

UNIVERSIDAD AUTÓNOMA DE MADRID
FACULTAD DE MEDICINA



Facultad
de Medicina

TRABAJO FIN DE MÁSTER

IMAGE ANALYSIS FOR THE
QUALITATIVE ANALYSIS OF CARDIAC
DEVELOPMENT

Máster Universitario en Bioinformática y Biología Computacional

Autor: Marcos Serrano, Ignacio

Director: Raiola, Morena

Departamento de Control genético del desarrollo y regeneración de órganos

Director: Jiménez Carretero, Daniel

Departamento de Bioinformática

Tutor: Sánchez Cabo, Fátima

Departamento de Bioinformática



CURSO: 2024/25
FECHA: Enero, 2025

Abstract

This study presents an integrated computational framework for analyzing heart tube morphogenesis during mice embryonic development from high-resolution 3D imaging datasets. The workflow incorporates 3D membrane-based cell and nuclei segmentation, single-cell 3D mesh reconstruction, cellular feature extraction alongside with cell division characterization, and mesh mapping and registration to enable comprehensive assessments of cell-level morphology and dynamics. Membrane-based cell and nuclei segmentations, along with 3D mesh reconstructions provided reliable identifications of individual cells in 3D confocal stacks. This was achieved despite challenges in signal decay in deeper tissue regions and densely-packed structures. A dedicated classifier was implemented to distinguish major cell proliferation stages, including calibration strategies to ensure robust probability estimates. Feature extraction and mapping allowed to project cell-level features onto the consensus geometry of a previously described staged 3D ATLAS of mouse heart development, facilitating cross-embryo and stage-by-stage comparisons within a shared 3D reference. While high computational demands and the need to adapt existing software to novel purposes constrained some aspects of the analysis, the results underscore the framework's potential for unlocking deeper insights into embryonic tissue organization and developmental patterns during heart tube morphogenesis. Future refinements will aim to address computational efficiency, more robust statistical assessments and dynamic live imaging integrations, extending the framework's capacity to broader developmental studies.

Keywords: Embryonic-development, Heart tube morphogenesis, Confocal microscopy, Cell segmentation, Cell division characterization, Feature extraction, Deep learning, Model calibration, Mesh reconstruction, Mesh registration

Contents

1	Introduction	1
1.1	Premise	1
1.2	Objectives	1
1.3	Biomedical background	2
1.3.1	Heart morphogenesis	2
1.3.2	Imaging Techniques in Developmental Biology	3
1.4	State of the art	4
1.4.1	Cell segmentation	4
1.4.2	Cell division characterization	7
2	Materials	8
2.1	Biological samples	8
2.2	Imaging techniques	8
2.3	Cell division	10
2.4	Computational Framework	10
3	Methods	11
3.1	Cell segmentation	12
3.1.1	Image Pre-processing	13
3.1.2	Membrane-based instance cell segmentation	15
3.1.2.1	Semantic membrane segmentation: 3D U-Net	15
3.1.2.2	Membrane-based instance cell segmentation: Graph partitioning	16
3.1.3	Nuclei instance segmentation	17
3.2	Single-cell 3D mesh reconstruction	18
3.3	Feature extraction	20
3.3.1	Radiomics features	20
3.3.1.1	Morphological features	20
3.3.1.2	First-order features	21
3.3.2	Complex features	21
3.4	Feature-based application: Cell division	23
3.4.1	Model architecture	23
3.4.1.1	Base model architecture selection	24
3.4.1.2	Discriminative Feature Localization: Class Activation Mapping	24
3.4.1.3	Top layers adaptations	24
3.4.2	Training strategy	26
3.4.3	Model calibration	26
3.4.3.1	Evaluation Metrics	28

3.4.4	Model 3D inference extension	28
3.5	Feature Mapping	28
3.5.1	Cell-Tissue Mesh Mapping	30
3.5.2	Tissue Surface Map Computation	30
3.5.3	Atlas Feature Mapping	31
4	Results	32
4.1	Cell Segmentation and Reconstruction	32
4.1.1	Membrane-based cell instance segmentation	33
4.1.2	Nuclei instance segmentation	33
4.1.3	Single-cell 3D mesh reconstruction	34
4.1.4	Exploratory data analysis of cell counts: myocardium	34
4.2	Feature-based application: Cell division	35
4.3	Feature mapping	36
5	Discussion	38
5.1	Single-cell segmentation and 3D reconstruction	38
5.2	Cell division characterization	38
5.3	Feature mapping and atlas-level comparisons	39
5.4	Limitations and future work	39
6	Conclusions	40
References		41
A	Appendix A: Supplementary model parameters	43
A.1	Cell segmentation models parameters	43
B	Appendix B: Supplementary Tables	44
B.1	Morphological radiomics features description	44
B.2	First-order radiomics features description	45
C	Appendix C: Supplementary algorithms	46
C.1	Graph-based segmentation quality assessment approach	46

Introduction

1.1 Premise

Heart tube formation takes place during early stages of cardiogenesis, leading to the development of the very first functional organ of the body: the heart. As the primary organ of the circulatory system, the heart is essential in the origin of life in early embryo stages, through its role in distributing oxygen and nutrients while removing metabolic waste. Essentially, the early stages of cardiogenesis (Figure 1.1) are necessary for subsequent normal heart formation and correct functioning of the cardiovascular system.

It is of great importance in life sciences to deeply understand the dynamics and mechanisms that contribute to the heart's normal functioning. This still remains a central challenge in developmental biology to be addressed, especially since some congenital diseases originate during the early stages of heart development, even before the heart itself forms (Guimier et al., 2015; Markwald et al., 1998) [1] [2]. This highlights the importance of understanding the mechanisms regulating cardiogenesis, proliferation, and differentiation from these initial phases, not only for research but also for clinical applications.

The mouse is a crucial model organism for studying cellular and tissue-level dynamics due to its genetic similarity to humans, rapid development, and the availability of advanced genetic tools. This makes it a valuable resource for understanding heart morphogenesis at cellular resolution. Advances in imaging technologies and genetic labeling in the mouse model have made it possible to study these cellular dynamics in greater detail. These innovations allow capture high-resolution, cellular-level images, enabling comprehensive visualization of the entire heart with exceptional quality. Further enhanced by image processing and deep learning algorithms, these tools transform images into valuable qualitative data, facilitating the extraction of meaningful information. Yet, it remains a demanding task that requires a high level of expertise in morphogenesis along with advanced engineering skills.

This project aims to build upon previous work and data from *Pseudodynamic analysis of heart tube formation in the mouse reveals strong regional variability and early left-right asymmetry* (Esteban et al., 2022) [3] from the CNIC's research group of Genetic Control of Organ Development and Regeneration led by Miguel Torres. Hence, the project seeks to exploit the potential of imaging and deep learning techniques to develop a new computational framework to extract, analyze, and classify features in time and space of myocardial cells using whole mouse embryo high-resolution confocal microscopy 3D images. This framework aims to advance the state of the art by addressing spatial and temporal variability in myocardial cell behavior, enabling deeper insights into heart morphogenesis.

1.2 Objectives

The primary objective of this project is to develop a comprehensive computational framework for analyzing heart tube formation at the cellular level during early cardiogenesis in mice. This framework will leverage high-resolution 3D confocal microscopy data and deep learning-based segmentation along with 3D mesh reconstruction to extract, classify, visualize and analyze static and dynamic features of

myocardial cells. Specifically, the framework will address the inherent challenges in image processing and segmentation through tailored deep learning models to capture critical aspects of cell morphology, behavior, and spatial-temporal organization throughout development.

More specifically, the project will first focus on generating 3D instance segmentations at the cellular level, capturing both membrane and nuclear structures from the high-resolution confocal microscopy 3D images. From these segmentations, both morphological and first-order/intensity-based features will be extracted from individual cells and nuclei respectively. Following this, the framework will reconstruct 3D meshes from the segmentations to support the extraction of more complex features, offering insights into specific cell-tissue relationships for each embryo in the dataset. Additionally, a deep learning-based classifier will be developed to detect and label cell division events, enhancing the framework's ability to track cellular processes.

By leveraging these capabilities, the extracted features and 3D meshes will facilitate an unsupervised spatial and temporal analysis of cellular dynamics, providing a new perspective on the morphogenesis of the heart tube based on high-resolution and temporally dense 3D imaging. To integrate these features across multiple embryos in a shared reference system. The previously described stages of the ATLAS (Esteban et. al., 2022)^[3] will serve as the final coordinate reference, enabling direct comparisons of morphological or molecular measurements. Finally, this project seeks to relate cellular characteristics to dynamic tissue deformation maps, providing a comprehensive view of heart formation. By overcoming segmentation challenges with customized deep learning approaches, this framework aims to offer valuable contributions to developmental biology, expanding the scope of high-resolution imaging studies in heart morphogenesis.

