor do *spot* $k$ corrigido, $c_k = f_k - b_k$.

Porém, caso o *background* seja local, isto é, estimado a partir dos *pixels* da vizinhança do *spot*, é comum que $c_k$ tenha valores negativos, especialmente quando o *probe* do *spot* $k$ não tem seu *target* correspondente na amostra, como é o caso dos controles negativos. Tal fenômeno pode ser evitado utilizando-se outras técnicas de estimativa do *background*, que levem em conta os controles negativos, ou que não estimem o valor do *background* acima do valor do sinal para muitos *spots*, como é o caso dos métodos apresentados na Seção 4.2.3.

Outro método de correção consiste em, quando um valor de $c_k$ for negativo, defini-lo como um inteiro positivo $\delta$ para evitar expressões relativas impossíveis:

$$c_k = \begin{cases} f_k - b_k & \text{se} f_k - b_k > \delta \\ \delta & \text{se} f_k - b_k \leq \delta \end{cases}$$

É sugerido por Chen et al. [29] usar um valor de $\delta$ igual a 1. Há uma versão mais sofisticada desse método [30] que substitui o $\delta$ do segundo caso por uma função monotônica derivável, linear na escala logarímica em função de $b_k$.

$$c_k = \begin{cases} f_k - b_k & \text{se} f_k - b_k > \delta \\ \delta \exp[1 - (b_k + \delta)/f_k] & \text{se} f_k - b_k \leq \delta \end{cases}$$

A idéia por trás dessa fórmula é permitir que *spots* com expressão muito baixa não sejam considerados todos com o mesmo nível de expressão, $\delta$, e evitar que *spots* com expressão muito baixa em ambos os canais tenham expressão relativa igual a 1 quando seus níveis de expressão são diferentes, como acontece no método anterior [30].

## 4.5 Influência do *background* no sinal

No contexto de lâminas de *microarray*, *background* se contrapõe a sinal, ou seja, é a parte da lâmina que não possui cDNA impresso e que, portanto, todo sinal é proveniente dos *targets* que aderiram ao substrato.

Na Seção 4.3 foram descritas várias formas de estimar o *background* que, supostamente, deve ser subtraído do valor do sinal como forma de eliminar a hibridização não específica. Yang et al. [2] compara os diversos métodos de cálculo do *background* e os classifica em quatro grupos.

- (1) *Background* local: o *background* é estimado para cada *spot* usando pequenas regiões que o rodeiam. É implementado na grande maioria dos programas no mercado.
- (2) Abertura morfológica: aplica-se uma abertura morfológica em toda a imagem usando

um elemento estruturante quadrado de lado maior ou igual à distância entre os centros dos *spots*. O valor do *background* estimado de um *spot* é igual ao valor do *pixel* do seu centro na imagem resultante. Dá uma estimativa menor que a dos outros métodos e não corrompida por *pixels* mais brilhantes.

- (3) *Background* constante: método global que subtrai o mesmo valor de *background* de todos os *spots*.
- (4) Sem ajuste: a possibilidade de não se fazer nenhuma correção de *background* também é considerada.

O método do *background* local tende a ser ruidoso, aumentando o desvio padrão das medidas de razão. O método de *background* constante teve o mesmo efeito. Já a ausência de ajuste reduz a habilidade de identificar genes diferencialmente expressos. A abertura morfológica é o método dentre os quatro que apresenta o melhor balanço entre variabilidade das medidas e habilidade de identificar genes diferencialmente expressos. Uma estimativa bem sucedida do valor do *background* deve ter valor maior que zero e menor que o do *spot* menos intenso da lâmina [2].

Além disso, o método do *background* local, largamente usado e implementado em diversos programas de análise de imagens, costuma resultar em valores de *background* maiores que o sinal para uma parcela não desprezível dos *spots*, e quando a correção é aplicada, tais *spots* ficam com intensidade negativa, o que sugere que o *background* deveria ser medido de outra forma.

Quando tais *spots* são observados cuidadosamente, pode-se notar que a região do sinal tem intensidade menor que o *background* correspondente, pois mais *target* adere à região ao redor do *spot* que ao *spot* em si. Dessa forma, pode-se concluir que a hibridização não específica dos *targets* ao substrato segue um modelo diferente da sua hibridização não específica aos *probes* presentes na lâmina.

# Capítulo 5

# O *software* desenvolvido

## 5.1 Introdução

Aqui descreveremos o *software* desenvolvido para automatizar o processo de segmentação e cálculo da expressão gênica a partir das imagens de *microarray*.

A maior parte do algoritmo já foi descrito no capítulo 3, sobre a segmentação. Resta descrever a interface com o usuário, e a interface com outros programas ou possíveis módulos que realizem outras partes do processo, ou *pipeline* [31].

O *software* foi projetado para que seja um módulo que possa ser conectado com facilidade a outros que façam parte do *pipeline*. Por isso, além da interface com o usuário, pode ser chamado como parte de um processo de lote e a saída e entrada podem ser feitas através de arquivos de texto com formato padronizado, que podem ser lidos ou criados automaticamente por outros processos.

A interface foi implementada em língua inglesa por ser o protótipo de um produto que visa o mercado internacional.

O arquivo de saída, uma matriz cujos dados são separados por tabulações, pode ainda ser facilmente importado por planilhas eletrônicas ou programas estatísticos.

## 5.2 Interface com o usuário

O protótipo foi totalmente implementado no MATLAB™[1], usando algumas rotinas de morfologia matemática da biblioteca SDC Morphology Toolbox[2].

Quando usado pela primeira vez, o usuário deve indicar a localização de um par de imagens de *microarray*, uma para cada canal, e, opcionalmente, um arquivo que tenha os parâmetros da respectiva família de experimentos, de extensão ".glo".

O programa, quando fechado, salva seu estado atual, que inclui o nome dos arquivos das imagens, parâmetros de segmentação e seus resultados, se houver, em um arquivo de experimento, de extensão ".exp". Reiniciado, este estado é recuperado e o usuário pode continuar o processo do ponto onde parou.

### 5.2.1 Interface principal

A interface principal (Figura 5.1) apresenta a imagem da lâmina em cor falsa, e o resultado da segmentação dos blocos. A interface é simples, procurando ser amigável. Os diversos parâmetros de segmentação, que são calculados a partir das distâncias entre blocos, *spots* e seu diâmetro esperado, são transparentes ao usuário, que precisa fornecer apenas os parâmetros de geometria da lâmina.

Clicando no botão "Start segmentation", o programa tenta começar a segmentação dos blocos, que consiste em calcular a posição das suas linhas de fronteira com o *background*. Caso esteja faltando algum parâmetro de geometria, o programa pára e pede ao usuário que insira tal parâmetro.

[1] www.mathworks.com
[2] www.mmorph.com

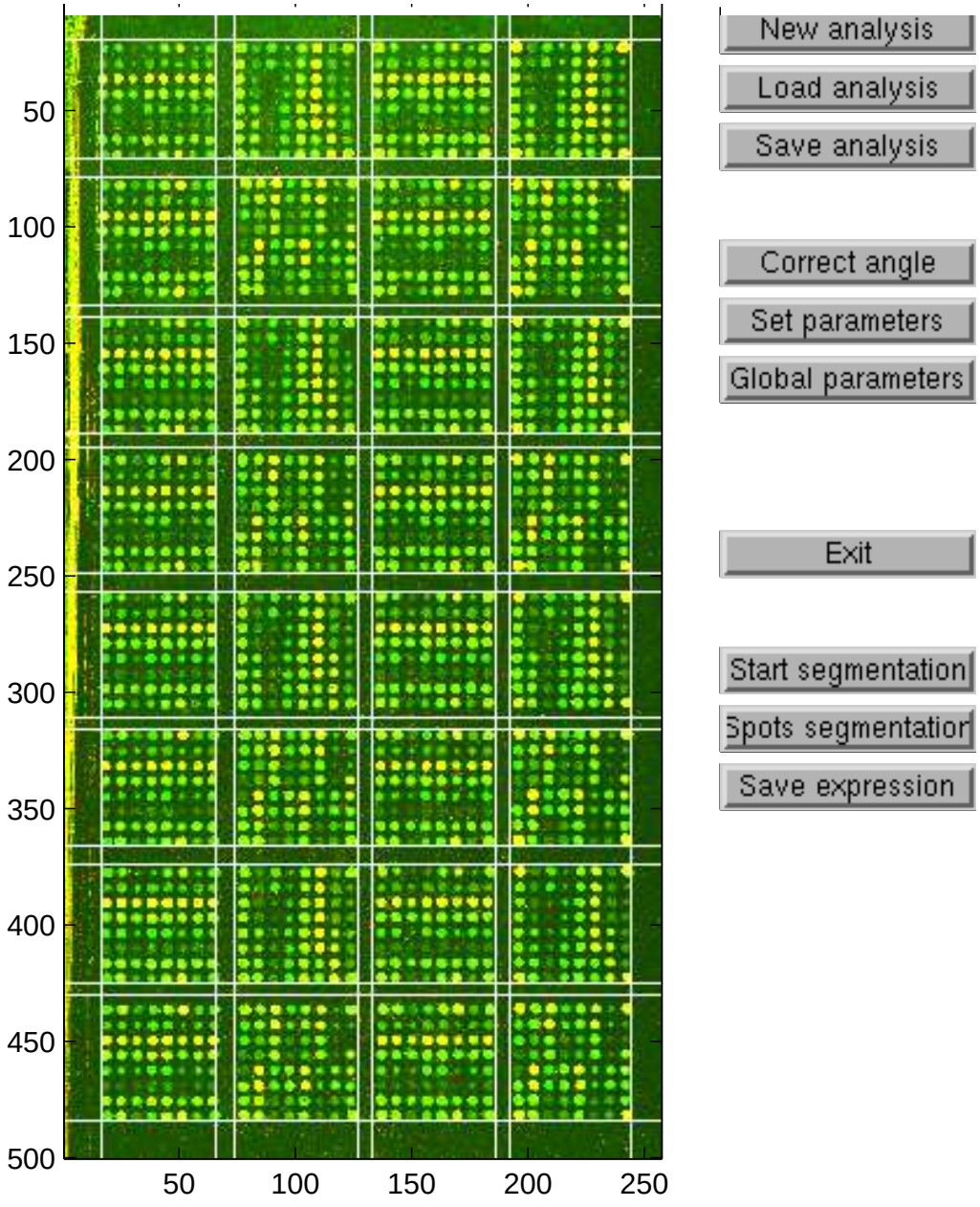



Figura 5.1: Interface principal do programa com a segmentação dos blocos terminada.

Clicando com o botão direito do *mouse* sobre a imagem, um pequeno menu é mostrado com algumas opções. Tais opções incluem a inserção ou exclusão de linhas de fronteira dos blocos, mudança do tamanho da janela e acesso à janela de análise individual do bloco.

Terminada a segmentação dos blocos, tendo o usuário feito alguma eventual correção manual na posição de alguma linha mal posicionada, e estando satisfeito com o resultado, pode começar a segmentação dos *spots* em modo automático clicando em "Spots segmentation". O usuário pode fazer o processo manualmente, bloco a bloco, clicando no bloco com o botão direito e selecionando a opção "Process block". Tal opção também pode ser usada ao final da segmentação automática dos *spots* se o usuário quiser inspecionar os resultados e recalcular algum bloco que apresente erros.

Por fim, quando toda a segmentação tiver sido completada, um arquivo com os dados de expressão de cada *spot* pode ser gerado clicando no botão "Save expression". Tal arquivo tem várias colunas com dados de posição do *spot* na lâmina, a estimativa de sua intensidade e do seu *background* em cada canal, alguns indicadores de qualidade, e outras medidas, descritas na seção 5.3.

