

UNIVERSIDADE FEDERAL DO PARANÁ

DANIEL FRANCELINO RIBEIRO

MORFOMETRIA DE OTÓLITOS DE PEIXES:
UMA NOVA FERRAMENTA PARA ESTUDOS APLICADOS.

PALOTINA - PR

2025

DANIEL FRANCELINO RIBEIRO

MORFOMETRIA DE OTÓLITOS DE PEIXES:
UMA NOVA FERRAMENTA PARA ESTUDOS APLICADOS.

Dissertação apresentada ao curso de Pós-Graduação em Aquicultura e Desenvolvimento Sustentável, Setor de Palotina, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Mestre em Aquicultura e Desenvolvimento Sustentável.

Orientador: Prof. Dr. André Martins
Vaz-dos-Santos

PALOTINA - PR

2025

Universidade Federal do Paraná. Sistemas de Bibliotecas.
Biblioteca UFPR Palotina.

R484 Ribeiro, Daniel Francelino
 Morfometria de otólitos de peixes: uma nova ferramenta para
 estudos aplicados / Daniel Francelino Ribeiro.
 – Palotina, PR, 2025.

 Dissertação (Mestrado) – Universidade Federal do Paraná,
 Setor Palotina, PR, Programa de Pós-Graduação em Engenharia
 e Tecnologia Ambiental.

 Orientador: Prof. Dr. André Martins Vaz-dos-Santos.

 1. Métodos automatizados. 2. Morfometria. 3. Peixes.
 I. Vaz-dos-Santos, André Martins.
 II. Universidade Federal do Paraná. III. Título.

CDU 62

Bibliotecária: Aparecida Pereira dos Santos – CRB 9/1653



MINISTÉRIO DA EDUCAÇÃO
SETOR PALOTINA
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO AQUICULTURA E
DESENVOLVIMENTO SUSTENTÁVEL - 40001016078P2

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação AQUICULTURA E DESENVOLVIMENTO SUSTENTÁVEL da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **DANIEL FRANCELINO RIBEIRO**, intitulada: **MORFOMETRIA DE OTÓLITOS DE PEIXES: UMA NOVA FERRAMENTA PARA ESTUDOS APLICADOS**, sob orientação do Prof. Dr. ANDRÉ MARTINS VAZ DOS SANTOS, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa. A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

PALOTINA, 12 de Setembro de 2025.

Assinatura Eletrônica

12/09/2025 15:48:53.0

ANDRÉ MARTINS VAZ DOS SANTOS

Presidente da Banca Examinadora

Assinatura Eletrônica

12/09/2025 15:50:20.0

MARIELE PASUCH DE CAMARGO

Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica

18/09/2025 11:21:39.0

MÁRCIA SANTOS DE MENEZES

Avaliador Externo (UNIVERSIDADE FEDERAL DO PARANÁ)

“Errar é humano, persistir no erro é diabólico.”

(SÊNECA, 65 d.C)

RESUMO

O presente trabalho aborda a crescente demanda por métodos eficientes e reprodutíveis na morfometria de otólitos de peixes, uma ferramenta crucial em estudos de idade, crescimento e identificação populacional. Tradicionalmente, a obtenção de medidas morfométricas tem se baseado em métodos manuais ou semi-automatizados, que demandam tempo considerável e podem comprometer a reprodutibilidade dos resultados. Diante disso, esse trabalho teve como objetivo desenvolver e validar o MORFO (Morfometria de Otólitos de Peixes), um programa com interface gráfica em linguagem Python para a extração automatizada de medidas morfométricas de otólitos. O estudo comparou o desempenho do método tradicional (M1) e do MORFO (M2) utilizando otólitos de sete espécies de peixes teleósteos. As análises estatísticas, incluindo testes de regressão potencial e comparações pareadas, revelaram que ambos os métodos são capazes de gerar dados com alta capacidade preditiva em relação ao tamanho corporal dos indivíduos ($R^2 > 0,95$). No entanto, as análises inferenciais apontaram para diferenças sistemáticas significativas entre M1 e M2, com o MORFO demonstrando maior reprodutibilidade, menor suscetibilidade a vieses observacionais e, em muitos casos, precisão ligeiramente superior. As vantagens do MORFO incluem a otimização do tempo de análise, a padronização das medições, a acessibilidade (sendo uma ferramenta gratuita e de código aberto) e a facilidade de uso, que permite a delegação de tarefas a alunos de graduação e iniciantes, democratizando o acesso a análises morfométricas avançadas. A ferramenta gera arquivos compatíveis com softwares de morfometria geométrica, ampliando suas aplicações. Conclui-se que o MORFO representa um avanço significativo para a morfometria de otólitos, posicionando-se como a abordagem mais indicada para estudos em larga escala e colaborativos, e abrindo novas perspectivas para a pesquisa em biologia pesqueira e áreas correlatas.

Palavras-chave: Otólitos; Morfometria; Métodos automatizados; Análise de imagem; Peixes.

ABSTRACT

This study addresses the growing demand for efficient and reproducible methods in fish otolith morphometry, a crucial tool in age, growth, and population identification studies. Traditionally, morphometric measurements have relied on manual or semi-automated methods, which, although effective, are time-consuming, and can compromise result reproducibility. Therefore, this dissertation aimed to develop and validate MORFO (Morphometry of Fish Otoliths), a graphical interface program in Python for automated otolith morphometric measurement extraction. The study compared the performance of MORFO (M2) with the traditional method (M1) using otoliths from seven teleost fish species. Statistical analyses, including potential regression tests and paired comparisons, revealed that both methods are capable of generating data with high predictive power regarding individual body size ($R^2 > 0.95$). However, inferential analyses indicated significant systematic differences between M1 and M2, with MORFO demonstrating greater reproducibility, lower susceptibility to observational biases, and, in many cases, slightly higher precision. The advantages of MORFO include optimized analysis time, measurement standardization, accessibility (being a free and open-source tool), and ease of use, which allows task delegation to undergraduate students and beginners, democratizing access to advanced morphometric analyses. The tool generates files compatible with geometric morphometry software, expanding its applications. It is concluded that MORFO represents a significant advance for otolith morphometry, positioning itself as the most suitable approach for large-scale and collaborative studies, and opening new perspectives for research in fisheries biology and related fields.

Keywords: Otoliths; Morphometry; Automated Methods; Image Analysis; Fishes.

LISTA DE FIGURAS

- FIGURA 1 – OTÓLITOS SAGITTAE DAS SETE ESPÉCIES UTILIZADAS. (A) *ATHERINELLA BRASILIENSIS*; (B) *CHLOROSCOMBRUS CHRYSURUS*; (C) *CYNOSCION JAMAICENSIS*; (D) *DIAPTERUS RHOMBEUS*; (E) *EUCINOSTOMUS ARGENTEUS*; (F) *HAEMULOPSIS CORVINAEFORMIS*; (G) *MUGIL CUREMA*. 15
- FIGURA 2 – ETAPAS DO TRATAMENTO MANUAL E PROCESSAMENTO DIGITAL DE IMAGENS NO PROGRAMA MORPHO, EXEMPLIFICADAS COM UM OTÓLITO DE *MUGIL CUREMA*. (A) IMAGEM ORIGINAL DO OTÓLITO; (B) IMAGEM TRATADA COM FUNDO PRETO; (C) IMAGEM PRÉ-PROCESSADA; (D) IMAGEM PROCESSADA NO PROGRAMA MORFO. 17
- FIGURA 3 – MENSURAÇÃO DAS MEDIDAS MORFOMÉTRICAS DE CO (AMARELO) E AO (VERMELHO) DO OTÓLITO DE *MUGIL CUREMA* NO IMAGEJ. 18
- FIGURA 4 – TELA DE SELEÇÃO DE DIRETÓRIO NO SOFTWARE MORFO PARA INÍCIO DO PROCESSAMENTO DAS IMAGENS DE OTÓLITOS. 19
- FIGURA 5 – TELA PARA SELEÇÃO DE ARQUIVO .TPS NO SOFTWARE MORFO. 21
- FIGURA 6 – TELA PARA INSERÇÃO DO VALOR DA ESCALA EM MILÍMETROS NO SOFTWARE MORFO. 21
- FIGURA 7 – INTERFACE PRINCIPAL DO SOFTWARE MORFO. 24

LISTA DE GRÁFICOS

- GRÁFICO 1 – VARIAÇÃO DOS PARÂMETROS DAS REGRESSÕES EM FUNÇÃO DA QUANTIDADE DE PONTOS HOMÓLOGOS UTILIZADOS. 25
- GRÁFICO 2 – *ANÁLISE COMPARATIVAS COM REGRESSÃO POTENCIAL DE COMPRIMENTO TOTAL (Ct, mm) E MEDIDAS MORFOMÉTRICAS (Aro e Co, mm) DOS OTÓLITOS DE ATHERINELLA BRASILIENSIS E DIAPTERUS RHOMBEUS.* 27
- GRÁFICO 3 – *ANÁLISE COMPARATIVAS COM REGRESSÃO POTENCIAL DE COMPRIMENTO TOTAL (Ct, mm) E MEDIDAS MORFOMÉTRICAS (Ao, mm) DOS OTÓLITOS DE CHLOROSCOMBRUS CHRYSURUS E CYNOSCION JAMAICENSIS.* 28
- GRÁFICO 4 – *ANÁLISE COMPARATIVAS COM REGRESSÃO POTENCIAL DE COMPRIMENTO TOTAL (Ct, mm) E MEDIDAS MORFOMÉTRICAS (Aro e Co, mm) DOS OTÓLITOS DE EUCINOSTOMUS ARGENTEUS E DIAPTERUS RHOMBEUS.* 30
- GRÁFICO 5 – *ANÁLISE COMPARATIVAS COM REGRESSÃO POTENCIAL DE COMPRIMENTO TOTAL (Ct, mm) E MEDIDAS MORFOMÉTRICAS (Ao, mm) DOS OTÓLITOS DE CHLOROSCOMBRUS CHRYSURUS E CYNOSCION JAMAICENSIS.* 31

LISTA DE TABELAS

TABELA 1 – <i>SOFTWARES</i> GRATUITOS E PAGOS UTILIZADOS PARA MENSURAÇÃO MORFOMÉTRICA DE OTÓLITOS.	12
TABELA 2 – ESTATÍSTICAS DESCRITIVAS DO COMPRIMENTO TOTAL (Ct, em mm) DAS ESPÉCIES ANALISADAS.	14
TABELA 3 – PARÂMETROS DAS EQUAÇÕES DE REGRESSÃO POTENCIAL ENTRE O COMPRIMENTO TOTAL (CT) E MEDIDAS MORFOMÉTRICAS DOS OTÓLITOS (ARO, PO, CO E AO) OBTIDOS PELO MÉTODO TRADICIONAL (M1).	28
TABELA 4 – PARÂMETROS DAS EQUAÇÕES DE REGRESSÃO POTENCIAL ENTRE O COMPRIMENTO TOTAL (CT) E MEDIDAS MORFOMÉTRICAS DOS OTÓLITOS (ARO, PO, CO E AO) OBTIDOS PELO MORFO (M2).	32
TABELA 5 – COMPARAÇÃO DE ESTATÍSTICA DESCRITIVA ENTRE DOIS MÉTODOS DE MEDIÇÃO MORFOMÉTRICA DE OTÓLITOS POR ESPÉCIE.	33

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS	13
1.1.1	Objetivo geral	13
1.1.2	Objetivos específicos	13
2	MATERIAL E MÉTODOS	14
2.1	SELEÇÃO DE OTÓLITOS DE ESPÉCIES MODELO	14
2.2	AQUISIÇÃO E TRATAMENTO DAS IMAGENS DOS OTÓLITOS	16
2.2.1	OBTENÇÃO DE DADOS PELO MÉTODO TRADICIONAL (M1)	17
2.2.2	DESENVOLVIMENTO DA FERRAMENTA AUTOMATIZADA E OBTENÇÃO DE DADOS (M2)	18
2.2.2.1	CÁLCULO DA ÁREA E PERÍMETRO	21
2.2.2.2	CÁLCULO DO COMPRIMENTO E DA ALTURA	22
2.2.2.3	CONVERSÃO PARA UNIDADES REAIS	22
2.2.3	COMPARAÇÃO ENTRE MÉTODOS	22
3	APRESENTAÇÃO DOS RESULTADOS	24
3.1	MORFO	24
3.2	RESULTADOS OBTIDOS PELO MÉTODO TRADICIONAL (M1)	27
3.3	RESULTADOS OBTIDOS OBTIDAS PELO MORFO (M2)	30
3.4	COMPARAÇÃO ENTRE OS MÉTODOS	33
4	DISCUSSÃO	34
5	CONCLUSÃO	38
	REFERÊNCIAS	40
	ANEXO 1 – SCRIPT EM R PARA ANÁLISE DE DIFERENTES QUANTIDADES DE PONTOS HOMÓLOGOS	44
	ANEXO 2 – SCRIPT EM R PARA COMPARAÇÃO ENTRE MÉTODOS M1 E M2	46
	ANEXO 3 – CÓDIGO FONTE DO PROGRAMA MORFO EM PYTHON	49

1 INTRODUÇÃO

As estruturas calcificadas de peixes — como otólitos, escamas, vértebras, espinhos de nadadeiras e certos ossos — são amplamente utilizadas em estudos de idade e crescimento, por apresentarem anéis de crescimento depositados periodicamente (PEGG et al., 2022). Especialmente os otólitos são formados por cristais de carbonato de cálcio organizados em uma matriz proteica, tendo a vantagem de serem morfologicamente inertes, formato espécie-específico e estrutura (forma e composição química) que podem refletir variações intraespecíficas associadas a diferenças genéticas, fisiológicas e ambientais (STURROCK et al., 2014). Assim, os otólitos funcionam como registros da história de vida dos peixes, sendo úteis na identificação de espécies, diferenciação de populações e estoques, além da inferência de idade (QUIST et al., 2023; CERNA et al., 2025).

De forma simplificada, os estudos com otólitos seguem uma sequência lógica que abrange sua descrição morfológica (qualitativa), caracterização morfométrica (quantitativa), análise de idade e crescimento, e por último são realizados estudos químicos, que destroem a estrutura (VOLPEDO & VAZ-DOS-SANTOS, 2015). A caracterização morfométrica, em especial, atua como ferramenta complementar às descrições morfológicas, além de representar um pré-requisito para análises de idade (SANTOS & VAZ-DOS-SANTOS, 2023).

Atualmente, existem diferentes abordagens para a análise morfométrica, como as morfometrias bivariada e geométrica, baseadas em marcos anatômicos e contornos (ondaletas e análise de Fourier) (TUSET et al., 2016; BIOLÉ et al., 2019). A morfometria bivariada utiliza medidas lineares dos otólitos (comprimento, altura, área e perímetro), sendo amplamente aplicada em análises de crescimento relativo, dimorfismo sexual e diferenciação de estoques (LOMBARTE; TUSET, 2015; VOLPEDO & VAZ-DOS-SANTOS, 2015). Entretanto, a morfometria linear não capta a complexidade da forma, o que pode ser feito com a morfometria geométrica (LOMBARTE; TUSET, 2015).

Em termos de mensuração, medidas como área (Aro), perímetro (Po), comprimento (Co) e altura (Ao) dos otólitos são utilizadas para o ajustes do modelo potencial ($y = a \cdot x^b$), quantificando as proporções entre as medidas e dando suporte a análises multivariadas diversas (LOMBARTE; TUSET, 2015). Um levantamento

bibliográfico dos últimos cinco anos realizado pelo autor revela que a maioria dos estudos que utilizam essas medidas ainda se baseiam em métodos manuais de mensuração, utilizando *softwares* gratuitos como ImageJ (MEDEIROS et al., 2021; ECHRESHAVI et al., 2021; OSMAN et al., 2021; BEZERRA-NETO et al., 2022; SAYGIN et al., 2022; YEDIER et al., 2023a; VEROCAI et al., 2023; CHARMPILA et al., 2024; BEZERRA-NETO et al., 2024), Digimizer (Yedier et al., 2023b), Pixel Pro-2.7 (DELGADO et al., 2023) e TPSdig2 (MACARENA et al., 2024), e softwares pagos como Image Pro Plus (LAM et al., 2021) (TABELA 1).

TABELA 1 – *SOFTWARES* GRATUITOS E PAGOS UTILIZADOS PARA MENSURAÇÃO MORFOMÉTRICA DE OTÓLITOS.

Software	Citações	Tipo
ImageJ	9	Gratuito
Image Pro Plus	1	Pago
Digimizer	1	Gratuito
Pixel Pro-2.7	1	Gratuito
TPSdig2	1	Gratuito

FONTE: O autor (2025).