1.3 Biomedical background

1.3.1 Heart morphogenesis

Heart morphogenesis is the developmental process by which the heart forms through complex cellular movements and differentiation events, eventually acquiring its complex structure and functionality. It is a fundamental area of study in developmental biology. Advances in this field are a cornerstone not only for understanding basic biological processes but also for addressing heart defects and regenerative therapies. (Buckingham et. al., 2005) [4] (Guimier et. al., 2015) [5]

As shown in Figure 1.1, myocardial progenitor cells originate in the primitive streak (PS) and migrate anteriorly around embryonic day E6.5. These cells settle beneath the head folds (HF) to form the cardiac crescent at approximately E7.5, where differentiated myocardial cells are first observed. By E8, the early cardiac tube forms through the fusion of the cardiac crescent at the midline (ML), and by E8.5, the cardiac tube begins to undergo looping. At E10.5, the heart develops well-defined chambers, but it remains a tubular structure with a ventral and dorsal orientation. By E14.5, septation divides the heart into distinct chambers connected to the pulmonary trunk (PT) and aorta (Ao), establishing separate pulmonary and systemic circulations. Deoxygenated blood enters the right atrium (RA) and is pumped to the lungs by the right ventricle (RV) via the PT. Oxygenated blood returns to the left atrium (LA) and is distributed to the body through the left ventricle (LV) and Ao (Buckingham et al., 2005) [4].

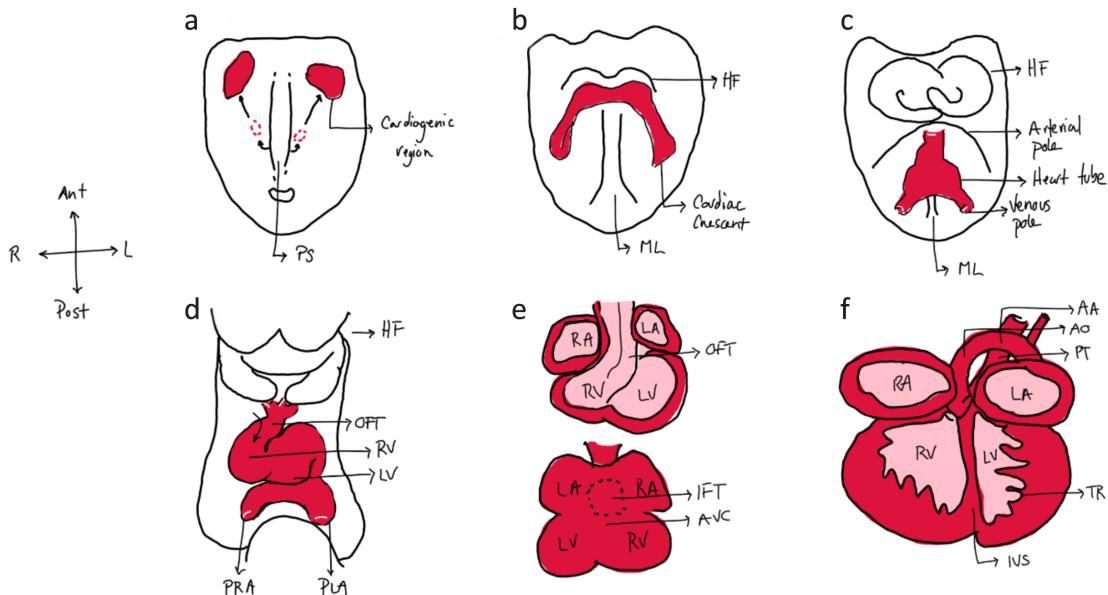


Figure 1.1: Morphogenesis of the Mouse Heart (Buckingham et al., 2005) [4]. **a**| Myocardial progenitor cells originate in the primitive streak (PS) and migrate anteriorly around embryonic day E6.5. **b-f**| Stages of heart development: cardiac crescent formation (E7.5), cardiac tube fusion (E8), looping (E8.5), chamber definition (E10.5), and septation into distinct chambers with pulmonary and systemic circulations (E14.5). Anterior–posterior and right (R)–left (L) axes are indicated.

1.3.2 Imaging Techniques in Developmental Biology

Studying early stages of heart morphogenesis presents significant challenges due to the dynamic nature of cardiac development, which involves cell migration, differentiation, and structural reorganization. High-resolution imaging is essential for capturing these processes with temporal and spatial precision. Techniques such as confocal microscopy or light-sheet microscopy are widely used, each offering distinct advantages and limitations. Confocal microscopy provides exceptional resolution and the ability to image thick tissue sections with optical sectioning, making it ideal for capturing cellular detail. However, it involves a trade-off between resolution and acquisition time and can experience signal loss at greater depths. By contrast, light-sheet microscopy can collect data quicker while reducing photobleaching, though setup complexity and alignment can lead to practical limitations. In this study, confocal microscopy was selected because it was readily available and could produce high-resolution 3D reconstructions of the embryonic heart.

Recent studies have leveraged these imaging technologies to unravel aspects of heart development. Esteban et al. (2022) [3] provided a high-resolution spatiotemporal description of heart tube morphogenesis, creating one of the first detailed 3D atlases of this process. At the same time, that study highlighted the need to delve deeper into cellular morphogenesis to fully explain the tissue-level dynamics observed, emphasizing the importance of integrating cell-scale investigations.

Esteban et al. collected and imaged samples at 20-minute temporal resolution during key developmental stages, capturing confocal images of sufficiently high resolution relative to the embryo's size to enable robust segmentation. Obtaining image datasets of this quality is particularly challenging, involving precise specimen preparation, meticulous optical configuration, and careful handling to minimize tissue deformation. Despite these efforts, morphological variability among embryos remains an intrinsic factor, complicating direct comparisons regardless of how densely or frequently samples are

acquired. The authors introduced a specific morphometric ratio as a staging parameter that captures the shape changes in the developing heart tube. This ratio enabled a continuous timeline for aligning specimens in the morphogenetic sequence, though it still relies on morphological calibration and may not capture every subtle detail across specimens.

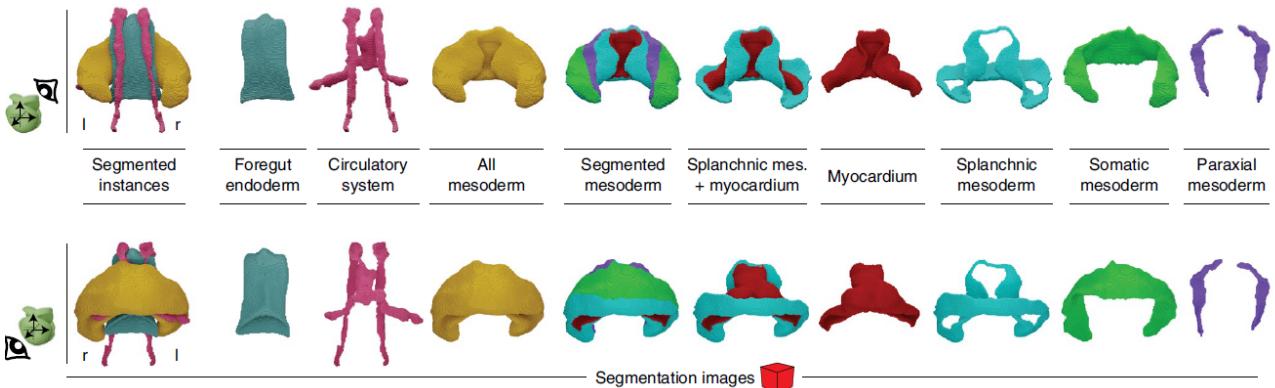


Figure 1.2: Segmentation of high-resolution confocal images reveals precise and detailed morphological representations for understanding heart development. Dorsal (top) and ventral (bottom) 3D representation of the segmented tissues. Modified from (Esteban et. al., 2022) [3]

Essentially, this study provided a detailed temporally dense confocal dataset that detailed key developmental events in heart morphogenesis. Segmentation focused on the mesodermal layers—specifically splanchnic and somatic mesoderm—with the anterior coelomic cavity, along with other pivotal structures (Figure 1.2). While this approach supported fine-grained tissue-level analysis, the sheer labor involved in extending segmentation down to the cellular scale underscored a broader need for both biological expertise and computational innovation to achieve comprehensive, cell-level resolution.

1.4 State of the art

1.4.1 Cell segmentation

Identification of individual cells is a well-studied problem for being a cornerstone of quantitative images analysis. Precise cell segmentation is crucial for tracking cellular behaviors, understanding tissue organization and identifying morphological changes. However, manual segmentation at cellular level is highly impractical, as it would require annotating hundreds of thousands of cells across many images. This labor-intensive requirement underscores the need for automated and scalable solutions to handle the intense volume and complexity of biological image data.