Caso a lâmina não tenha sido digitalizada no mesmo ângulo em que foi impressa, os *spots* não estarão alinhados paralelamente às bordas da imagem, o que é necessário para a segmentação, já que as linhas das grades, tanto da segmentação dos blocos quanto dos *spots*, são paralelas. Tal inclinação causa problemas quando a projeção de algum *spot* está muito próxima ou tem interseção com a projeção de algum *spot* de outro bloco e os mesmos devem estar separados pela grade dos blocos. Nesse caso, a grade corre o risco de deixar parte do sinal como *background* e a rotação deve ser corrigida.

Caso isso ocorra, o usuário deve corrigir a rotação através do botão "Correct angle". Aqui o usuário seleciona dois pontos que deveriam estar alinhados e o programa desloca os *pixels* da imagem, sem alterar seus valores, de modo que os tais pontos se alinhem.

Os botões "Set parameters" e "Global parameters", que serão descritos respectivamente nas seções 5.2.2 e 5.2.3 levam a janelas onde o usuário ajusta parâmetros de visualização e configuração geométrica da lâmina.

Para analisar uma nova lâmina, o usuário deve clicar no botão "New analysis" e lhe serão pedidos os nomes das imagens e, opcionalmente, o nome de um arquivo ".glo", que contém os respectivos parâmetros de geometria.

O usuário também pode salvar um experimento a qualquer momento em um arquivo de experimento para fazer outras análises e voltar ao primeiro posteriormente. Tais ações estão disponíveis na interface principal através dos botões "Load analysis" e "Save analysis".

### 5.2.2 Interface de parâmetros específicos

Esta janela (Figura 5.2) está disponível através do botão "Set parameters" da interface principal. Através desta janela o usuário pode ajustar alguns parâmetros específicos da lâmina. Pode ativar a equalização de histograma, mudar os pontos de saturação mínimo e máximo ou aumentar o brilho da imagem para que possa visualizar e comparar *spots* de todas as intensidades. Note que esses ajustes afetam apenas a visualização, não influindo nos valores reais dos *pixels* ou no resultado.

Aqui o usuário também pode definir uma região de interesse onde se encontram todos os blocos e *spots*, eliminando manualmente o ruído da borda, ou definir o ângulo de rotação da imagem.

Para maior conveniência do usuário, todas as alterações são descartadas caso se pressione o botão "Cancel". Pressionando "Ok" as alterações são salvas e retorna-se à janela principal.

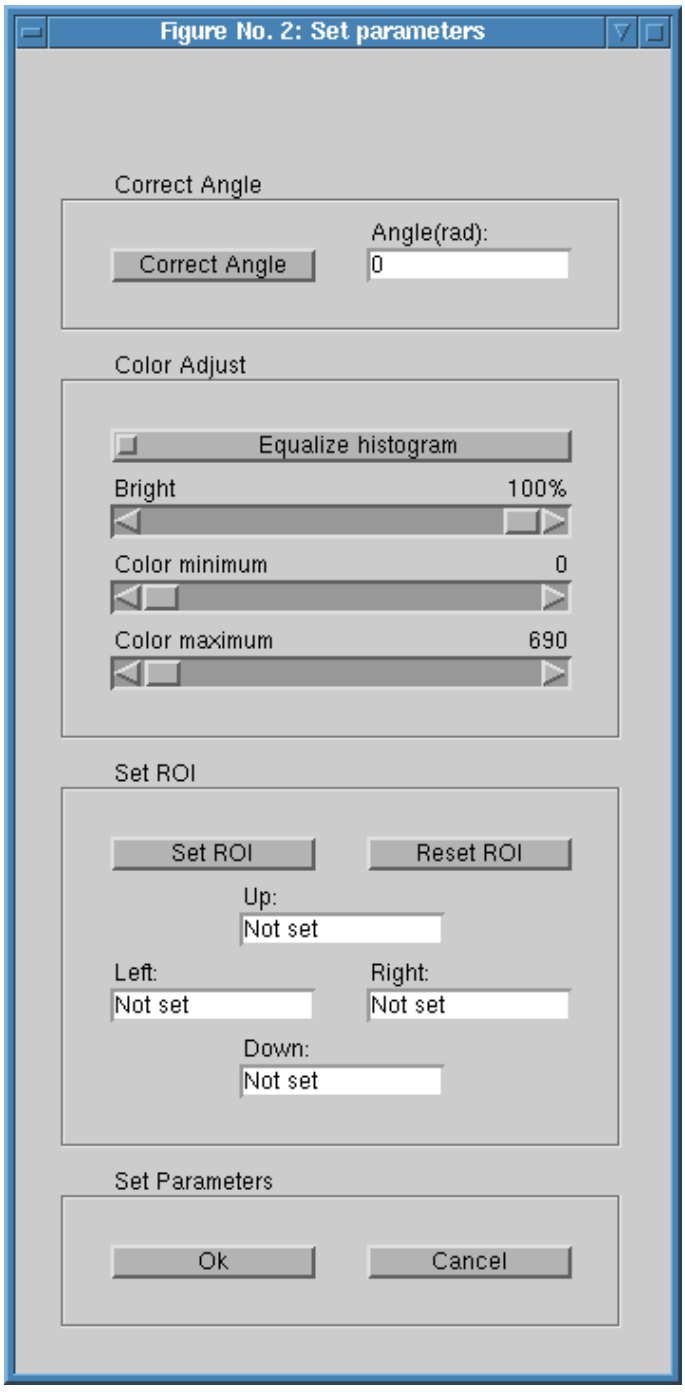



Figura 5.2: Interface de parâmetros específicos da lâmina.

### 5.2.3 Interface de parâmetros globais

Esta janela (Figura 5.3) aparece ao se pressionar o botão "Global parameters" da interface principal. Nesta o usuário define os parâmetros geométricos da lâmina. São os nove parâmetros necessários para segmentar a lâmina automaticamente: distâncias horizontal e vertical entre os blocos, distâncias horizontal e vertical entre os *spots*, número de linhas e colunas de blocos, número de linhas e colunas de *spots* por bloco e diâmetro aproximado do *spot*.

Ao pressionar o botão "Output options", uma pequena caixa de diálogo aparece para que o usuário defina os parâmetros de cálculo da expressão dos métodos do histograma e da região fixa. Tais parâmetros são os percentis que serão considerados para a estimação do sinal e do *background*. Para a definição dos percentis, os valores dos *pixels* podem ser ordenados de duas formas, independentemente nos dois canais, ou segundo o valor da distância *city-block* até a origem.

Para maior conveniência do usuário, todas as alterações são descartadas caso se pressione o botão "Cancel". Pressionando "Ok" as alterações são salvas e retorna-se à janela principal.

Nesta janela, o usuário também pode gravar os parâmetros em um arquivo ".glo" para uso posterior em outras lâminas do mesmo lote de experimentos. A gravação é feita pelo botão "Save" e o carregamento, pelo botão "Load".

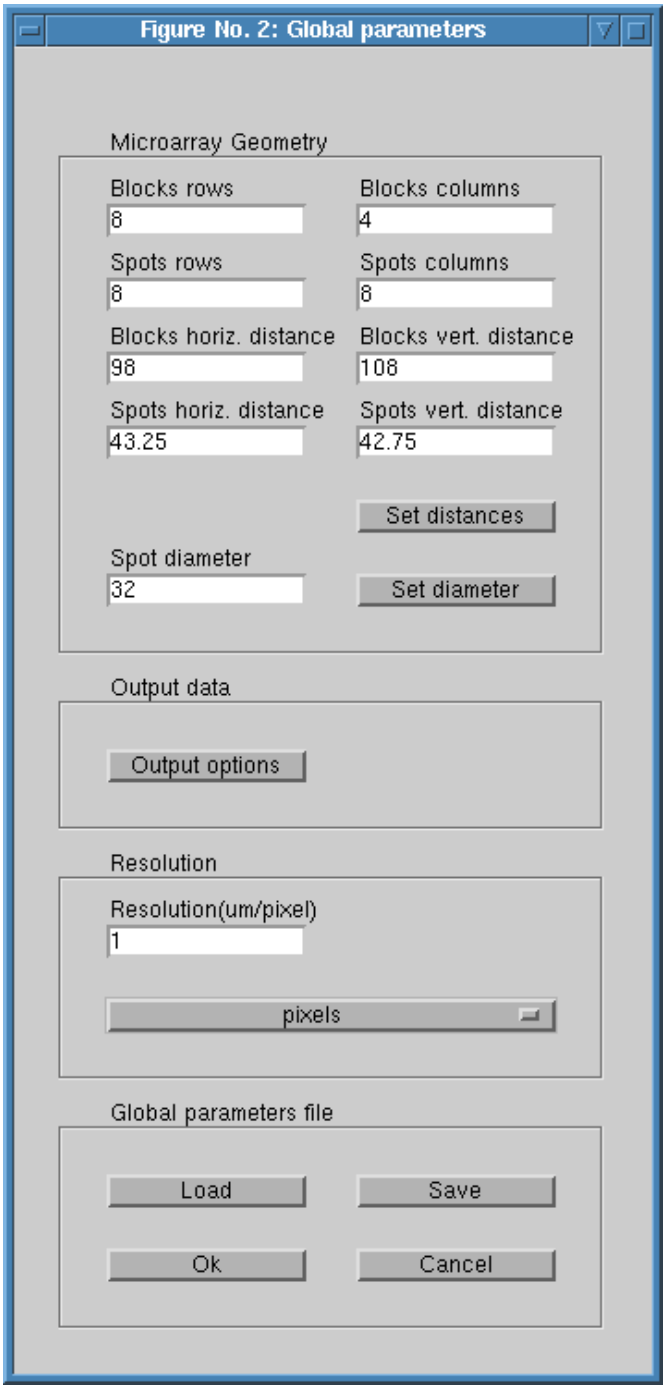



Figura 5.3: Interface de parâmetros globais da família de experimentos.

### 5.2.4 Interface de análise do bloco

Esta interface (Figura 5.4) é mostrada quando o usuário clica em algum bloco na interface principal e seleciona a opção "Process block" do menu. Como na interface principal, o usuário também tem acesso a um pequeno menu quando clica com o botão direito na imagem. Neste menu são apresentadas as opções de incluir ou excluir linhas da grade, marcar ou desmarcar algum *spot*, por estar ruim ou por merecer atenção especial, com um *flag bit*, analisar algum *spot* (opção "Process spot") ou mudar o tamanho da imagem na tela.

No arquivo de dados de saída, os *spots* marcados com o *flag bit* possuem valor diferente de zero na coluna apropriada.

Selecionando a opção "Process spot", alguns gráficos e valores são apresentados para o usuário em seqüência. Primeiro um *scatter plot* dos valores dos *pixels* de um canal pelo outro é mostrado, onde pontos que representam *pixels* do *background* aparecem com cor diferente dos pontos do sinal. Ao mesmo tempo, aparecem os valores de correlação, e as derivadas de uma regressão e de um ajuste robusto (*robust fit*). O segundo gráfico mostra quatro curvas: para os canais verde e vermelho, mostra as curvas dos valores dos *pixels*, do sinal e de toda a região de interesse ou ROI (*region of interest*) do *spot* (sinal junto do *background*), depois de ordenados. O terceiro gráfico mostra duas curvas: uma é o quociente entre os valores dos *pixels* do sinal ordenados de um canal pelo outro; a outra é o quociente entre as curvas dos *pixels* da ROI do *spot*. O quarto gráfico é um histograma do logaritmo na base dois dos quocientes entre pixels de um canal pelo outro. As barras que representam pontos do sinal estão com cor diferente das que representam pontos do *background*.