Apesar de consolidados e amplamente aceitos, esses procedimentos exigem que cada amostra seja analisada individualmente pelo pesquisador, o que torna o processo demorado, sujeito a erro humano e dependente da experiência do operador, o que pode comprometer a reprodutibilidade dos dados entre laboratórios e coleções (ARNQVIST & MÅRTENSSON, 1998). Além disso, o treinamento para iniciantes no estudo de otólitos e a disponibilização de uma maneira padronizada e fácil de uso são relevantes para o processamento de muitas amostras.

Nesse contexto, o desenvolvimento de uma ferramenta automatizada ou semi-automatizada para a obtenção de medidas de otólitos, com níveis de precisão e reprodutibilidade comparáveis aos métodos tradicionais, tem o potencial de padronizar a qualidade das informações armazenadas em bancos de dados de otólitos, otimizar o tempo de análise, reduzir a variação entre observadores e garantir maior consistência entre coleções digitais (EVANS, 2013). Além disso, sua

integração a plataformas *online* de coleções biológicas poderia ampliar significativamente o acesso a esses dados e padronizar procedimentos entre diferentes projetos de pesquisa em âmbito nacional e internacional.

1.1 OBJETIVOS

1.1.1 Objetivo geral

- Desenvolver e validar uma ferramenta automatizada para extração de medidas morfométricas de otólitos de peixes.

1.1.2 Objetivos específicos

- Avaliar a consistência estatística dos resultados obtidos com o método desenvolvido;
- Comparar a eficiência, precisão e reprodutibilidade do novo método em relação ao método tradicional nas extrações de medidas de otólitos;
- Testar a aplicação do novo método em espécies modelo de diferentes hábitos e habitats.

2 MATERIAL E MÉTODOS

2.1 SELEÇÃO DE OTÓLITOS DE ESPÉCIES MODELO

Para o presente estudo foram selecionados os otólitos *sagitta* de sete espécies de peixes teleósteos do Acervo de Estruturas Calcificadas - FishCAST da Universidade Federal do Paraná (VAZ-DOS-SANTOS, 2025). Estas espécies foram selecionadas por disponibilidade e representatividade de diferentes hábitos e habitats, incluindo uma espécie costeira de zona litoral, *Atherinella brasiliensis* (QUOY & GAIMARD, 1825), uma pelágica nerítica com importância comercial, *Chloroscombus chrysurus* (LINNAEUS, 1766); duas pelágicas costeiras, *Diapterus rhombeus* (CUVIER, 1829) e *Eucinostomus argenteus* (BAIRD & GIRARD, 1855); duas demersais, uma mais costeira, *Haemulopsis corvinaeformis* (STEINDACHNER, 1868) e outra de plataforma, *Cynoscion jamaicensis* (VAILLANT & BOCOURT, 1883); e uma costeira-estuarina, *Mugil curema* (VALENCIENNES, 1836). A estrutura em comprimento total (Ct, mm) destas espécies está presente na Tabela 1.

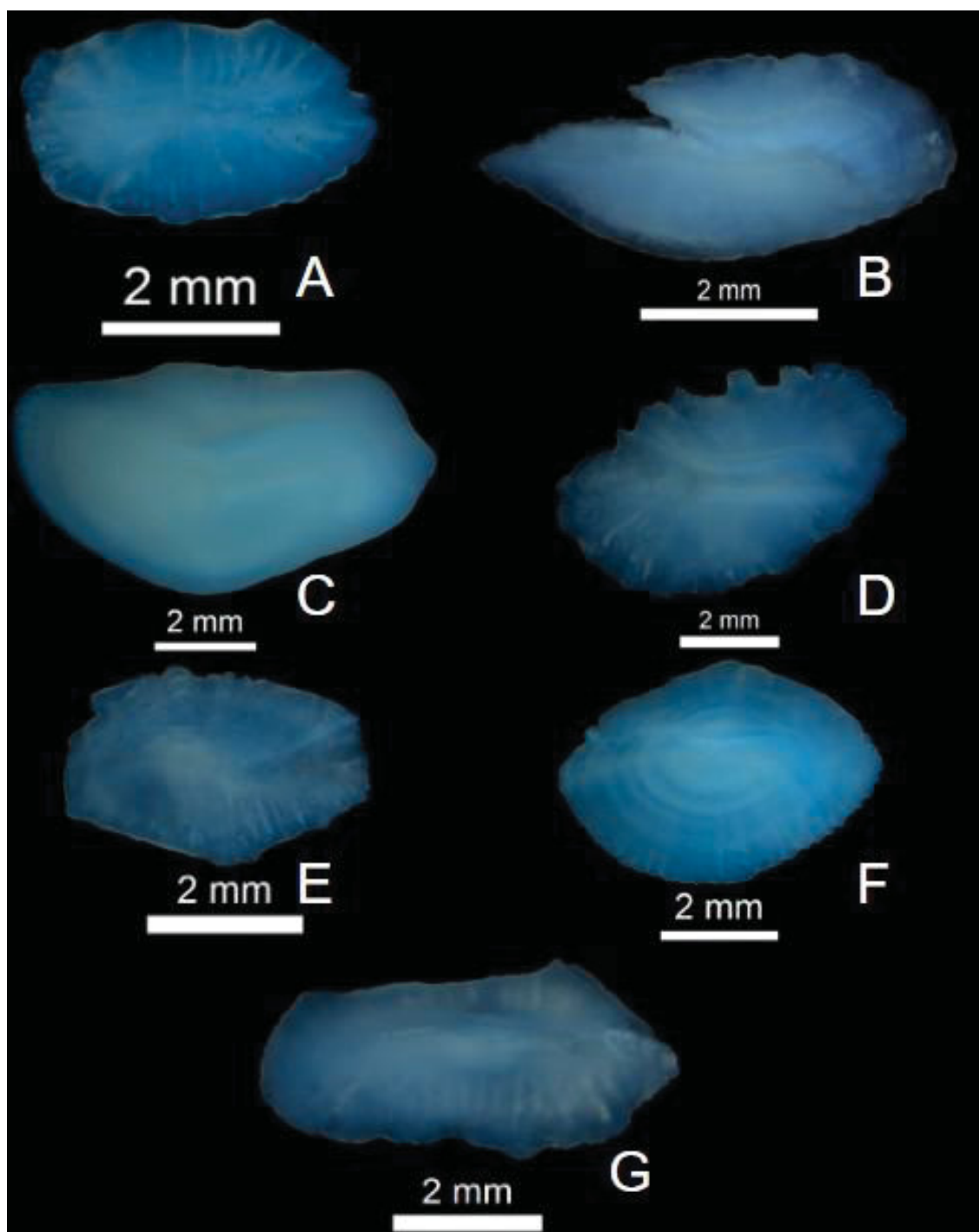
Ao todo, foram selecionados os otólitos de 589 indivíduos para a realização das análises morfométricas (FIGURA 1). A seleção foi realizada por classes de comprimento total, para garantir representatividade morfológica ao longo do gradiente de tamanho. Procurou-se obter ao menos dez indivíduos por classe de comprimento total caso o número de amostras disponíveis permitisse (GOTELLI & ELLISON, 2011; VAZ-DOS-SANTOS & SILVEIRA, 2021).

TABELA 2 – ESTATÍSTICA DESCRITIVA DO COMPRIMENTO TOTAL (Ct, em mm) DAS ESPÉCIES ANALISADAS (s: desvio padrão, n: quantidade de indivíduos).

Comprimento total					
Espécie	mín.	méd.	máx.	s	n
<i>Atherinella brasiliensis</i>	31	90,72	141	28,32	97
<i>Chloroscombus chrysurus</i>	235	326,82	391	37,77	50
<i>Cynoscion jamaicensis</i>	154	195,12	229	16,39	34
<i>Diapterus rhombeus</i>	20	118,26	215	58,37	114
<i>Eucinostomus argenteus</i>	30	85,98	145	28,04	84
<i>Haemulopsis corvinaeformis</i>	75	123,81	176	28,38	90
<i>Mugil curema</i>	36	182,87	389	97,53	119

FONTE: O autor (2025).

FIGURA 1 - OTÓLITOS SAGITTAE DAS ESPÉCIES ANALISADAS (A) *ATHERINELLA BRASILIENSIS*; (B) *CHLOROSCOMBRUS CHRYSURUS*; (C) *CYNOSCION JAMAICENSIS*; (D) *DIAPTERUS RHOMBEUS*; (E) *EUCINOSTOMUS ARGENTEUS*; (F) *HAEMULOPSIS CORVINAEFORMIS*; (G) *MUGIL CUREMA*.



FONTE: O autor (2025).

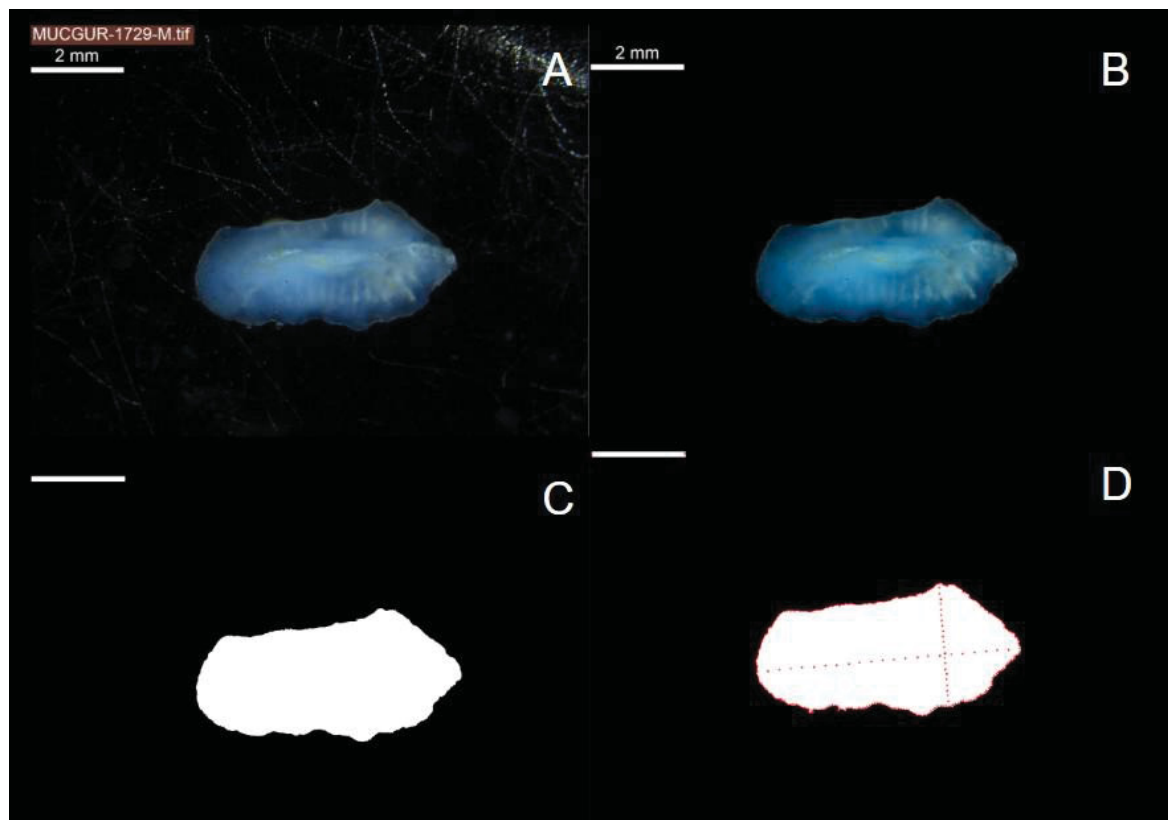
2.2 AQUISIÇÃO E TRATAMENTO DAS IMAGENS DOS OTÓLITOS

Os otólitos *sagitta* foram posicionados individualmente em um recipiente com água destilada e fundo negro, sob luz refletida, de modo a favorecer o contraste visual da estrutura (FIGURA 2-A) (VOLPEDO & VAZ-DOS-SANTOS, 2015). As capturas de imagem iniciaram-se pelo maior indivíduo de cada espécie, a fim de padronizar a escala para aquela espécie.

Foram capturadas imagens da face medial dos otólitos, priorizando o otólito esquerdo (ECHRESHAVI et al., 2021), uma vez que nestas espécies as *sagittae* são simétricas. As imagens foram obtidas com o auxílio de um estereomicroscópio Leica M205 C com câmera digital Leica MC170 HD acoplada, utilizando aumentos entre 10-20x de acordo com a espécie. As imagens foram posteriormente tratadas utilizando o software Adobe Photoshop (versão 22.0.0, Adobe Systems Incorporated, 2020), com o objetivo de realçar o contorno dos otólitos em relação ao fundo negro, garantindo maior precisão na etapa de segmentação e extração dos dados morfológicos (FIGURA 2-B).

As imagens tratadas foram analisadas por meio de dois métodos distintos, um considerando método manual assistido por softwares de imagem (M1) e um outro método automatizado de análise morfométrica (M2).

FIGURA 2 – ETAPAS DO TRATAMENTO MANUAL E PROCESSAMENTO DIGITAL DE IMAGENS NO PROGRAMA MORFO, EXEMPLIFICADAS COM UM OTÓLITO DE *MUGIL CUREMA*. (A) IMAGEM ORIGINAL DO OTÓLITO; (B) IMAGEM TRATADA COM FUNDO PRETO; (C) IMAGEM PRÉ-PROCESSADA; (D) IMAGEM PROCESSADA NO PROGRAMA MORFO.

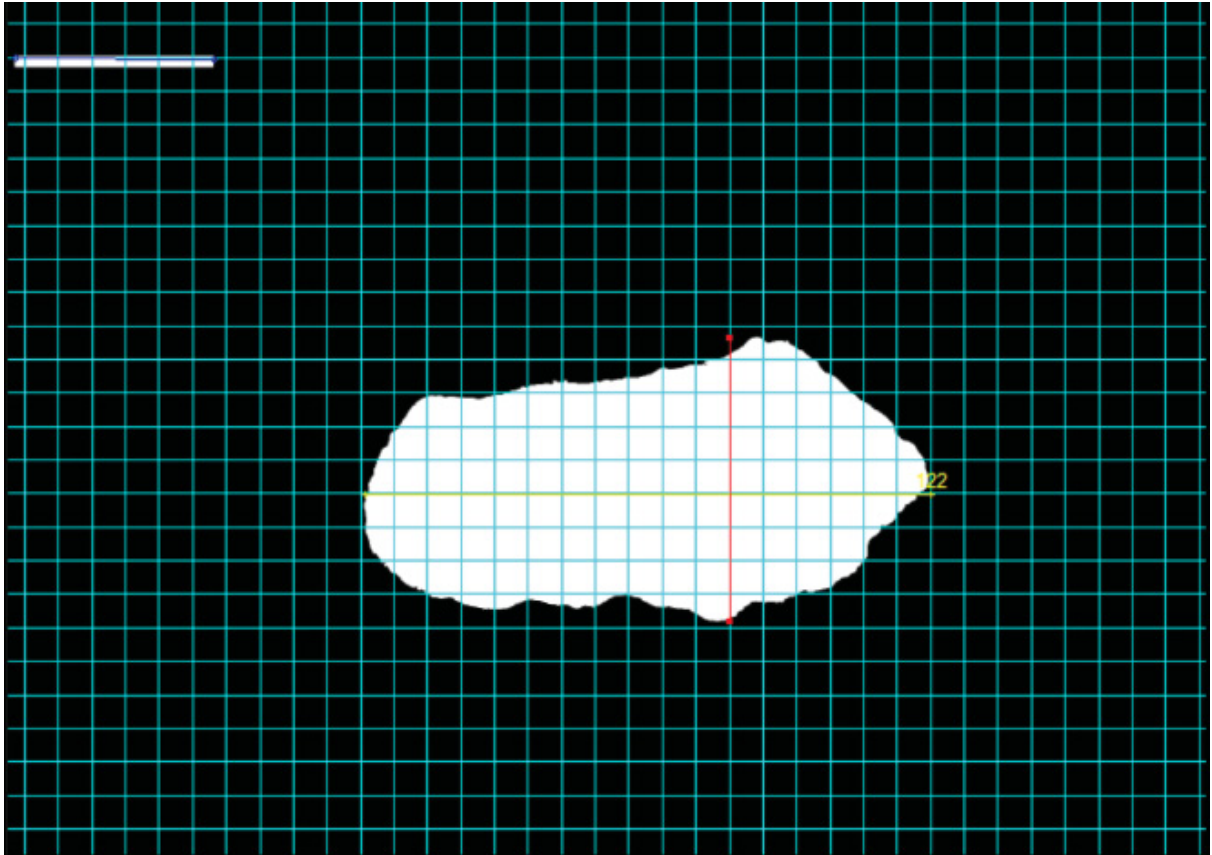


FONTE: O autor (2025).