Although still being a challenging task, deep learning has emerged as a powerful tool for image analysis in developmental biology. Convolutional Neural Networks (CNN) (Figure 1.3) have revolutionized image analysis by leveraging specialized layers that process image data hierarchically.

Convolutional layers apply filters (kernels) to the input data to extract spatial features like edges, textures, and patterns, enabling the network to recognize structures at different scales. Batch normalization is often applied after convolutional layers to normalize activations, stabilizing and accelerating training by reducing internal covariate shifts. Activation functions, such as ReLU (Rectified Linear Unit), are then applied to introduce non-linearity, allowing the network to learn more complex patterns. Pooling layers reduce the spatial dimensions of feature maps (outputs of convolutional

layers) by summarizing regions, improving computational efficiency and capturing invariant features, often using operations like max pooling (taking the maximum value in a region) or average pooling (taking the average value in a region).

Beyond these core building blocks, more sophisticated network architectures introduce additional layer types to handle more advanced tasks like image segmentation. Deconvolutional (transposed convolution) layers, for instance, enable upsampling of reduced feature maps to reconstruct higher-resolution representations. Architectures like U-Net (Ronneberger et al., 2015) [6] employ an encoder-decoder structure, where skip connections directly link encoder and decoder layers, preserving detailed spatial information while incorporating larger-scale contextual features. Instance segmentation architectures, like Mask R-CNN (He et. al., 2017) [7], build on these principles to precisely identify and delineate individual cells within complex tissues.

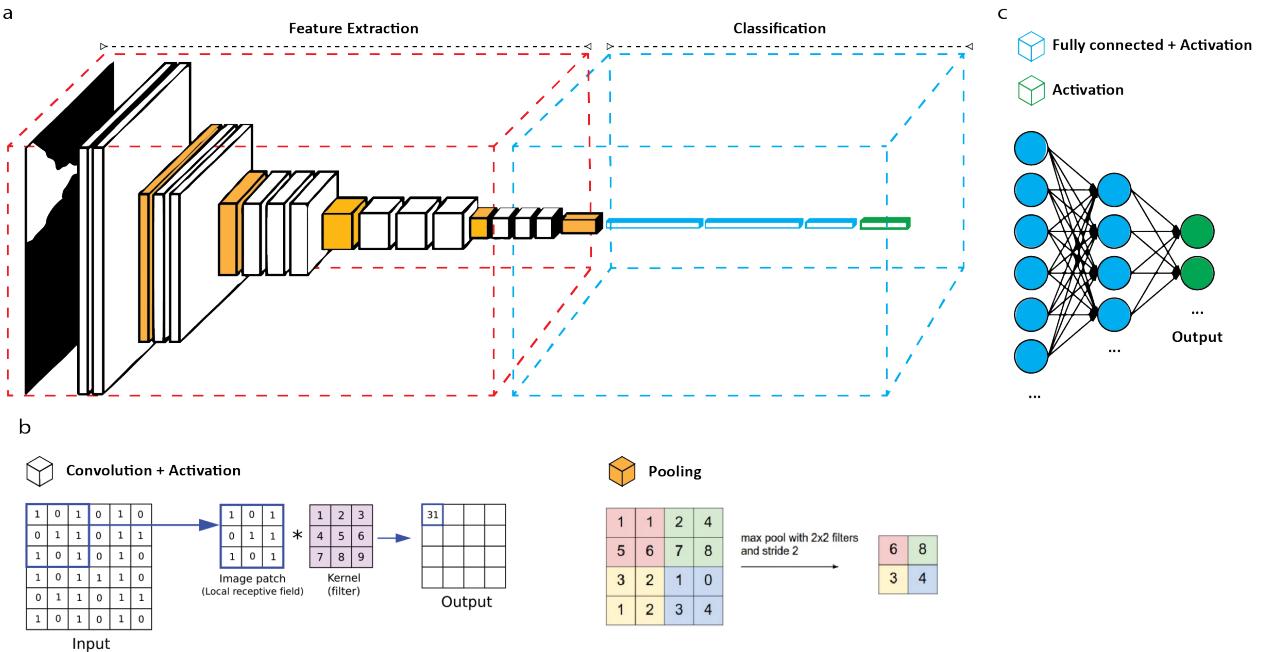


Figure 1.3: Overview of a CNN architecture for classification. **a|** The architecture is divided into two main stages: Feature Extraction (red box) and Classification (blue box). The feature extraction stage utilizes convolutional and pooling layers to learn spatial hierarchies of features from input data, while the classification stage is composed by a conventional fully connected Neural Network (NN) **b|** Convolutional layers apply filters to the input image to detect features, with activation functions enhancing non-linear properties. Pooling layers reduce spatial dimensions, improving computational efficiency while retaining essential information. **c|** Fully connected layers in the classification stage aggregate extracted features for decision-making, with activation functions mapping the outputs to probabilities or class scores.

These models highlight where traditional image processing techniques often fail, such as in highly crowded or overlapping cellular regions, making them indispensable in biomedical image analysis. For example, Wolny et al. (2020) [8] demonstrated accurate membrane segmentation in plants using U-Net-inspired architectures, showcasing their versatility for segmenting cells in various biological systems. Advancements in nuclei segmentation represent another leap in cell detection accuracy, Stringer et al. (2024) [9] introduced Cellpose 3.0, a model capable of segmenting diverse cell types across imaging modalities. Similarly, Weigert et al. (2022) [10] developed StarDist, a robust model for segmenting nuclei in crowded biological environments (Figure 1.4), overcoming challenges in cell boundary detection.

Beyond these approaches, multiple additional segmentation pipelines have been compared by González-Benito et al. (2021) [11], including DeepCell (Vanvalen wt. al., 2016) [12], 3DUnet+WS (Ciccek et. al., 2016), [13]), Mask R-CNN+WS (He et. el., 2017)[7]), and MARS (Cutrale et. al., 2017)[14]. Each method presents distinct trade-offs in terms of dimensional support (2D vs. 3D), memory usage, and adaptability to different cell morphologies, making them more or less suitable for particular imaging conditions. Table 1.1 provides a brief overview of these pipelines and highlights their core features.

Method	Key Features	Dimensionality
DeepCell	Pre-trained models Interactive correction	2D
3DUnet+WS	3D U-Net backbone Watershed for instance separation	3D
Mask R-CNN+WS	Mask R-CNN for instance seg. Watershed-based refinement	2D (extensible to 3D)
MARS	Multi-class, multi-scale Post-processing for complex tissues	2D/3D
Cellpose	Domain-general approach Flexible across various cell types	2D (extensible to 3D)
PlantSeg	Optimized for plant tissues 3D-based segmentation of large samples	2D/3D
StarDist	Star-convex shape priors Effective in crowded nuclei	2D/3D

Table 1.1: A brief comparison of additional segmentation pipelines (adapted from González-Benito et al., 2021).

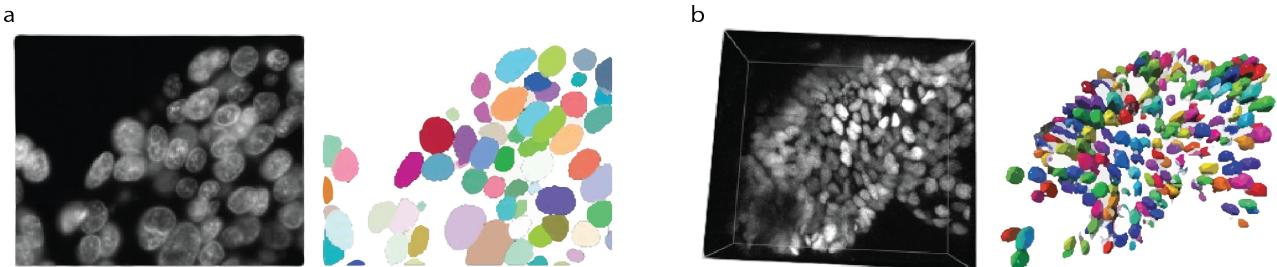


Figure 1.4: StarDist instance segmentation example (Weigert et al., 2022) [10]. **a|** 2D nuclei instance segmentation. **b|** 3D nuclei instance segmentation.

These methodologies facilitate the extraction of geometrical and morphological features from imaging data. However, despite these advancements, deep learning-based segmentation has inherit limitations, particularly the need to tailor models to specific types of biological data. For optimal performance, segmentation models must be trained or fine-tuned on images that closely match the structure and appearance of the target data. Variations in imaging techniques, cellular environments or cellular morphology can significantly affect cell visibility and boundary clarity, requiring unique model adaptations and further model customization to maintain segmentation accuracy. This process, can be restrictively time-consuming and computationally demanding.

