

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Breno Henrique de Oliveira Ferreira

**Caracterização e análise de regiões celulares em
imagens de marcação fluorescente de larvas da
*Drosophila melanogaster***

Uberlândia, Brasil

2023

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Breno Henrique de Oliveira Ferreira

**Caracterização e análise de regiões celulares em imagens
de marcação fluorescente de larvas da *Drosophila
melanogaster***

Trabalho de Conclusão de Curso.

Orientador: Dr. Bruno Augusto Nassif Travençolo

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2023

Agradecimentos

Primeiramente, gostaria de agradecer ao grupo PET pelo suporte contínuo durante todo o período de realização deste trabalho. Aos membros e ao tutor, Dr. Wendel Melo. Gostaria também de agradecer às pesquisadoras da UFRJ, Dra. Maira Arruda Cardoso e Dra. Helena Araujo, que cederam a base de imagens para o processamento das informações, sem as quais este trabalho não teria sido possível. Não posso deixar de agradecer também aos professores que me ensinaram ao longo de minha graduação, cujos conhecimentos e experiências foram fundamentais para minha formação. Por fim, agradeço a todos os que, de alguma forma, contribuíram para a realização deste trabalho. Sua ajuda, incentivo e apoio foram essenciais para que eu pudesse concluir este projeto.

Resumo

A *Drosophila melanogaster* (conhecida como mosca-da-fruta), é considerada um dos principais modelos de estudos usados em biomedicina. Seus padrões de expressão gênica são amplamente estudados na forma de imagens digitais, e permitem investigar uma ampla gama de fenômenos biológicos. Neste projeto, imagens de larvas dessa mosca são utilizadas para identificar o papel da proteína Calpaína A na resposta imune inata de *Drosophila*. Usando técnicas de segmentação de imagens no conjunto de imagens cedidas foram geradas máscaras para quantificação nuclear no contexto do corpo gorduroso (tecido responsável pela produção de peptídeos antimicrobianos e antifúngicos em resposta à ativação da via de Toll). Por fim, uma ferramenta foi implementada usando a biblioteca D3.js para visualização integrada dos resultados da segmentação e das principais características das regiões nucleares e citoplasmáticas. Como resultado, teve-se a segmentação correta das imagens obtendo máscaras, que permitiram identificar os núcleos celulares nas amostra com precisão e quantificar a intensidade de expressão da proteína nos núcleos.

Palavras-chave: Segmentação de imagens, Quantificação nuclear, *Drosophila melanogaster*, D3.js, Visualização.

Abstract

Drosophila melanogaster, commonly known as fruit fly, is considered one of the main study models used in biomedicine. Its gene expression patterns are widely studied in the form of digital images, allowing for investigation of a wide range of biological phenomena. In this project, images of larvae from this fly are used to identify the role of Calpain A protein in the innate immune response of *Drosophila*. Using image segmentation techniques on the provided image set, masks were generated for nuclear and cytoplasmic quantification in the context of the fat body (tissue responsible for the production of antimicrobial and antifungal peptides in response to Toll pathway activation). Finally, new visualization tool was implemented using the D3.js library for integrated visualization of segmentation results and main features of nuclear and cytoplasmic regions. As a result, nuclear segmentation was achieved, obtaining masks that allowed for accurate identification of cellular nuclei and quantification of protein expression intensity in the nuclei.

Keywords: *Image segmentation, Nuclear quantification, Drosophila melanogaster, D3.js, Visualization.*

Lista de ilustrações

Figura 1 – Lâminas coradas de tecidos de larvas de <i>Drosophila melanogaster</i>	17
Figura 2 – Processo de segmentação.	18
Figura 3 – Remoção de regiões nucleares fora de foco.	20
Figura 4 – Lâminas coradas de tecidos de larvas de <i>Drosophila</i> com bordas realçadas.	20
Figura 5 – Regiões de expressão da proteína e seus, respectivos, histogramas de frequência de intensidade de proteína de diferentes amostras.	22
Figura 6 – Amostra do grupo Cg_+_Myd.	25
Figura 7 – Resultado do processo de segmentação e análise dos núcleos.	27
Figura 8 – Ferramenta Web de visualização.	28

Sumário

1	INTRODUÇÃO	8
1.1	Objetivos	9
2	REFERENCIAL TEÓRICO	10
2.1	Processamento Digital de Imagens	10
2.1.1	Pré-processamento	10
2.1.2	Segmentação	10
2.1.3	Pós-processamento	10
2.1.4	Filtro Gaussiano	11
2.1.5	Dilatação	11
2.1.6	Erosão	12
2.1.7	Gradiente Morfológico	12
2.1.8	Histograma de Frequência	13
2.2	Tecnologias	13
2.2.1	Matlab	13
2.2.2	HTML	13
2.2.3	CSS	14
2.2.4	JavaScript	14
2.2.5	D3.js	14
2.2.6	CSV	14
2.2.7	Git	14
2.2.8	<i>GitHub</i>	15
2.2.9	<i>GitHub Pages</i>	15
3	METODOLOGIA	16
3.1	Aquisição das imagens	16
3.2	Organização das imagens	16
3.3	Leitura das Imagens	17
3.4	Aplicação de Filtro Gaussiano	17
3.5	Segmentação das Imagens	18
3.6	Remoção de regiões nucleares	19
3.7	Realce das bordas de regiões nucleares	19
3.8	Quantificação da intensidade	21
3.9	Desenvolvimento da Ferramenta de Visualização	23
4	RESULTADOS	25

4.1	Imagens obtidas	25
4.2	Extração das Propriedades dos Núcleos	26
4.3	Ferramenta de visualização	26
5	CONCLUSÃO	29
5.1	Trabalhos Futuros	29
5.1.1	Melhoria da Ferramenta de Visualização	29
5.1.2	Melhoria das Técnicas de Processamento	29
5.1.3	Pesquisas Relacionadas à Base de Dados Coletados	30
	REFERÊNCIAS	31

1 Introdução

Os padrões de expressão gênica são fundamentais para a diferenciação celular e o entendimento do desenvolvimento de um organismo. A análise desses padrões pode ser realizada por meio de imagens digitais, que permitem a caracterização temporal e espacial da expressão gênica (TRAVENÇOLO, 2007). Em particular, a *Drosophila melanogaster* é amplamente utilizada como modelo de estudo em biomedicina, e seus padrões de expressão gênica são estudados por meio de imagens digitais (SOUSA et al., 2020).

Nesse contexto, este projeto tem como objetivo caracterizar e analisar regiões celulares em imagens de marcação fluorescente de larvas da *Drosophila melanogaster* por meio de técnicas de segmentação e quantificação de intensidade de sinal de fluorescência. Foram ao todo analisadas 258 amostras, cada amostra é formada por dois canais. A primeira imagem de canal 1 representa os núcleos celulares, e a segunda imagem de canal 2 representa a expressão da proteína. Essas amostras foram organizadas em 10 grupos de coletas diferentes, são eles: *CgToll10B*, *CgToll10B CactE10eGFP*, *Cg +*, *Cg Cactus CalpAeGFP*, *Cg CactE10eGFP CalpAeGFP*, *CgToll10B CalpARNAi*, *CgToll10B CalpAeGFP*, *CgToll10B CactusRNAi*, *CgToll10B CactuseGFP*.

Para atingir os objetivos deste trabalho, foram aplicados diferentes processos de processamento digital de imagens nas amostras. Na etapa de pré-processamento das imagens, foi aplicado um filtro Gaussiano para redução de ruídos e aumentar a definição das bordas dos núcleos. Na etapa de segmentação, optamos por utilizar a técnica *Points Segmentation*. A segmentação de imagens é um processo que visa dividir uma imagem em regiões para identificar objetos (GONZALEZ; WOODS, 2008). Existem diversos algoritmos de segmentação de imagens, que muitas vezes são combinados para adaptação ao domínio do problema. No pós-processamento, utilizamos operações morfológicas para transformar a imagem e destacar as bordas dos núcleos, permitindo uma melhor identificação das regiões nucleares nas imagens de marcação fluorescente da proteína.