2.2.1 OBTENÇÃO DE DADOS PELO MÉTODO TRADICIONAL (M1)

No primeiro método (M1), os otólitos foram previamente segmentados no Adobe Photoshop, sendo pintados inteiramente de branco sobre um fundo preto para facilitar a extração das informações morfométricas (FIGURA 2-C). Em seguida, as medidas de área (Aro, mm²) e perímetro (Po, mm) foram obtidas por meio do software Image-Pro Plus (versão 4.5.0.29), empregando-se a ferramenta Magic Wand, que permite a seleção precisa do contorno do otólito com base no contraste da imagem (TIANYANG et al., 2024). As medidas de altura (Ao, mm) e comprimento (Co, mm) dos otólitos foram obtidas com o uso dos softwares ImageJ (versão 1.54g) em conjunto com o plugin ObjectJ (versão 1.05o), utilizando-se a ferramenta grid para a marcação dos pontos de referência (FIGURA 3) (MEDEIROS et al., 2021).

FIGURA 3 – MENSURAÇÃO DAS MEDIDAS MORFOMÉTRICAS DE CO (AMARELO) E AO (VERMELHO) DO OTÓLITO DE *MUGIL CUREMA* NO IMAGEJ.



FONTE: O autor (2025).

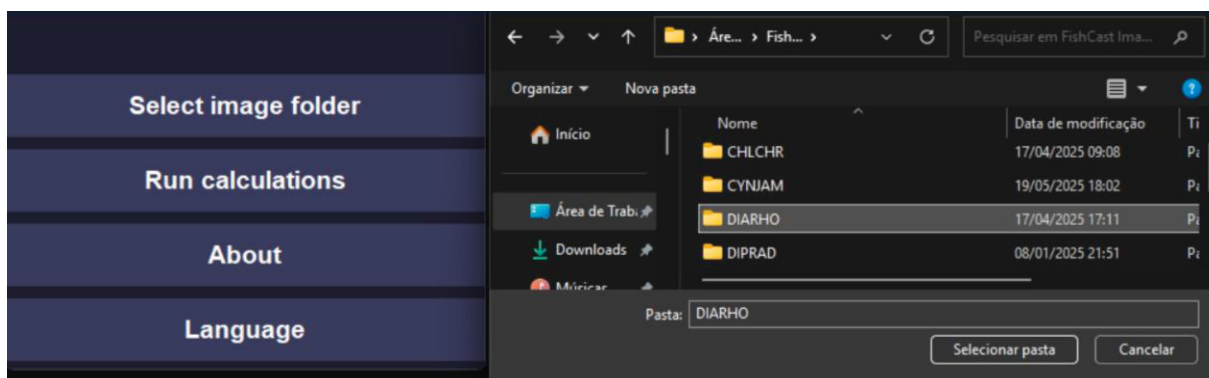
2.2.2 DESENVOLVIMENTO DA FERRAMENTA AUTOMATIZADA E OBTENÇÃO DE DADOS (M2)

Um programa com interface gráfica em linguagem Python (PYTHON SOFTWARE FOUNDATION, 2024), foi desenvolvido para extração de medidas morfométricas de otólitos de forma automatizada.

O software foi estruturado em duas etapas principais: (1) processamento e geração de pontos homólogos digitais e (2) cálculo automatizado das medidas morfométricas. A implementação foi feita com o suporte das bibliotecas OpenCV (BRADSKI, 2000), NumPy (HARRIS et al., 2020), Pandas (REBACK et al., 2020) e tkinter para a interface com o usuário.

Na primeira etapa, o usuário seleciona a pasta contendo os arquivos (.tif) das imagens dos otólitos. Cada imagem é então convertida para tons de cinza e binarizada para destacar o objeto de interesse em contraste com o fundo (FIGURA 4).

FIGURA 4 – TELA DE SELEÇÃO DE DIRETÓRIO NO SOFTWARE MORFO PARA INÍCIO DO PROCESSAMENTO DAS IMAGENS DE OTÓLITOS.



FONTE: Autor (2025).

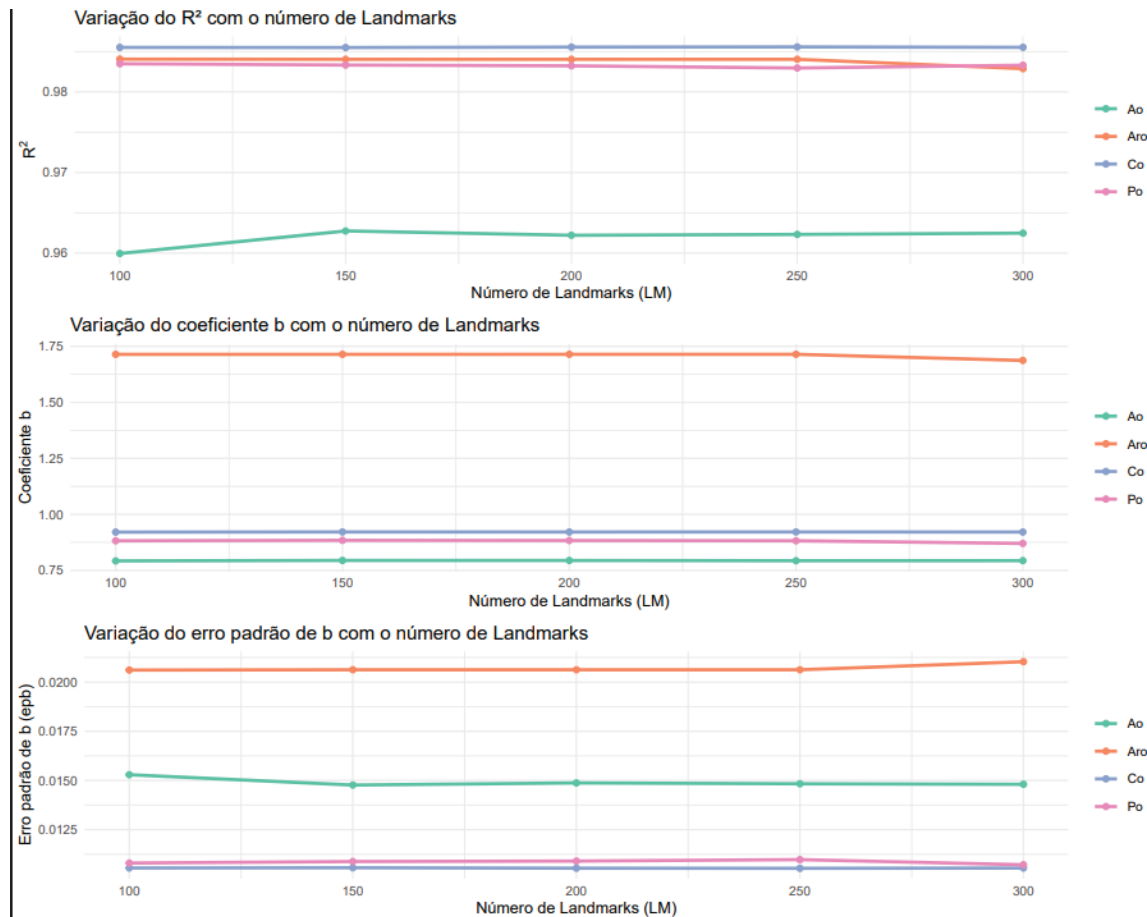
O otólito é identificado com base na detecção do maior contorno centralizado na imagem, enquanto a barra de escala é reconhecida por seu formato alongado, posicionamento na imagem e pela sua cor branca. Ambos os elementos são convertidos em máscaras binárias que, posteriormente, são unificadas em uma única imagem (FIGURA 2-C). Em seguida, o programa realiza o posicionamento automático dos pontos homólogos.

A fim de determinar o número ideal de pontos homólogos a serem utilizados no contorno, foram conduzidas análises de regressão potencial entre as medidas morfométricas (Aro, Po, Co e Ao) e o comprimento total dos indivíduos, utilizando diferentes quantidades de pontos homólogos (100, 150, 200, 250 e 300). Essas análises foram realizadas exclusivamente com as amostras de *Diapterus rhombeus*, por se tratar da espécie com o maior número de amostras disponíveis. As análises estatísticas foram conduzidas no software R, com o auxílio dos pacotes readxl (WICKHAM & BRYAN, 2023), openxlsx (SCHAUBERGER & WALKER, 2023), dplyr (WICKHAM et al., 2023), gridExtra, RColorBrewer e ggplot2 (WICKHAM, 2023).

Os resultados das análises de regressão potencial entre as medidas morfométricas extraídas (Aro, Po, Co e Ao, em mm) e o comprimento total (Ct, em mm) dos indivíduos de *D. rhombeus* indicaram que os valores de R^2 , os coeficientes alométricos (b) e seus erros-padrão (epb) se estabilizaram a partir de 150 pontos homólogos, não havendo ganhos significativos na precisão estatística com o uso de maiores quantidades (GRÁFICO 1). Esses resultados demonstram que o aumento da resolução além desse limite não contribui substancialmente para a qualidade do

ajuste dos modelos, podendo inclusive aumentar o tempo de processamento de forma desnecessária.

GRÁFICO 1 - VARIAÇÃO DOS PARÂMETROS DAS REGRESSÕES EM FUNÇÃO DA QUANTIDADE DE PONTOS HOMÓLOGOS UTILIZADOS.



FONTE: Autor (2025).

Diante disso, e considerando a necessidade de representar adequadamente as variações morfológicas entre otólitos de diferentes formatos, optou-se pelo uso de 250 pontos homólogos, por oferecer uma resolução suficientemente alta para capturar contornos complexos sem comprometer a eficiência computacional. Essa escolha garante a aplicabilidade do programa a uma ampla variedade de espécies, com diferentes padrões de forma, mantendo a padronização e a reprodutibilidade das medidas extraídas.

A barra de escala é representada por dois pontos homólogos adicionais posicionados em suas extremidades (FIGURA 2-D). A linha correspondente ao maior comprimento do otólito é definida a partir da maior distância euclidiana entre

dois pontos pertencentes ao contorno, sendo a distância entre cada par de pontos calculada com base na fórmula da distância euclidiana:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

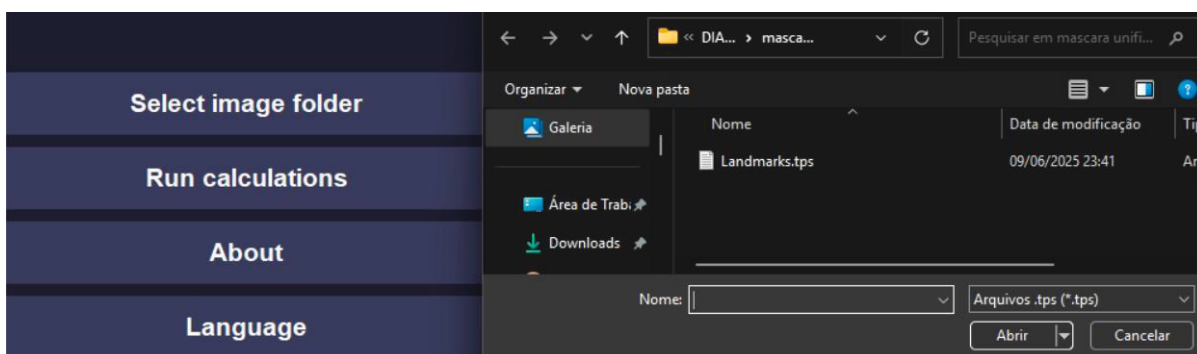
em que (x_1, y_1) e (x_2, y_2) correspondem às coordenadas cartesianas de dois pontos homólogos distintos. O par de pontos com a maior distância registrada foi utilizado para representar o eixo de maior comprimento do otólito.

A partir desta linha, foi calculado um vetor perpendicular em relação a linha de comprimento. Considerando o vetor $v = (dx, dy)$, o vetor ortogonal correspondente foi definido como $v \perp = (-dy, dx)$. Esse vetor serviu como referência para identificar a maior distância entre dois pontos do contorno ao longo da direção perpendicular, por meio da projeção ortogonal dos pontos homólogos sobre o vetor. Os dois pontos que delimitam essa distância máxima foram utilizados para traçar a linha da altura do otólito, representando o eixo perpendicular ao comprimento de maior extensão.

Ambas as linhas são convertidas em sequências compostas por 24 pontos homólogos interpolados cada. No total, cada amostra contém 300 pontos homólogos organizados em blocos, que são automaticamente exportados em arquivo chamado “Landmarks” no formato .tps e identificados com base no nome da imagem original (FIGURA 2-D).

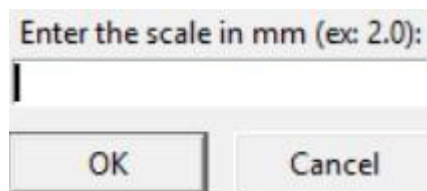
Na segunda etapa, o usuário pode selecionar o arquivo .tps gerado (FIGURA 5) e inserir o valor da escala da barra de referência (mm) (FIGURA 6). As medidas morfométricas são calculadas a partir do posicionamento dos Pontos homólogos.

FIGURA 5 - TELA PARA SELEÇÃO DE ARQUIVO .TPS NO SOFTWARE MORFO.



FONTE: Autor (2025).

FIGURA 6 - TELA PARA INSERÇÃO DO VALOR DA ESCALA EM MILÍMETROS NO SOFTWARE MORFO.



FONTE: Autor (2025).

2.2.2.1 CÁLCULO DA ÁREA E PERÍMETRO

A área do otólito foi calculada pelos pontos homólogos do contorno [1, 250] utilizando a fórmula de Shoelace (LOMBARTE; TUSET, 2015):

$$Aro = (1/2) \times |\sum (x_i \times y_{i+1} - y_i \times x_{i+1})|$$

onde (x_{i+1}, y_{i+1}) representa o primeiro ponto do contorno, usado para fechá-lo.

O perímetro do otólito foi obtido pela soma das distâncias euclidianas entre pontos homólogos do contorno [1, 250] (LOMBARTE; TUSET, 2015):

$$Po = \sum \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

2.2.2.2 CÁLCULO DO COMPRIMENTO E DA ALTURA

O comprimento e a altura foram calculados pela soma das distâncias euclidianas entre os pontos homólogos [252, 276] e [276, 300], respectivamente:

$$Co = \sum \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}, \text{ para } i \text{ de } 252 \text{ a } 276$$

$$Ao = \sum \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}, \text{ para } i \text{ de } 276 \text{ a } 300$$

2.2.2.3 CONVERSÃO PARA UNIDADES REAIS

As medidas morfométricas foram normalizadas com base na escala (S), determinada a partir da distância entre dois Pontos homólogos [251, 252], calculada pela fórmula:

$$S = d(\text{pontos homólogos}) / F$$

onde F é o valor real da escala.

Após o cálculo da escala, todos os valores obtidos (Aro, Po, Co e Ao) foram convertidos para unidades reais, e exportados automaticamente para um arquivo chamado “Landmarks” no formato .xlsl (STEIMBRUCH; WINTERLE, 1987).

2.2.3 COMPARAÇÃO ENTRE MÉTODOS

Com o objetivo de avaliar a consistência entre os dois métodos utilizados para a extração das medidas morfométricas dos otólitos, para cada espécie analisada foi realizada uma comparação estatística pareada entre os valores obtidos para as quatro variáveis analisadas: Aro, Po, Co e Ao (CALLEGARI-JACQUES, 2003; LOMBARTE; TUSET, 2015). Inicialmente, para cada par de variáveis (M1 vs M2), foi calculada a diferença entre os valores individuais e aplicado o teste de Shapiro-Wilk com o intuito de verificar a normalidade da distribuição das diferenças. Em caso de normalidade, foi aplicado o teste *t* pareado; caso contrário, utilizou-se o teste de Wilcoxon (ZAR, 2010).

Para avaliar a relação alométrica entre o comprimento total (Ct) dos indivíduos e as variáveis morfométricas dos otólitos obtidas por ambos os métodos, foi realizada uma análise de regressão potencial ($y = a x^b$) (ZAR, 2010) entre as medidas de área (Aro), perímetro (Po), comprimento (Co) e altura (Ao) dos otólitos e o logaritmo do comprimento total dos peixes (BERVIAN et al., 2006; PATHAK et al., 2024). Todas as análises estatísticas foram conduzidas no software R, com o auxílio dos pacotes readxl (WICKHAM & BRYAN, 2023), openxlsx (SCHAUBERGER & WALKER, 2023), dplyr (WICKHAM et al., 2023), broom (ROBINSON et al., 2023) e ggplot2 (WICKHAM, 2023).