1.4.2 Cell division characterization

Cell division is a fundamental biological process through which one cell splits into two daughter cells, ensuring growth, tissue maintenance, and regeneration. In eukaryotic cells, mitosis is the primary mechanism, typically divided into four main stages. During prophase, the cell's genetic material condenses, and the nuclear envelope breaks down. In metaphase, the chromosomes line up in the cell's center. Anaphase follows, as each chromosome pair splits and moves to opposite sides of the cell. Finally, telophase sees new nuclear membranes form around these separated chromosomes. Once cytokinesis completes, two distinct daughter cells emerge, each bearing a full set of genetic material (Alberts et. al., 2002) [15]

The assessment of cellular proliferation is a cornerstone in developmental biology. Understanding the dynamics and regulation of cell division is crucial for figuring fundamental biological processes and uncovering the mechanisms underlying normal development and disease. Automated methods for classifying cell division events have become essential tools, overcoming the limitations of manual approaches, which are highly time-consuming, prone to inter-user variability, and poorly adapted for the large-scale analyses required in modern research. Recent advancements in deep learning and computational imaging have provided scalable and highly accurate solutions, enabling detailed analysis of cellular behaviors in both 2D and 3D imaging data.

Hailstone et al. developed CytoCensus [16], a supervised machine learning framework designed to map cell divisions within tissues. The model supports both 2D and 3D datasets and utilizes user-provided annotations for training. This approach achieves high accuracy while significantly reducing the time required for analysis compared to manual methods. Another innovative method, DetectDiv (Aspert et al., 2022) [17], combines high-throughput microfluidic imaging with deep learning-based analysis to track cell divisions over time. This approach employs time-series classification and semantic segmentation to reconstruct cellular replicative lifespans and extract temporal metrics such as division duration. Operating on 2D datasets, DetectDiv demonstrates high accuracy in detecting cell division events across large-scale data.

These methods leverage deep learning to classify cell division stages using CNN-based architectures (Figure 1.3), in these cases Inception V1 (Szegedy et al., 2016) [18], optimized for 2D images. By integrating domain knowledge into the classification process, these techniques enhance predictive accuracy, facilitating the detailed characterization of cell division dynamics. Their ability to handle diverse datasets and provide high-throughput analyses makes them invaluable tools for embryonic development research.

Nevertheless, applying such existing approaches directly can prove challenging if the imaging protocols or tissue morphologies differ enough from their original training data. Consequently, custom classifiers remain necessary to address particular constraints of embryonic heart development, such as combining membrane and nuclear signals or distinguishing closely related cell-cycle stages in high-resolution 3D volumes.

Materials

2.1 Biological samples

The mouse transgenic line used in this study carried *Mesp1*^{Cre} and *R26R*^{mTmG} alleles, allowing the labeling of mesodermal cells, specifically in the head and cardiogenic areas, with membrane-localized GFP (mGFP) fluorescent protein. DAPI was used as a nuclear stain across all tissues to facilitate the identification of cellular structures and organization within the developing heart regions. [3] Hence, the raw images consist of two fluorescent channels: the green channel for the labeled membranes and the blue channel to label the nuclei (Figure 2.1b).

Within the cardiogenic region, four main tissues were segmented by Esteban et. al.: myocardium (MYO), splanchnic mesoderm (SPL), somatic mesoderm (SOM), and the proximal heart mesoderm (HPM). These tissues play distinct roles in heart morphogenesis and were identified using manual segmentation (Figure 2.1d-e). Their precise identification is crucial for understanding the structural organization and dynamics of the developing heart tube. This project focuses in the myocardium Figure 2.1f

A total of 52 embryos were collected and imaged at distinct stages of cardiac development, covering phases from the cardiac crescent to the heart looping stages. These developmental stages align with a temporal range between embryonic window E7.5 and E8.5, capturing approximately 18 hours of embryonic growth. Within this dataset, embryos were classified into 10 different developmental stages (Figure 2.1e-f), where stages 1 through 4 correspond to cardiac crescent formation, stages 5 through 8 represent the linear heart tube phase, and stages 9 and 10 cover the initial stage of heart looping. [3]

2.2 Imaging techniques

Imaging was conducted using high-resolution confocal microscopy to capture 3D image stacks of the heart tube regions from whole mouse specimens to obtain a complete description of the tissue anatomy of the forming heart tube. Embryos were fixed using the procedure shown in Figure 2.1a, fixation in 2% paraformaldehyde (PFA) overnight, permeabilization in phosphate-buffered saline with 0.5% PBT for 30 minutes, nuclear staining using DAPI at a dilution of 1:500 overnight and optical cleaning in CUBIC I reagent for three days. The embryos were then mounted in 1% agarose and imaged in glycerol. [3]

Each image volume extends approximately 1mm in both X and Y axes, with a depth of up to $677\mu m$, ensuring coverage of the entire cardiogenic region, where these values slightly varies across specimens. Image resolution varies on the developmental stage, with XY resolution ranging from $0.38\mu m$ to $0.49\mu m$ and Z-axis resolution between $0.49\mu m$ and $2.0\mu m$. For downstream segmentation, a resized version of the original raw images, set to 1024x1024 pixels, was used, balancing the need for manageable data size with sufficient resolution for accurate analysis. Additional auxiliary files were also included, consisting of images with hand-made segmentations of the different tissues for each embryo, 3D reconstructions (meshes), and an atlas mesh representing the average shape of each group for posterior feature analysis (Figure 2.1d-f). [3]