Também foram quantificados os valores de intensidade presentes nos núcleos celulares identificados durante o processo de segmentação. A partir desses valores, foi gerado um histograma de frequência para cada amostra. Por fim, desenvolvemos uma ferramenta web para visualização dos resultados obtidos. Essa ferramenta foi criada utilizando HTML, CSS, Javascript e a biblioteca D3.js para manipulação de dados.

Os resultados da rotina de segmentação apresentaram boa precisão na detecção das regiões celulares para a maioria das amostras de imagens de marcação fluorescente de larvas da *Drosophila melanogaster*. Para o grupo de amostras que não tiveram um resultado tão preciso esse trabalho sugere que a aplicação de um filtro de fechamento

poderia preencher os espaços não segmentados nas regiões celulares identificadas.

1.1 Objetivos

O objetivo principal é realizar a caracterização e análise de regiões celulares em imagens de marcação fluorescente de larvas da *Drosophila melanogaster* gerando máscaras nucleares para quantificação da intensidade de sinal de fluorescência nessas regiões. O objetivo foi dividido nos seguintes itens:

- Desenvolvimento de rotinas para segmentação automática de núcleos celulares;
- Quantificação da intensidade de expressão das proteínas analisadas em diferentes canais das imagens;
- Desenvolvimento de ferramenta de visualização para apresentação integrada dos resultados da segmentação e das principais características das regiões nucleares;

2 Referencial Teórico

Neste capítulo serão apresentados os principais conceitos utilizados no desenvolvimento do trabalho.

2.1 Processamento Digital de Imagens

O Processamento Digital de Imagens é uma área da computação que se dedica à manipulação de imagens digitais com o uso de técnicas e algoritmos específicos. O objetivo principal do processamento de imagens é extrair informações relevantes a partir das imagens, melhorar a qualidade visual, corrigir distorções e fazer análises quantitativas (GONZALEZ; WOODS, 2008). Essa área possui uma grande variedade de aplicações, abrangendo setores como medicina, astronomia, biologia, segurança, dentre outros.

2.1.1 Pré-processamento

No pré-processamento, as imagens são submetidas a técnicas com o objetivo de melhorar a sua qualidade e torná-las mais adequadas para o processamento. Isso pode incluir operações como filtragem, redução de ruído, correção de brilho e contraste, equalização de histograma e outros ajustes. Entre as técnicas de pré-processamento, foi aplicado neste trabalho o filtro Gaussiano.

2.1.2 Segmentação

A segmentação de imagens consiste em separar a imagem em regiões ou objetos com características semelhantes. Existem várias técnicas de segmentação de imagens, incluindo segmentação baseada em limiar, segmentação por crescimento de região, segmentação por borda e outras técnicas mais avançadas.

2.1.3 Pós-processamento

O pós-processamento é uma etapa importante para melhorar a qualidade da imagem e destacar características relevantes. O objetivo é transformar a imagem para torná-la mais adequada para a finalidade desejada (GONZALEZ; WOODS, 2008). Da etapa de pós-processamento foram utilizadas duas operações morfológicas, sendo elas dilatação e erosão.

2.1.4 Filtro Gaussiano

O filtro Gaussiano funciona aplicando uma convolução entre a imagem original e um kernel Gaussiano, que é uma matriz de valores centrada em torno do pixel atual. A convolução é calculada multiplicando os valores dos pixels da imagem original com os valores do kernel e somando os resultados para gerar um novo valor para o pixel central. Esse processo é repetido para cada pixel na imagem, resultando em uma nova imagem suavizada e com ruído reduzido (GONZALEZ; WOODS, 2008).

A fórmula do kernel Gaussiano é dada por:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

onde x e y são as coordenadas do pixel atual em relação ao centro do kernel, e σ é o desvio padrão que determina a largura da curva Gaussiana. Quanto maior o valor de σ , mais suave será a imagem resultante.

O filtro Gaussiano é amplamente utilizado em diversas aplicações, como no processamento de imagens médicas, visão computacional e reconhecimento de padrões (GONZALEZ; WOODS, 2008). Ele é considerado um filtro de baixa passagem, o que significa que ele suaviza as áreas de baixa frequência na imagem, reduzindo o ruído e preservando as bordas e detalhes de alta frequência (GONZALEZ; WOODS, 2008).

2.1.5 Dilatação

A operação morfológica de dilatação é uma técnica comum de processamento de imagens que amplia as regiões de uma imagem segmentada. A dilatação pode ser realizada por meio de um elemento estruturante, que é uma pequena matriz usada para percorrer a imagem e expandir as regiões.

O processo de dilatação posiciona o elemento estruturante de forma centralizada em cada pixel da imagem. Em seguida, é realizada uma operação lógica entre o elemento estruturante e a região da imagem correspondente aos pixels que o elemento cobre. Se pelo menos um pixels dessa região possui intensidade igual a “branco”, ou seja, valor igual a 1, então o pixel central é considerado como pertencente à região e é alterado para o valor “branco” na imagem resultante. Caso contrário, se nenhum pixel dessa região tiver intensidade igual a “branco”, então o pixel central é considerado como não pertencente à região e é alterado para o valor “preto”, ou seja, valor igual a 0, na imagem resultante (GONZALEZ; WOODS, 2008).

A operação de dilatação pode ser expressa em termos de operações de conjuntos, como:

$$A \oplus B = z \mid (\hat{B})_z \cap A \neq \emptyset$$

onde A representa a imagem original, B representa o elemento estruturante e \oplus denota a operação de dilatação. O símbolo $(\hat{B})_z$ representa a translação do elemento estruturante para a posição z na imagem e \cap denota a operação de interseção de conjuntos.

A dilatação é útil para várias aplicações de processamento de imagens, como preenchimento de lacunas em objetos, remoção de ruído e detecção de bordas. No entanto, é importante notar que a dilatação também pode expandir as regiões de uma imagem de tal forma que os objetos adjacentes se fundam, podendo resultar em uma imagem distorcida (GONZALEZ; WOODS, 2008).

2.1.6 Erosão

Ao contrário da dilatação, a erosão é uma técnica que diminui as regiões de uma imagem segmentada. Assim como a dilatação, a erosão pode ser realizada por meio de um elemento estruturante.

O processo de erosão, assim como a dilatação, posiciona o elemento estruturante de forma centralizada em cada pixel da imagem. Em seguida, é realizada uma operação lógica entre o elemento estruturante e a região da imagem correspondente aos pixels que o elemento cobre. Se todos os pixels dessa região possuem intensidade igual a “branco”, ou seja, valor igual a 1, então o pixel central é considerado como pertencente à região e é alterado para o valor “branco” na imagem resultante. Caso contrário, se pelo menos um pixel dessa região tiver intensidade igual a “preto”, ou seja, valor igual a 0, então o pixel central é considerado como não pertencente à região e é alterado para o valor “preto” na imagem resultante (GONZALEZ; WOODS, 2008).

A operação de erosão pode ser expressa em termos de operações de conjuntos, como:

$$A \ominus B = z \mid (B)_z \subseteq A$$

onde A representa a imagem original, B representa o elemento estruturante e \ominus denota a operação de erosão. O símbolo $(B)_z$ representa a translação do elemento estruturante para a posição z na imagem e \subseteq denota a operação de inclusão de conjuntos.

A erosão é útil para remoção de detalhes finos, separação de objetos próximos e detecção de bordas. No entanto, assim como a dilatação, a erosão pode resultar em uma imagem distorcida. Já que a erosão pode diminuir as regiões de uma imagem de tal forma que os objetos se tornem desconexos (GONZALEZ; WOODS, 2008).

2.1.7 Gradiente Morfológico

O gradiente morfológico é uma operação utilizada para destacar as bordas dos objetos presentes na imagem. Ele é obtido a partir da diferença entre a imagem dilatada

e a imagem erodida.

O processo de cálculo do gradiente morfológico é feito utilizando o elemento estruturante. Primeiramente, é realizada a dilatação e a erosão da imagem original pelo elemento estruturante. Em seguida, é feita a diferença entre a imagem dilatada e a imagem erodida ([GONZALEZ; WOODS, 2008](#)).