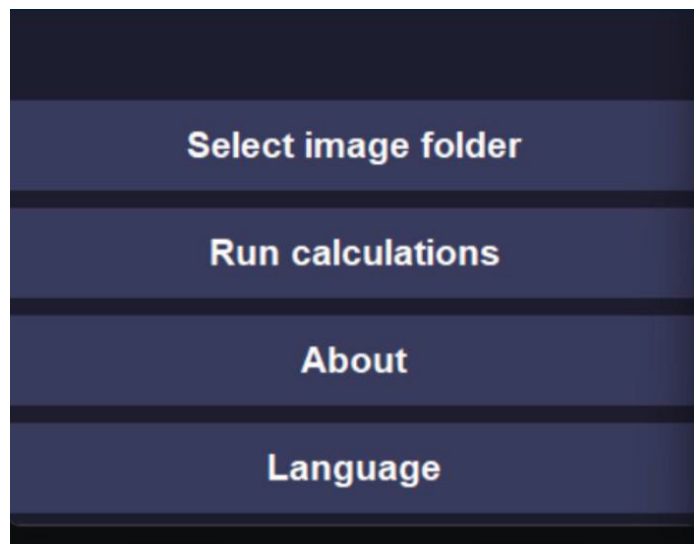
3 RESULTADOS

3.1 MORFO

O principal produto deste estudo foi o MORFO (Morfometria de Otólitos de Peixes), um programa com interface gráfica criado em linguagem Python para a extração automatizada de medidas morfométricas de otólitos. A ferramenta foi estruturada para realizar o processamento de imagens digitais no formato .tif, identificação automática do contorno do otólito e da barra de escala, posicionamento de pontos homólogos digitais e exportação dos dados em formatos compatíveis com demais análises morfométricas.

O programa foi projetado com foco em facilidade de utilização e praticidade, permitindo que usuários com pouca familiaridade em programação possam utilizá-lo com facilidade. Sua interface intuitiva exige mínimo treinamento, o que reduz significativamente o tempo e o esforço despendidos na análise morfométrica de grandes conjuntos de dados (FIGURA 7).

FIGURA 7 - INTERFACE PRINCIPAL DO SOFTWARE MORFO.



FONTE: Autor (2025).

Além da extração de medidas morfométricas de otólitos, o MORFO gera automaticamente arquivos no formato .tps, compatíveis com softwares tradicionais em análises morfométricas, como o tpsDig2, possibilitando a visualização gráfica das linhas de altura e de comprimento do otólito diretamente sobre as imagens

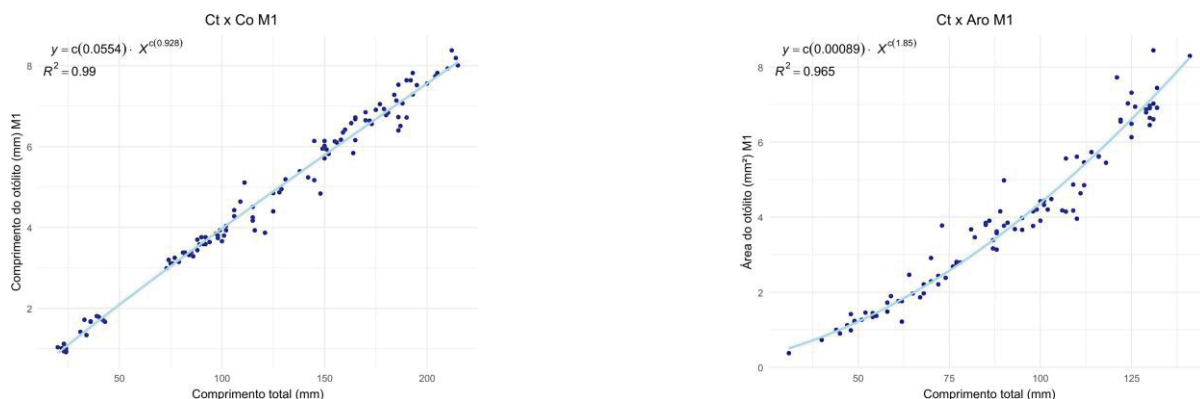
tratadas. Esses arquivos também podem ser utilizados em pacotes R que realizam análises de forma, como Momocs ou geomorph, ampliando as possibilidades de aplicação da ferramenta. Além de oferecer precisão e padronização nos dados gerados, o programa representa uma alternativa gratuita e prática aos métodos manuais e semi automáticos.

3.2 RESULTADOS OBTIDOS PELO MÉTODO TRADICIONAL (M1)

Os resultados obtidos indicaram fortes relações alométricas para a maioria das espécies, com coeficientes de determinação (R^2) superiores a 0,95 em grande parte das regressões, especialmente para as variáveis de área e perímetro. Esses resultados demonstram a consistência e precisão do Método 1, mesmo considerando sua natureza parcialmente manual.

As espécies *Atherinella brasiliensis*, *Diapterus rhombeus*, *Eucinostomos argenteus*, *Haemulopsis corvinaeformis* e *Mugil curema* apresentaram R^2 acima de 0,95 em todas as variáveis, com destaque para *A. brasiliensis* (Ct × Aro: $R^2 = 0,965$) e *D. rhombeus* (Ct × Co: $R^2 = 0,99$) (GRÁFICO 2).

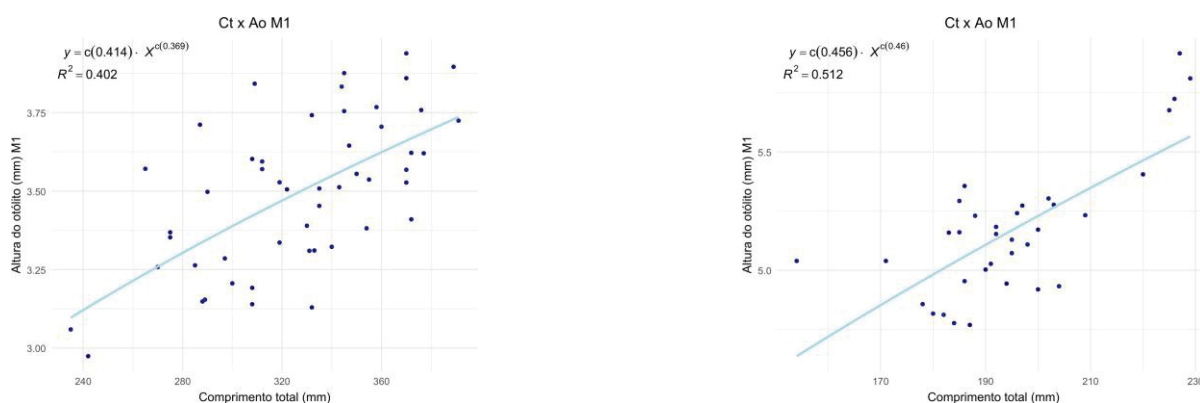
GRÁFICO 2 - ANÁLISE COMPARATIVAS COM REGRESSÃO POTENCIAL DE COMPRIMENTO TOTAL (Ct, mm) E MEDIDAS MORFOMÉTRICAS (Aro e Co, mm) DOS OTÓLITOS DE *ATHERINELLA BRASILIENSIS* E *DIAPTERUS RHOMBEUS*.



FONTE: Autor (2025).

Por outro lado, as espécies *Chloroscombrus chrysurus* e *Cynoscion jamaicensis* apresentaram valores mais baixos de R^2 , sobretudo nas variáveis de altura (Ct $\times$ Ao: $R^2 = 0,402$ e $0,512$, respectivamente) e perímetro (GRÁFICO 3).

GRÁFICO 3 - ANÁLISE COMPARATIVAS COM REGRESSÃO POTENCIAL DE COMPRIMENTO TOTAL (Ct, mm) E MEDIDAS MORFOMÉTRICAS (Ao, mm) DOS OTÓLITOS DE *CHLOROSCOMBRUS CHRYSURUS* E *CYNOSCIION JAMAICENSIS*.



FONTE: Autor (2025).

Os coeficientes de alometria (b) obtidos variaram de 0,369 a 1,846, com valores próximos ou superiores a 1 nas relações com a área, especialmente em *A. brasiliensis*, *D. rhombeus* e *H. corvinaeformis*, caracterizando alometria positiva.

Para as variáveis de comprimento e altura, os coeficientes tendem a valores próximos da isometria ou abaixo de 1, sugerindo alometria negativa, principalmente em *C. chrysurus* e *C. jamaicensis* (TABELA 3).

TABELA 3 – PARÂMETROS DAS EQUAÇÕES DE REGRESSÃO POTENCIAL ENTRE O COMPRIMENTO TOTAL (CT) E MEDIDAS MORFOMÉTRICAS DOS OTÓLITOS: ARO (área do otólito), PO (perímetro do otólito), CO (comprimento do otólito) e AO (altura do otólito), OBTIDOS PELO MÉTODO TRADICIONAL (M1).

Espécie	Relação Ct x	M1				
		<i>a</i>	<i>epa</i>	<i>b</i>	<i>epb</i>	<i>r</i> ²
<i>Atherinella brasiliensis</i>	Aro	0,001	0	1,846	0,036	0,965
	Po	0,143	0,013	0,889	0,02	0,956
	Co	0,042	0,004	0,927	0,021	0,953
	Ao	0,035	0,003	0,857	0,018	0,961
<i>Chloroscombrus chrysurus</i>	Aro	0,166	0,087	0,826	0,091	0,633
	Po	1,456	0,515	0,468	0,061	0,549
	Co	0,566	0,195	0,457	0,06	0,551
	Ao	0,414	0,155	0,369	0,065	0,402
<i>Cynoscion jamaicensis</i>	Aro	0,117	0,062	1,121	0,101	0,794
	Po	0,934	0,271	0,639	0,055	0,808
	Co	0,236	0,078	0,709	0,063	0,799
	Ao	0,456	0,191	0,46	0,079	0,512
<i>Diapterus rhombeus</i>	Aro	0,003	0	1,773	0,018	0,989
	Po	0,163	0,007	0,944	0,01	0,988
	Co	0,055	0,002	0,928	0,009	0,99
	Ao	0,071	0,004	0,813	0,012	0,976
<i>Eucinostomus argenteus</i>	Aro	0,008	0,001	1,422	0,023	0,979
	Po	0,349	0,019	0,731	0,013	0,977
	Co	0,123	0,007	0,719	0,013	0,972
	Ao	0,101	0,007	0,667	0,016	0,953
<i>Haemulopsis corvinaeformis</i>	Aro	0,01	0,001	1,576	0,029	0,972
	Po	0,347	0,026	0,823	0,015	0,97
	Co	0,098	0,008	0,781	0,018	0,956
	Ao	0,124	0,008	0,815	0,014	0,975
<i>Mugil curema</i>	Aro	0,024	0,002	1,205	0,017	0,976
	Po	0,554	0,028	0,646	0,01	0,973
	Co	0,187	0,009	0,655	0,009	0,976
	Ao	0,167	0,007	0,541	0,009	0,971

LEGENDA: *a* = coeficiente da equação; *epa* = erro padrão do coeficiente *a*; *b* = expoente da equação; *epb* = erro padrão do coeficiente *b*; *r*² = coeficiente de determinação.

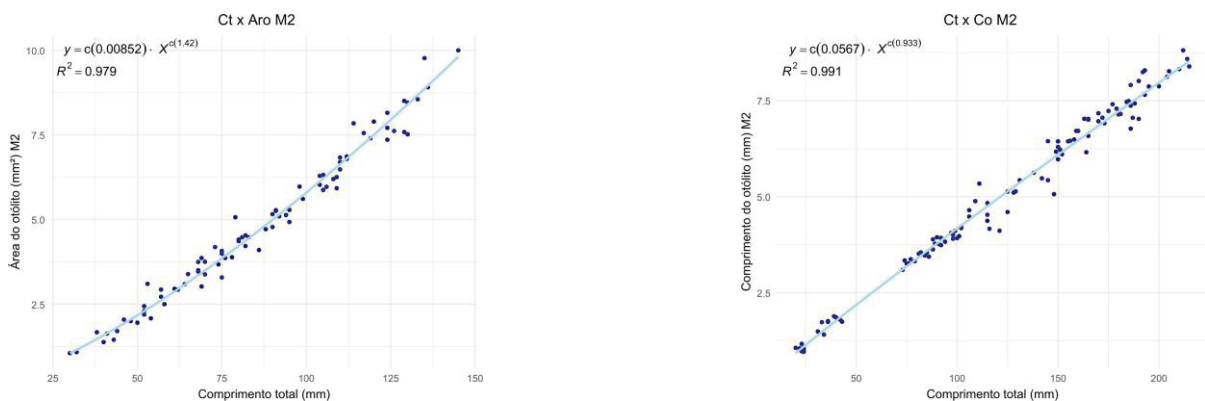
FONTE: Autor (2025).

3.3 RESULTADOS OBTIDOS PELO MORFO (M2)

Os resultados indicaram fortes relações alométricas para a maioria das espécies, com valores de coeficiente de determinação (R^2) superiores a 0,95 para as variáveis de área e perímetro em praticamente todos os casos.

Espécies como *Atherinella brasiliensis*, *Diapterus rhombeus*, *Eucinostomos argenteus*, *Haemulopsis corvinaeformis* e *Mugil curema* apresentaram R^2 acima de 0,96 em todas as variáveis, com destaque para *E. argenteus* (Ct × Aro: $R^2 = 0,979$) e *D. rhombeus* (Ct × Co: $R^2 = 0,991$) (GRÁFICO 4).

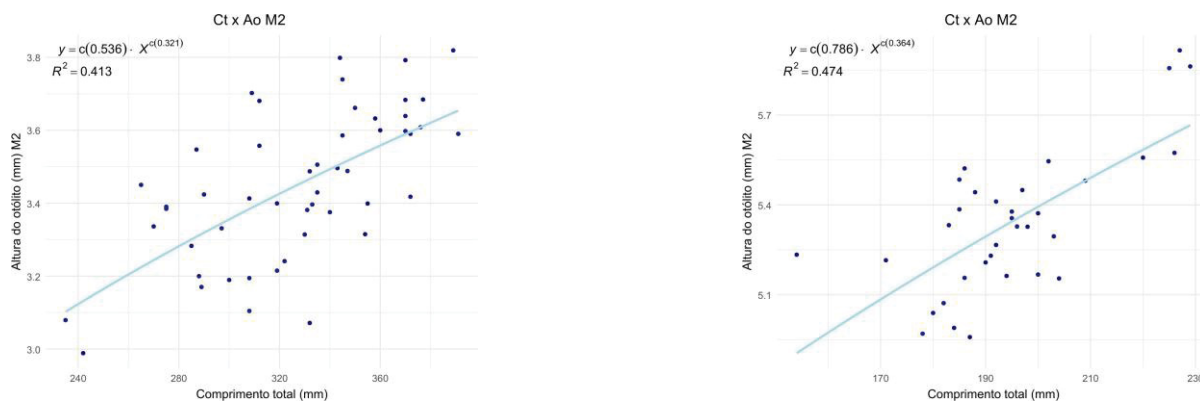
GRÁFICO 4 - ANÁLISE COMPARATIVAS COM REGRESSÃO POTENCIAL DE COMPRIMENTO TOTAL (Ct, mm) E MEDIDAS MORFOMÉTRICAS (Aro e Co, mm) DOS OTÓLITOS DE *EUCINOSTOMUS ARGENTEUS* E *DIAPTERUS RHOMBEUS*.



FONTE: Autor (2025).

Por outro lado, para *Chloroscombrus chrysurus* e *Cynoscion jamaicensis*, assim como visto nos resultados do método 1, foram observados R^2 mais baixos em comparação com as demais espécies, especialmente nas variáveis de altura (Ct × Ao: $R^2 = 0,413$ para *C. chrysurus* e $R^2 = 0,474$ para *C. jamaicensis*) e perímetro.

GRÁFICO 5 - ANÁLISE COMPARATIVAS COM REGRESSÃO POTENCIAL DE COMPRIMENTO TOTAL (Ct, mm) E MEDIDAS MORFOMÉTRICAS (Ao, mm) DOS OTÓLITOS DE *CHLOROSCOMBRUS CHRYSURUS* E *CYNOSCIION JAMAICENSIS*.



FONTE: Autor (2025).

Os coeficientes de alometria (b) variaram entre 0,321 e 1,829, com valores próximos ou superiores a 1 nas relações envolvendo a área, especialmente para *A. brasiliensis*, *D. rhombeus* e *H. corvinaeformis*, indicando alometria positiva. Já para o comprimento e a altura, os coeficientes tendem a valores menores, próximos da isometria ou indicando alometria negativa, principalmente em *C. chrysurus* e *C. jamaicensis* (TABELA 4).