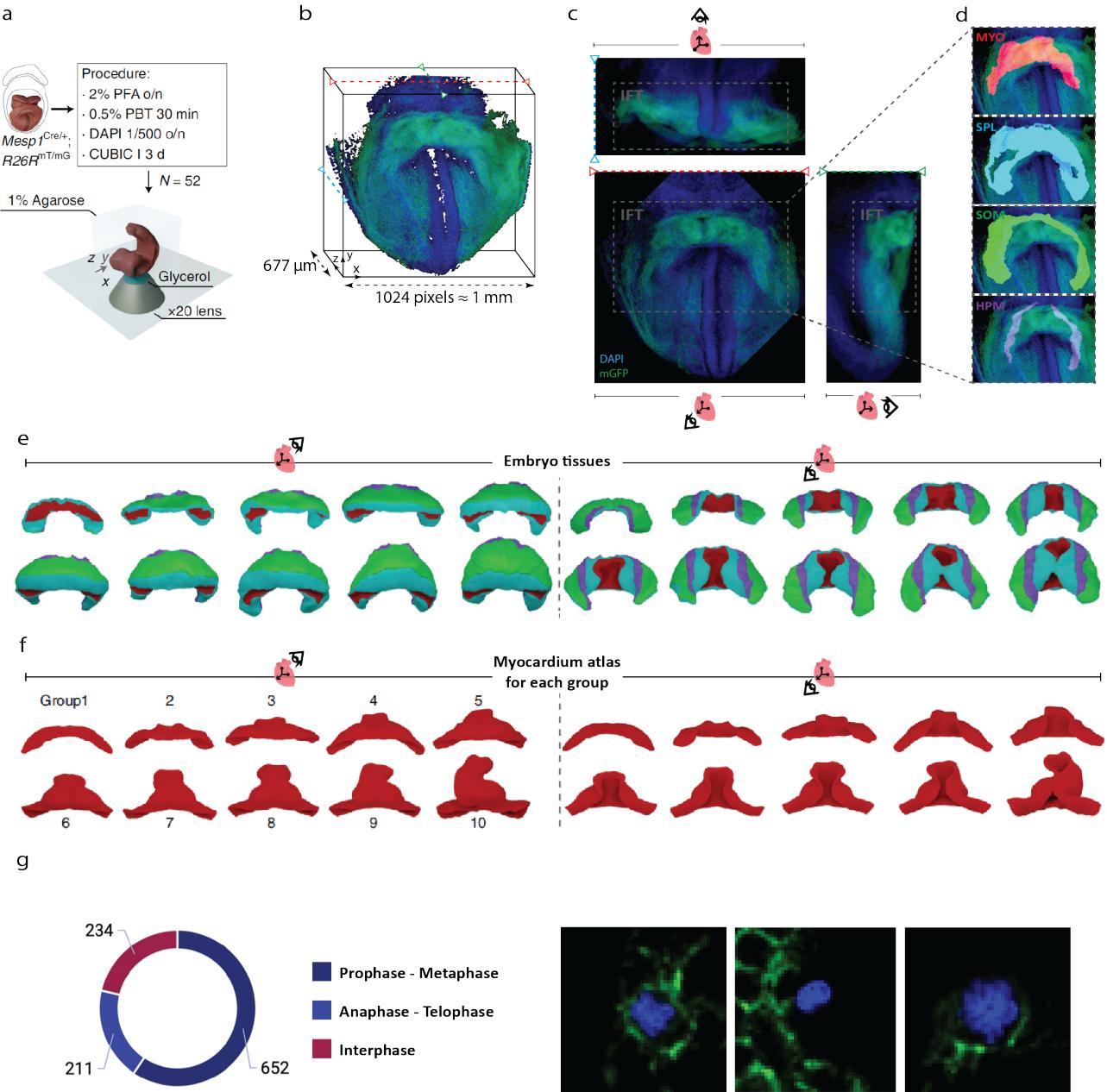


Figure 2.1: Overview of Experimental Workflow and Dataset Characteristics **a**| Schematic representation of the experimental workflow. Embryos carrying *Mesp1^{Cre/+}; R26R^{mT/mG}* alleles were fixed, stained with DAPI, and optically cleared using the CUBIC I protocol. Samples were embedded in 1% agarose and imaged using high-resolution confocal microscopy with a 20x lens. **b-c**| Example of a 3D confocal imaging volume showing multi-view projections (frontal, side, and top views) of the cardiogenic region. **d**| Tissue segmentation key structures of the cardiac inflow tract (IFT) region: myocardium (red), splanchnic mesoderm (blue), somatic mesoderm (green), and the proximal heart region (purple). **e-f**| Examples of the different tissues across the 10 developmental stages (E7.75–E8.5) derived from Esteban et. al. [3]. Top row displays 3D tissue reconstructions (color-coded by tissue type), while the bottom row isolates the myocardium atlas used as average shape between groups. **g**| Examples of single-cell manual annotations (50x50 pixels) used for assessing cellular division, including nuclei (DAPI) and membrane (mGFP) channels, along with their division labels.

2.3 Cell division

Single-cell manual annotations were performed to obtain 2D cropped images (50x50px) containing both nuclear (DAPI) and membrane (mGFP) fluorescent channels. Each annotation included labels for cell division stages, providing ground truth data for classification tasks. To minimize bias and ensure reliability, each image was annotated by at least three independent professionals from the CNIC’s research group of Genetic Control of Organ Development and Regeneration. In cases of discrepancies, annotations were resolved using majority voting, and cells were classified into three categories: Prophase/Metaphase, Anaphase/Telophase, and Interphase (Figure 2.1g).

2.4 Computational Framework

The computational framework for this study required significant processing power due to the high resolution and complexity of the imaging data. The analyses were carried out using both a local workstation and a high-performance computing (HPC) cluster provided by the CNIC’s Bioinformatics unit, enabling efficient data processing and model training and inference. The local workstation consisted of the following specifications:

Component	Specification
Operating System	Ubuntu 22.04.5 LTS
Processor	Intel Xeon E5-1620 v3 @ 3.50GHz (8 cores)
Memory	64 GiB RAM
GPU 1	NVIDIA RTX 2060 SUPER (8GiB VRAM)
GPU 2	NVIDIA GTX 1070 (8GiB VRAM)

For tasks involving heavy computational loads, such as 3D image segmentation, the HPC cluster was employed. The cluster provided extensive computational resources, including high-memory nodes and advanced GPUs, ensuring efficient processing and model scalability. The specifications of the HPC are the following:

Component	Specification
GPUs	4 x NVIDIA A100 GPUs (80GiB per GPU)
CPU Cores	190 cores
Memory	750 GiB RAM

The combination of local and HPC resources was critical to effectively handle the size and complexity of the 3D datasets. The local workstation was sufficient for most processing tasks. However, the HPC environment was essential for large-scale tasks requiring intensive computational power, such as inference of deep learning models on high-resolution data and performing batch segmentation jobs across the 52 embryos. This division of tasks ensured both scalability and efficiency in the analysis pipeline.

Methods

In this section, a comprehensive pipeline for the analysis of the 3D high-resolution confocal images of the developing mouse embryos is described. In Figure 3.1 is outlined the general workflow along with the interactions between the different sections. The pipeline is designed to capture, segment, reconstruct and analyze single-cell morphology and division patterns during early cardiac development. All the mentioned steps of the pipeline will be described along this section.

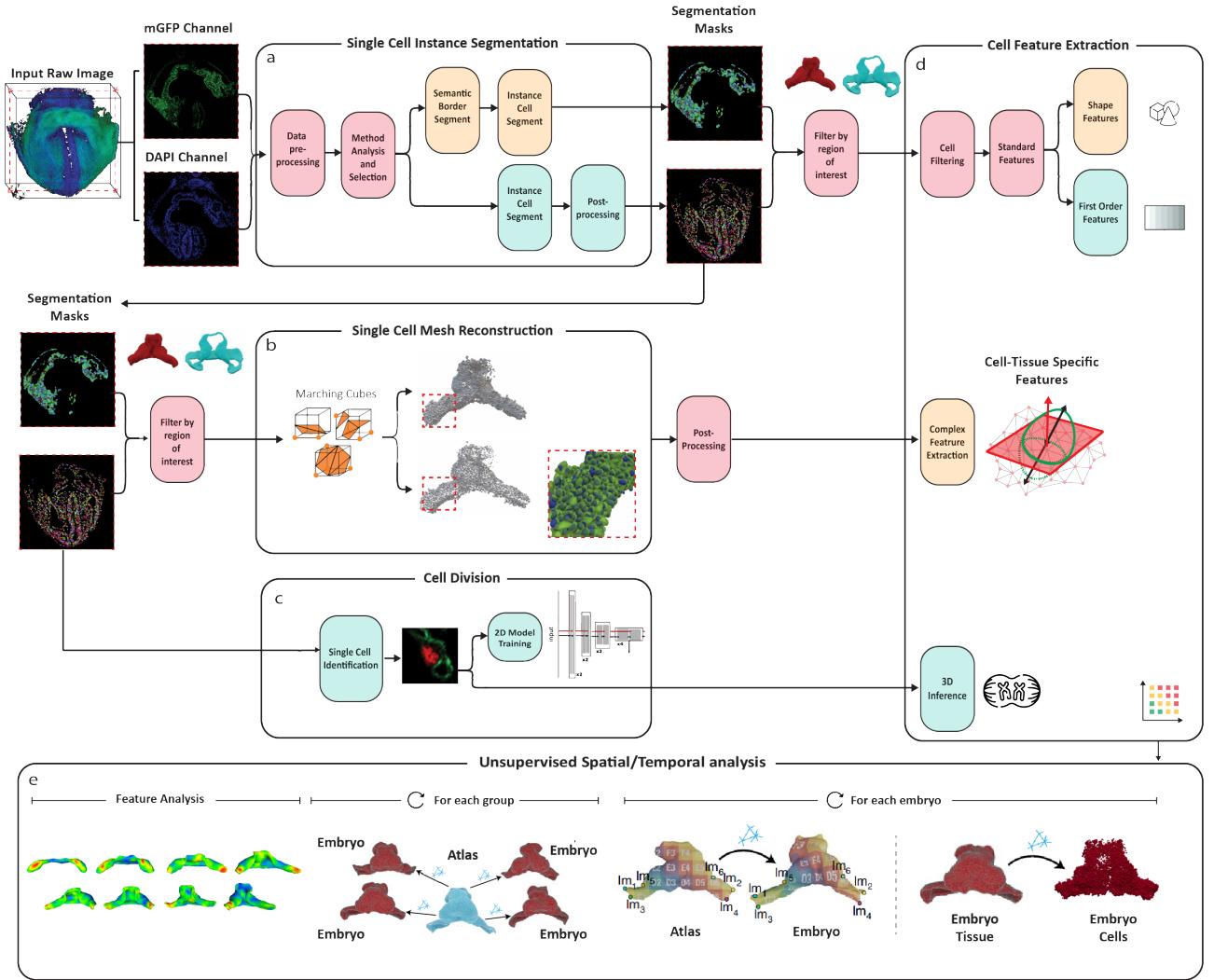


Figure 3.1: General pipeline for the development of the project. The steps are highlighted in orange for membrane analysis, blue for nuclei analysis and red involving both cellular levels. The pipeline starts with the input of the 3D image stacks from the mGFP and DAPI channels, with other additional files as the tissue manual segmentations. **a|** Single-cell instance segmentation for both membrane and nuclei along with the corresponding pre and post-processing techniques. **b|** 3D mesh reconstruction and post-processing from the segmentation masks. **c|** Cell division analysis, including the construction and training of a 2D classifier from single-nuclei images. **d|** Cell feature extraction from the images and 3D shapes along with a 3D analysis of the cell division classifier to finally being analyzed in space and time across the different specimens and groups. **e|** Unsupervised spatial and temporal analysis of the extracted features.