Nesta interface o usuário também pode ajustar os parâmetros de visualização de saturação mínima, máxima e equalização de histograma. São úteis para verificar se as bordas de *spots* muito apagados estão coerentes com a borda detectada.

O botão "Grid & segment" inicia o processo de gradeamento e segmentação dos *spots*. No final do processo, caso haja alguma linha de grade mal posicionada, o usuário pode corrigi-la clicando na imagem e pressionar o botão "Segment only", que mantém a grade definida pelo usuário e calcula apenas a segmentação dos *spots*.

O gradeamento e segmentação efetuados nesta janela são salvos automaticamente.

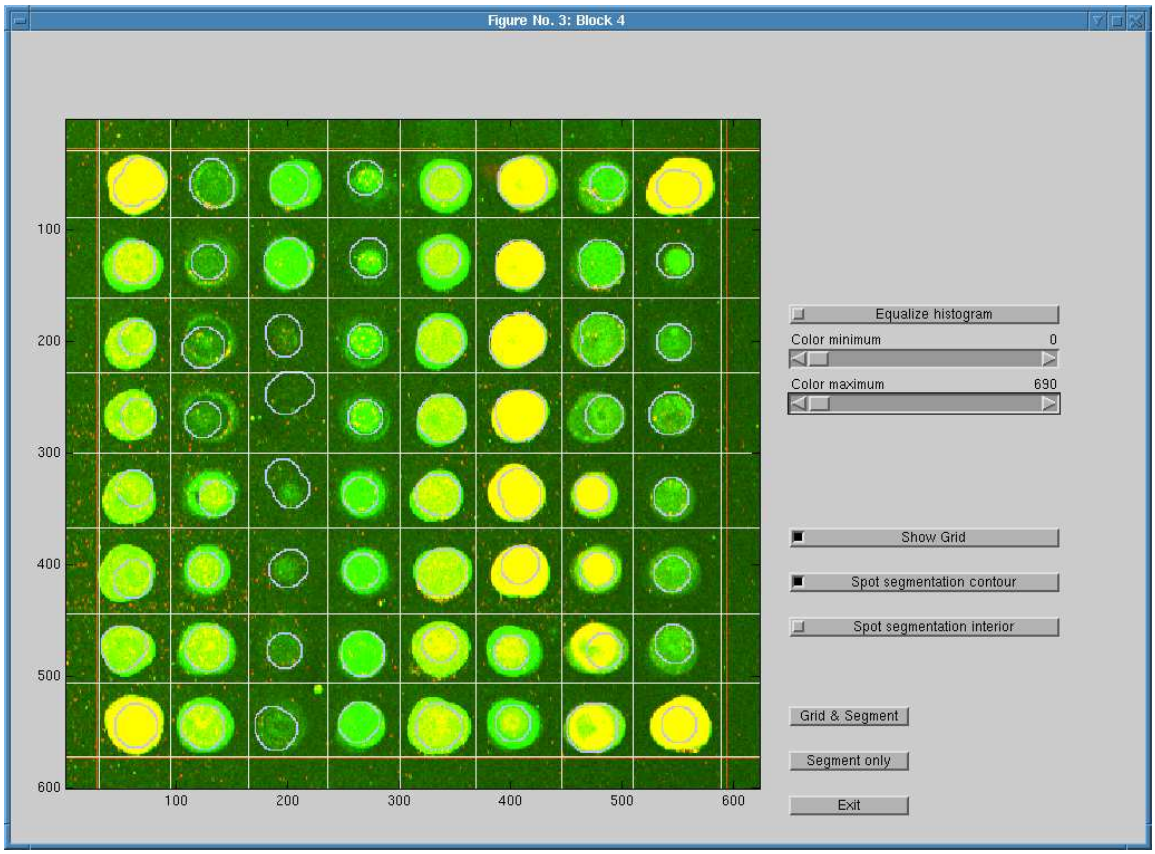



Figura 5.4: Interface para análise de blocos individuais.

### 5.2.5 Correção de rotação

Para que o gradeamento seja bem sucedido, os *spots* devem estar alinhados paralelamente às bordas da imagem. Caso isso não ocorra, os perfis horizontal e vertical não terão um bom contraste, o que dificulta o gradeamento. Além disso, a partir de um certo ângulo de rotação, *spots* de uma coluna de blocos podem estar alinhados verticalmente com *spots* de outra coluna de blocos, e nesse caso, como as linhas de grade são retas paralelas à borda da imagem, uma separação perfeita das colunas de blocos se torna impossível.

O método proposto para correção de rotação depende do usuário. Pode corrigir pequenos ângulos e é feito por uma interface onde o usuário escolhe dois pontos da imagem que deveriam

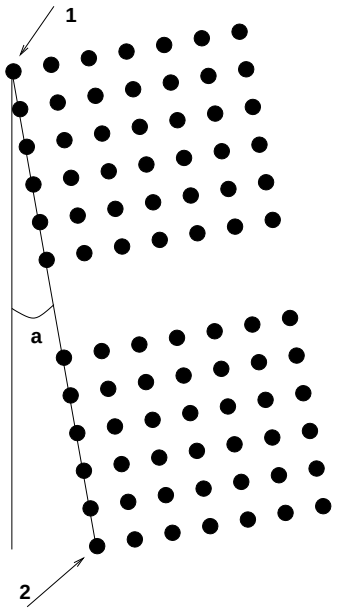



Figura 5.5: O usuário seleciona dois pontos para definir o ângulo de rotação.

estar verticalmente alinhados, definindo o ângulo de rotação. A Figura 5.5 ilustra o processo de escolha dos pontos, por exemplo, o centro do *spot* no extremo superior esquerdo e o centro do *spot* inferior esquerdo.

O programa calcula o ângulo $a$ e corrige a posição dos *pixels* $p = (x, y), \forall p \in E$, onde $E$ é a imagem, transladando-o verticalmente do inteiro mais próximo a $\tan(a)(x - x_0)$ e horizontalmente do inteiro mais próximo a $\tan(a)(y - y_0)$, onde $(x_0, y_0)$ é o centro da imagem.

É importante notar que não podemos aplicar métodos de rotação convencionais (linear, bilinear, bicúbico etc) pois estes alteram o valor dos *pixels*. Nosso método é equivalente a utilizar linhas de grade que formam um ângulo $a$ com as bordas da imagem.

Parte das bordas da imagem é perdida no processo de rotação, o que não é problema se nessa região não houver sinal, ou seja, *spots*.

## 5.3 Arquivo de saída

O arquivo de saída é um arquivo de texto que pode ser aberto em qualquer editor de texto ou programa estatístico. Em cada linha estão todas as informações do respectivo *spot* e as colunas são separadas por tabulações. O formato do arquivo é igual ao do ScanAlyze [25], com algumas colunas a mais e outras a menos.

A primeira coluna pode apresentar três valores: "HEADER" se a linha contém os nomes das colunas, "REMARK" se a linha contiver algum comentário, ou "SPOT" se a linha for referente a algum *spot*.

Na saída do programa, a primeira linha é do tipo "HEADER", que mostra o nome das colunas de dados que aparecem nas linhas do tipo "SPOT". Em seguida vêm as linhas "REMARK", cuja segunda coluna apresenta os seguintes valores: "SOFTWARE" com o nome do programa que gerou o arquivo, "SOFTVERS" com a versão do programa, "GLO FILE" com o nome do arquivo ".glo" se houver, "CH1 IMAGE" e "CH2 IMAGE" com os nomes das imagens verde e vermelha respectivamente, e "DATE" e "TIME" com a data e hora de geração do arquivo.

Finalmente temos as linhas do tipo "SPOT", com tantas colunas quanto a linha "HEADER". Atualmente, as colunas gravadas no arquivo de saída são as seguintes:

- SPOT (inteiro): é um índice único para cada *spot*. Começa com 1 (um) e aumenta de um para cada novo *spot* processado. Os *spots* são processados em uma ordem bem definida e a partir de tal índice pode-se encontrar o bloco e a posição do *spot* dentro dele. Todos os spots de cada linha são processados, da esquerda para a direita, antes de passar para a linha seguinte, imediatamente abaixo.
- GRID (inteiro): é um índice único para cada bloco. Começa com 1 (um) e aumenta de um para cada novo bloco processado. Os blocos são processados em uma ordem bem definida e todos seus *spots* são processados antes de se começar outro bloco. Todos os blocos de cada linha são processados, da esquerda para a direita, antes de se passar para a linha seguinte, imediatamente abaixo.
- TOP (inteiro): a ROI de cada *spot* é um pequeno retângulo que o contém. Para defini-lo bastam quatro valores numéricos da posição das linhas que o limitam: limites superior,

inferior, direito e esquerdo. O valor desta coluna é a posição do limite superior da ROI do *spot* na imagem.

- LEFT (inteiro): a posição do limite esquerdo da ROI do *spot* na imagem.
- BOT (inteiro): a posição do limite inferior da ROI do *spot* na imagem.
- RIGHT (inteiro): a posição do limite direito da ROI do *spot* na imagem.
- ROW (inteiro): linha em que se localiza o *spot* no respectivo bloco. A linha do extremo superior tem valor igual a 1 (um) e cresce de um para cada nova linha processada.
- COL (inteiro): coluna em que se localiza o *spot* no respectivo bloco. A coluna da extrema esquerda tem valor igual a 1 (um) e cresce de um para cada nova coluna processada.
- CH1I (inteiro): valor médio arredondado dos *pixels* do sinal no canal verde.
- CH1B (inteiro): valor mediano arredondado dos *pixels* do *background* no canal verde.
- CH1AB (inteiro): valor médio arredondado dos *pixels* do *background* no canal verde.
- CH2I (inteiro): valor médio arredondado dos *pixels* do sinal no canal vermelho.
- CH2B (inteiro): valor mediano arredondado dos *pixels* do *background* no canal vermelho.
- CH2AB (inteiro): valor médio arredondado dos *pixels* do *background* no canal vermelho.
- SPIX (inteiro): número de *pixels* do sinal.
- BGPIX (inteiro): número de *pixels* do *background*, que é o complementar do sinal na ROI do *spot*.
- MRAT (inteiro): contém a mediana de

$$\frac{\text{CH2PI} - \text{CH2B}}{\text{CH1PI} - \text{CH1B}}$$

onde CH1PI e CH2PI representam valores de *pixels* isolados do sinal.

- REGR (real): contém a derivada da reta ajustada aos valores dos *pixels* por mínimos quadrados.
- CORR (real): coeficiente de correlação entre os valores dos *pixels*. O que aparece como uma nuvem circular no *scatter plot* apresenta correlação próxima de zero, e uma nuvem alongada apresenta correlação próxima de um.
- CH1GTB1 e CH2GTB1 (percentual): fração dos pixels do sinal maiores que CH1B e CH2B respectivamente.