TABELA 4 – PARÂMETROS DAS EQUAÇÕES DE REGRESSÃO POTENCIAL ENTRE O COMPRIMENTO TOTAL (CT) E MEDIDAS MORFOMÉTRICAS DOS OTÓLITOS: ARO (área do otólito), PO (perímetro do otólito), CO (comprimento do otólito) e AO (altura do otólito), OBTIDOS PELO MÉTODO TRADICIONAL (M2).

Espécie	Relação Ct x	M2				
		a	epa	b	epb	r ²
<i>Atherinella brasiliensis</i>	Aro	0,001	0	1,829	0,035	0,966
	Po	0,149	0,013	0,872	0,019	0,957
	Co	0,044	0,004	0,925	0,021	0,955
	Ao	0,035	0,003	0,864	0,018	0,961
<i>Chloroscombrus chrysurus</i>	Aro	0,159	0,085	0,83	0,092	0,629
	Po	1,548	0,524	0,448	0,059	0,549
	Co	0,599	0,2	0,455	0,058	0,564
	Ao	0,536	0,172	0,321	0,055	0,413
<i>Cynoscion jamaicensis</i>	Aro	0,114	0,062	1,121	0,102	0,79
	Po	0,988	0,279	0,619	0,054	0,807
	Co	0,259	0,087	0,701	0,064	0,79
	Ao	0,786	0,28	0,364	0,068	0,474
<i>Diapterus rhombeus</i>	Aro	0,003	0	1,753	0,016	0,99
	Po	0,192	0,008	0,901	0,008	0,99
	Co	0,057	0,002	0,933	0,008	0,991
	Ao	0,069	0,004	0,825	0,012	0,975
<i>Eucinostomus argenteus</i>	Aro	0,009	0,001	1,416	0,023	0,979
	Po	0,354	0,018	0,72	0,012	0,978
	Co	0,126	0,007	0,725	0,013	0,974
	Ao	0,104	0,008	0,669	0,016	0,953
<i>Haemulopsis corvinaeformis</i>	Aro	0,01	0,001	1,582	0,029	0,972
	Po	0,368	0,024	0,801	0,014	0,975
	Co	0,1	0,009	0,784	0,018	0,955
	Ao	0,131	0,009	0,813	0,014	0,975
<i>Mugil curema</i>	Aro	0,025	0,002	1,196	0,017	0,977
	Po	0,597	0,026	0,624	0,008	0,979
	Co	0,176	0,007	0,534	0,008	0,972
	Ao	0,197	0,009	0,653	0,009	0,976

LEGENDA: a = coeficiente da equação; epa = erro padrão do coeficiente a; b = expoente da equação; epb = erro padrão do coeficiente b; r² = coeficiente de determinação.

FONTE: Autor (2025).

3.4 COMPARAÇÃO ENTRE OS MÉTODOS

Os dados das Tabela 2 e 3 mostram que ambos os métodos apresentaram coeficientes de determinação (R^2) superiores a 0,95 para a maioria das espécies analisadas.

Espécies como *Atherinella brasiliensis*, *Diapterus rhombeus*, *Eucinostomos argenteus*, *Haemulopsis corvinaeformis* e *Mugil curema* mantiveram altos valores de R^2 em todas as variáveis, independentemente do método aplicado. Por outro lado, *Chloroscombus chrysurus* e *Cynoscion jamaicensis* apresentaram desempenho inferior em ambos os métodos, com valores reduzidos de R^2 nas variáveis de altura e perímetro.

Os coeficientes de alometria (b) também se mostraram similares entre os métodos, com alometria positiva nas relações com a área e tendência à isometria ou alometria negativa para altura e comprimento. Isso demonstra que, além da qualidade do ajuste estatístico, a interpretação biológica das relações permanece consistente, independentemente da metodologia empregada.

Já a Tabela 4 mostra que o teste de *Shapiro-Wilk* indicou desvios de normalidade em 21 dos 28 pares de medidas morfométricas, evidenciando que a maioria dos pares de dados não seguem uma distribuição normal. Os testes de *Wilcoxon* e *t pareado*, por sua vez, apresentaram valores de p extremamente significativos para todas as medidas avaliadas ($p < 0,05$). Os resultados reforçam que as discrepâncias entre os métodos não são fruto do acaso, mesmo quando as médias aparentam ser próximas.

TABELA 4 – COMPARAÇÃO DE ESTATÍSTICA DESCRITIVA ENTRE DOIS MÉTODOS DE MEDIÇÃO MORFOMÉTRICA DE OTÓLITOS POR ESPÉCIE.

Espécie	Medida	M1			M2			Estatística		
		méd.	mediana	s	méd.	mediana	s	Shapiro-Wilk	Wilcoxon	teste t pareado
<i>Atherinella brasiliensis</i>	Aro	3,9443	3,8460	2,0836	3,8663	3,7548	2,0342	4,15E-03	4,75E-12	
	Po	7,8632	8,2108	2,1954	7,5434	7,8554	2,0739	2,19E+01	1,36E-13	
	Co	2,7103	2,8165	0,7964	2,8254	2,9248	0,8331	1,28E+02	1,24E-13	
	Ao	1,6522	1,7119	0,4509	1,6947	1,7444	0,4636	4,36E+03		3,77E-25
<i>Chloroscombrus chrysurus</i>	Aro	19,8191	19,8136	2,4078	19,4581	19,3336	2,3878	3,20E-02	8,80E-06	
	Po	21,8126	21,7612	1,6238	20,6707	20,6504	1,4777	1,02E+00	7,79E-06	
	Co	7,9792	8,0387	0,5759	8,3222	8,3602	0,5906	3,23E-07	7,79E-06	
	Ao	3,4965	3,5200	0,2420	3,4478	3,4269	0,2051	7,73E+02		1,03E+01
<i>Cynoscion jamaicensis</i>	Aro	43,2746	42,9118	4,7477	42,4084	41,9050	4,6773	1,89E-02	2,33E-06	
	Po	27,1287	27,0111	1,6532	25,8996	25,7216	1,5280	5,67E+02		8,09E-25
	Co	9,9344	9,8451	0,6787	10,4542	10,3687	0,7074	2,41E+03		3,20E-25
	Ao	5,1696	5,1561	0,2847	5,3442	5,3302	0,2389	2,70E-01	2,91E-05	
<i>Diapterus rhombeus</i>	Aro	13,9384	12,3354	9,6093	13,7586	12,1816	9,4828	9,63E-11	1,20E-11	
	Po	14,6729	15,0215	6,8104	13,9822	14,0525	6,3156	2,22E+01	8,73E-14	
	Co	4,6106	4,4700	2,1869	4,8446	4,7402	2,3059	1,16E+02	1,94E-16	
	Ao	3,3662	3,6150	1,3357	3,4544	3,6929	1,3828	4,40E+01	1,79E-10	
<i>Eucinostomus argenteus</i>	Aro	4,8730	4,5486	2,2308	4,8405	4,5063	2,2107	3,18E+02	7,78E-11	
	Po	8,9715	8,8415	2,2871	8,6225	8,4979	2,1613	2,43E+03		5,83E-33
	Co	3,0049	2,9616	0,7648	3,1528	3,1097	0,8076	6,32E+02		1,98E-39
	Ao	1,9464	1,9643	0,4459	2,0231	2,0381	0,4665	1,16E-02	1,94E-11	
<i>Haemulopsis corvinaeformis</i>	Aro	21,1448	19,9958	7,5463	20,8592	19,5934	7,4721	4,17E-01	1,83E-12	
	Po	18,2657	17,9176	3,5729	17,4003	17,1596	3,2889	2,84E-04	1,77E-12	
	Co	4,2074	4,1837	0,7803	4,3547	4,3137	0,8141	1,00E+03		6,06E-37
	Ao	6,2865	6,3108	1,1939	6,5739	6,5780	1,2458	1,60E+01	1,77E-12	
<i>Mugil curema</i>	Aro	12,8210	10,0700	7,4789	12,7470	10,0052	7,4069	1,65E+02	7,12E-09	
	Po	15,4869	14,0525	5,4652	14,8711	13,4128	5,0942	2,94E-03	9,54E-17	
	Co	5,4673	5,0860	1,9477	2,7367	2,5577	0,8261	6,13E+00	2,92E-17	
	Ao	2,6988	2,4964	0,8219	5,7174	5,3065	2,0341	7,47E+00	2,92E-17	

FONTE: O autor (2025).

4 DISCUSSÃO

O desenvolvimento do MORFO representa um avanço significativo na morfometria de otólitos, oferecendo uma solução robusta e acessível à comunidade científica. A ferramenta foi concebida para simplificar o processo de análise, desde o tratamento de imagens digitais no formato .tif até a identificação automática do contorno do otólito e da barra de escala, culminando no posicionamento de pontos homólogos digitais e na exportação de dados em formatos compatíveis com análises subsequentes. A interface gráfica intuitiva do programa é um de seus maiores trunfos, minimizando a curva de aprendizado e permitindo que pesquisadores com pouca ou nenhuma experiência em programação o utilizem com facilidade.

Essa característica é particularmente relevante, pois democratiza o acesso a análises morfométricas avançadas, possibilitando a delegação de tarefas a estudantes de graduação, bolsistas de Iniciação Científica (IC), Programas de Voluntariado Acadêmico (PVA) e estagiários, sem comprometer a qualidade dos dados. Tal otimização é crucial para estudos de larga escala, contribuindo para a eficiência de tempo e recursos humanos. A redução da dependência da experiência do operador, um problema comum em métodos manuais, contribui diretamente para a padronização das medições e a consistência dos dados gerados, elementos essenciais para a reprodutibilidade científica (ARNQVIST & MÅRTENSSON, 1998).

Além da eficiência na extração de medidas, o MORFO se destaca pela sua capacidade de gerar arquivos no formato .tps, que podem ser utilizados em softwares de morfometria geométrica, como o tpsDig2. Essa compatibilidade não apenas facilita a visualização gráfica das linhas de altura e comprimento do otólito diretamente sobre as imagens processadas, mas também expande as possibilidades de análise, permitindo a integração com pacotes R como Momocs e geomorph para estudos mais aprofundados da forma.

A natureza gratuita e de código aberto do MORFO o posiciona como uma alternativa viável e prática aos softwares comerciais, que muitas vezes representam um custo proibitivo para muitos laboratórios e pesquisadores, especialmente em regiões com recursos limitados. Essa acessibilidade, combinada com a precisão e padronização dos dados, reforça o potencial do MORFO para se tornar uma ferramenta padrão em estudos de morfometria de otólitos, promovendo a

colaboração e a comparabilidade de resultados entre diferentes instituições (EVANS, 2013).

A comparação entre o método tradicional (M1) e o MORFO (M2) revelou que ambos são capazes de capturar com alta fidelidade as relações alométricas entre o comprimento total dos peixes (Ct) e as medidas morfométricas dos otólitos. Os coeficientes de determinação (R^2) superiores a 0,95 observados na maioria das regressões para ambos os métodos, especialmente para as variáveis de área e perímetro, sublinham a robustez dessas relações. Isso indica que, independentemente da metodologia empregada, os otólitos mantêm uma forte correlação com o tamanho corporal do indivíduo, validando a premissa fundamental da morfometria de otólitos como um proxy para o crescimento e a idade dos peixes (VOLPEDO & VAZ-DOS-SANTOS, 2015).

Espécies como *Atherinella brasiliensis*, *Diapterus rhombeus*, *Eucinostomus argenteus*, *Haemulopsis corvinaeformis* e *Mugil curema* demonstraram consistentemente altos valores de R^2 em todas as variáveis analisadas, tanto com o M1 quanto com o M2. *D. rhombeus*, em particular, apresentou os maiores valores de R^2 (até 0,991), sugerindo que as relações alométricas em seus otólitos são extremamente bem definidas e menos suscetíveis à variabilidade metodológica. Essa consistência em diversas espécies reforça a aplicabilidade geral da morfometria de otólitos e a capacidade de ambos os métodos em extrair informações biológicas significativas.

No entanto, *Chloroscombrus chrysurus* e *Cynoscion jamaicensis* apresentaram um desempenho inferior em ambos os métodos, com valores de R^2 reduzidos, especialmente para as variáveis de altura e perímetro. Essa menor capacidade preditiva pode ser atribuída a uma maior variabilidade morfológica intrínseca dos otólitos nessas espécies, que pode dificultar a identificação precisa de pontos homólogos ou a delimitação de contornos em métodos manuais, e a interpretação automática em métodos automatizados. Adicionalmente, a literatura sugere que o número reduzido de amostras disponíveis para análise, pode impactar significativamente a robustez das regressões, levando a coeficientes de determinação mais baixos (BISSONETTE., 1999). Este achado ressalta a importância de um tamanho amostral adequado para garantir a validade estatística e a interpretabilidade biológica dos resultados, independentemente da ferramenta utilizada.

Os coeficientes de alometria (b) também se mostraram notavelmente similares entre os dois métodos, com alometria positiva nas relações envolvendo a área e uma tendência à isometria ou alometria negativa para as variáveis de altura e comprimento. Essa consistência nos padrões alométricos é um indicativo crucial de que o MORFO não apenas replica, mas valida as interpretações biológicas obtidas pelo método tradicional. A alometria positiva da área sugere que o otólito cresce proporcionalmente mais rápido em superfície do que o corpo do peixe, enquanto a isometria ou alometria negativa para comprimento e altura pode indicar um crescimento mais conservador nessas dimensões, refletindo adaptações morfológicas específicas da espécie ou do ambiente (LOMBARTE; TUSET, 2015).

Apesar da similaridade nos padrões alométricos gerais, análises inferenciais mais detalhadas, como o teste de Shapiro-Wilk, indicaram desvios de normalidade em 21 dos 28 pares de medidas morfométricas, sugerindo que a maioria dos dados não segue uma distribuição normal, o que é comum em dados biológicos. Mais importante, os testes de *Wilcoxon* e *t pareado* apresentaram valores de *p* extremamente significativos ($p < 0,05$) para todas as medidas avaliadas. Esses resultados reforçam que as discrepâncias entre os métodos não são meramente aleatórias, mas sim sistemáticas, mesmo quando as médias aparentam ser próximas. Isso implica que, embora ambos os métodos forneçam resultados que refletem as relações alométricas, a precisão e a reprodutibilidade podem variar (TAKÁCS et al., 2016).

Nesse contexto, o MORFO demonstrou maior reprodutibilidade e menor suscetibilidade a vieses observacionais. A automação inerente ao programa elimina a subjetividade humana na delimitação de contornos e no posicionamento de pontos, que são fontes comuns de erro e variabilidade em métodos manuais (BOYKIN e NELSON, 1981). A padronização das medições, uma característica central do MORFO, garante que cada otólito seja processado da mesma maneira, reduzindo a variância entre diferentes operadores e ao longo do tempo. Em muitos casos, o MORFO também apresentou uma precisão ligeiramente superior, o que pode ser atribuído à sua capacidade de processar imagens e aplicar algoritmos de detecção de bordas mais consistentes do que o olho humano. Essas vantagens são cruciais para estudos que exigem alta precisão e comparabilidade de dados, como aqueles que envolvem a diferenciação de estoques ou a análise de pequenas variações morfológicas.

Em suma, enquanto o método tradicional (M1) provou ser eficaz na identificação de relações alométricas robustas, o MORFO (M2) emerge como uma ferramenta superior em termos de reprodutibilidade, padronização, acessibilidade e, em muitos casos, precisão. A capacidade do MORFO de otimizar o tempo de análise e reduzir a dependência da experiência do operador o torna uma solução ideal para a pesquisa moderna em biologia pesqueira, onde a análise de grandes volumes de dados é cada vez mais comum. A sua natureza de código aberto e a compatibilidade com outros softwares de análise morfométrica ampliam ainda mais o seu potencial de aplicação e colaboração científica.

Uma área para aprimoramento reside na validação do MORFO em uma gama ainda mais ampla de espécies de peixes, abrangendo diferentes famílias, hábitos de vida e habitats. Embora o presente estudo tenha incluído sete espécies, a diversidade morfológica dos otólitos é vasta, e a generalização da eficácia do MORFO se beneficiaria de testes em cenários mais variados. A inclusão de espécies com otólitos de formas mais complexas ou com características morfológicas distintas poderia revelar desafios adicionais e guiar o desenvolvimento de algoritmos mais sofisticados para a detecção de contornos e pontos homólogos.