3.1 Cell segmentation

Image segmentation is the process of partitioning an image into meaningful localized regions to simplify analysis. There are two main kind of segmentations, semantic and instance segmentation. The key distinction between both methods is that, semantic segmentation assigns a class label to every pixel in the image, grouping all pixels belonging to the same object category (e.g. labeling by background and cell) without distinguishing between individual objects. Instance segmentation takes a step further by, not only classifying pixels into object categories but also identifies and splits each instance of an object, providing unique labels for different objects of the same category (e.g. background, cell 1, cell 2, and so on). Instance segmentation is particularly useful compared to semantic segmentation, especially for scenarios involving same-class touching objects, such as the identification and distinction of individual cells within the developing heart tube.

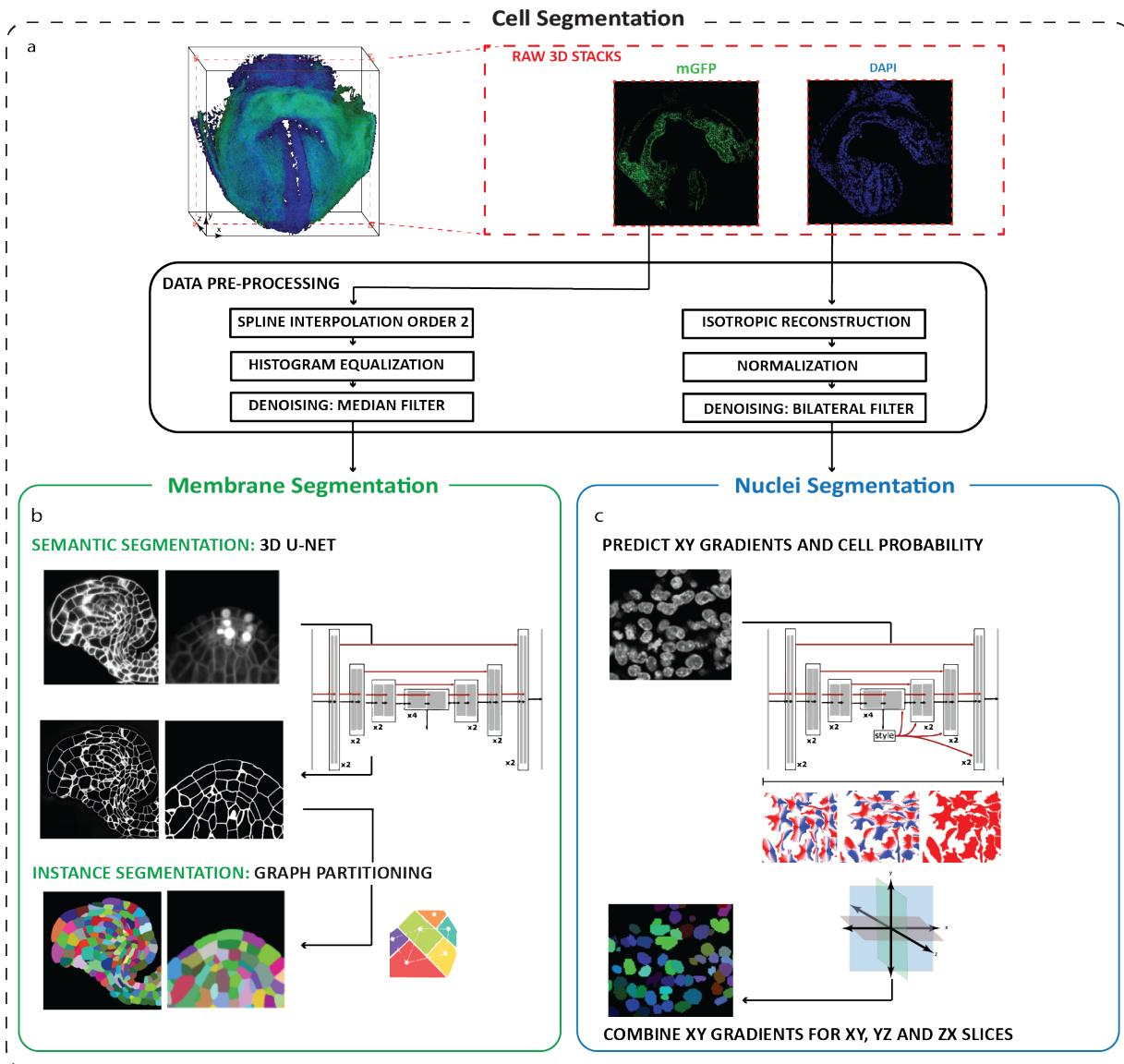


Figure 3.2: General cell segmentation workflow. **a|** The pipeline starts with the extraction of the mGFP and DAPI channels along with pre-processing steps applied differently for membranes and nuclei. **b|** Membrane 3D semantic and instance segmentation using U-Nets and Graph partitioning algorithms. **c|** Nuclei 2D instance segmentation extended into 3D segmentation.

In order to transform the images into valuable qualitative data to facilitate the extraction of meaningful information, instance cell segmentation serves as a critical step for accurately delineating individual cells within the developing heart tube, allowing for detailed analysis of cellular morphology and spatial organization. In this case, leveraging the different fluorescent markers - mGFP for labeling cell membranes (green channel) and DAPI for labeling nuclei (blue channel), allowing to segment independently nuclei and cells, using the membrane boundaries. This approach of segmenting them independently was used to extract more information per cell by extracting characteristics needing both information, see Section 3.3 for further details.

Due to the absence of ground truth labels to train our own model and manual labeling is an extreme time-restrictive approach and subject to specific-observer variability, we opted for using pre-trained models for cell segmentation. However, due to Deep Learning-specific limitations such as tailored model parameters for very specific kind of data, data pre-processing is a crucial step to enhance the image quality such as spline interpolation to improve image continuity, histogram equalization to enhance contrast, denoising filters or image normalization.

After testing multiple state-of-the-art open-source algorithms for both membrane and nuclei approaches, we selected the ones able to segment the higher number of cells with the highest boundary quality along with other metrics further explained. This initial search was assessed by visually comparing the predicted masks for the membranes and nuclei for each embryo along with their corresponding raw image using an standard software such as Fiji (ImageJ) (Schindeling et. al., 2012) [19]. After selecting the objective algorithm to perform the segmentation, the data and model parameters were further tuned to enhance the accuracy of the methods, further explained for membrane and nuclei-specific assessment.

3.1.1 Image Pre-processing

We applied the following pipeline of pre-processing steps to prepare raw image data for segmentation, enhancing quality and reducing noise. Representations of the impact of these steps on the images are shown in Figure 3.3.

For membrane segmentation (Fig 3.3a), the pre-processing includes three key steps:

Spline Interpolation of Order 2	This step smooths and interpolates image data to ensure spatial continuity, compensating for inconsistencies in voxel spacing or missing data points, particularly in the Z-axis. Ideally solving image anisotropy in Z-axis. Implemented in Plant-Seg's internal code.
Histogram Equalization	This adjusts image contrast by redistributing pixel intensities, emphasizing differences between foreground and background, and enhancing the visibility of membrane boundaries for more accurate segmentation. Implemented in 2D using Scikit-Image python's library (Van et. al., 2014)[20].
Median Filter	This reduces noise caused by imaging artifacts or background fluorescence while preserving edge integrity. By replacing each pixel's value with the median of its local neighborhood, this filter maintains critical membrane structures while removing pixel-level anomalies. Implemented in 3D using SciPy python's library (Virtanen et. al., 2020)[21].

For nuclei segmentation (Fig 3.3b), the pre-processing pipeline begins with:

Isotropic Reconstruction

This step ensures uniform voxel dimensions across all axes. Raw confocal images often exhibit anisotropic resolution (e.g., higher resolution in the XY-plane than along the Z-axis), leading to distortions in 3D reconstructions. Isotropic reconstruction homogenizes voxel dimensions, enabling appropriate 3D analysis of nuclei shapes and positions. Implemented in 3D using SciPy python's library (Virtanen et. al., 2020)[21].