- CH1GTB2 (percentual): fração dos pixels do sinal do sinal maiores que 1.5 vezes CH1B e CH2B respectivamente.
- CH1EDGEA e CH2EDGEA (real): valor médio dos vetores de Sobel vertical e horizontal no interior do *spot* nos canais 1 e 2.
- FLAG (binário): igual a um somente se o respectivo *spot* for marcado pelo usuário.
- CH1KSD, CH1KSP, CH2KSD e CH2KSP: comparam as distribuições dos valores dos *pixels* do sinal com os do *background*. Os valores CH1KSD e CH2KSD são os valores da estatística de Kolmogorov-Smirnov, e CH1KSP e CH2KSP são as probabilidades de que os *pixels* do sinal têm a mesma distribuição do *background*.
- CH1ABB e CH2ABB (inteiro): valor arredondado da média dos valores dos *pixels* no retângulo que limita todo o bloco. Não se mostrou uma boa aproximação do valor do *background*.
- CH1AB10 e CH2AB10 (inteiro): valor arredondado da média dos valores dos *pixels* do primeiro decil na ROI do *spot*.
- CH1B10 e CH2B10 (inteiro): valor arredondado da mediana dos valores dos *pixels* do primeiro decil na ROI do *spot*.
- CH1AF10 e CH2AF10 (inteiro): valor arredondado da média dos valores dos *pixels* do último decil na ROI do *spot*.
- CH1F10 e CH2F10 (inteiro): valor arredondado da mediana dos valores dos *pixels* do último decil na ROI do *spot*.
- CH1ERO e CH2ERO (inteiro): valor arredondado da média dos valores dos *pixels* da região do sinal erodida por um elemento estruturante circular com um sexto do diâmetro esperado do *spot*.
- CH1FHIST_80_95 e CH2FHIST_80_95 (inteiro): valor arredondado da média dos valores dos *pixels* na ROI do *spot* entre os percentis indicados no nome da coluna. Tais percentis são escolhidos pelo usuário na janela "Global parameters", pressionando o botão "Output parameters". O sinal se distingue do *background* apenas pelo valor maior dos percentis, já que a região considerada é a mesma.
- CH1BHIST_5_20 e CH2BHIST_5_20 (inteiro): valor arredondado da média dos valores dos *pixels* na ROI do *spot* entre os percentis indicados no nome da coluna. Tais percentis

são escolhidos pelo usuário na janela "Global parameters", pressionando o botão "Output parameters".

- CH1FFIXR_45_95 e CH2FFIXR_45_95 (inteiro): valor arredondado da média dos valores dos *pixels* do sinal do *spot* entre os percentis indicados no nome da coluna. Tais percentis são escolhidos pelo usuário na janela "Global parameters", pressionando o botão "Output parameters".
- CH1BFIXR_5_55 e CH2BFIXR_5_55 (inteiro): valor arredondado da média dos valores dos *pixels* do *background* do *spot* entre os percentis indicados no nome da coluna. Tais percentis são escolhidos pelo usuário na janela "Global parameters", pressionando o botão "Output parameters".

## 5.4 Próximos passos

Numa versão definitiva, a interface principal deve também apresentar a segmentação dos *spots*. Deve ainda ser implementada num ambiente que ofereça mais recursos de interface, já que o MATLAB™ é bastante limitado. Por exemplo, o usuário não precisaria clicar na imagem e indicar um tamanho, mas o faria clicando e arrastando os cantos da janela.

O programa, na sua versão atual, já apresenta um protótipo de correção de rotação automática, e a versão final deverá ter uma versão melhorada deste procedimento.

# Capítulo 6

# Validação

Existem diversos métodos para estimação da expressão gênica, sendo que os principais foram apresentados no Capítulo 4. Em sua dissertação de mestrado, Gustavo H. Esteves [32], do Instituto Ludwig de Pesquisa contra o Câncer, compara as diversas técnicas propostas por meio de experimentos controlados projetados para tal.

O primeiro experimento, denominado *exp1/1*, foi feito sem *swap*, e a razão esperada entre as amostras teste e referência foi igual a um. No entanto, tal exprimento não permite que as técnicas de medida sejam comparadas para genes diferencialmente expressos. Para contornar esse problema, o experimento foi refeito, agora com *swap* e com quantidades diferentes de cDNA marcado, apresentando razões esperadas de três e seis. Os experimentos foram denominados *exp3/1* e *exp6/1* respectivamente.

Tais experimentos permitem que se faça a normalização por *swap*, mas não permitem que se verifique a corretude dos valores encontrados por não ter genes com a mesma razão esperada, os genes de *housekeeping*. Por essa razão, projetamos novos experimentos que contivessem tanto genes com razão 1 (um) quanto genes diferencialmente expressos. Foram feitos mais três experimentos onde os genes diferencialmente expressos tinham razões cinco, dois e dez, denominados respectivamente *exp1/1-1/5*, *exp1/1-1/2* e *exp1/1-1/10*, todos com *swap*.

Para estimar o erro, comparou-se a razão obtida nos experimentos com a razão esperada usando a função

$$E = \frac{\sum_{i=1}^{p} |r_i - r_e|}{p}$$

onde $p$ é o número de *spots* de um dado gene, $r_i$ é a razão observada em cada *spot* e $r_e$ é a razão esperada.

As expressões foram medidas pelas técnicas:

- *Circfix*: Segmentação por círculo fixo, descrita em 4.3.2. Tem a desvantagem de não separar bem o sinal do *background*, o que pode resultar em superestimação do *background*, especialmente nos *spots* mais intensos, ou acréscimo de ruído aos valores de expressão. Foi a metodologia adotada que incorreu em maiores erros nos genes mais expressos (genes 3, 5 e 6) de *exp1/6* e de *exp1/3*. Nos três últimos experimentos, *exp1/1-1/5*, *exp1/1-1/2* e *exp1/1-1/10*, também apresentou comportamento semelhante a *Segment-100-100*, mostrando um erro médio menor nos genes diferencialmente expressos (genes 1, 3 e 5) e erro médio maior nos genes menos expressos e com razão igual a um (genes 2 e 4).
- *Adap*: Segmentação adaptativa, descrita em 4.3.3. Foi a técnica que apresentou em *exp1/1* um índice de dispersão muito maior que as outras técnicas nos genes pouco expressos, apesar de se mostrar um pouco menor nos outros genes. Esse alto índice de dispersão aparece também em outros experimentos e pode ser observado pelos gráficos. Na Figura 6.3 pode-se ver que os maiores erros nos genes menos expressos (genes 1, 2 e 4) são cometidos por esta metodologia em cinco dos seis casos. Esta metodologia também apresentou erros maiores em *exp1/1-1/10*, nos genes onde a razão esperada entre teste e referência era igual a dez (genes 1, 3 e 5).
- *Circhist-50-50*: Variação da metodologia de segmentação por círculo fixo, descrita em 4.3.2, eliminando-se alguns percentis. Aqui os *pixels* usados estão entre os percentis 45 e 95 da distribuição do sinal e entre os percentis 5 e 55 da distribuição do *background*. Os *pixels* usados podem ser diferentes nos dois canais.
- *Circhist-100-20*: Variação da metodologia de segmentação por círculo fixo, descrita em 4.3.2, eliminando-se alguns percentis. Aqui usam-se todos os *pixels* do sinal e os que estejam en-

tre os percentis 1 e 20 da distribuição do *background*. Os *pixels* usados podem ser diferentes nos dois canais.

- *Circhist-30-10*: Variação da metodologia de segmentação por círculo fixo, descrita em 4.3.2, eliminando-se alguns percentis. Aqui os *pixels* usados são os 30% mais intensos do sinal e os 10% menos intensos do *background*. Faz-se um único histograma e os *pixels* usados são os mesmos nos dois canais.
- *Hist-15-15*: Método descrito em 4.3.1. Os *pixels* usados para estimar o sinal estão entre os percentis 80 e 95 e para o *background*, entre 5 e 20.
- *Segment-50-50*: Metodologia de segmentação morfológica, descrita em 4.3.5, usando os mesmos percentis de *Circhist-50-50*.
- *Segment-100-20*: Metodologia de segmentação morfológica, descrita em 4.3.5, usando os mesmos percentis de *Circhist-100-20*.
- *Segment-100-100*: Metodologia de segmentação morfológica, descrita em 4.3.5, usando todos os percentis do sinal e do *background*. Mostra um comportamento semelhante ao de *Circfix*, nos três últimos experimentos, e em *exp1/3*, apesar de ser o melhor em cinco dos seis genes de *exp1/6*.

As metodologias *Circhist-50-50*, *Circhist-100-20*, *Circhist-30-10*, *Hist-15-15*, *Segment-50-50* e *Segment-100-20* apresentaram erros semelhantes em todos os experimentos e respectivos genes.

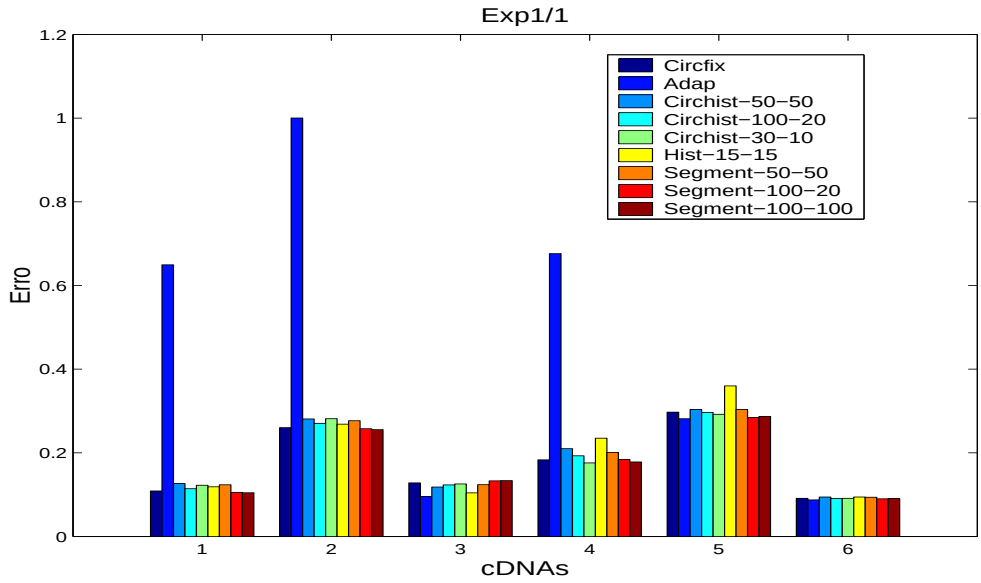



**Figura 6.1: Erros cometidos no experimento exp1/1 (com diluição 5).**
Erros cometidos pelas diferentes metodologias empregadas para a quantificação dos experimentos. Nesta figura são mostrados os erros cometidos para *spots* de todos os cDNAs utilizados com diluição cinco no experimento exp1/1. Os cDNAs estão indicados no eixo $x$ na seguinte ordem: 1 - LysA, 2 - TrpC, 3 - Gene Q, 4 - ST0280, 5 - Il-6, 6 - Irf-1.