Adicionalmente, a integração do MORFO com outras ferramentas de análise de imagem e plataformas de dados biológicos representa uma promissora avenida para o futuro. A capacidade de exportar dados em formato .tps já é um passo importante, mas a criação de módulos de interoperabilidade direta com bancos de dados online poderia ampliar ainda mais o impacto da ferramenta. Isso facilitaria a análise de padrões espaciais e temporais na morfometria de otólitos, contribuindo para uma compreensão mais holística da ecologia e conservação de peixes.

Por fim, o desenvolvimento de funcionalidades adicionais no MORFO, como a capacidade de realizar análises de morfometria geométrica diretamente na interface do programa, ou a incorporação de algoritmos de aprendizado de máquina para a classificação automática de espécies com base na forma do otólito, poderia transformar a ferramenta em uma plataforma ainda mais completa e poderosa. A contínua evolução do MORFO, guiada pela colaboração com a comunidade científica, tem o potencial de estabelecer novos padrões para a morfometria de otólitos, impulsionando a pesquisa e a gestão de recursos pesqueiros em escala global.

5 CONCLUSÃO

O presente estudo foi concebido com o objetivo primordial de desenvolver e validar uma ferramenta automatizada para a extração de medidas morfométricas de otólitos de peixes, visando superar as limitações inerentes aos métodos tradicionais. As análises e os resultados obtidos demonstram de forma inequívoca que todos os objetivos propostos foram plenamente alcançados, culminando na criação e validação do programa MORFO (Morfometria de Otólitos de Peixes) como uma solução inovadora e superior para a pesquisa em biologia pesqueira.

O objetivo geral de desenvolver e validar uma ferramenta automatizada foi concretizado com sucesso através da concepção do MORFO. Este programa provou ser eficaz na automação do processamento de imagens digitais, identificação de contornos de otólitos e barras de escala, posicionamento de pontos homólogos e exportação de dados em formatos compatíveis com análises morfométricas avançadas. Em relação aos objetivos específicos, as análises estatísticas conduzidas neste estudo forneceram evidências robustas:

1.

Avaliação da consistência estatística dos resultados obtidos com o método desenvolvido: O MORFO demonstrou alta consistência estatística, com coeficientes de determinação (R^2) superiores a 0,95 para a maioria das relações alométricas entre o comprimento total dos peixes e as medidas morfométricas dos otólitos. Essa robustez foi observada em diversas espécies, indicando que o método desenvolvido é capaz de gerar dados confiáveis e com alta capacidade preditiva.

2.

Comparação da eficiência, precisão e reprodutibilidade do novo método em relação ao método tradicional: A comparação detalhada entre o MORFO (M2) e o método tradicional (M1) revelou que, embora ambos sejam eficazes na identificação de relações alométricas, o MORFO se destaca significativamente em termos de reprodutibilidade e precisão. As análises inferenciais, incluindo testes de Shapiro-Wilk, Wilcoxon e t pareado, evidenciaram diferenças sistemáticas entre os métodos, com o MORFO apresentando menor suscetibilidade a vieses observacionais e, em muitos casos, uma precisão ligeiramente superior. A automação inerente ao MORFO elimina a subjetividade humana, padronizando as medições e garantindo maior consistência dos dados.

3.

Teste da aplicação do novo método em espécies modelo de diferentes hábitos e habitats: O MORFO foi testado com sucesso em otólitos de sete espécies de peixes teleósteos, representando uma variedade de hábitos e habitats. Embora algumas espécies tenham apresentado menor capacidade preditiva devido à variabilidade morfológica intrínseca ou ao tamanho amostral, a ferramenta demonstrou aplicabilidade geral e a capacidade de extrair informações biológicas significativas em diferentes contextos. Isso valida a versatilidade do MORFO para uma ampla gama de estudos em biologia pesqueira.

Diante do exposto, concluímos que o programa MORFO é o método mais indicado para a extração de medidas morfométricas de otólitos de peixes. Suas vantagens, que incluem a otimização do tempo de análise, a padronização das medições, a acessibilidade (sendo uma ferramenta gratuita e de código aberto) e a facilidade de uso, o posicionam como uma ferramenta essencial para estudos em larga escala e colaborativos. O MORFO não apenas aprimora a qualidade e a reprodutibilidade dos dados morfométricos, mas também abre novas perspectivas para a pesquisa em biologia pesqueira e áreas correlatas, impulsionando a gestão e a conservação de recursos pesqueiros em escala global. A disponibilização pública do MORFO contribuirá para a disseminação e o avanço do conhecimento científico na área.

O MORFO estará disponível para acesso público por meio do site do Laboratório de Pesca, Recursos Aquáticos e Pessoas (LAPESC UFPR), favorecendo sua disseminação e utilização por outros pesquisadores interessados em estudos morfométricos.

REFERÊNCIAS

- ADOBE SYSTEMS INCORPORATED.** Adobe Photoshop (Version 22.0.0) [Software]. San Jose: Adobe, 2020. Disponível em: <https://www.adobe.com/products/photoshop.html>. Acesso em: 10 abr. 2025.
- ARNQVIST, G.; MÅRTENSSON, T.** Measurement error in geometric morphometrics: empirical strategies to assess and reduce its impact on measures of shape. *Acta Zoologica Academiae Scientiarum Hungaricae*, v. 44, n. 1–2, p. 73–96, 1998.
- BERVIAN, G.; FONTOURA, N.F.; HAIMOVICI, M.** Statistical model of variable allometric growth: otolith growth in *Micropogonias furnieri* (Actinopterygii, Sciaenidae). *Journal of Fish Biology*, Porto Alegre, v. 68, p. 196–208, 2006.
- BEZERRA-NETO, E. B. et al.** Atlas de otólitos de peixes capturados na bacia do rio Verde, estado do Mato Grosso do Sul, Brasil. *Acta of Fisheries and Aquatic Resources (Acta Pesca News)*, v. 10, n. 2, p. 27–39, 2022. DOI: 10.46732/actafish.2022.10.2.27-39.
- BEZERRA-NETO, E. B. et al.** Otolith morphometry for the determination of the theoretical growth curve of fish. *Boletim do Instituto de Pesca*, 2024, v. 50, e842. DOI: 10.20950/1678-2305/bip.2024.50.e842.
- BISSONETTE, J. A.** Small sample size problems in wildlife ecology: a contingent analytical approach. *Wildlife Biology*, v. 5, n. 2, p. 65–71, 1999. <http://dx.doi.org/10.2981/wlb.1999.010>.
- BIOLÉ, F.G. et al.** Application of otolith morphometry for the study of ontogenetic variations of *Odontesthes argentinensis*. *Environmental Biology of Fishes*, v. 102, n. 10, p. 1301–1310, 2019. <https://doi.org/10.1007/s10641-019-00908-0>
- BOYKIN, R. A.; NELSON, R. O.** The effects of instructions and calculation procedures on observers' accuracy, agreement, and calculation correctness. *Journal of Applied Behavior Analysis*, v. 14, n. 4, p. 479–489, 1981. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1308235/>.
- BRADSKI, G.** The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- CALLEGARI-JACQUES, S. M.** *Bioestatística: Princípios e Aplicações*. Porto Alegre: Artmed, 2003.

DELGADO, L. B. ; PIN, J. C. ; SANTANA, L. Z. Análisis morfométrico de otolitos en pelágicos pequeños de la familia Carangidae capturados en el Pacífico Ecuatoriano. *Revista de Ciencias del Mar y Acuicultura*, v. 6, n. 11, p. 1-?, jan.-dez. 2023. ISSN 2600-5824. DOI: <https://doi.org/10.56124/yaku.v6i11.536>.

ECHRESHAVI, S.; ESMAEILI, H. R.; TEIMORI, A.; SAFAIE, M. Otolith morphology: a hidden tool in the taxonomic study of goatfishes (Teleostei: Perciformes: Mullidae). *Zoological Studies*, v. 60, p. 36, 2021. DOI: 10.6620/ZS.2021.60-36.

EVANS, A. R. *Shape descriptors as ecometrics in dental ecology*. *Hystrix, the Italian Journal of Mammalogy*, v. 24, n. 1, p. 133–140, 2013. DOI: [10.4404/hystrix-24.1-6363](https://doi.org/10.4404/hystrix-24.1-6363).

GOTELLI, N. J.; ELLISON, A. M. *Princípios de estatística em ecologia*. Porto Alegre: Editora Artmed, 2011.

HARRIS, C. R. et al. Array programming with NumPy. *Nature*, 585, 357–362, 2020. <https://doi.org/10.1038/s41586-020-2649-2>

LOMBARTE, A.; TUSET, V. Morfometría de otolitos. In: VOLPEDO, A. V.; VAZ-DOS-SANTOS, A. M. (Eds.). *Métodos de estudos com otólitos: princípios e aplicações*. Buenos Aires: CAFP-BA-PIESCI, 2015. p. 269-302.

MACARENA, Z. M. Morphogeometric analysis of tooth plates of *Lepidosiren paradoxa* (Sarcopterygii, Dipnoi) to infer Late-Holocene paleoenvironmental variations in the South American Gran Chaco. *The Holocene*, 2024, p. 1-11. DOI: <https://doi.org/10.1177/09596836241285796>.

MEDEIROS, R. et al. Growth stanza in fish life history using otoliths shape. *Neotropical Ichthyology*, v. 19, n. 4, e200145, 2021. DOI: 10.1590/1982-0224-2020-0145.

OSMAN, Y. A. A. et al. Relationship between body and otolith morphological characteristics of Sabre Squirrelfish (*Sargocentron spiniferum*) from the Southern Red Sea: Difference between right and left otoliths. *Journal of Marine Science and Engineering*, v. 10, n. 1, p. 1–13, 2022. DOI: 10.3390/jmse10010001.

PATHAK, V. et al. Discrimination of five gobiid species (Family: Gobiidae) based on otolith morphometry. *Indian Journal of Animal Research*, 2023. DOI: 10.18805/IJAR.B-4458.

PEGG, Mark A. et al. Creating a Digital Repository of Calcified Structures from Known-Age Fishes, a Century in the Making. *Fisheries*, v. 47, n. 8, p. 357-360, 20 maio 2022.
<http://dx.doi.org/10.1002/fsh.10773>.

PYTHON SOFTWARE FOUNDATION. Python Language Reference, version 3.x. 2024.
Disponível em: <https://www.python.org/>

QUIST, M. C. et al. Precision of Structures Used to Estimate Age and Growth of Apache Trout from Arizona. *Journal Of Fish and Wildlife Management*, v. 14, n. 1, p. 188-194, 16 abr. 2023. <http://dx.doi.org/10.3996/jfwm-22-021>.

REBACK, J. et al. pandas-dev/pandas: Pandas (v1.0.5). *Zenodo*, 2020.
<https://doi.org/10.5281/zenodo.3509134>

SANTOS, L.; VAZ-DOS-SANTOS, A.M. Insights of Otoliths Morphology to Reveal Patterns of Teleostean Fishes in the Southern Atlantic. *Fishes* 2023, 8, 21. <https://doi.org/10.3390/fishes8010021>

SAYGIN, S. et al. The relationships between fish length and otolith measurements in *Alburnus derjugini* (Leuciscidae) and *Vimba vimba* (Leuciscidae) from the Harşit Stream, Türkiye. *Acta Biologica Turcica*, v. 37, n. 1, p. J1-J1-8, 2024.

SCHAUBERGER, P.; WALKER, A. openxlsx: Read, Write and Edit XLSX Files. *R package version 4.2.5.2*, 2023. <https://CRAN.R-project.org/package=openxlsx>

STEIMBRUCH, A.; WINTERLE, P. *Geometria Analítica*. 2. ed. São Paulo: McGraw-Hill, 1987.

TAKÁCS, P. et al. Repeatability, reproducibility, separative power and subjectivity of different fish morphometric analysis methods. *PLoS ONE*, v. 11, n. 6, e0157890, 2016.
<https://doi.org/10.1371/journal.pone.0157890>.

TUSET, V. M. et al. Testing otolith morphology for measuring marine fish biodiversity. *Marine and Freshwater Research*, v. 67, n. 7, p. 1037-1048, 2016. <https://doi.org/10.1071/MF15052>

VAZ-DOS-SANTOS, A.M.; SILVEIRA, E.L. Idade e crescimento de peixes de riacho: métodos e desafios para a obtenção de estimativas robustas. *Oecologia Australis*, v. 25, n. 2, pp. 344-366, 2021.

VAZ-DOS-SANTOS, A. M. FishCAST: Collection of Fish Calcified Structures. World Wide Web electronic publication. www.fishcast.info, version 06/2025.

VOLPEDO, A. V.; VAZ-DOS-SANTOS, A. M. (Eds.). *Métodos de estudos com otólitos: princípios e aplicações*. Buenos Aires: CAFB-BA-PIESCI, 2015.

WICKHAM, H. et al. dplyr: A Grammar of Data Manipulation. *R package version 1.1.4*, 2023. <https://CRAN.R-project.org/package=dplyr>

WICKHAM, H. ggplot2: Elegant graphics for data analysis. *R package, version 3.5.2*, 2023. <https://ggplot2.tidyverse.org>.

WICKHAM, H.; HESTER, J.; BRYAN, J. readr: Read Rectangular Text Data. *R package version 2.1.4*, 2023. <https://CRAN.R-project.org/package=readr>

YEDIER, Serdar; YALÇINKAYA, Seda Konaş; BOSTANCI, Derya; POLAT, Nazmi. Morphologic, morphometric and contour shape variations of sagittal otoliths of *Lepidorhombus* spp. in the Aegean Sea. *Anatomia, Histologia, Embryologia*, [S.l.], v. 52, n. 1, p. 48–57, jan. 2023. Wiley. DOI: <https://doi.org/10.1111/ahe.12881>.

YEDIER, Serdar; BOSTANCI, Derya; TÜRKER, Dilek. Morphological and morphometric features of the abnormal and normal saccular otoliths in flatfishes. *The Anatomical Record*, [S.l.], v. 306, n. 2, p. 412–423, fev. 2023. Wiley. DOI: <https://doi.org/10.1002/ar.25106>.

ZAR, J.H. *Bioestatistical analysis*. 5th Ed. New Jersey: Pearson, 2010.