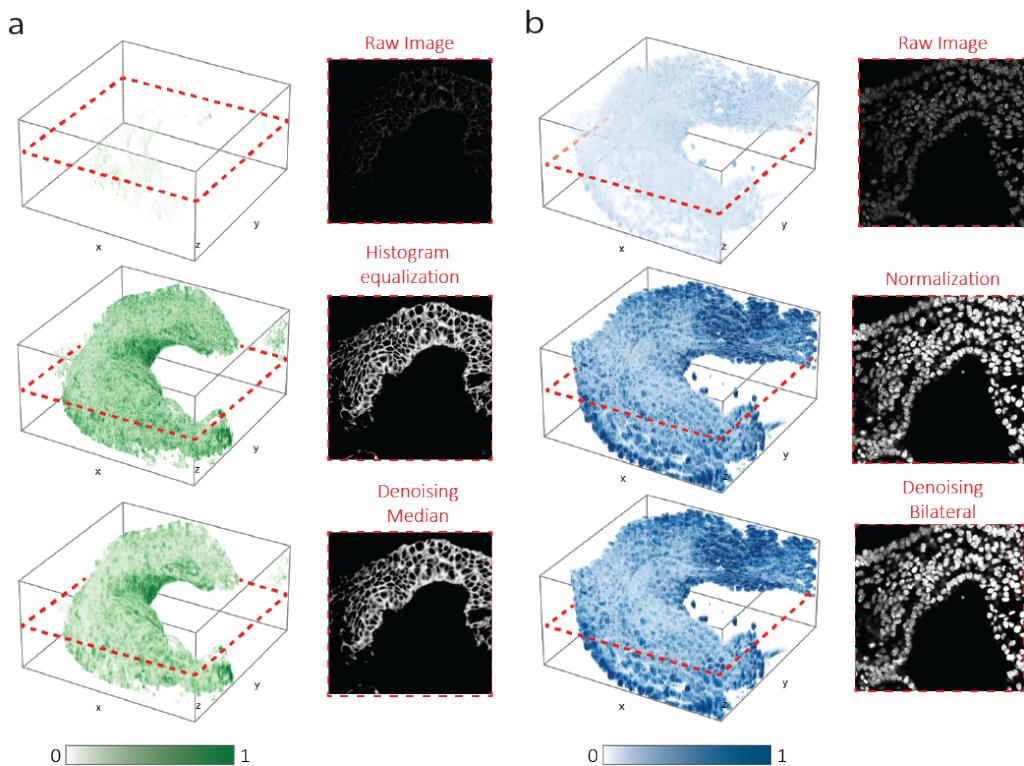
Normalization

This scales pixel intensities to a consistent range across images, reducing variability introduced by differences in imaging conditions. It compensates for signal degradation due to optical limitations, where light penetration diminishes with depth, ensuring comparability across datasets. Implemented in 2D using Scikit-Image python's library (Van et. al., 2014)[20].

Bilateral Filter

This removes noise while preserving fine edges and structural details, crucial for delineating nuclear boundaries. Unlike other smoothing filters, the bilateral filter respects intensity gradients, avoiding over-smoothing critical regions like nuclear contours. Implemented in 2D using Scikit-Image python's library (Van et. al., 2014)[20].

Together, these pre-processing steps reduce noise, standardize image quality, and enhance structural features, ensuring that the data is optimized for precise cell instance segmentation.



NOTE

Volumes are not real images, just volumetric representations of them to visualize the impact of some of the preprocessing algorithms involved in the pipeline.

Figure 3.3: Image preprocessing examples for **a**| mGFP membrane channel; Histogram Equalization and Median Filter **b**| DAPI nuclei channel; Normalization and Bilateral Filter.

3.1.2 Membrane-based instance cell segmentation

For membrane instance segmentation, PlantSeg pre-trained model was utilized (Wolny et al., 2020) [8]. Originally developed for accurate membrane segmentation in plants, the model was trained using confocal and light-sheet microscopy data. Despite being trained on plant tissues, the model demonstrated promising applicability for segmenting animal tissues, including the mouse embryos studied in this project, due to its ability to generalize across different biological structures.

The model does not directly perform instance segmentation on the image. Instead, it begins with a semantic segmentation step to enhance and better connect membrane boundaries, utilizing a pre-trained 3D U-Net (Cicek et al., 2016) [22]. The resulting prediction map is then processed with a graph partitioning approach to generate the final instance segmentation mask (Figure 3.4).

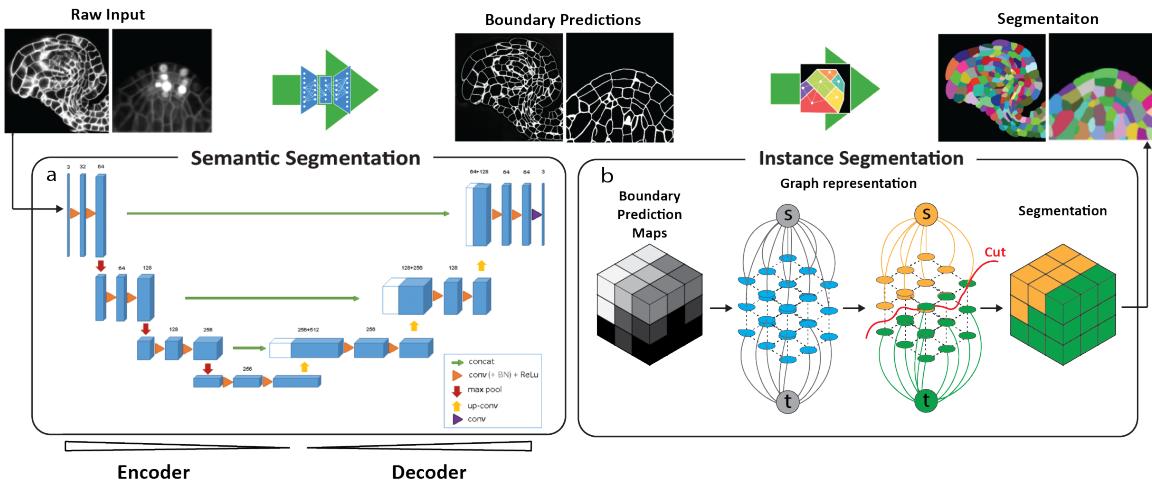


Figure 3.4: General workflow for the membrane instance segmentation approach, using as input the pre-processed mGFP membrane channel. **a**| Membrane semantic segmentation with 3D U-Net obtaining as output a prediction map. In blue representing the feature maps, orange triangles the convolutions (+ BN + ReLU) layers, in red arrows the max pooling layers, in yellow arrows the deconvolutions and in green arrows the skipping connections between the encoder and the decoder. **b**| Instance cell segmentation with graph partitioning over the prediction map.

3.1.2.1 Semantic membrane segmentation: 3D U-Net

The 3D U-Net (Figure 3.4a) extends the original U-Net architecture by replacing 2D operations with their 3D counterparts, enabling volumetric data segmentation. The classical architecture is composed of layers that include 3D convolutions, 3D max pooling, and 3D deconvolutions, each playing a distinct role. The layers are distinguished into the different sections; The encoder captures hierarchical features by progressively reducing spatial resolution while increasing the depth of feature maps. It begins with two $3 \times 3 \times 3$ convolutions that extract local patterns in the volumetric data, followed by a $2 \times 2 \times 2$ max pooling layer with strides of two in each dimension, which reduces spatial dimensions and retains key features. The decoder performs upsampling to reconstruct spatial details. It starts with a deconvolution layer of $2 \times 2 \times 2$ with strides of two in each dimension to expand feature maps, followed by two $3 \times 3 \times 3$ convolutions to refine and restore details. All convolutions use a Rectified Linear Unit (ReLU) activation function to introduce non-linearity, with batch normalization applied to stabilize training. Skip connections link corresponding layers in the encoder and decoder, preserving high-resolution features and improving the network's ability to reconstruct fine details. For binary segmentation, the architecture concludes with a $1 \times 1 \times 1$ convolution to reduce the output channels to

a single probability map, where each value represents the likelihood of a voxel belonging to the target class.

As aforementioned, leveraging deep-learning pre-trained models, in this case confocal_3D_unet_sa_meristem_cells, eliminates the necessity of manually segmenting and training new models from scratch, saving significant time and resources. However, in exchange, the prediction quality might be compromised due to domain-specific differences within the data. In addition, open boundaries in many image regions often result in segmentation masks with incomplete contours and reduced performance. To address these issues, morphological operations such as erosion, dilation, and contouring were applied as post-processing steps to improve segmentation quality. Specifically, morphological opening—erosion followed by dilation—helps removing spurious segmentations by expanding or contracting object edges, while contouring ensures smoother and more coherent segmentations.

3.1.2.2 Membrane-based instance cell segmentation: Graph partitioning

After predicting the cell boundary probability maps with the 3D U-Net, it is still necessary to group the voxels into distinct regions corresponding to individual cells for instance segmentation. This is achieved using a graph-based partitioning approach (Figure 3.4b), where the images are represented as a graph $G = (N_G, \epsilon_G)$, being N_G the set of nodes, representing the voxels, and ϵ_G the set of edges connecting the adjacent nodes, with weights derived from the predicted boundary probabilities.

The process begins with a distance transform watershed algorithm (Beucher et. al., 1992) [23], which uses the inverted distance map of the boundary predictions to seed the watershed segmentation. Watershed algorithm is a popular method for image segmentation that treats the image as a topographic surface where the intensity of each pixel represents its height. The algorithm "floods" the surface from the intensity minima and builds barriers where different water sources meet, segmenting the image into distinct regions. Therefore, being particularly useful for separating touching or overlapping objects in an image.