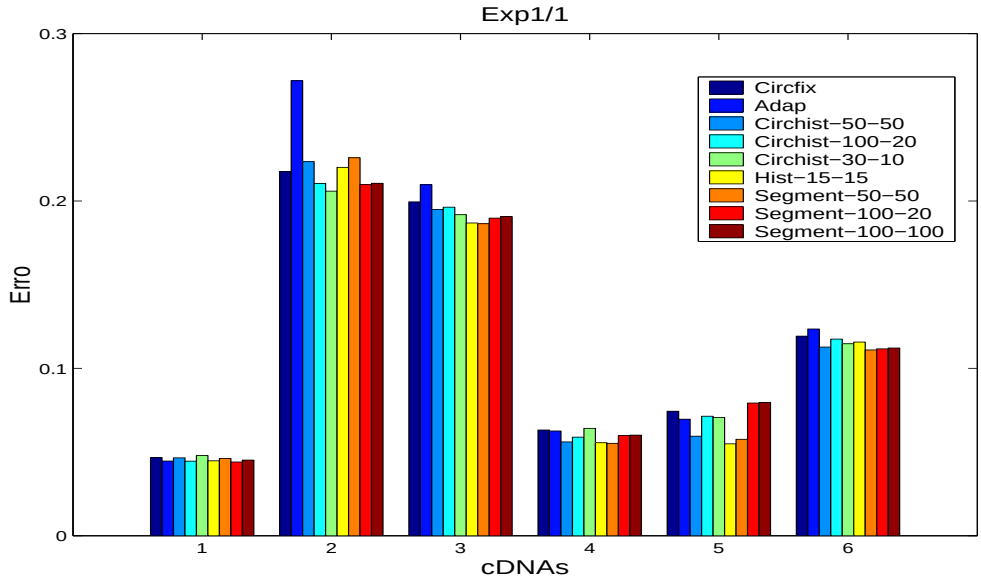



**Figura 6.2: Erros cometidos no experimento exp1/1.**
Esta figura ilustra os erros cometidos pelas diferentes metodologias de quantificação para o experimento exp1/1. Os cDNAs estão indicados no eixo $x$ na seguinte ordem: 1 - LysA, 2 - TrpC, 3 - Gene Q, 4 - ST0280, 5 - Il-6, 6 - Irf-1.

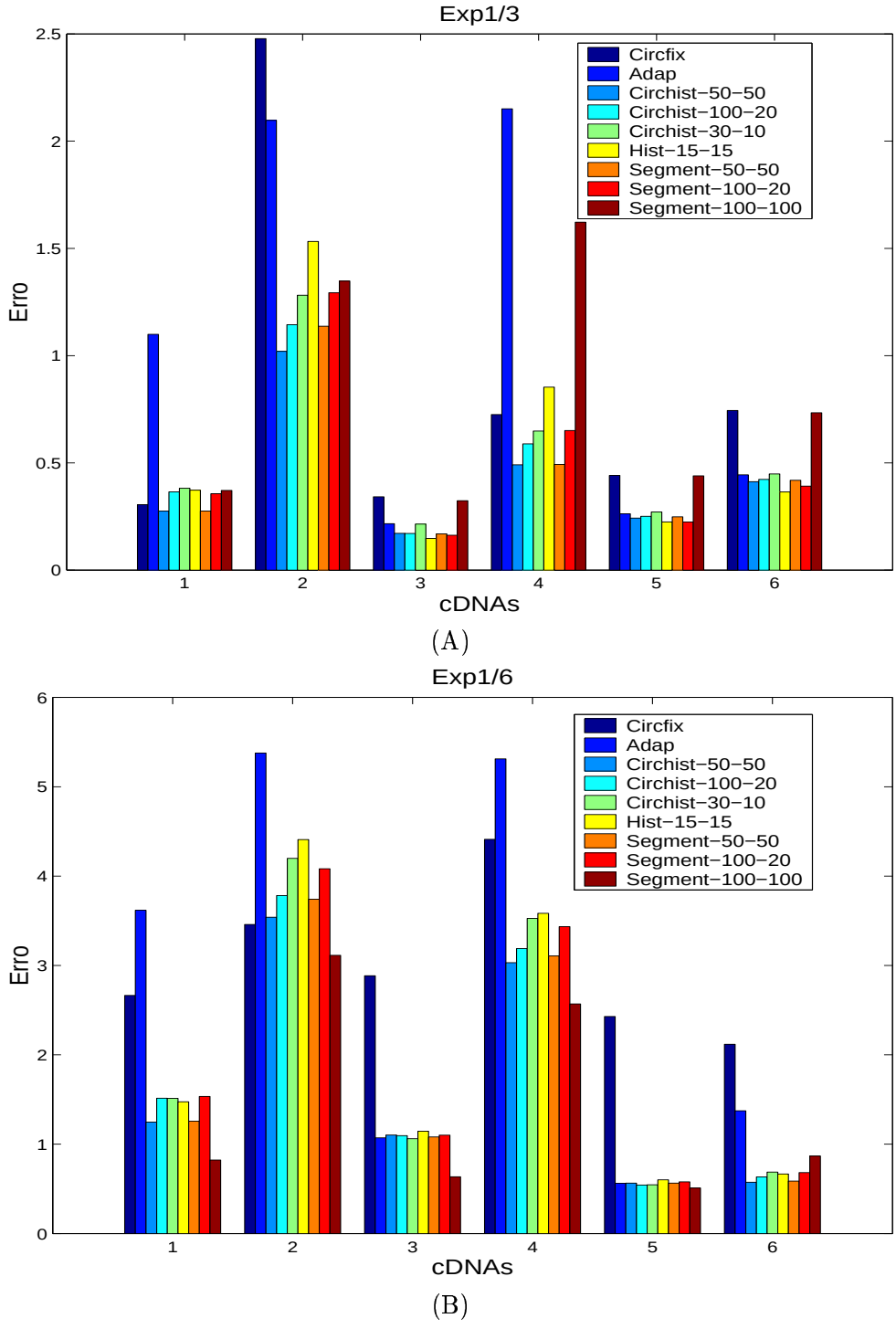



**Figura 6.3: Erros cometidos nos experimentos exp3/1 e exp6/1.**
Erros cometidos pelas diferentes metodologias empregadas para a quantificação dos experimentos (A) - exp3/1 e (B) - exp6/1. Os cDNAs estão indicados no eixo $x$ na seguinte ordem: 1 - LysA, 2 - TrpC, 3 - Gene Q, 4 - ST0280, 5 - Il-6, 6 - Irf-1.

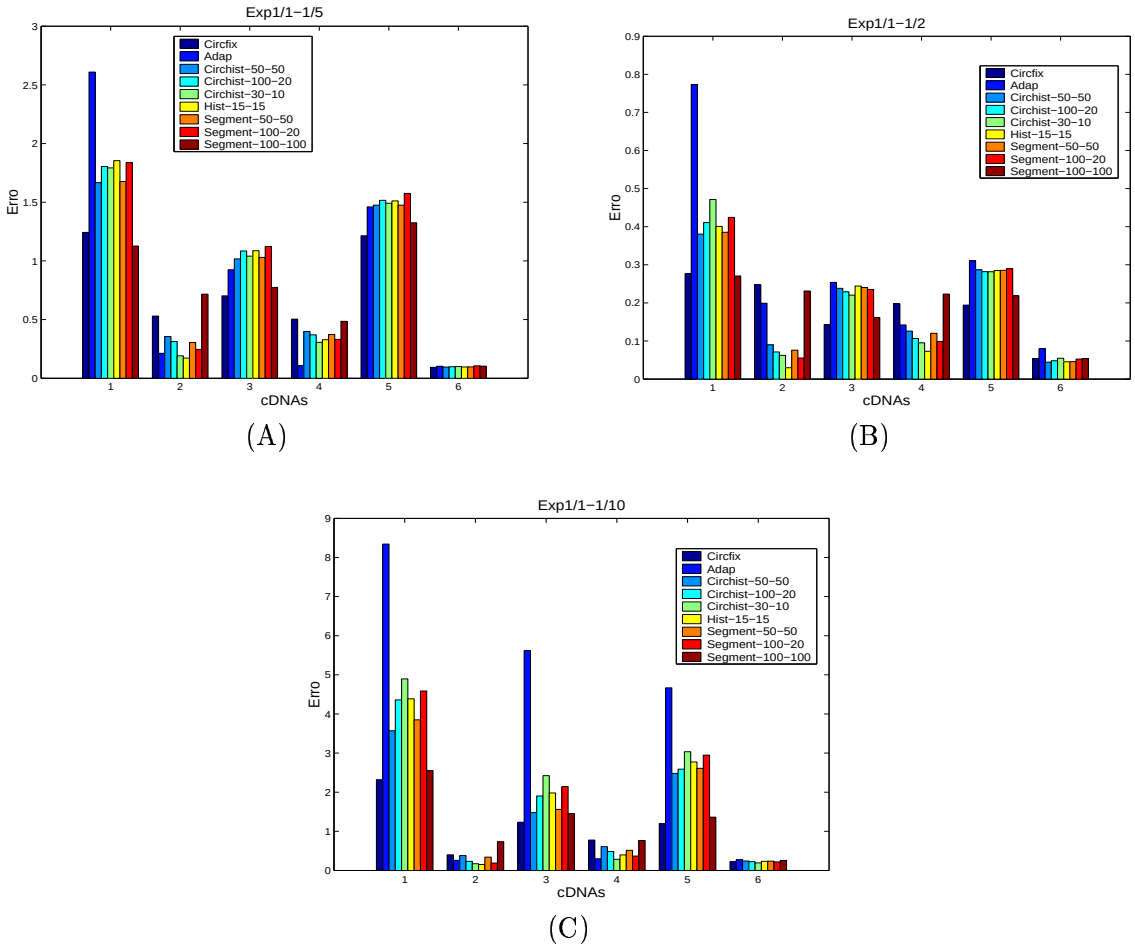



**Figura 6.4: Erros cometidos nos experimentos exp1/1-5/1, exp1/1-2/1 e exp1/1-10/1.**
Essa figura mostra os erros cometidos pelas diferentes metodologias nos experimentos (A) - exp1/1-5/1, (B) - exp1/1-2/1 e (C) - exp1/1-10/1. Os cDNAs estão indicados no eixo $x$ na seguinte ordem: 1 - LysA, 2 - TrpC, 3 - Gene Q, 4 - ST0280, 5 - Il-6, 6 - Irf-1.

**Tabela 6.1: Dados obtidos para o experimento exp1/1 (diluição cinco).**
Médias, desvios padrão e erros obtidos para todos os fragmentos da sonda que apresentam diluição cinco no experimento exp1/1. Neste experimento é esperada razão um para todos os cDNAs.

| Experimento exp1/1, com diluição cinco | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Soft** | **LysA** | | | **TrpC** | | | **Gene Q** | | | **ST0280** | | | **Il6** | | | **Irf1** | | |
| | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** |
| *Circfix* | 1.09 | 0.11 | 0.12 | 1.25 | 0.26 | 0.18 | 0.88 | 0.13 | 0.08 | 1.16 | 0.18 | 0.19 | 1.30 | 0.30 | 0.16 | 1.03 | 0.09 | 0.13 |
| *Adap* | 1.65 | 0.65 | 0.29 | 2.00 | 1.00 | 0.30 | 0.93 | 0.10 | 0.08 | 1.67 | 0.68 | 0.30 | 1.28 | 0.28 | 0.16 | 1.04 | 0.09 | 0.12 |
| *Circhist-50-50* | 1.12 | 0.13 | 0.12 | 1.28 | 0.28 | 0.17 | 0.90 | 0.12 | 0.08 | 1.20 | 0.21 | 0.19 | 1.30 | 0.30 | 0.17 | 1.03 | 0.09 | 0.13 |
| *Circhist-100-20* | 1.11 | 0.11 | 0.12 | 1.27 | 0.27 | 0.16 | 0.89 | 0.12 | 0.08 | 1.18 | 0.19 | 0.18 | 1.30 | 0.30 | 0.16 | 1.03 | 0.09 | 0.13 |
| *Circhist-30-10* | 1.11 | 0.12 | 0.13 | 1.28 | 0.28 | 0.18 | 0.88 | 0.13 | 0.08 | 1.12 | 0.18 | 0.19 | 1.29 | 0.29 | 0.16 | 1.03 | 0.09 | 0.13 |
| *Hist-15-15* | 1.11 | 0.12 | 0.12 | 1.27 | 0.27 | 0.16 | 0.92 | 0.10 | 0.08 | 1.23 | 0.23 | 0.19 | 1.36 | 0.36 | 0.17 | 1.04 | 0.09 | 0.13 |
| *Segment-50-50* | 1.12 | 0.12 | 0.12 | 1.28 | 0.28 | 0.16 | 0.89 | 0.12 | 0.09 | 1.19 | 0.20 | 0.20 | 1.30 | 0.30 | 0.17 | 1.03 | 0.09 | 0.13 |
| *Segment-100-20* | 1.09 | 0.11 | 0.12 | 1.26 | 0.26 | 0.16 | 0.88 | 0.13 | 0.08 | 1.17 | 0.18 | 0.18 | 1.28 | 0.28 | 0.16 | 1.03 | 0.09 | 0.13 |
| *Segment-100-100* | 1.09 | 0.10 | 0.12 | 1.26 | 0.26 | 0.17 | 0.88 | 0.13 | 0.08 | 1.15 | 0.18 | 0.19 | 1.29 | 0.29 | 0.16 | 1.03 | 0.09 | 0.13 |