ANEXO 1 – SCRIPT EM R PARA ANÁLISE DE DIFERENTES QUANTIDADES DE PONTOS HOMÓLOGOS

```
library(readxl)
library(ggplot2)
library(openxlsx)
library(dplyr)
library(gridExtra)
library(RColorBrewer)
arquivo <- ""
planilhas <- excel_sheets(arquivo)
resultados_totais <- list()
ajustar_regressao_potencial <- function(df, medida, TL, planilha_nome) {
  log_TL <- log(df[[TL]])
  log_y <- log(df[[medida]])
  modelo <- lm(log_y ~ log_TL)
  coef <- coefficients(modelo)
  a <- exp(coef[1])
  b <- coef[2]
  epa <- summary(modelo)$coefficients[1,2] * a
  epb <- summary(modelo)$coefficients[2,2]
  r2 <- summary(modelo)$r.squared
  x_vals <- seq(min(df[[TL]]), max(df[[TL]]), length.out = 100)
  y_pred <- a * x_vals^b
  curva_df <- data.frame(x = x_vals, y = y_pred)
  p <- ggplot(df, aes_string(x = TL, y = medida)) +
    geom_point(color = "blue", size = 2) +
    geom_line(data = curva_df, aes(x = x, y = y), color = "red") +
    labs(
      title = paste(medida, "vs Ct |", planilha_nome, "LM"),
      subtitle = paste0("y = ", round(a, 4), " * x^", round(b, 4),
        " | R² = ", round(r2, 4))
    ) +
    theme_minimal()
  return(list(
    grafico = p,
    resultado = data.frame(
      Medida = medida,
      Planilha = as.numeric(planilha_nome),
      a = a, epa = epa,
      b = b, epb = epb,
      R2 = r2
    )
  ))
}
for (planilha in planilhas) {
  dados <- read_excel(arquivo, sheet = planilha)
  medidas <- c("Aro", "Po", "Ao", "Co", "Ct")
  for (m in medidas) {
    dados[[m]] <- as.numeric(gsub(",", ".", dados[[m]]))
  }
  res_aro <- ajustar_regressao_potencial(dados, "Aro", "Ct", planilha)
  res_po <- ajustar_regressao_potencial(dados, "Po", "Ct", planilha)
  res_ao <- ajustar_regressao_potencial(dados, "Ao", "Ct", planilha)
  res_co <- ajustar_regressao_potencial(dados, "Co", "Ct", planilha)
  resultados_totais[[length(resultados_totais)+1]] <- res_aro$resultado
  resultados_totais[[length(resultados_totais)+1]] <- res_po$resultado
  resultados_totais[[length(resultados_totais)+1]] <- res_ao$resultado
  resultados_totais[[length(resultados_totais)+1]] <- res_co$resultado
  if (!exists("graficos")) graficos <- list()
  graficos[["Aro"]][[planilha]] <- res_aro$grafico
  graficos[["Po"]][[planilha]] <- res_po$grafico
  graficos[["Ao"]][[planilha]] <- res_ao$grafico
  graficos[["Co"]][[planilha]] <- res_co$grafico
}
```

```

}
resultado_final <- do.call(rbind, resultados_totais)
resultado_final <- resultado_final %>%
  arrange(Medida, Planilha)
write.xlsx(resultado_final, "")
pdf("", width = 10, height = 12)
for (medida in c("Aro", "Po", "Ao", "Co")) {
  plots <- graficos[[medida]]
  plots_ordenados <- plots[order(as.numeric(names(plots)))]
  grid.arrange(grobs = plots_ordenados, ncol = 2, top = paste("Regressões Ct x", medida))
}
dev.off()
gerar_analise_desempenho <- function(resultados) {
  cores <- brewer.pal(4, "Set2")
  g1 <- ggplot(resultados, aes(x = Planilha, y = R2, color = Medida)) +
    geom_line(size = 1) +
    geom_point(size = 1.5) +
    scale_color_manual(values = cores) +
    labs(
      title = "Variação do R² com o número de Pontos Homólogos",
      x = "Número de Pontos Homólogos",
      y = expression(R^2)
    ) +
    theme_minimal(base_size = 10) +
    theme(legend.title = element_blank())
  g2 <- ggplot(resultados, aes(x = Planilha, y = b, color = Medida)) +
    geom_line(size = 1) +
    geom_point(size = 1.5) +
    scale_color_manual(values = cores) +
    labs(
      title = "Variação do coeficiente b com o número de Pontos Homólogos",
      x = "Número de Pontos Homólogos",
      y = "Coeficiente b"
    ) +
    theme_minimal(base_size = 10) +
    theme(legend.title = element_blank())
  g3 <- ggplot(resultados, aes(x = Planilha, y = epb, color = Medida)) +
    geom_line(size = 1) +
    geom_point(size = 1.5) +
    scale_color_manual(values = cores) +
    labs(
      title = "Variação do erro padrão de b com o número de Pontos Homólogos",
      x = "Número de Pontos Homólogos",
      y = "Erro padrão de b (epb)"
    ) +
    theme_minimal(base_size = 10) +
    theme(legend.title = element_blank())
  pdf("", width = 10, height = 8)
  grid.arrange(g1, g2, g3, ncol = 1, heights = c(1, 1, 1))
  dev.off()
  message("☑️ Análise de desempenho comparativa gerada com sucesso!")
}
gerar_analise_desempenho(resultado_final)
message("☑️ Script completo executado com sucesso: Excel e gráficos gerados.")

```

ANEXO 2 – SCRIPT EM R PARA COMPARAÇÃO ENTRE MÉTODOS M1 E M2

```
library(readxl)
library(openxlsx)
library(dplyr)
library(broom)
library(ggplot2)

arquivo_entrada <- ""
arquivo_saida_excel <- ""
diretorio_saida_graficos <- ""
nome_planilha_entrada <- 1
dados <- tryCatch({
  read_excel(arquivo_entrada, sheet = nome_planilha_entrada)
}, error = function(e) {
  stop(paste("Erro ao ler o arquivo Excel:", arquivo_entrada, "\nVerifique o caminho e se o arquivo existe. Erro original:", e$message))
})

nomes_colunas <- c("ID", "Ct", "M2OA", "M2OP", "M2OL", "M2OH", "M1OA", "M1OP", "M1OL", "M1OH")
if (ncol(dados) != length(nomes_colunas)) {
  warning(paste("Número de colunas no arquivo (", ncol(dados), ") diferente do esperado (", length(nomes_colunas), "). Verifique a planilha."))
}

colnames(dados) <- nomes_colunas
pares <- list(
  c("M1OA", "M2OA", "Aro", "Área do otólito", "mm²"),
  c("M1OP", "M2OP", "Po", "Perímetro do otólito", "mm"),
  c("M1OL", "M2OL", "Co", "Comprimento do otólito", "mm"),
  c("M1OH", "M2OH", "Ao", "Altura do otólito", "mm")
)

colunas_necessarias <- unique(c(unlist(lapply(pares, `[`, 1:2)), "Ct"))
colunas_faltantes <- setdiff(colunas_necessarias, colnames(dados))
if (length(colunas_faltantes) > 0) {
  stop(paste("Colunas necessárias não encontradas nos dados:", paste(colunas_faltantes, collapse=" ")))
}

if (!dir.exists(diretorio_saida_graficos)) {
  dir.create(diretorio_saida_graficos, recursive = TRUE)
}

wb <- createWorkbook()
addWorksheet(wb, "Resultados_Estatisticos")
addWorksheet(wb, "Resultados_Regressoes")
resultados_estatisticos <- data.frame()
resultados_regressoes <- data.frame()
for (i in 1:length(pares)) {
  par <- pares[[i]]
  m1_col <- par[1]
  m2_col <- par[2]
  nome_curto <- par[3]
  nome_completo <- par[4]
  unidade_y <- par[5]
  dados_pareados <- dados[!is.na(dados[[m1_col]]) & !is.na(dados[[m2_col]]), ]
  m1_pareado <- dados_pareados[[m1_col]]
  m2_pareado <- dados_pareados[[m2_col]]
  diff_pareado <- m1_pareado - m2_pareado
  if (length(m1_pareado) > 2) {
    m1_mean <- round(mean(m1_pareado), 4); m1_median <- round(median(m1_pareado), 4); m1_sd <- round(sd(m1_pareado), 4)
    m2_mean <- round(mean(m2_pareado), 4); m2_median <- round(median(m2_pareado), 4); m2_sd <- round(sd(m2_pareado), 4)
    shapiro_result <- tryCatch(shapiro.test(diff_pareado), error = function(e) NULL)
    shapiro_p <- if (!is.null(shapiro_result)) formatC(shapiro_result$p.value, format = "e", digits = 4) else NA
    wilcox_p <- NA; t_p <- NA
    if (!is.na(shapiro_p) && as.numeric(shapiro_p) >= 0.05) {
      test_result <- tryCatch(t.test(m1_pareado, m2_pareado, paired = TRUE), error = function(e) NULL)
      if (!is.null(test_result)) t_p <- formatC(test_result$p.value, format = "e", digits = 4)
    } else {
      test_result <- tryCatch(wilcox.test(m1_pareado, m2_pareado, paired = TRUE), error = function(e) NULL)
      if (!is.null(test_result)) wilcox_p <- formatC(test_result$p.value, format = "e", digits = 4)
    }
  }
  resultados_estatisticos <- rbind(resultados_estatisticos, data.frame(
    Medida = nome_curto, N_Pares = length(m1_pareado),
    Metodo1_Media = m1_mean, Metodo1_Mediana = m1_median, Metodo1_SD = m1_sd,
```

```

Metodo2_Media = m2_mean, Metodo2_Mediana = m2_median, Metodo2_SD = m2_sd,
Shapiro_Wilk_Diff_PValue = shapiro_p, Wilcoxon_PValue = wilcox_p, T_Test_Pareado_PValue = t_p,
stringsAsFactors = FALSE ))
} else { warning(paste("Não há dados pareados suficientes para testes na medida:", nome_curto)) }
df_m1_reg <- dados %>% select(Ct, y = all_of(m1_col)) %>% filter(!is.na(Ct) & !is.na(y) & Ct > 0 & y > 0)
df_m2_reg <- dados %>% select(Ct, y = all_of(m2_col)) %>% filter(!is.na(Ct) & !is.na(y) & Ct > 0 & y > 0)
a_m1 <- b_m1 <- r2_m1 <- epa_m1 <- epb_m1 <- NA
a_m2 <- b_m2 <- r2_m2 <- epa_m2 <- epb_m2 <- NA
modelo_m1 <- modelo_m2 <- NULL
n_m1 <- nrow(df_m1_reg)
n_m2 <- nrow(df_m2_reg)
if (n_m1 > 2) {
  df_m1_reg <- df_m1_reg %>% mutate(log_Ct = log(Ct), log_y = log(y))
  modelo_m1 <- tryCatch(lm(log_y ~ log_Ct, data = df_m1_reg), error = function(e) {warning(paste("Erro ao ajustar modelo M1 para", nome_curto, ":", e$message));
NULL}))
  if (!is.null(modelo_m1)) {
    sumario_m1 <- summary(modelo_m1); coefs_m1 <- coef(sumario_m1)
    log_a_m1 <- coef(modelo_m1)[1]; b_m1 <- coef(modelo_m1)[2]
    a_m1 <- exp(log_a_m1)
    epa_m1 <- a_m1 * coefs_m1[1, 2]; epb_m1 <- coefs_m1[2, 2]
    r2_m1 <- sumario_m1$r.squared
  }
}
if (n_m2 > 2) {
  df_m2_reg <- df_m2_reg %>% mutate(log_Ct = log(Ct), log_y = log(y))
  modelo_m2 <- tryCatch(lm(log_y ~ log_Ct, data = df_m2_reg), error = function(e) {warning(paste("Erro ao ajustar modelo M2 para", nome_curto, ":", e$message));
NULL}))
  if (!is.null(modelo_m2)) {
    sumario_m2 <- summary(modelo_m2); coefs_m2 <- coef(sumario_m2)
    log_a_m2 <- coef(modelo_m2)[1]; b_m2 <- coef(modelo_m2)[2]
    a_m2 <- exp(log_a_m2)
    epa_m2 <- a_m2 * coefs_m2[1, 2]; epb_m2 <- coefs_m2[2, 2]
    r2_m2 <- sumario_m2$r.squared
  }
}
resultados_regressoes <- rbind(resultados_regressoes, data.frame(
  Espécie = "ATHBRA", Relacao = paste("Ct x", nome_curto),
  N_Pontos_M1 = n_m1, a_M1 = round(a_m1, 3), epa_M1 = round(epa_m1, 3), b_M1 = round(b_m1, 3), epb_M1 = round(epb_m1, 3), r2_M1 = round(r2_m1, 3),
  N_Pontos_M2 = n_m2, a_M2 = round(a_m2, 3), epa_M2 = round(epa_m2, 3), b_M2 = round(b_m2, 3), epb_M2 = round(epb_m2, 3), r2_M2 = round(r2_m2, 3),
  stringsAsFactors = FALSE
))
criar_anotacao <- function(a, b, r2) {
  if (any(is.na(c(a, b, r2)))) return("Modelo não ajustado")
  eq_expression <- substitute(italic(y) == a %.% italic(X)^b,
    list(a = format(a, digits = 3, scientific = FALSE),
      b = format(b, digits = 3, scientific = FALSE)))
  r2_expression <- substitute(italic(R)^2 == r2,
    list(r2 = format(r2, digits = 3)))
  list(eq = eq_expression, r2 = r2_expression)
}
curva_potencial <- function(x, a, b) { a * x^b }
salvar_grafico <- function(plot_obj, filename) {
  tryCatch({
    ggsave(filename = filename, plot = plot_obj, width = 6, height = 5, units = "in", device = cairo_pdf)
    print(paste("Gráfico salvo com sucesso em:", filename))
  }, error = function(e) {
    warning(paste("Erro ao salvar o gráfico:", filename, "\nErro original:", e$message))
  })
}
eixo_x_label <- "Comprimento total (mm)"
anotacao_m1 <- criar_anotacao(a_m1, b_m1, r2_m1)
titulo_m1 <- paste0("Ct x", nome_curto, " M1")
eixo_y_label_m1 <- paste0(nome_completo, " (", unidade_y, ") M1")
g1 <- ggplot(df_m1_reg, aes(x = Ct, y = y)) +
  geom_point(color = "darkblue", shape = 16) +
  labs(title = titulo_m1, x = eixo_x_label, y = eixo_y_label_m1) +
  theme_minimal(base_family = "DejaVu Sans")

```

```

if (!is.list(anotacao_m1)) {
  g1 <- g1 + annotate("text", x = Inf, y = Inf, label = anotacao_m1, hjust = 1.1, vjust = 2, size = 4, color = "red", family = "DejaVu Sans")
} else {
  if (!any(is.na(c(a_m1, b_m1)))) {
    g1 <- g1 + stat_function(fun = curva_potencial, args = list(a = a_m1, b = b_m1), color = "lightblue", linewidth = 1)
  }
  g1 <- g1 + annotate("text", x = -Inf, y = Inf, label = deparse(anotacao_m1$eq), parse = TRUE, hjust = -0.1, vjust = 1.1, size = 4, family = "DejaVu Sans", colour =
"black") +
  annotate("text", x = -Inf, y = Inf, label = deparse(anotacao_m1$r2), parse = TRUE, hjust = -0.1, vjust = 2.6, size = 4, family = "DejaVu Sans", colour = "black")
}

nome_arquivo_g1 <- file.path(diretorio_saida_graficos, paste0("Grafico_Ct_vs_", nome_curto, "_M1.pdf"))
salvar_grafico(g1, nome_arquivo_g1)
anotacao_m2 <- criar_anotacao(a_m2, b_m2, r2_m2)
titulo_m2 <- paste0("          Ct x ", nome_curto, " M2")
eixo_y_label_m2 <- paste0(nome_completo, " (", unidade_y, ") M2")
g2 <- ggplot(df_m2_reg, aes(x = Ct, y = y)) +
  geom_point(color = "darkblue", shape = 16) +
  labs(title = titulo_m2, x = eixo_x_label, y = eixo_y_label_m2) +
  theme_minimal(base_family = "DejaVu Sans")
if (!is.list(anotacao_m2)) {
  g2 <- g2 + annotate("text", x = Inf, y = Inf, label = anotacao_m2, hjust = 1.1, vjust = 2, size = 4, color = "red", family = "DejaVu Sans")
} else {
  if (!any(is.na(c(a_m2, b_m2)))) {
    g2 <- g2 + stat_function(fun = curva_potencial, args = list(a = a_m2, b = b_m2), color = "lightblue", linewidth = 1) # Conforme user_6
  }
  g2 <- g2 + annotate("text", x = -Inf, y = Inf, label = deparse(anotacao_m2$eq), parse = TRUE, hjust = -0.1, vjust = 1.1, size = 4, family = "DejaVu Sans", colour =
"black") + # Conforme user_12
  annotate("text", x = -Inf, y = Inf, label = deparse(anotacao_m2$r2), parse = TRUE, hjust = -0.1, vjust = 2.6, size = 4, family = "DejaVu Sans", colour = "black") #
Conforme user_12
}
nome_arquivo_g2 <- file.path(diretorio_saida_graficos, paste0("Grafico_Ct_vs_", nome_curto, "_M2.pdf"))
salvar_grafico(g2, nome_arquivo_g2)
}
tryCatch({
  writeData(wb, "Resultados_Estatisticos", resultados_estatisticos)
  writeData(wb, "Resultados_Regressoes", resultados_regressoes)
  saveWorkbook(wb, arquivo_saida_excel, overwrite = TRUE)
  print(paste("Resultados salvos com sucesso em:", arquivo_saida_excel))
}, error = function(e) {
  warning(paste("Erro ao salvar o arquivo Excel:", arquivo_saida_excel, "\nVerifique as permissões e o caminho. Erro original:", e$message))
})
print(paste("Processing completed. Individual charts saved at:", diretorio_saida_graficos))