O resultado do gradiente morfológico é uma imagem binária que representa as bordas dos objetos presentes na imagem original. As bordas são destacadas porque são os pixels que se mantêm após a diferença entre as operações de dilatação e erosão.

2.1.8 Histograma de Frequência

O histograma de frequência é uma ferramenta para análise da distribuição de intensidades de pixels em imagens digitais. Trata-se de uma representação gráfica da distribuição de frequências das intensidades de pixels, em que as intensidades são apresentadas no eixo horizontal e as frequências no eixo vertical. É amplamente utilizado em aplicações de processamento digital de imagens ([GONZALEZ; WOODS, 2008](#)).

2.2 Tecnologias

Existem diversas tecnologias disponíveis, com diferentes objetivos e funcionalidades, que podem ser utilizadas para fins diversos. Algumas das mais populares incluem linguagens de programação, formatos de arquivos, bibliotecas para desenvolvimento, softwares de análise de dados, entre outros.

2.2.1 Matlab

O MATLAB é uma plataforma de software de computação numérica e visualização de dados amplamente utilizada em processamento digital de imagens. Ele oferece uma ampla variedade de ferramentas e funções para processar, analisar e visualizar imagens ([MATHWORKS, 2022](#)). Alguns exemplos incluem a biblioteca de processamento de imagens, a biblioteca de segmentação de imagens, a biblioteca de reconhecimento de padrões e a biblioteca de aprendizado de máquina.

2.2.2 HTML

HTML é a linguagem de marcação padrão usada para criar páginas da web. Ele permite definir a estrutura e o conteúdo de uma página, incluindo imagens, texto e outros elementos. As imagens podem ser incorporadas diretamente na página ou referenciadas por meio de URLs ([MOZILLA, 2021b](#)).

2.2.3 CSS

CSS é uma linguagem de folha de estilo usada para definir a aparência e o layout de uma página da web. Ele permite definir a cor, fonte, tamanho e posicionamento de elementos, incluindo imagens. Isso pode ser usado para criar imagens interativas, como mapas de imagem, em que diferentes áreas da imagem são destacadas quando o usuário passa o mouse sobre elas ([MOZILLA, 2021a](#)).

2.2.4 JavaScript

JavaScript é uma linguagem de programação de alto nível usada para adicionar interatividade e dinamismo a páginas da web. Ele permite responder a eventos do usuário, como cliques e movimentos do mouse, e manipular elementos da página em tempo real. Isso pode ser usado para criar animações e efeitos visuais em imagens, bem como para criar interfaces de usuário interativas ([MOZILLA, 2021c](#)).

2.2.5 D3.js

D3.js é uma biblioteca JavaScript popular para visualização de dados na web. Ele permite criar gráficos e visualizações interativas com base em dados, incluindo imagens. Ele oferece uma ampla variedade de recursos para manipulação de dados e interação do usuário, permitindo a criação de visualizações complexas e personalizadas. D3.js pode ser usado para criar imagens interativas, como mapas de calor e gráficos de redes ([D3.JS, 2023](#)).

2.2.6 CSV

CSV (*Comma-Separated Values*) é um formato de arquivo de texto simples e amplamente utilizado na armazenagem de dados tabulares, em que cada linha representa uma entrada da tabela e as colunas são separadas por vírgulas ([WIKIPÉDIA, 2018](#)).

2.2.7 Git

O Git é um sistema de controle de versão amplamente utilizado na indústria de desenvolvimento de software para gerenciar o código-fonte de um projeto ([CHACON; STRAUB, 2014](#)). Ele permite que várias pessoas trabalhem em um mesmo projeto simultaneamente, mantendo um histórico completo de todas as alterações feitas no código. Com o Git, os desenvolvedores podem trabalhar em seus próprios ramos de código e depois mesclá-los de volta para o ramo principal, mantendo um registro de quem fez quais alterações e quando ([CHACON; STRAUB, 2014](#)).

2.2.8 *GitHub*

O *GitHub* é uma plataforma online que utiliza o Git como sistema de controle de versão e fornece uma interface amigável para gerenciar repositórios de código-fonte. Com o *GitHub*, os desenvolvedores podem facilmente colaborar em projetos, abrir e revisar solicitações de alteração, gerenciar problemas e bugs, além de publicar e compartilhar projetos com outros usuários da plataforma ([GITHUB](#), Accessed: 2023).

Entre muitos dos serviços que o *GitHub* oferece, existe, o serviço de hospedagem de páginas estáticas, chamado *GitHub Pages*.

2.2.9 *GitHub Pages*

O *GitHub Pages* é uma solução gratuita oferecida pelo *GitHub* para hospedar páginas da web diretamente de um repositório do *GitHub* ([GITHUB](#), 2021). Essa funcionalidade permite que os desenvolvedores criem sites estáticos, como blogs pessoais, portfólios ou documentação de projetos, sem a necessidade de contratar serviços de hospedagem ou infraestrutura ([GITHUB](#), 2021).

Um bom uso para o *GitHub Pages*, é que os desenvolvedores podem hospedar documentação de projetos diretamente no repositório do projeto. Isso torna mais fácil para outros usuários do *GitHub* acessar a documentação e entender como utilizar o software.

3 Metodologia

Esta seção inclui os métodos e técnicas aplicados no trabalho, justificando a seleção dos métodos, materiais e procedimentos usados.

3.1 Aquisição das imagens

Neste trabalho foram utilizadas imagens de marcação fluorescente de larvas da *Drosophila melanogaster*, adquiridas por pesquisadoras da Universidade Federal do Rio de Janeiro (UFRJ) e cedidas para segmentação com o objetivo da quantificação da intensidade de expressão das proteínas presentes nos núcleos celulares.

As imagens fornecidas apresentam alta qualidade, tornando-as ideais para uso em algoritmos de segmentação. Ao todo, foram disponibilizados 258 de imagens para processamento e análise, organizadas em 10 grupos de coleta distintos: *CgToll10B*, *CgToll10B CactE10eGFP*, *Cg +*, *Cg Cactus CalpAeGFP*, *Cg CactE10eGFP CalpAeGFP*, *CgToll10B CalpARNAi*, *CgToll10B CalpAeGFP*, *CgToll10B CactusRNAi*, *CgToll10B CactuseGFP*. Cada imagem tem resolução de 1024 pixels por 1024 pixels e, em média, ocupa cerca de 1,00 MB de espaço em memória.

3.2 Organização das imagens

A organização das imagens adquiridas foi realizada afim de facilitar a análise dos dados. Cada amostra contém duas imagens de lâminas coradas do canal 1 (azul) e Canal 2 (vermelho), como mostra a Figura 1. O Canal 1 (azul) apresenta as imagens dos núcleos das larvas de *Drosophila melanogaster*, enquanto o Canal 2 (vermelho) identifica a presença da proteína analisada.

Primeiramente, foi definido o diretório onde as imagens se encontram. Em seguida, foi criado uma função, *getFolders*, para obter a lista de pastas dentro do diretório, divididas de acordo com os grupos de coleta. Com a lista de pastas em mãos, foi criada uma função *getAllFiles* para buscar todos os arquivos dentro das pastas de dados e filtrar os arquivos das imagens dos canais 1 e 2, e armazenar seus nomes nas variáveis *ch00_files* e *ch02_files*.

Após a leitura dos arquivos, foi criado um diretório para armazenar os resultados da análise. Caso o diretório já existisse, o mesmo é removido utilizando a função *rmdir* para evitar possíveis conflitos na criação de novos arquivos. Por fim, foi criado um loop para percorrer a lista de pastas de grupos coletados, criando um diretório para cada grupo dentro do diretório de resultados utilizando a função *mkdir*.

A partir dessa estrutura, usando técnicas de segmentação, foi possível gerar uma máscara para identificar os núcleos celulares no Canal 1 e quantificar a expressão da proteína nas imagens do Canal 2. Possibilitando a identificação de padrões e alterações na distribuição da proteína em questão.