**Tabela 6.2: Dados obtidos para o experimento exp1/1.**
Médias, desvios padrão e erros obtidos para todos os fragmentos com diluição um da sonda no experimento exp1/1. Neste experimento é esperada razão um para todos os cDNAs.

| Experimento exp1/1, com diluição um | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Soft** | **LysA** | | | **TrpC** | | | **Gene Q** | | | **ST0280** | | | **Il6** | | | **Irf1** | | |
| | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** |
| *Circfix* | 1.02 | 0.05 | 0.06 | 1.22 | 0.22 | 0.07 | 0.80 | 0.20 | 0.05 | 1.02 | 0.06 | 0.07 | 1.07 | 0.07 | 0.05 | 0.89 | 0.12 | 0.07 |
| *Adap* | 1.00 | 0.04 | 0.06 | 1.27 | 0.27 | 0.18 | 0.79 | 0.21 | 0.05 | 1.02 | 0.06 | 0.07 | 1.05 | 0.07 | 0.06 | 0.88 | 0.12 | 0.07 |
| *Circhist-50-50* | 1.02 | 0.05 | 0.05 | 1.22 | 0.22 | 0.07 | 0.81 | 0.19 | 0.04 | 1.02 | 0.06 | 0.07 | 1.05 | 0.06 | 0.05 | 0.89 | 0.11 | 0.06 |
| *Circhist-100-20* | 1.01 | 0.04 | 0.06 | 1.21 | 0.21 | 0.07 | 0.80 | 0.20 | 0.04 | 1.02 | 0.06 | 0.07 | 1.06 | 0.07 | 0.05 | 0.89 | 0.12 | 0.07 |
| *Circhist-30-10* | 1.01 | 0.05 | 0.06 | 1.20 | 0.21 | 0.10 | 0.81 | 0.19 | 0.04 | 1.03 | 0.06 | 0.07 | 1.06 | 0.07 | 0.05 | 0.89 | 0.11 | 0.07 |
| *Hist-15-15* | 1.02 | 0.04 | 0.06 | 1.22 | 0.22 | 0.08 | 0.81 | 0.19 | 0.04 | 1.02 | 0.06 | 0.06 | 1.04 | 0.05 | 0.05 | 0.89 | 0.12 | 0.06 |
| *Segment-50-50* | 1.02 | 0.05 | 0.05 | 1.23 | 0.23 | 0.07 | 0.81 | 0.19 | 0.04 | 1.02 | 0.06 | 0.06 | 1.05 | 0.06 | 0.05 | 0.90 | 0.11 | 0.06 |
| *Segment-100-20* | 1.01 | 0.04 | 0.06 | 1.21 | 0.21 | 0.07 | 0.81 | 0.19 | 0.04 | 1.02 | 0.06 | 0.07 | 1.07 | 0.08 | 0.05 | 0.90 | 0.11 | 0.07 |
| *Segment-100-100* | 1.01 | 0.05 | 0.06 | 1.21 | 0.21 | 0.07 | 0.81 | 0.19 | 0.04 | 1.02 | 0.06 | 0.07 | 1.07 | 0.08 | 0.05 | 0.90 | 0.11 | 0.07 |

**Tabela 6.3: Dados obtidos para o experimento exp3/1.**
Médias, desvios padrão e erros obtidos para todos os fragmentos com diluição um da sonda no experimento exp3/1. Neste experimento é esperada razão três para todos os cDNAs.

| Experimento exp3/1, com diluição um | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Soft** | **LysA** | | | **TrpC** | | | **Gene Q** | | | **ST0280** | | | **Il6** | | | **Irf1** | | |
| | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** |
| *Circfix* | 3.16 | 0.31 | 0.38 | 4.93 | 2.48 | 8.37 | 3.34 | 0.34 | 0.18 | 3.61 | 0.73 | 0.75 | 3.44 | 0.44 | 0.24 | 3.72 | 0.74 | 0.31 |
| *Adap* | 1.90 | 1.10 | 0.38 | 0.99 | 2.10 | 0.72 | 3.19 | 0.22 | 0.17 | 0.85 | 2.15 | 0.22 | 3.24 | 0.26 | 0.21 | 3.41 | 0.44 | 0.29 |
| *Circhist-50-50* | 2.75 | 0.27 | 0.25 | 1.98 | 1.02 | 0.29 | 3.15 | 0.17 | 0.15 | 2.53 | 0.49 | 0.29 | 3.22 | 0.24 | 0.19 | 3.41 | 0.41 | 0.20 |
| *Circhist-100-20* | 2.64 | 0.36 | 0.27 | 1.86 | 1.14 | 0.31 | 3.14 | 0.17 | 0.14 | 2.42 | 0.59 | 0.31 | 3.22 | 0.25 | 0.20 | 3.39 | 0.42 | 0.26 |
| *Circhist-30-10* | 2.65 | 0.38 | 0.38 | 1.72 | 1.28 | 0.41 | 3.19 | 0.21 | 0.17 | 2.36 | 0.65 | 0.37 | 3.26 | 0.27 | 0.19 | 3.41 | 0.45 | 0.26 |
| *Hist-15-15* | 2.64 | 0.37 | 0.27 | 1.47 | 1.53 | 0.22 | 3.12 | 0.15 | 0.13 | 2.15 | 0.85 | 0.26 | 3.19 | 0.22 | 0.19 | 3.36 | 0.36 | 0.19 |
| *Segment-50-50* | 2.75 | 0.28 | 0.25 | 1.86 | 1.14 | 0.36 | 3.15 | 0.17 | 0.14 | 2.52 | 0.49 | 0.30 | 3.23 | 0.25 | 0.19 | 3.41 | 0.42 | 0.20 |
| *Segment-100-20* | 2.65 | 0.36 | 0.27 | 1.71 | 1.29 | 0.33 | 3.14 | 0.16 | 0.13 | 2.35 | 0.65 | 0.30 | 3.20 | 0.22 | 0.18 | 3.36 | 0.39 | 0.24 |
| *Segment-100-100* | 3.23 | 0.37 | 0.47 | 3.41 | 1.35 | 1.97 | 3.32 | 0.32 | 0.18 | 4.44 | 1.62 | 2.91 | 3.44 | 0.44 | 0.23 | 3.71 | 0.73 | 0.31 |

**Tabela 6.4: Dados obtidos para o experimento exp6/1.**
Médias, desvios padrão e erros obtidos para todos os fragmentos com diluição um da sonda no experimento exp6/1. Neste experimento é esperada razão seis para todos os cDNAs.

| Experimento exp6/1, com diluição um | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Soft** | **LysA** | | | **TrpC** | | | **Gene Q** | | | **ST0280** | | | **Il6** | | | **Irf1** | | |
| | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** |
| *Circfix* | 3.33 | 2.67 | 0.57 | 2.78 | 3.46 | 1.90 | 3.11 | 2.89 | 0.27 | 4.08 | 4.41 | 8.74 | 3.57 | 2.43 | 0.40 | 3.88 | 2.12 | 0.58 |
| *Adap* | 2.38 | 3.62 | 0.61 | 0.62 | 5.38 | 0.24 | 4.93 | 1.07 | 0.38 | 0.69 | 5.31 | 0.29 | 5.59 | 0.56 | 0.52 | 4.65 | 1.37 | 0.64 |
| *Circhist-50-50* | 4.75 | 1.25 | 0.59 | 2.46 | 3.54 | 0.85 | 4.90 | 1.10 | 0.36 | 2.97 | 3.03 | 0.65 | 5.56 | 0.56 | 0.52 | 5.67 | 0.57 | 0.61 |
| *Circhist-100-20* | 4.49 | 1.51 | 0.66 | 2.22 | 3.78 | 0.82 | 4.90 | 1.10 | 0.36 | 2.81 | 3.19 | 0.63 | 5.61 | 0.54 | 0.51 | 5.62 | 0.63 | 0.68 |
| *Circhist-30-10* | 4.49 | 1.51 | 0.74 | 1.80 | 4.20 | 0.52 | 4.94 | 1.06 | 0.38 | 2.47 | 3.53 | 0.66 | 5.60 | 0.55 | 0.51 | 5.56 | 0.69 | 0.71 |
| *Hist-15-15* | 4.52 | 1.48 | 0.59 | 1.59 | 4.41 | 0.29 | 4.85 | 1.15 | 0.36 | 2.42 | 3.58 | 0.49 | 5.51 | 0.60 | 0.52 | 5.49 | 0.67 | 0.64 |
| *Segment-50-50* | 4.74 | 1.26 | 0.60 | 2.26 | 3.74 | 0.71 | 4.92 | 1.08 | 0.37 | 2.89 | 3.11 | 0.62 | 5.59 | 0.56 | 0.52 | 5.68 | 0.59 | 0.63 |
| *Segment-100-20* | 4.47 | 1.53 | 0.62 | 1.92 | 4.08 | 0.52 | 4.90 | 1.10 | 0.36 | 2.57 | 3.43 | 0.56 | 5.54 | 0.58 | 0.51 | 5.53 | 0.68 | 0.69 |
| *Segment-100-100* | 5.98 | 0.82 | 1.09 | 4.11 | 3.11 | 2.94 | 5.37 | 0.64 | 0.42 | 5.20 | 2.57 | 3.09 | 6.14 | 0.51 | 0.66 | 6.63 | 0.87 | 0.95 |

**Tabela 6.5: Dados obtidos para o experimento exp1/1-5/1.**
Médias, desvios padrão e erros obtidos para todos os fragmentos com diluição um da sonda no experimento exp1/1-5/1. Neste experimento é esperada razão um para os cDNAs de TrpC, ST0280 e Irf-1 e razão cinco para os demais.