```

ANEXO 3 – CÓDIGO FONTE DO PROGRAMA MORFO EM PYTHON

```
import os
import cv2
import pandas as pd
import numpy as np
import tkinter as tk
from tkinter import filedialog, simpledialog
import tkinter.font as tkFont
import threading

textos = {
    "en": {
        "titulo": "MORFO",
        "seleccionar_pasta": "Select image folder",
        "executar_calculos": "Run calculations",
        "sobre": "About",
        "idioma": "Language",
        "sobre_texto": "MORFO\nVersion 1.0\n\nDeveloped by:\n\nDaniel Francelino Ribeiro\n\n\nAdvised by:\n\nDr. André Martins Vaz dos Santos",
        "idioma_titulo": "Languages",
        "portugues": "Português",
        "ingles": "English",
        "espanhol": "Español",
        "espera": "",
        "nomao": "Scale factor",
        "escala": "Enter the scale in mm (ex: 2.0):",
        "columnas_excel": ["ID", "Area (mm²)", "Perimeter (mm)", "Height (mm)", "Length (mm)"],
        "progresso_preprocessamento": "{atual} of {total} images pre-processed",
        "progresso_processamento": "{atual} of {total} images processed"
    },
    "pt": {
        "titulo": "MORFO",
        "seleccionar_pasta": "Selecionar pasta de imagens",
        "executar_calculos": "Executar cálculos",
        "sobre": "Sobre",
        "idioma": "Idioma",
        "sobre_texto": "MORFO\nVersão 1.0\n\nDesenvolvido por:\n\nDaniel Francelino Ribeiro\n\n\nOrientado por:\n\nDr. André Martins Vaz dos Santos",
        "idioma_titulo": "Idiomas",
        "portugues": "Português",
        "ingles": "English",
        "espanhol": "Español",
        "espera": "",
        "nomao": "Fator de escala",
        "escala": "Digite o valor da escala em milímetros (ex: 2.0):",
        "columnas_excel": ["ID", "Área (mm²)", "Perímetro (mm)", "Altura (mm)", "Comprimento (mm)"],
        "progresso_preprocessamento": "{atual} de {total} imagens pré-processadas",
        "progresso_processamento": "{atual} de {total} imagens processadas"
    },
    "es": {
        "titulo": "MORFO",
        "seleccionar_pasta": "Seleccionar carpeta de imágenes",
        "executar_calculos": "Ejecutar cálculos",
        "sobre": "Acerca de",
        "idioma": "Idioma",
        "sobre_texto": "MORFO\nVersión 1.0\n\nDesarrollado por:\n\nDaniel Francelino Ribeiro\n\n\nOrientado por:\n\nDr. André Martins Vaz dos Santos",
        "idioma_titulo": "Idiomas",
        "portugues": "Português",
        "ingles": "English",
        "espanhol": "Español",
        "espera": "",
        "nomao": "Factor de escala",
        "escala": "Ingresa el valor de la escala en milímetros (ej.: 2.0):",
        "columnas_excel": ["ID", "Área (mm²)", "Perímetro (mm)", "Altura (mm)", "Longitud (mm)"],
        "progresso_preprocessamento": "{atual} de {total} imágenes preprocesadas",
        "progresso_processamento": "{atual} de {total} imágenes procesadas"
    }
}
```

```

idioma_atual= "en"
def selecionar_pasta():
    base_dir = filedialog.askdirectory()
    input_directory = base_dir
    mask_output_directory = os.path.join(base_dir, "mascara unificada")
    image_directory = mask_output_directory
    output_tps_file = os.path.join(mask_output_directory, "Landmarks.tps")
    os.makedirs(mask_output_directory, exist_ok=True)
def preprocess_image(image):
    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    _, binary_image = cv2.threshold(gray_image, 1, 255, cv2.THRESH_BINARY)
    return binary_image
def find_central_object(binary_image):
    h, w = binary_image.shape
    image_center = (w // 2, h // 2)
    contours, _ = cv2.findContours(binary_image, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    closest_contour, max_area = None, 0
    for contour in contours:
        area = cv2.contourArea(contour)
        if area > max_area:
            m = cv2.moments(contour)
            if m["m00"] > 0:
                cx, cy = int(m["m10"] / m["m00"]), int(m["m01"] / m["m00"])
                if abs(cx - image_center[0]) < w // 2 and abs(cy - image_center[1]) < h // 2:
                    max_area, closest_contour = area, contour
    return closest_contour
def find_and_draw_bar(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    _, thresh = cv2.threshold(gray, 200, 255, cv2.THRESH_BINARY)
    contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    bar_mask = np.zeros_like(gray)
    for contour in contours:
        x, y, w, h = cv2.boundingRect(contour)
        aspect_ratio = w / h
        if aspect_ratio > 5 and h < 100:
            cv2.drawContours(bar_mask, [contour], -1, (255,), thickness=cv2.FILLED)
    return bar_mask
def create_object_mask(image_shape, contour):
    mask = np.zeros(image_shape, dtype=np.uint8)
    if contour is not None:
        cv2.drawContours(mask, [contour], -1, (255,), thickness=cv2.FILLED)
    return mask
def process_image(image):
    binary_image = preprocess_image(image)
    closest_contour = find_central_object(binary_image)
    otolith_mask = create_object_mask(binary_image.shape, closest_contour)
    bar_mask = find_and_draw_bar(image)
    unified_mask = cv2.bitwise_or(otolith_mask, bar_mask)
    return unified_mask
def process_all_images(input_directory, mask_output_directory):
    arquivos = [f for f in os.listdir(input_directory) if f.endswith(".tif")]
    total = len(arquivos)
    for index, filename in enumerate(arquivos):
        mensagem = textos[idioma_atual][f"progresso_preprocessamento"].format(atual=index + 1, total=total)
        progress_label.config(text=mensagem)
        janela.update_idletasks()
        image_path = os.path.join(input_directory, filename)
        image = cv2.imread(image_path)
        unified_mask = process_image(image)
        mask_output_path = os.path.join(mask_output_directory, filename)
        cv2.imwrite(mask_output_path, unified_mask)
process_all_images(input_directory, mask_output_directory)
def get_contour_landmarks(image, num_landmarks=250):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    _, thresh = cv2.threshold(gray, 127, 255, cv2.THRESH_BINARY)
    contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
    if len(contours) == 0:

```

```

        raise ValueError("")
    contour = max(contours, key=cv2.contourArea)
    contour_length = cv2.arcLength(contour, True)
    step = contour_length / num_landmarks
    landmarks = []
    dist = 0
    for i in range(num_landmarks):
        while dist < i * step and len(contour) > 1:
            p1 = contour[0][0]
            p2 = contour[1][0]
            dist += np.linalg.norm(p2 - p1)
            contour = np.delete(contour, 0, axis=0)
            landmarks.append(contour[0][0])
    landmarks = np.array(landmarks)
    height = image.shape[0]
    landmarks[:, 1] = height - landmarks[:, 1]
    return landmarks

def find_scale_bar(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    _, thresh = cv2.threshold(gray, 127, 255, cv2.THRESH_BINARY)
    contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
    contours = sorted(contours, key=cv2.contourArea, reverse=True)
    if len(contours) < 2:
        raise ValueError("")
    scale_bar = contours[1]
    x_min = np.min(scale_bar[:, 0, 0])
    x_max = np.max(scale_bar[:, 0, 0])
    y_coord = np.mean(scale_bar[:, 0, 1])
    height = image.shape[0]
    lm201 = [x_min, height - y_coord]
    lm202 = [x_max, height - y_coord]
    return np.array([lm201, lm202])

def find_max_distance_landmarks_with_angle(landmarks, lm1_length, lm2_length):
    dist_matrix = np.linalg.norm(landmarks[:, np.newaxis] - landmarks, axis=2)
    max_distance = 0
    lm1 = None
    lm2 = None
    vector_length = lm2_length - lm1_length
    perp_vector = np.array([-vector_length[1], vector_length[0]])
    perp_vector = perp_vector / np.linalg.norm(perp_vector)
    for i in range(len(landmarks)):
        for j in range(len(landmarks)):
            distance = dist_matrix[i, j]
            vector = landmarks[j] - landmarks[i]
            vector_norm = np.linalg.norm(vector)
            if vector_norm > 0:
                vector_unit = vector / vector_norm
                angle_between = np.arccos(np.clip(np.dot(vector_unit, perp_vector), -1.0, 1.0))
                if angle_between < np.radians(1) and distance > max_distance:
                    max_distance = distance
                    lm1 = landmarks[i]
                    lm2 = landmarks[j]
    return lm1, lm2

def generate_dotted_line(lm1, lm2, num_points=24):
    return np.linspace(lm1, lm2, num_points)

def find_max_distance_landmarks(landmarks):
    dist_matrix = np.linalg.norm(landmarks[:, np.newaxis] - landmarks, axis=2)
    max_distance = 0
    lm1 = None
    lm2 = None
    for i in range(len(landmarks)):
        for j in range(i + 1, len(landmarks)):
            distance = dist_matrix[i, j]
            if distance > max_distance:
                max_distance = distance
                lm1 = landmarks[i]
                lm2 = landmarks[j]

```

```

        return lm1, lm2
def generate_distance_line(lm1, lm2, num_points=24):
    return np.linspace(lm1, lm2, num_points)
def save_to_tps(landmarks, scale_landmarks, dotted_line, distance_line, image_name, output_file):
    with open(output_file, 'a') as f:
        f.write("LM=300\n")
        for lm in landmarks:
            f.write(f"{lm[0]:.5f}".replace('.', ',') + " " + f"{lm[1]:.5f}".replace('.', ',') + "\n")
        for lm in scale_landmarks:
            f.write(f"{lm[0]:.5f}".replace('.', ',') + " " + f"{lm[1]:.5f}".replace('.', ',') + "\n")
        for lm in dotted_line:
            f.write(f"{lm[0]:.5f}".replace('.', ',') + " " + f"{lm[1]:.5f}".replace('.', ',') + "\n")
        for lm in distance_line:
            f.write(f"{lm[0]:.5f}".replace('.', ',') + " " + f"{lm[1]:.5f}".replace('.', ',') + "\n")
        f.write(f"IMAGE={image_name}\n")
        f.write(f"ID=0\n\n")
def process_images(directory, output_file):
    results = []
    extensoes_permitidas = (".tif")
    arquivos = [f for f in os.listdir(directory) if f.endswith(extensoes_permitidas)]
    total = len(arquivos)
    for index, filename in enumerate(arquivos):
        mensagem = textos[idioma_atual]["progresso_processamento"].format(atual=index + 1, total=total)
        progress_label.config(text=mensagem)
        janela.update_idletasks()
        image_path = os.path.join(directory, filename)
        image = cv2.imread(image_path)
        try:
            landmarks = get_contour_landmarks(image, num_landmarks=250)
            scale_landmarks = find_scale_bar(image)
            lm1_length, lm2_length = find_max_distance_landmarks(landmarks)
            lm1_perpendicular, lm2_perpendicular = find_max_distance_landmarks_with_angle(
                landmarks, lm1_length, lm2_length)
            dotted_line = generate_dotted_line(lm1_perpendicular, lm2_perpendicular, num_points=24)
            lm1_distance, lm2_distance = find_max_distance_landmarks(landmarks)
            distance_line = generate_distance_line(lm1_distance, lm2_distance, num_points=24)
            save_to_tps(landmarks, scale_landmarks, dotted_line, distance_line, filename, output_file)
            results.append(filename)
        except Exception as e:
            print(f"Erro ao processar {filename}: {e}")
    process_images(image_directory, output_tps_file)
def executar_calculos():
    a = filedialog.askopenfilename(title="Escolha o arquivo", filetypes=[("Arquivos .tps", "*.tps")])
    scale_mm = simpledialog.askfloat(
        title=textos[idioma_atual]["nomao"],
        prompt=textos[idioma_atual]["escala"])
    output_excel_file = os.path.splitext(a)[0] + "_xlsx"
def parse_tps(file_path):
    samples = []
    current_landmarks = []
    current_id = None
    with open(file_path, 'r') as file:
        for line in file:
            line = line.strip()
            if line.startswith("IMAGE="):
                current_id = line.split('=')[1].split(".tif")[0]
            elif line.startswith("LM="):
                continue
            elif line.startswith("ID="):
                continue
            elif line == "":
                if current_landmarks and current_id:
                    samples.append({
                        "id": current_id,
                        "landmarks": np.array(current_landmarks)
                    })
                current_landmarks = []

```

```

        current_id = None
    else:
        coords = list(map(float, line.replace(',', '.').split()))
        current_landmarks.append(coords)
    if current_landmarks and current_id:
        samples.append({
            "id": current_id,
            "landmarks": np.array(current_landmarks)
        })
    return samples
def calculate_area_perimeter(landmarks, scale_landmarks, scale_factor):
    scale_length = np.linalg.norm(scale_landmarks[0] - scale_landmarks[1]) / scale_factor
    contour = landmarks[:250]
    area = 0.5 * np.abs(np.dot(contour[:, 0], np.roll(contour[:, 1], 1)) -
        np.dot(contour[:, 1], np.roll(contour[:, 0], 1)))
    perimeter = np.sum(np.linalg.norm(np.diff(contour, axis=0, append=[contour[0]]), axis=1))
    height = np.sum(np.linalg.norm(np.diff(landmarks[252:276], axis=0), axis=1))
    length = np.sum(np.linalg.norm(np.diff(landmarks[276:300], axis=0), axis=1))
    return area / (scale_length ** 2), perimeter / scale_length, height / scale_length, length / scale_length
def save_to_excel(results, output_excel_file):
    output_directory = os.path.dirname(output_excel_file)
    if not os.path.exists(output_directory):
        os.makedirs(output_directory)
    df = pd.DataFrame(results, columns=textos[jidioma_atual]["colunas_excel"])
    df.to_excel(output_excel_file, index=False)
def process_tps_file(tps_file, output_excel_file, scale_factor):
    samples = parse_tps(tps_file)
    results = []
    for sample in samples:
        sample_id = sample["id"]
        landmarks = sample["landmarks"]
        contour_landmarks = landmarks[:250]
        scale_landmarks = landmarks[250:252]
        height_landmarks = landmarks[252:276]
        length_landmarks = landmarks[276:300]
        area, perimeter, height, length = calculate_area_perimeter(landmarks, scale_landmarks, scale_factor)
        results.append([sample_id, area, perimeter, height, length])
    save_to_excel(results, output_excel_file)
if __name__ == "__main__":
    tps_file = a
    output_excel_file = os.path.splitext(a)[0] + "_xlsx"
    scale_factor = scale_mm
    process_tps_file(tps_file, output_excel_file, scale_factor)
janela = tk.Tk()
janela.geometry("400x300")
janela.configure(bg="#1e1e2f")
fonte_titulo = tkFont.Font(family="Helvetica", size=16, weight="bold")
fonte_normal = tkFont.Font(family="Helvetica", size=12)
progress_label = tk.Label(janela, text=textos[jidioma_atual]["espera"], font=fonte_titulo, fg="white", bg="#1e1e2f",
    activeforeground="white", relief="flat", width=30)
progress_label.pack(pady=5, ipady=5)
btn1 = tk.Button(janela, font=fonte_titulo, fg="white", bg="#3a3a5c", activebackground="#50507a",
    activeforeground="white", relief="flat", width=30,
    command=lambda: threading.Thread(target=selecionar_pasta).start())
btn1.pack(pady=5, ipady=5)
btn2 = tk.Button(janela, font=fonte_titulo, fg="white", bg="#3a3a5c", activebackground="#50507a",
    activeforeground="white", relief="flat", width=30,
    command=executar_calculos)
btn2.pack(pady=5, ipady=5)
def mostrar_sobre():
    sobre_janela = tk.Toplevel(janela)
    sobre_janela.title(textos[jidioma_atual]["sobre"])
    sobre_janela.geometry("400x250")
    sobre_janela.configure(bg="#1e1e2f")
    label_sobre = tk.Label(sobre_janela, text=textos[jidioma_atual]["sobre_texto"], fg="white", bg="#1e1e2f",
        font=fonte_normal, justify="center")
    label_sobre.pack(expand=True, pady=20)

```

```

btn3 = tk.Button(janela, font=fonte_titulo, fg="white", bg="#3a3a5c", activebackground="#50507a",
                 activeforeground="white", relief="flat", width=30, command=mostrar_sobre)
btn3.pack(pady=5, ipady=5)
def mostrar_idioma():
    idioma_janela = tk.Toplevel(janela)
    idioma_janela.title(textos[idioma_atual]["idioma_titulo"])
    idioma_janela.geometry("130x150")
    idioma_janela.configure(bg="#1e1e2f")
    def mudar_idioma(novo_idioma):
        global idioma_atual
        idioma_atual = novo_idioma
        atualizar_textos()
        idioma_janela.destroy()
    for cod in ["en", "pt", "es"]:
        btn = tk.Button(idioma_janela, text=textos[idioma_atual][{"en": "ingles", "pt": "portugues", "es": "espanhol"}[cod]],
                       font=fonte_normal, fg="white", bg="#3a3a5c", activebackground="#50507a",
                       activeforeground="white", relief="flat", width=20,
                       command=lambda c=cod: mudar_idioma(c))
        btn.pack(pady=5, ipady=5)
btn4 = tk.Button(janela, font=fonte_titulo, fg="white", bg="#3a3a5c", activebackground="#50507a",
                 activeforeground="white", relief="flat", width=30, command=mostrar_idioma)
btn4.pack(pady=5, ipady=5)
def atualizar_textos():
    janela.title(textos[idioma_atual]["titulo"])
    progress_label.config(text=textos[idioma_atual]["espera"])
    btn1.config(text=textos[idioma_atual]["selecionar_pasta"])
    btn2.config(text=textos[idioma_atual]["executar_calculos"])
    btn3.config(text=textos[idioma_atual]["sobre"])
    btn4.config(text=textos[idioma_atual]["idioma"])
atualizar_textos()
janela.mainloop()

```