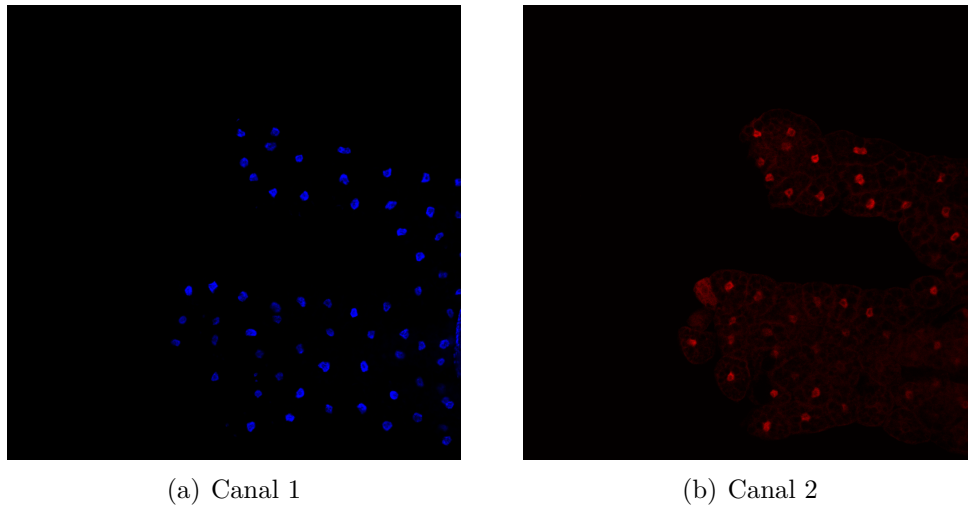


Figura 1 – Lâminas coradas de tecidos de larvas de *Drosophila melanogaster*

3.3 Leitura das Imagens

Para a abertura das imagens foi utilizada a função *imread*, que carrega as imagens em uma variável para que elas possam ser manipuladas pelo código. A *imread* retorna uma matriz de pixels que representa a imagem, onde cada elemento da matriz corresponde a um pixel da imagem (MATHWORKS, 2022). As imagens de cada canal são lidas de forma separada e armazenadas em variáveis distintas, uma amostra de cada vez. Ao final do processamento da amostra ela é removida da memória.

3.4 Aplicação de Filtro Gaussiano

Para este trabalho, tornou-se necessária a utilização de um filtro de suavização, no caso, utilizamos uma função de filtro Gaussiano com desvio padrão 1,5. Sua aplicação teve como objetivo reduzir ruídos e suavizar as bordas, sem prejudicar a nitidez da imagem original. Para aplicação do filtro foi utilizada a função do Matlab *imgaussfilt*, que recebe a imagem original e um valor de desvio padrão retornando como resultado uma matriz de pixels (MATHWORKS, 2022).

3.5 Segmentação das Imagens

Foram realizadas uma série de testes empíricos comparando técnicas de limiarização mais simples. No entanto, os resultados desses testes não foram satisfatórios. Isso se deve ao fato de que as diferentes técnicas de limiarização mais simples não conseguiram lidar adequadamente com os diferentes planos de corte dos núcleos nas imagens. Portanto, optamos por utilizar uma técnica mais avançada de limiarização.

Para a segmentação dos núcleos utilizamos uma técnica chamada de *Points Segmentation*. Foi necessário a implementação dessa técnica, sendo dividida, em duas rotinas a *PointsSegmentation.m* e *CleanCC.m*. Essa técnica têm seu funcionamento similar a técnica *Watershed* (GONZALEZ; WOODS, 2008) e pode ser descrito nos seguintes parágrafos.

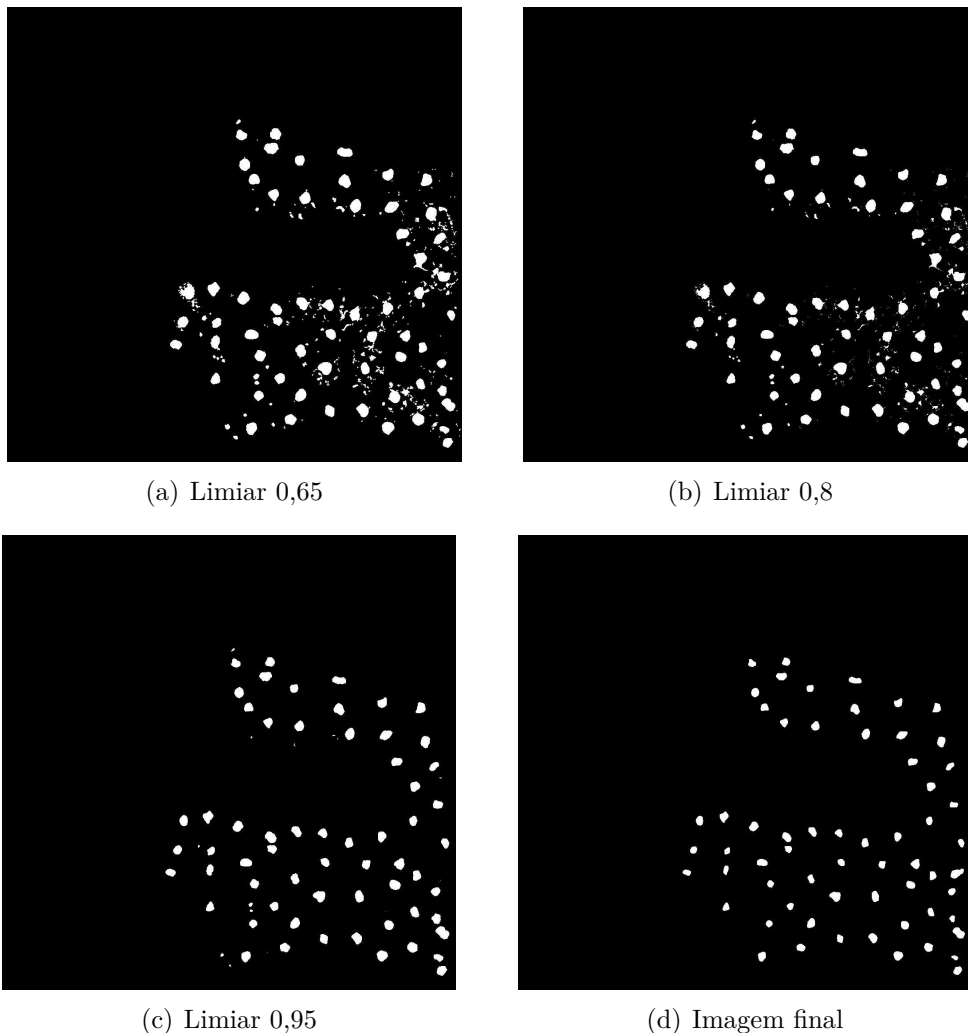


Figura 2 – Processo de segmentação.

A primeira etapa é a normalização dos valores dos pixels na imagem para garantir que todos os valores estejam dentro do intervalo $[0,255]$. Em seguida é definido um

elemento estruturante em formato de esfera com raio 1, que será usado na dilatação da imagem.

É realizada a limiarização da imagem, em que é feita uma varredura dos valores de limiar da imagem, utilizando uma faixa de valores definidos entre 0,5 e 1 com uma variação de 0,15 entre cada valor definido. A primeira limiarização é feita utilizando o valor mínimo desse intervalo, e as demais são realizadas a partir dos valores seguintes até o valor máximo do intervalo.

Após a segmentação, é feita uma limpeza dos componentes conexos identificados, que tem como objetivo remover componentes conexos que se encontram em contato com a borda da imagem.

A limiarização então é repetida para o próximo valor do intervalo. Neste momento, é realizada a verificação se algum componente conexo da nova limiarização sobrepõe mais de um componente conexo da anterior. Isso é feito comparando os identificadores de cada componente conexo nas imagens anterior e atual. Se houver algum identificador duplicado, significa que dois componentes conexos da imagem anterior se fundiram na nova imagem, o que não é permitido. Neste caso, o componente conexo da nova imagem é ignorado.

Por fim, a imagem final é obtida pela união dos componentes conexos da imagem anterior e atual, que são dilatados para se tornarem conexos.