| **Experimento exp1/1-5/1, com diluição um** | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Soft** | **LysA** | | | **TrpC** | | | **Gene Q** | | | **ST0280** | | | **Il6** | | | **Irf1** | | |
| | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** |
| *Circfix* | 3.76 | 1.24 | 0.37 | 1.53 | 0.53 | 0.23 | 4.30 | 0.70 | 0.21 | 1.50 | 0.50 | 0.17 | 3.80 | 1.21 | 0.32 | 0.91 | 0.09 | 0.04 |
| *Adap* | 2.39 | 2.61 | 0.36 | 0.79 | 0.21 | 0.12 | 4.09 | 0.93 | 0.25 | 1.08 | 0.11 | 0.11 | 3.54 | 1.46 | 0.18 | 0.90 | 0.10 | 0.03 |
| *Circhist-50-50* | 3.33 | 1.67 | 0.26 | 1.36 | 0.36 | 0.11 | 3.98 | 1.02 | 0.15 | 1.40 | 0.40 | 0.09 | 3.52 | 1.48 | 0.17 | 0.90 | 0.10 | 0.03 |
| *Circhist-100-20* | 3.20 | 1.80 | 0.25 | 1.31 | 0.31 | 0.11 | 3.92 | 1.08 | 0.13 | 1.37 | 0.37 | 0.09 | 3.48 | 1.52 | 0.16 | 0.90 | 0.10 | 0.03 |
| *Circhist-30-10* | 3.21 | 1.79 | 0.28 | 1.18 | 0.19 | 0.17 | 3.96 | 1.04 | 0.19 | 1.31 | 0.31 | 0.11 | 3.51 | 1.49 | 0.17 | 0.90 | 0.10 | 0.03 |
| *Hist-15-15* | 3.15 | 1.85 | 0.23 | 1.17 | 0.17 | 0.12 | 3.91 | 1.09 | 0.14 | 1.33 | 0.33 | 0.11 | 3.49 | 1.51 | 0.16 | 0.90 | 0.10 | 0.03 |
| *Segment-50-50* | 3.32 | 1.68 | 0.24 | 1.30 | 0.30 | 0.12 | 3.97 | 1.03 | 0.14 | 1.37 | 0.37 | 0.10 | 3.52 | 1.48 | 0.17 | 0.90 | 0.10 | 0.03 |
| *Segment-100-20* | 3.16 | 1.84 | 0.26 | 1.24 | 0.25 | 0.14 | 3.88 | 1.12 | 0.16 | 1.33 | 0.33 | 0.09 | 3.42 | 1.58 | 0.15 | 0.89 | 0.11 | 0.03 |
| *Segment-100-100* | 3.88 | 1.13 | 0.41 | 1.70 | 0.72 | 0.87 | 4.24 | 0.77 | 0.28 | 1.49 | 0.49 | 0.17 | 3.67 | 1.33 | 0.18 | 0.90 | 0.10 | 0.03 |

**Tabela 6.6: Dados obtidos para o experimento exp1/1-2/1.**
Médias, desvios padrão e erros obtidos para todos os fragmentos com diluição um da sonda no experimento exp1/1-2/1. Neste experimento é esperada razão um para os cDNAs de TrpC, ST0280 e Irf-1 e razão dois para os demais.

| Experimento exp1/1-2/1, com diluição um | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Soft** | **LysA** | | | **TrpC** | | | **Gene Q** | | | **ST0280** | | | **Il6** | | | **Irf1** | | |
| | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** |
| *Circfix* | 1.72 | 0.28 | 0.17 | 1.22 | 0.25 | 0.37 | 1.87 | 0.14 | 0.09 | 1.20 | 0.20 | 0.08 | 1.81 | 0.19 | 0.09 | 0.98 | 0.05 | 0.07 |
| *Adap* | 1.23 | 0.77 | 0.17 | 0.87 | 0.20 | 0.21 | 1.75 | 0.25 | 0.08 | 0.97 | 0.14 | 0.21 | 1.69 | 0.31 | 0.09 | 0.95 | 0.08 | 0.09 |
| *Circhist-50-50* | 1.62 | 0.38 | 0.10 | 1.08 | 0.09 | 0.06 | 1.76 | 0.24 | 0.07 | 1.13 | 0.13 | 0.05 | 1.71 | 0.29 | 0.08 | 0.97 | 0.04 | 0.05 |
| *Circhist-100-20* | 1.59 | 0.41 | 0.10 | 1.06 | 0.07 | 0.06 | 1.77 | 0.23 | 0.07 | 1.11 | 0.11 | 0.05 | 1.72 | 0.28 | 0.08 | 0.97 | 0.05 | 0.05 |
| *Circhist-30-10* | 1.53 | 0.47 | 0.18 | 1.01 | 0.06 | 0.08 | 1.78 | 0.22 | 0.07 | 1.08 | 0.10 | 0.08 | 1.72 | 0.28 | 0.09 | 0.97 | 0.05 | 0.06 |
| *Hist-15-15* | 1.60 | 0.40 | 0.09 | 1.01 | 0.03 | 0.04 | 1.76 | 0.24 | 0.07 | 1.07 | 0.07 | 0.04 | 1.71 | 0.29 | 0.08 | 0.97 | 0.05 | 0.05 |
| *Segment-50-50* | 1.61 | 0.39 | 0.10 | 1.07 | 0.08 | 0.06 | 1.76 | 0.24 | 0.07 | 1.12 | 0.12 | 0.05 | 1.71 | 0.29 | 0.08 | 0.97 | 0.05 | 0.05 |
| *Segment-100-20* | 1.58 | 0.42 | 0.12 | 1.04 | 0.06 | 0.06 | 1.76 | 0.24 | 0.06 | 1.10 | 0.10 | 0.05 | 1.71 | 0.29 | 0.08 | 0.97 | 0.05 | 0.06 |
| *Segment-100-100* | 1.79 | 0.27 | 0.22 | 1.22 | 0.23 | 0.24 | 1.84 | 0.16 | 0.07 | 1.22 | 0.22 | 0.12 | 1.78 | 0.22 | 0.10 | 0.98 | 0.05 | 0.08 |

**Tabela 6.7: Dados obtidos para o experimento exp1/1-10/1.**
Médias, desvios padrão e erros obtidos para todos os fragmentos com diluição um da sonda no experimento exp1/1-10/1. Neste experimento é esperada razão um para os cDNAs de TrpC, ST0280 e Irf-1 e razão dez para os demais.

| **Experimento exp1/1-10/1, com diluição um** | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Soft** | **LysA** | | | **TrpC** | | | **Gene Q** | | | **ST0280** | | | **Il6** | | | **Irf1** | | |
| | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** | $\mu$ | **E** | **DP** |
| *Circfix* | 7.82 | 2.32 | 1.83 | 1.39 | 0.40 | 0.31 | 10.55 | 1.23 | 1.52 | 1.78 | 0.78 | 0.34 | 8.87 | 1.20 | 1.17 | 1.23 | 0.23 | 0.08 |
| *Adap* | 1.66 | 8.34 | 0.47 | 0.76 | 0.26 | 0.17 | 4.38 | 5.62 | 1.00 | 0.70 | 0.30 | 0.17 | 5.33 | 4.67 | 0.80 | 1.28 | 0.28 | 0.10 |
| *Circhist-50-50* | 6.43 | 3.57 | 0.81 | 1.38 | 0.38 | 0.15 | 8.52 | 1.48 | 0.70 | 1.61 | 0.61 | 0.14 | 7.52 | 2.48 | 0.41 | 1.24 | 0.24 | 0.05 |
| *Circhist-100-20* | 5.64 | 4.36 | 0.98 | 1.23 | 0.23 | 0.16 | 8.10 | 1.90 | 0.86 | 1.48 | 0.48 | 0.18 | 7.41 | 2.59 | 0.61 | 1.23 | 0.22 | 0.07 |
| *Circhist-30-10* | 5.10 | 4.90 | 1.18 | 1.07 | 0.17 | 0.21 | 7.58 | 2.42 | 1.21 | 1.27 | 0.29 | 0.25 | 6.97 | 3.03 | 0.94 | 1.19 | 0.19 | 0.09 |
| *Hist-15-15* | 5.61 | 4.39 | 0.74 | 1.13 | 0.15 | 0.13 | 8.02 | 1.98 | 0.73 | 1.40 | 0.40 | 0.12 | 7.23 | 2.77 | 0.48 | 1.23 | 0.23 | 0.06 |
| *Segment-50-50* | 6.15 | 3.85 | 0.87 | 1.34 | 0.34 | 0.14 | 8.44 | 1.56 | 0.70 | 1.51 | 0.51 | 0.17 | 7.39 | 2.61 | 0.46 | 1.24 | 0.24 | 0.06 |
| *Segment-100-20* | 5.41 | 4.59 | 1.01 | 1.18 | 0.19 | 0.15 | 7.86 | 2.14 | 0.86 | 1.36 | 0.37 | 0.22 | 7.05 | 2.95 | 0.74 | 1.22 | 0.22 | 0.07 |
| *Segment-100-100* | 8.68 | 2.55 | 3.36 | 1.74 | 0.74 | 0.72 | 10.94 | 1.45 | 2.29 | 1.77 | 0.77 | 0.83 | 8.66 | 1.36 | 0.91 | 1.26 | 0.26 | 0.07 |

# Capítulo 7

# Conclusão

Esse trabalho foi motivado pela dificuldade de alguns pesquisadores no uso das soluções de segmentação de imagens disponíveis. Tais soluções, em geral, requerem uma demorada manipulação das imagens para que se obtenha uma segmentação de qualidade, já que a grade é colocada manualmente ou a segmentação automática não funciona bem. A estratégia de solução do problema se baseia na análise dos perfis da imagem: o sinal é identificado pelos valores do perfil muito mais altos que os do *background*.

A metodologia desenvolvida segmenta automaticamente o sinal de imagens de cDNA sem a necessidade de intensa manipulação normalmente requerida ao se usarem os programas comerciais disponíveis no mercado, além de serem reprodutíveis. Uma lâmina analisada com os mesmos parâmetros retorna resultados idênticos. Os resultados obtidos são equivalentes aos de outros programas em relação ao erro observado.

Uma das deficiências da solução proposta é a necessidade, em algumas imagens, de uma correção de rotação, ou seja, os *spots* devem estar alinhados com as bordas da imagem. Além disso, o passo de correção do gradeamento dos *spots* não aproveita a informação do perfil, o que pode ocasionar erros, especialmente em imagens onde cada bloco possui muitas fileiras de *spots*.

Entre os passos futuros da pesquisa estão: a correção automática da rotação da imagem, identificação automática de *spots* ruins, testar estatisticamente se os experimentos controlados representam realmente experimentos reais e fazer a escolha automática do melhor método de

estimação da expressão gênica para cada *spot*.

Como se trata de uma medida física, com erro experimental associado, gostaríamos de associar barras de erro a cada medida de expressão gênica.

O *software* de segmentação deverá se tornar parte de um sistema completo de análise de *microarray*, com módulos de normalização, análise estatística e de agrupamentos, e integrado a um bancos de dados para armazenar informações sobre o experimento, seus genes e resultados.

# Apêndice A

# Publicações associadas a esta dissertação

- Roberto Hirata Jr., Junior Barrera, Ronaldo F. Hashimoto, Daniel O. Dantas, and Gustavo Esteves. Segmentation of microarray images by mathematical morphology. *Real-Time Imaging*, 8(6):491-505, December 2002.
- `http://www.vision.ime.usp.br/~ddantas/slides/icobicobi2003/icobicobi2003.ppt`.
  Daniel Oliveira Dantas, Junior Barrera, Gustavo Henrique Esteves and Roberto Hirata Junior. Poster 10.33: A software for automatically measuring gene expression from microarray images. 1st International Conference on Bioinformatics and Computational Biology.
- Junior Barrera, R. M. Cesar Jr., Daniel O. Dantas, and D. C. Martins Jr. From microarray images to biological knowledge. *II Brazilian Symposium of Mathematical and Computational Biology*, 2002.
- Roberto Hirata Jr., Junior Barrera, Ronaldo F. Hashimoto, and Daniel O. Dantas. Microarray gridding by mathematical morphology. *In: Proc. SIBGRAPI, Florianópolis. IEEE*, pages 112–119, 2001.