Em resumo, a função de segmentação de imagens apresentada realiza uma segmentação iterativa da imagem, seguida de uma limpeza das componentes conexas identificadas, e por fim uma união dessas componentes conexas para obtenção da imagem segmentada final.

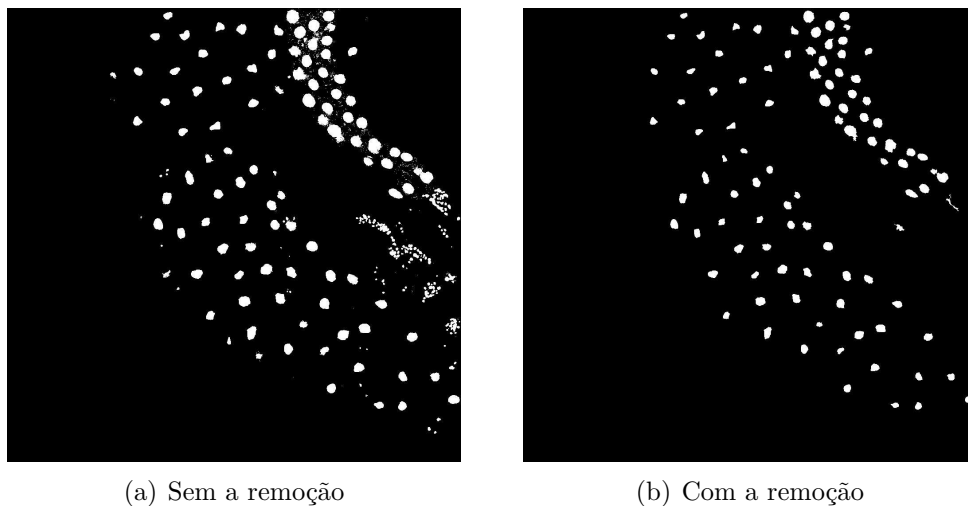
3.6 Remoção de regiões nucleares

Observamos que regiões com áreas muito pequenas ou muito grandes estavam fora de foco e não eram relevantes para a análise. Sendo assim, definiu-se um intervalo fechado de 100 a 2500 unidades para manter apenas as regiões nucleares consideradas relevantes para a análise. A determinação dos valores de área em pixels utilizados para filtrar as regiões nucleares foi obtida de forma empírica após uma análise inicial das imagens.

Na Figura 3, é possível visualizar o efeito da remoção dos núcleos celulares presentes no intervalo.

3.7 Realce das bordas de regiões nucleares

Buscando melhorar a visibilidade das regiões nucleares nas imagens do canal 1 e 2 foram realçadas suas bordas, como mostra a Figura 4. Para isso, foi utilizada a máscara de

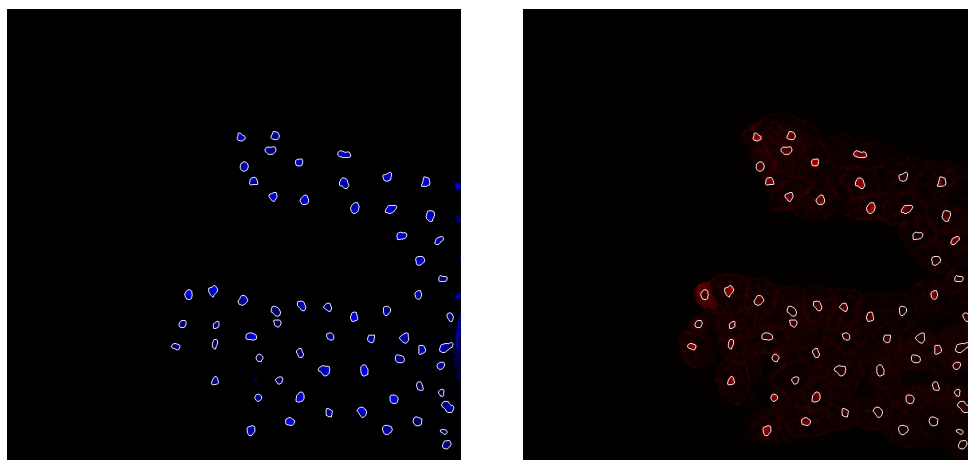


(a) Sem a remoção

(b) Com a remoção

Figura 3 – Remoção de regiões nucleares fora de foco.

regiões nucleares obtida na etapa de segmentação. As bordas dos núcleos foram destacadas por meio da operação de gradiente morfológico, que faz uso das operações de dilatação e erosão. O elemento estruturante utilizado pode ser representado por uma matriz 3x3 com os valores: $[0 \ 1 \ 0 ; 1 \ 1 \ 1 ; 0 \ 1 \ 0]$, que pode ser entendido como uma cruz.



(a) Canal 1 com bordas realçadas

(b) Canal 2 com bordas realçadas

Figura 4 – Lâminas coradas de tecidos de larvas de *Drosophila* com bordas realçadas.

Para realizar a operação de gradiente morfológico, utilizamos as funções *imdilate* e *imerode* do MATLAB. A imagem resultante é a diferença entre a dilatação e a erosão da imagem original, que representa as bordas da imagem, que é então armazenada na variável *border*. Para realçar as bordas na imagem, utilizamos a função $im_ch0_bd(border > 0) = 255$, que define o valor máximo de intensidade para os pixels correspondentes às bordas. Alterando o valor máximo no *color map* da imagem para branco é possível visualizar as bordas sobre as imagens resultantes para os canais 1 e 2.

3.8 Quantificação da intensidade

A quantificação da intensidade de proteína nos núcleos celulares é uma etapa essencial para compreender a expressão dessas proteínas. Para realizar a quantificação foi utilizada a máscara obtida na etapa de segmentação que permitiu, primeiramente, calcularmos a média da intensidade dos pixels na área de cada região nuclear no Canal 2 da imagem. Para isso, utilizamos a função *regionprops* do MATLAB, que nos permite obter as informações de área e intensidade média de cada região nuclear. A intensidade média de cada núcleo celular é armazenado em um vetor para posteriormente serem salvos.

Além disso, geramos um histograma da frequência da intensidade média das regiões nucleares na imagem da amostra. O histograma é uma representação gráfica da distribuição de frequência de dados (GONZALEZ; WOODS, 2008) e nos permitiu visualizar a distribuição de intensidade de proteínas nos núcleos celulares de cada imagem. Para gerar o histograma, utilizamos a função *mnthist* do MATLAB, que nos permitiu especificar os vetores de intensidade média, bem como o título e as legendas do eixo *x* e *y*.

Os dados das médias de intensidade foram salvos em um arquivo CSV com o nome do arquivo da imagem e o sufixo *_data.CSV*. O arquivo CSV contém as informações de identificador do núcleo, intensidade média, área, em pixel, e *bounding box* de cada núcleo celular identificado. Esse arquivo foi útil para a análise posterior dos dados obtidos. A Figura 5 mostra exemplos de histogramas gerados a partir dos dados de intensidade média das regiões nucleares da amostra.

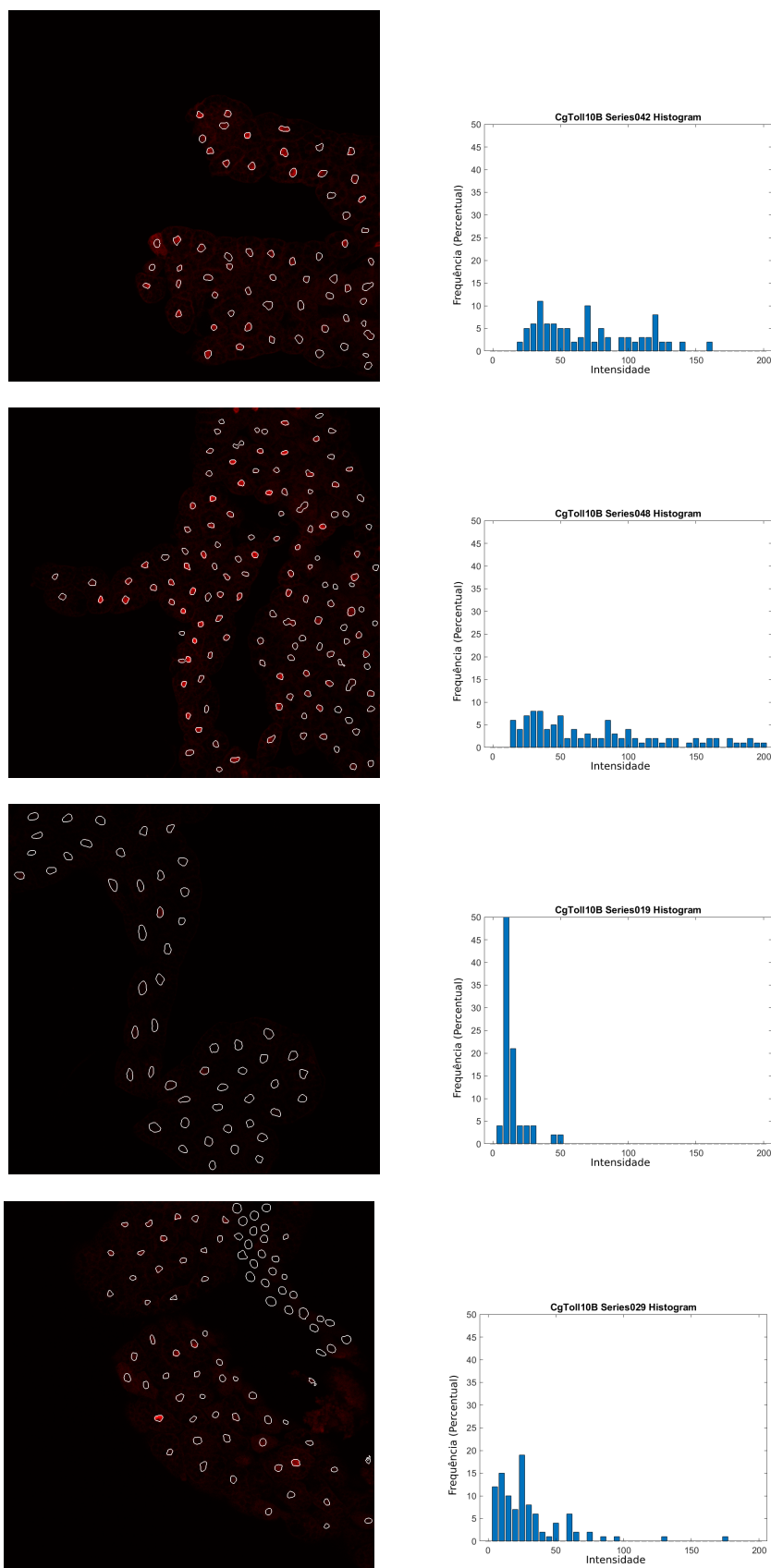


Figura 5 – Regiões de expressão da proteína e seus, respectivos, histogramas de frequência de intensidade de proteína de diferentes amostras.

3.9 Desenvolvimento da Ferramenta de Visualização

A ferramenta de visualização foi desenvolvida usando a biblioteca D3.js e permite que o usuário visualize várias imagens e seus dados associados.

Todos os dados e imagens são armazenados juntamente com o código em um repositório privado no GitHub. E são carregados de acordo com a utilização da ferramenta. Ao carregar a página, o código Javascript é executado, onde, preenchemos a seleção de grupo de coleta com opções, que são os nomes dos grupos. E inserimos os dados associados à imagem atual. Os nomes dos grupos de coleta são salvos em vetor de *strings* como uma constante global.

A implementação foi feita seguindo um paradigma funcional e orientação a eventos. A organização do código está dividido em funções bem definidas, cada uma com uma responsabilidade específica. O código desenvolvido utilizou recursos da linguagem JavaScript para manipulação do DOM, juntamente com a biblioteca D3.js, para criar e manipular elementos HTML e CSS, bem como adicionar comportamentos dinâmicos, como as funções *mouseover* e *mouseout*. A seguir descrevemos o funcionamento do código e funções desenvolvidos para a implementação das funcionalidades da ferramenta.

Foi criada uma função *getCurrentSample*. Esta função recebe o tipo do grupo de coleta de imagens como parâmetro e retorna um objeto que contém as informações relevantes para as imagens desse grupo. Essas informações incluem as imagens com bordas realçadas dos dois canais, a máscara, o histograma, os dados associados às imagens e o número total de imagens no grupo.

Foi criada uma função *changeImage*. Essa função agrupa chamadas para outras três funções: *removeData*, *insertImages* e *insertData*. Essas funções são descritas a seguir. A função *removeData* remove todas as imagens e dados atuais da página. Enquanto a função *insertImages* insere as novas imagens na página, que são as imagens de bordas realçada do canal 1 e 2, a máscara e o histograma.

A função *insertData* implementa a funcionalidade de inserir um *tooltip* contendo informações sobre uma imagem específica, que é carregada do arquivo CSV. Esta função insere os dados associados à imagem atual na página, são eles a área da região, a intensidade da proteína, os parâmetros de delimitação da caixa, *bounding box*, e o identificador do núcleo.

A função utiliza a biblioteca D3.js para selecionar o elemento com o id *data* e adicionar um novo elemento HTML *div* com a classe *tooltipdiv*. Em seguida, a posição e as dimensões do *tooltip* são definidas com base nos parâmetros de delimitação da caixa presentes no arquivo CSV. Após a criação do *tooltip*, são adicionados três elementos HTML *p* contendo informações sobre o núcleo, a intensidade e a área da imagem. Por fim, são definidos dois eventos para o *tooltip*: *mouseover* e *mouseout*. Quando o cursor do mouse

está sobre o *tooltip*, ele é exibido mostrando as informações contidas nos elementos HTML *p*. Quando o cursor é removido do *tooltip*, as informações são escondidas novamente.

Depois disso, definimos três eventos usando a biblioteca D3.js. O primeiro evento é acionado quando o botão *next* é clicado aumentando o índice da imagem atual em um. Se o índice exceder o tamanho total das imagens no grupo, o índice é definido como zero. O segundo evento é acionado quando o botão *prev* é clicado diminuindo o índice da imagem atual em um. Se o índice for menor que zero, o índice é definido como o tamanho total das imagens menos um. O terceiro evento é acionado quando a seleção de grupo é alterada e define o grupo de amostras atual como o valor da seleção. O índice da imagem atual é definido como zero. Em todos os eventos a função *changeImage* é chamada para atualizar a visualização com as novas imagens e dados.

Após o desenvolvimento, foi necessário realizar a publicação em uma plataforma que possibilitasse o acesso público e de fácil compartilhamento. Para isso, optamos pelo *GitHub Pages*, que permite a hospedagem de sites estáticos diretamente de um repositório no *GitHub*.

4 Resultados

Esta seção tem o propósito de apresentar os resultados obtidos na aplicação da método de segmentação e análise dos núcleos, bem como mostrar o resultado final da ferramenta de visualização.

4.1 Imagens obtidas

Os resultados obtidos mostraram que a segmentação de imagens e a remoção de regiões nucleares foram fundamentais para a análise das imagens. A limiarização iterativa permitiu obter uma máscara que destacava claramente as regiões de interesse. Ao todo dos 10 grupos de coleta processados 9 tiveram resultados muito bem definidos e precisos.

Com exceção de um dos grupo de coleta, *Cg_+_Myd* que não apresentou bons resultados na sua segmentação. Como mostra Figura 6, a rotina teve dificuldade de identificar as regiões internas dos núcleos. Essa dificuldade pode ter ocorrido devido aos parâmetros utilizados para esse conjunto de dados ou à falta de um pré-processamento ou de pós-processamento mais rígido para esse conjunto, que apresenta diferenças em relação aos outros conjuntos de dados processados. Como é possível visualizar, suas regiões internas não tem um preenchimento bem definido.