# Apêndice B

# Normalização

## B.1 Introdução

A tecnologia de *microarray* visa medir as diferenças biológicas entre a expressão de RNA de duas amostras. No entanto, tais diferenças podem ser causadas por ruído ou outros fatores não biológicos e isso deve ser ajustado. Tais diferenças são facilmente observadas quando duas amostras idênticas de mRNA são marcadas com corantes diferentes e hibridizadas na mesma lâmina. A intensidade no canal vermelho tende a ser menor que a do canal verde. Tal diferença também varia entre *spots* mais e menos intensos [33].

Isso se deve a diferentes características físicas e químicas dos corantes fluorescentes (meia vida, eficiência da incorporação, emissão de fluorescência), propriedades ou configurações do *scanner*. Outra fonte de ruído vem da posição do *spot* na lâmina, causado por diferenças entre o tamanho das agulhas ou por uma hibridização não uniforme. Entre lâminas diferentes, condições diferentes na hora do experimento também devem causar variações [34, 35].

Para que se obtenha uma medida de melhor qualidade, isso deve ser corrigido. As técnicas de normalização são usadas para isso, e aqui descreveremos algumas das mais utilizadas.

Usaremos a letra $R$ para representar os valores dos *spots* no canal vermelho, e $G$ para os valores do canal verde. Uma forma bastante conveniente de se representar os valores de $R$ e $G$ é o chamado *MA-plot* [33], que é um gráfico de dispersão onde a abscissa é dada por $A = \frac{1}{2}\log_2 RG$

e a ordenada por $M = \log_2 R/G$.

## B.2 Normalização intralâmina

Esse tipo de normalização é feita separadamente para cada lâmina, utilizando apenas as intensidades dos *spots* nos dois canais.

No caso de experimentos de membrana, onde cada lâmina possui um único canal, a normalização é feita utilizando-se as intensidades dos *spots* de duas lâminas diferentes.

### B.2.1 Normalização global

Tais métodos assumem que existe um fator constante que relaciona as intensidades nos dois canais [27], ou seja, $R = kG$. Para encontrar o valor de $k$, pode-se assumir que a média e/ou média aparada (média após desprezar percentis extremos) e/ou a mediana das razões das intensidades deve ser igual a um. Este é o método mais simples e também o mais utilizado.

### B.2.2 Normalização por genes de *housekeeping*

É um tipo de normalização global que usa genes dos quais se espera que a expressão nos dois canais sejam as mesmas. Assumindo que a expressão de genes igualmente expressos deva ser constante independente da intensidade, tais genes podem ser usados para se estimar tal constante.

Yang et al. [36] usa genes dos quais se espera, por experiências anteriores, que tenham a mesma expressão em suas amostras teste e referência.

### B.2.3 Normalização segundo a intensidade

Em muitos casos, a variação causada pelos corantes parece depender da intensidade dos *spots*. Nesses casos, é preferível usar um método de normalização que dependa da intensidade. Uma forma de se fazê-lo é usando o método de regressão conhecido como *lowess* (Robust Locally

Weighted Regression) [36, 37].

Tal função é um alisador de gráficos de dispersão que faz um ajuste diferente para cada ponto do gráfico segundo os pontos que o circundam. A fração dos pontos considerados depende do parâmetro $f$, que é usado tipicamente entre 20% e 40%. Seja $c(A)$ o valor da curva para cada valor de $A$. Os valores de $M$ são normalizados fazendo $M \rightarrow M - c(A)$.

## B.3 Normalização por *swap*

Se supusermos que a hibridização segue um certo modelo matemático, e fizermos o experimento em duplicata, mas trocando os corantes das amostras, a normalização por *swap* anula os efeitos causados pelas diferentes características físicas dos corantes [36].

Sejam $R/kG$ as expressões relativas normalizadas dos genes da primeira lâmina, e $R'/k'G'$ as expressões da segunda lâmina. Os valores de $k$ e $k'$ são funcões de normalização para as duas lâminas que poderiam ser obtidos por algum método intralâmina já descrito. A expressão relativa, denotada por $x$, dos genes da primeira lâmina deve ser, igual ao inverso da segunda lâmina:

$$\frac{R}{kG} \approx \frac{k'G'}{R'}$$

Se $k \approx k'$ então

$$\sqrt{\frac{R}{kG}\frac{k'G'}{R'}} \approx \sqrt{\frac{RG'}{GR'}}$$

nos dá a expressão relativa dos genes normalizada sem que tenhamos que estimar o valor de $k$.

# Referências Bibliográficas


[1] D. J. Duggan, M. Bittner, Y. Chen, P. Meltzer, and J. M. Trent. Expression Profiling Using cDNA Microarrays. *Nature (Genetics Supplement)*, 21:10–14, January 1999.

[2] Yee Hwa Yang, Michael J. Buckley, Sandrine Dudoit, and Terence P. Speed. Comparison of methods for image analysis on cdna microarray data. Technical report, Department of Statistics, University of California at Berkeley.

[3] Aimée M. Dudley, John Aach, Martin A. Steffen, and George M. Church. Measuring absolute expression with microarrays with a calibrated reference sample and an extended signal intensity range. *PNAS*, 99(11):7554–7559, May 2002.

[4] Affymetrix. Affymetrix® GeneChip® Technology Overview. page `http://www.affymetrix.com/technology/`.

[5] Robert J. Lipshutz, Stephen P. A. Fodor, Thomas R. Gingeras, and David J. Lockhart. High density synthetic onigonucleotide arrays. *Supplement to Nature Genetics*, 21:20–24, January 1999.

[6] The chipping forecast. *Supplement to Nature Genetics*, 21, January 1999.

[7] Carl S. Brown, Paul C. Goodwin, and Peter K. Sorger. Image metrics in the statistical analysis of dna microarray data. *PNAS*, 98(16):8944–8949, July 2001.

[8] Jesús Angulo and Jean Serra. Automatic analysis of DNA microarray images using mathematical morphology. *Bioinformatics*, 19(5):553–562, 2003.

[9] A. Brazma and J. Vilo. Gene expression data analysis. *FEBS Letters*, 480(1):17–24, August 2000.

[10] P. D'haeseleer. *Reconstructing Gene Networks from Large Scale Gene Expression Data*. PhD thesis, The University of New Mexico, 2000.

[11] Zbynek Bozdech, Jingchun Zhu, Brian Pulliam, Marcin Joachimiak, Fred Cohen, and Joseph DeRisi. Expression profiling the schizont and trophozoite stages of plasmodium falciparum with a long oligonucleotide microarray. *Genome Biology*, 4(2):`http://derisilab.ucsf.edu/falciparum/`, 2003.

[12] Almut Schulze and Julian Downward. Navigating gene expression using microarrays - a technology review. *Nature Cell Biology*, 3:E190–E195, August 2001.

[13] François Bertucci, Karine Bernard, Béatrice Loriod, Yi-Chung Chang, Samuel Granjeaud, Daniel Birnbaum, Catherine Nguyen, Konan Peck, and Bertrand R. Jordan. Sensitivity issues in dna array-based expression measurements and performance of nylon microarrays for small samples. *Human Molecular Genetics*, 8(9):1715–1722, 1999.

[14] Xiangqin Cui, M. Kathleen Kerr, and Gary A. Churchill. Data transformations for cDNA microarray data. *(in press)*.

[15] G. J. F. Banon and J. Barrera. Bases da Morfologia Matemática para Análise de Imagens Binárias. IX Escola de Computação, Pernambuco, Julho 1994.

[16] Gerald Jean Francis Banon. Formal introduction to digital image processing. Deposited in the URLib collection., 2000. Second edition. This material is used as class notes for an INPE posgraduate course. This work has been supported by CNPq under contract 300966/90-3.

[17] J. Barrera, G. J. F. Banon, R. A. Lotufo, and R. Hirata Jr. MMach: a Mathematical Morphology Toolbox for the Khoros System. *Electronic Imaging*, 7(1):174–210, 1998.

[18] F. Meyer and S. Beucher. Morphological Segmentation. *Journal of Visual Communication and Image Representation*, 1(1):21–46, September 1990.

[19] R. Hirata Jr. Segmentação de Imagens por Morfologia Matemática. Master's thesis, Instituto de Matemática e Estatística - USP, março 1997.

[20] S. Beucher. Watersheds of Functions and Picture Segmentation. In *ICASSP 82, Proc. IEEE Intern. Conf. on Acoustics, Speech and Signal Processing*, pages 1928–1931, Paris, May 1982.

[21] P. Soille and L. Vincent. Determining Watersheds in Digital Pictures via Flooding Simulations. In *Visual Communications and Image Processing*, pages 240–250. SPIE, 1990. volume 1360.

[22] L. Vincent and P. Soille. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, June 1991.

[23] D. Rocke and B. Durbin. A model for measurement error for gene expression arrays. *J Comput Biol*, 8(6):557–569, 2001.

[24] *QuantArray Operating Manual Version 3.0.* Packard Bioscience, January 2001.

[25] Michael Eisen. *ScanAlyze User Manual.* Stanford University, 1999.

[26] M. J. Buckley. The spot user's guide. *CSIRO Mathematical and Information Sciences*, page `http://www.cmis.csiro.au/IAP/Spot/spotmanual.htm`, August 2000.

[27] Yidong Chen, Edward R. Dougherty, and Michael L. Bittner. Ratio-Based Decisions and the Quantitative Analysis of cDNA Microarray Images. *Biomedical Optics*, 2(4):364–374, October 1997.

[28] Ajay N. Jain, Taku A. Tokuyasu, Antoine N. Snijders, Richard Segraves, Donna G. Albertson, and Daniel Pinkel. Fully automatic quantification of microarray image data. *Genome Research*, 12(2):325–332, February 2002.

[29] Yidong Chen, Edward R. Dougherty, Michael L. Bittner, Paul Meltzer, and Jeffrey M. Trent. *Computational and Statistical Approaches to Genomics*, chapter 1 (Microarray Image Analysis and Gene Expression Ratio Statistics). Kluer Academic Publishers, 2002.

[30] David Edwards. Non-linear normalization and background correction in one-channel cDNA microarray studies. *Bioinformatics*, 19(7):825–833, 2003.

[31] Junior Barrera, Roberto Marcondes Cesar Junior, João Eduardo Ferreira, and Marco Dimas Gubitoso. An environment for knowledge discovery in biology. *Computers in Medicine*, (in press).

[32] Gustavo Henrique Esteves. Validação de procedimentos para medida de expressão gênica a partir de imagens de cDNA *microarray*. Master's thesis, Instituto Ludwig de Pesquisa contra o Câncer, Dezembro 2002.

[33] Sandrine Dudoit, Yee Hwa Yang, Matthew J. Callow, and Terence P. Speed. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. Technical Report 578, Stanford University, August 2000.

[34] Wolfgang Huber, Anja von Heydebreck, and Martin Vingron. *Handbook of Statistical Genetics*, chapter Analysis of microarray gene expression data. Wiley, second edition, 2003.

[35] Gordon K. Smyth, Yee Hwa Yang, and Terry Speed. Statistical issues in cDNA microarray data analysis.

[36] Yee Hwa Yang, Sandrine Dudoit, Percy Luu, and Terence P. Speed. Normalization for cDNA microarray data. Technical Report 589, January 2001.

[37] Yee Hwa Yang, Sandrine Dudoit, Percy Luu, David M. Lin, Vivian Peng, John Ngai, and Terence P. Speed. Normalization for cDNA microarray data: a robust composite method adressing single and multiple slide systematic variation. *Nucleic Acids Research*, 30(4), 2002.

# Índice Remissivo

Impresso na Gráfica do IME-USP

4 de Agosto de 2004