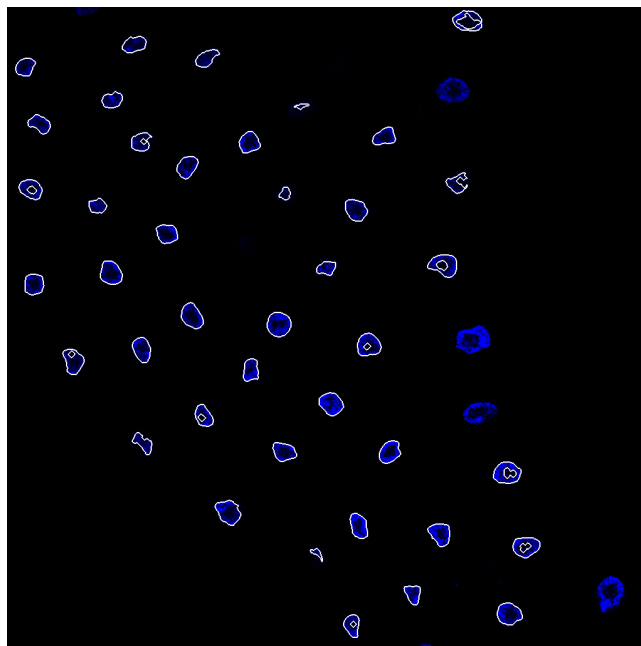


Figura 6 – Amostra do grupo *Cg_+_Myd*.

Também, a remoção das regiões nucleares foi um passo importante para garantir

a precisão da análise de dados. O intervalo definido entre 100 a 2500 unidades para filtrar as regiões nucleares mostrou-se eficiente para eliminar regiões com áreas muito pequenas ou muito grandes, que não eram relevantes para a análise.

A determinação dos valores de área em pixels utilizados para filtrar as regiões nucleares foi obtida de forma empírica após uma análise inicial das imagens. No entanto, a utilização de métodos mais objetivos, como a análise de histogramas, pode ser uma abordagem interessante para a definição desses valores de forma mais precisa e reproduzível.

Sobre o realce das bordas, os resultados obtidos demonstraram que a técnica foi eficiente para detecção das bordas nas imagens, o que pode facilitar a identificação e análise das imagens em trabalhos posteriores.

Por fim, como resultado foram geradas 1032 imagens, são elas: as máscaras dos núcleos celulares, as imagens com bordas realçadas do Canal 1 e do Canal 2, e os histogramas de frequência de intensidade. Um exemplo pode ser visualizados na Figura 7.

Também, recebemos *feedbacks* positivos das pesquisadoras da UFRJ sobre os resultados, que ficaram muito satisfeitas com os resultados obtidos na primeira versão e solicitaram o uso da ferramenta em novos conjuntos de dados.

4.2 Extração das Propriedades dos Núcleos

Foram coletados e armazenados diversos propriedades das regiões nucleares identificadas em cada imagem. Essas propriedades incluem informações como a área da região, a intensidade da proteína, os parâmetros de delimitação da caixa de borda e um identificador exclusivo para cada região nuclear. Os parâmetros de delimitação da caixa de borda e os identificadores desses dados foram utilizados para melhorar a identificação dos núcleos na ferramenta de visualização. Para facilitar a manipulação, os dados coletados foram armazenados no formato CSV. Esses dados serão utilizados para análise posteriormente pelas pesquisadoras da UFRJ para avaliação da influência das proteínas no desenvolvimento da *Drosophila*.

4.3 Ferramenta de visualização

Como um dos resultados deste trabalho podemos citar a criação da ferramenta para visualização dos resultados. Essa ferramenta pode ser utilizada para interação com os resultados em trabalhos posteriores. A seguir descrevemos as funcionalidades que a ferramenta permite executar e uma visão da sua interface.

Entre as funcionalidades estão: (i) a exibição interativa de informações sobre cada

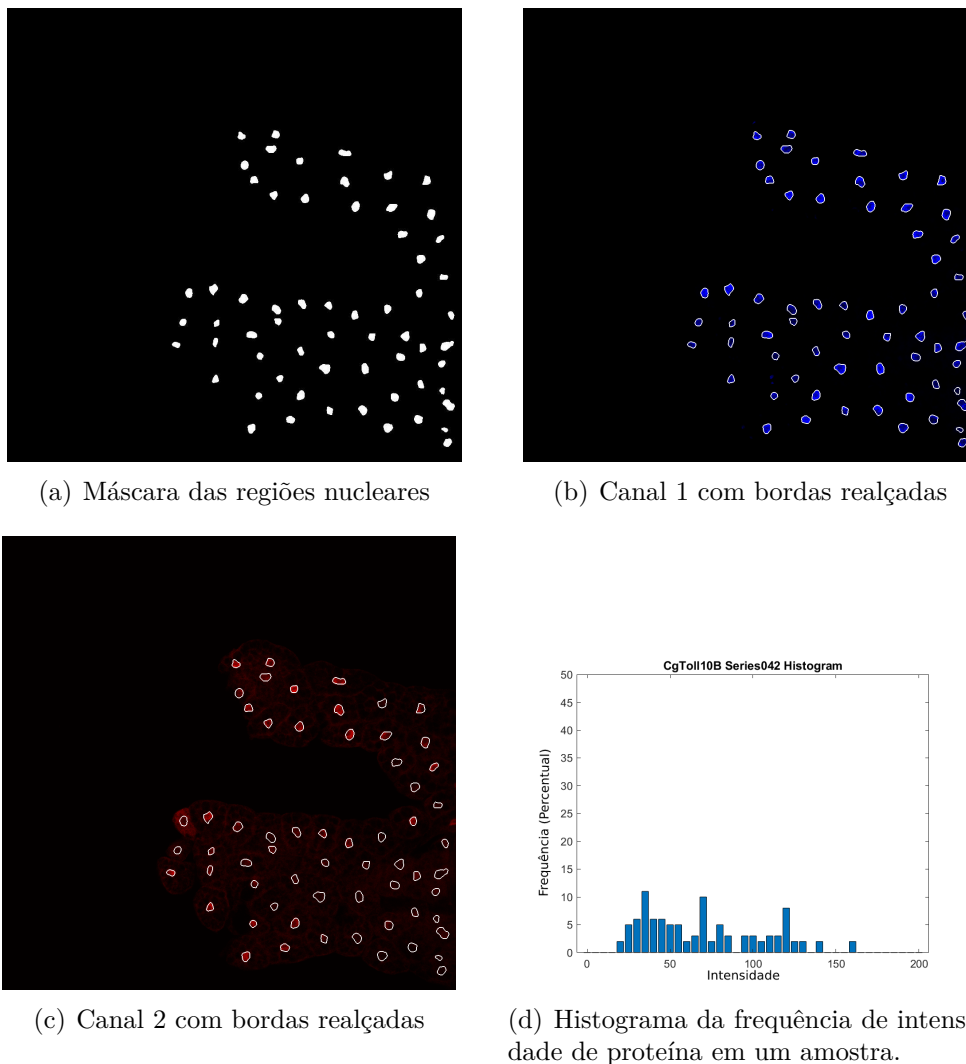


Figura 7 – Resultado do processo de segmentação e análise dos núcleos.

região nuclear ao posicionar o cursor sobre a região; (ii) a capacidade de visualizar as imagens de Canal 1 e Canal 2, máscaras da região nuclear e histograma; (iii) a opção de segregar e selecionar grupos de amostras por tipo de proteína; (iv) e a capacidade de navegar entre grupos de amostras da mesma série. Essas etapas são detalhadas nos próximos parágrafos.

(i) A partir dos parâmetros de posição da caixa de borda coletados, foi implementada a funcionalidade de exibição de informações sobre cada região nuclear quando posicionado o cursor sobre elas nas imagens de Canal 2. Esta funcionalidade consiste na apresentação de uma caixa de texto contendo informações relevantes sobre a região nuclear, tais como o seu identificador, área e intensidade da proteína presente. Essa interação pode ser visualizada na área destacada em vermelho da Figura 8.

(ii) Essas visualizações são apresentadas na área principal da página web, com a imagem de Canal 2 sendo exibida na maior parte à esquerda, seguida da imagem de Canal

1, como é possível visualizar na área destacada em azul da Figura 8. Ao lado das imagens, há um histograma que representa a frequência de intensidade das regiões nucleares na imagem da amostra juntamente com a máscara das regiões nucleares.

(iii) A opção de segregar e selecionar grupos de amostras por tipo de proteína permite ao usuário filtrar e selecionar apenas as amostras de interesse para análise. Isso é feito por um de um menu de seleção que lista todos os grupos de amostras. Ao selecionar um grupo específico, a ferramenta exibe apenas as amostras referentes a aquele grupo.

(iv) Ao lado do menu de seleção também há botões para navegação entre as séries de imagens permitindo a visualização individual dos resultados de cada amostra do grupo selecionado. Ambas as interações, (iii) e (iv), podem ser visualizadas na área destacada em amarelo da Figura 8.

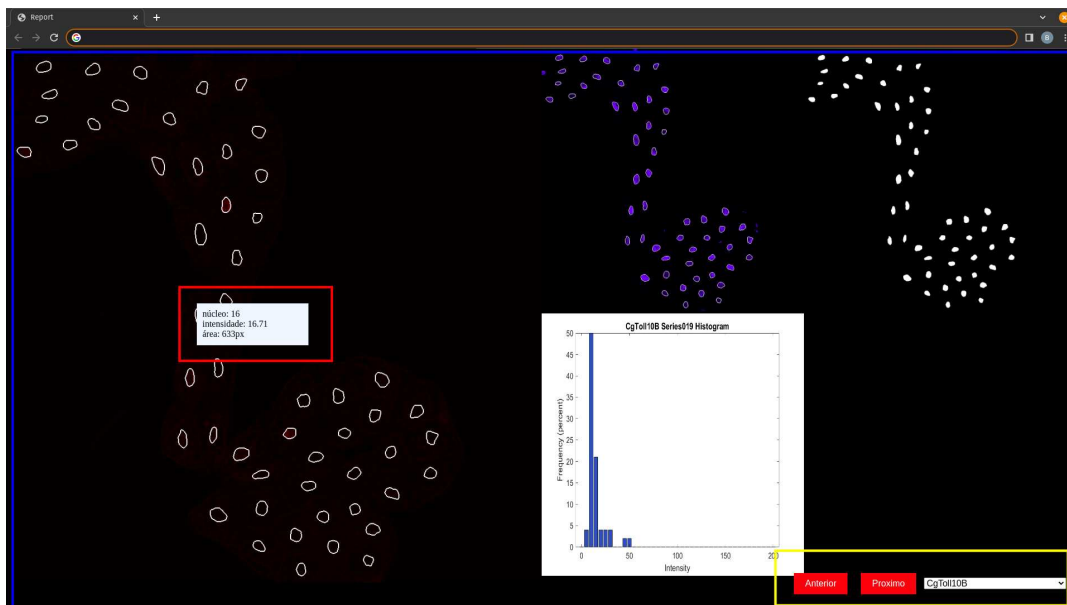


Figura 8 – Ferramenta Web de visualização.

5 Conclusão

Este trabalho teve como objetivo principal desenvolver uma rotina computacional para segmentar imagens de lâminas contendo tecidos de larvas de *Drosophila melanogaster*, quantificar as proteínas presentes nas regiões nucleares e criar uma ferramenta interativa de visualização dos resultados. Durante o processo de desenvolvimento, foi necessário entender conceitos fundamentais de processamento de imagens, como limiarização, segmentação e operações morfológicas. Além disso, foi preciso aplicar esses conceitos em um contexto real, lidando com imagens reais e resolvendo problemas práticos de segmentação e extração de dados. Também permitiu a aplicação e aprendizado na área de desenvolvimento Web com o desenvolvimento da ferramenta de visualização.

A principal conclusão deste trabalho foi o sucesso no desenvolvimento de uma rotina computacional eficiente para a segmentação de imagens de lâminas coradas com tecidos nucleares de larvas de *Drosophila*, bem como na quantificação das proteínas presentes nas regiões nucleares. Além disso, a ferramenta interativa de visualização desenvolvida permite uma melhor compreensão e análise dos resultados obtidos, tornando possível identificar padrões e tendências na expressão proteica.

Esse processo permitiu uma compreensão mais profunda dos conceitos teóricos e sua aplicação em um ambiente de pesquisa real, o que é muito valioso para a formação acadêmica e profissional.

5.1 Trabalhos Futuros

5.1.1 Melhoria da Ferramenta de Visualização

A diversas melhorias que poderiam ser desenvolvidas para a ferramenta de visualização. É possível criar um sistema com interação com banco de dados e APIs para fornecer recursos adicionais, como a capacidade de filtrar imagens, área da região, intensidade da proteína ou outras informações coletadas. Além disso, pode-se incluir opções para manipulação de imagens, como rotação, ampliação ou ajuste de contraste, para melhor visualização e análise dos dados.

5.1.2 Melhoria das Técnicas de Processamento

As técnicas de processamento de imagem utilizadas poderiam ser aprimoradas para o grupo de coleta que não teve bons resultados. Uma possível abordagem seria testar a operação morfológica de fechamento para melhorar a identificação das regiões internas dos

núcleos e, conseqüentemente, a segmentação da imagem. Além disso, outras técnicas de processamento de imagem, como filtragem adaptativa ou a utilização de diferentes algoritmos de limiarização, podem ser exploradas para melhorar a qualidade da segmentação.

5.1.3 Pesquisas Relacionadas à Base de Dados Coletados

É importante explorar mais profundamente a base de dados analisada e buscar correlações entre as informações coletadas. Por exemplo, pode-se investigar se existe alguma relação entre a intensidade da proteína e a área da região ou se há diferenças significativas entre os grupos de coleta em relação à expressão da proteína analisada. Além disso, é possível buscar outras fontes de dados para comparar com as informações coletadas neste estudo e expandir a análise para diferentes espécies ou tecidos.

Referências

- CHACON, S.; STRAUB, B. *Pro Git*. Apress, 2014. Disponível em: <<https://git-scm.com/book/pt-br/v2>>. Cited on page 14.
- D3.JS. *D3.js*. 2023. Disponível em: <<https://d3js.org/>>. Cited on page 14.
- GITHUB. *About GitHub Pages*. 2021. Disponível em: <<https://docs.github.com/en/pages/getting-started-with-github-pages/about-github-pages>>. Cited on page 15.
- GITHUB. *Hello World - GitHub Docs*. Accessed: 2023. <<https://docs.github.com/en/get-started/quickstart/hello-world>>. Cited on page 15.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. [S.l.]: Pearson, 2008. Cited 7 times on pages 8, 10, 11, 12, 13, 18 e 21.
- MATHWORKS, I. T. *Read image from graphics file - MATLAB imread*. 2022. Disponível em: <https://www.mathworks.com/help/matlab/ref/imread.html?s_tid=doc_ta>. Acesso em: 10 jan. 2023. Cited 2 times on pages 13 e 17.
- MOZILLA. *CSS*. 2021. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/CSS>>. Cited on page 14.
- MOZILLA. *HTML*. 2021. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/HTML>>. Cited on page 13.
- MOZILLA. *JavaScript*. 2021. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/JavaScript>>. Cited on page 14.
- SOUSA, D. J. de; CARDOSO, M. A.; BISCH, P. M.; LOPES, F. J. P.; TRAVENÇOLO, B. A. N. Automated standardization of images of *Drosophila* embryos. *Journal of Visual Communication and Image Representation*, v. 71, p. 102758, 2020. ISSN 1047-3203. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1047320320300080>>. Cited on page 8.
- TRAVENÇOLO, B. A. N. *Métodos computacionais para a caracterização e análise da relação entre anatomia e expressão gênica em sistemas biológicos*. Tese (Doutorado) — Universidade de São Paulo, 2007. Cited on page 8.
- WIKIPÉDIA. *Comma-separated values* — *Wikipédia, a enciclopédia livre*. 2018. Disponível em: <https://pt.wikipedia.org/w/index.php?title=Comma-separated_values&oldid=53784594>. Acesso em: 10 jan. 2023. Cited on page 14.