The distance transform computes the distance of each voxel from the nearest boundary, creating a gradient map that guides the watershed algorithm to partition the image into over-segmented regions, or supervoxels. This result is used to build a Region Adjacency Graph (RAG), where nodes represent these supervoxels, and edges between nodes capture their spatial adjacency and boundary weights.

The RAG is then processed using the Generalized Algorithm for Signed Graph Partitioning (GASP) (Bailoni et. al., 2019) [24]. GASP employs an iterative clustering strategy that starts with each supervoxel as an independent cluster. The algorithm reformulates the graph as $G = (N_G, \epsilon_G, w+, w-)$, where $w+$ and $w-$ represents the attraction and repulsion weights respectively. These weights reflect the probability of merging two clusters (originally supervoxels) or maintaining their separation. During the iterative process, clusters with the strongest attractive interactions (positive weights) are merged, while those with repulsive interactions (negative weights) remain separated. The process concludes once the clusters share only mutual repulsive interactions, resulting in a final partition of the graph.

The interaction between clusters is defined by a linkage criterion, in this case, the average linkage, which determines how weights are aggregated as clusters merge. The average linkage criterion $W = (S_u, S_v)$ computes the interaction weight between clusters S_u and S_v as $W(S_u, S_v) = \frac{1}{|S_u||S_v|} \sum_{i \in S_u} \sum_{j \in S_v} w_{ij}$. Where $|S_u|$ and $|S_v|$ are the sizes (number of nodes/voxels) of the clusters and w_{ij} are the edge weights between a pair of nodes of $|S_u|$ and $|S_v|$. These edge weights represent

the probabilities derived from the predicted boundary probability maps, indicating the likelihood of two neighboring nodes belonging to the same cluster. Essentially, they quantify the "strength" of the connection between clusters, often influenced by the shared boundary or proximity between them. During each iteration, GASP updates the interactions between the clusters and their neighbors according to the linkage criterion. The cannot-link constraint, a concept used in clustering and graph-based partitioning methods to explicitly prevent certain nodes from being grouped into the same cluster, was disabled in this implementation to allow a more flexible merging process. This choice accounts for potential imperfections in the initial boundary probabilities, as more strict implementations could lead to over-segmentation. More detailed model parameters are provided in Appendix A.

3.1.3 Nuclei instance segmentation

In confocal microscopy 3D images, the DAPI nuclei channel signal intensity decays along the Z-axis slices due to light scattering, absorption and photobleaching as the laser penetrates deeper into the tissue sample. This decay, intrinsic to confocal microscopy, results in loss of signal strength and differences in resolution along the Z-axis. This anisotropy, where resolution differs between axes, can distort the spatial representation of the nuclei structures. To address this issue, an additional pre-processing step is necessary for isotropic reconstruction, ensuring uniform voxel dimensions across all axes. This can be achieved with interpolation methods, nearest neighbors in this case, which estimate intermediate voxel intensities to balance the resolution and restore the geometric representation of the 3D image stacks, with the addition or deletion of z-slices. The addition or deletion of z-slices is based on a resampling factor of $\text{resolution}_z/\text{resolution}_{xy}$ and its inverse $\text{resolution}_{xy}/\text{resolution}_z$ for recovering the images to its original shapes.

For direct nuclei instance segmentation, Cellpose3 pretrained model was used (Stringer et. al., 2024) [9]. Cellpose3 is a generalist algorithm for cell and nuclei segmentation trained over diverse datasets. This model does not directly implement 3D operations but an extension of the 2D models to work with 3D data using a U-Net architecture (Ronneberger et. al., 2015) [6] with some modifications as baseline (Figure 3.5).

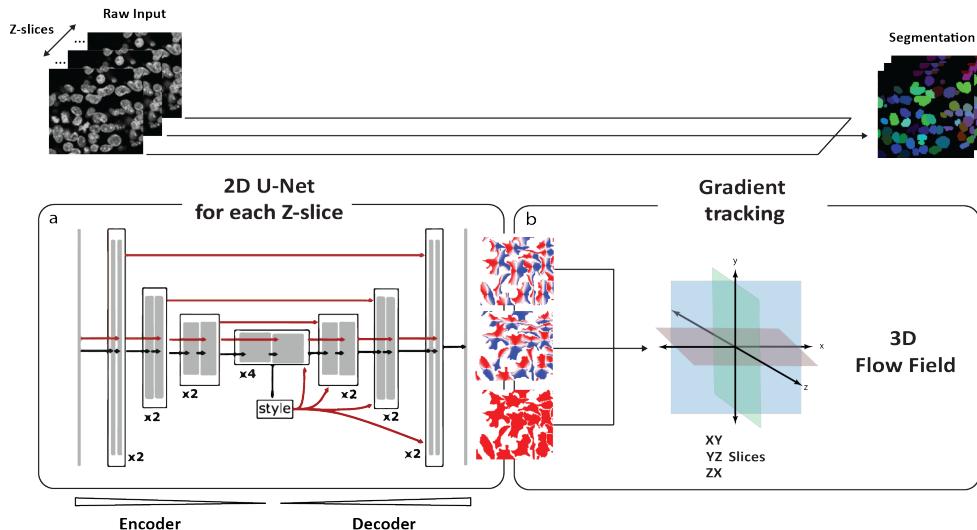


Figure 3.5: General workflow for the nuclei instance segmentation approach, using as input the pre-processed DAPI nuclei channel. **a|** 2D XY gradients and cell probability estimation with 2D U-Net. **b|** Combination of XY gradients for XY, YX and ZX slices to get a flow field in 3D to estimate cell regions.

The 2D U-Net (Figure 3.5a) used for nuclei segmentation is an extension of the classical architecture with some modifications to enhance its performance. Instead of employing feature concatenation when mixing the feature maps from the encoder to the decoder, the model uses direct summation, which simplifies the network by directly combining features. The main difference between these methods lies in the feature representation: feature concatenation expands the dimensionality of the feature space by appending features, whereas direct summation merges them, preserving the original feature dimensions, potentially enhancing efficiency while maintaining feature integration. Additionally, the standard convolutional blocks are replaced with residual blocks (He et.al., 2016) [25], which introduce shortcut connections to bypass layers, addressing the common degradation problem in deep networks, while potentially reducing overfitting and increasing convergence efficiency. Finally, this architecture leverage *style vectors* during decoding, which are obtained by applying Global Average Pooling (GAP) to the smallest feature maps. These style vectors helps the model to adapt to each specific image by tailoring the process to focus on the unique details of the image.

This model was trained to predict the XY gradients representing directional information for pixel grouping, and a binary probability map indicating whether a pixel is in or out the cell region. The training annotated cells were used alongside topological maps created through simulated diffusion, which provides the model the spatial relationships in cell structures. When making a prediction, gradient tracking is applied on the vector fields generated from the XY gradients, grouping together all pixels converging in the same point in order to differentiate the individual cells. The binary probability map is then applied to refine the segmentation, enhancing cell boundaries.

The model 3D extension leverages from the 2D implementation by predicting in 2D all XY, YZ and ZX slices independently. By predicting over the different axis, the U-Net returns six different gradient estimates per voxel. All six gradients are averaged together (Figure 3.5b) to obtain a final set of 3D vector gradients. A 3D extension of the gradient tracking step is then used to cluster the voxels converging in the same point, enabling segmentation of 3D cells. See more detailed information about the model parameters in Appendix A.

3.2 Single-cell 3D mesh reconstruction

A mesh is a representation of a 3D object defined as $M = (V, \epsilon, F)$ a set of vertices, edges and faces (Figure 3.6a). Vertices are points $V = \{v_1..v_n\}$ where each $v_i \in \mathbb{R}^3$ represents a point in a 3D space, edges are a set $\epsilon = \{e_1..e_m\}$ where each edge $e_i = (v_j, v_k)$ connects a pair of vertices v_j and v_k , and faces (often referred as triangles) are a set $F = \{f_1..f_p\}$ where each face $f_i = (v_j, v_k, v_l)$ represents a triangular element by enclosing three vertices. Meshes are widely used to approximate complex shapes with a network of simple geometric elements, commonly employed to reconstruct and visualize 3D cellular or tissular structures from volumetric data, providing a detailed and analyzable representation of morphological and spatial organization.

To reconstruct 3D meshes of individual cells from the segmentation masks, it was implemented a pipeline based on the Marching Cubes Algorithm (Lorensen et. al., 1987) [26] (Figure 3.6c). This method generates meshes by extracting isosurfaces from the 3D segmented cell images by processing each voxel independently. For each voxel, it determines which vertices (corners of the voxel) lies above or below a certain value (isovalue), using a lookup table to identify intersected edges based on different possible configurations. The algorithm interpolates the exact surface crossing points on these edges and connects them into triangles, forming local surface patches. By iteratively repeating

the process for all voxels (cubes) and joining the triangles together, it builds a continuous 3D mesh.

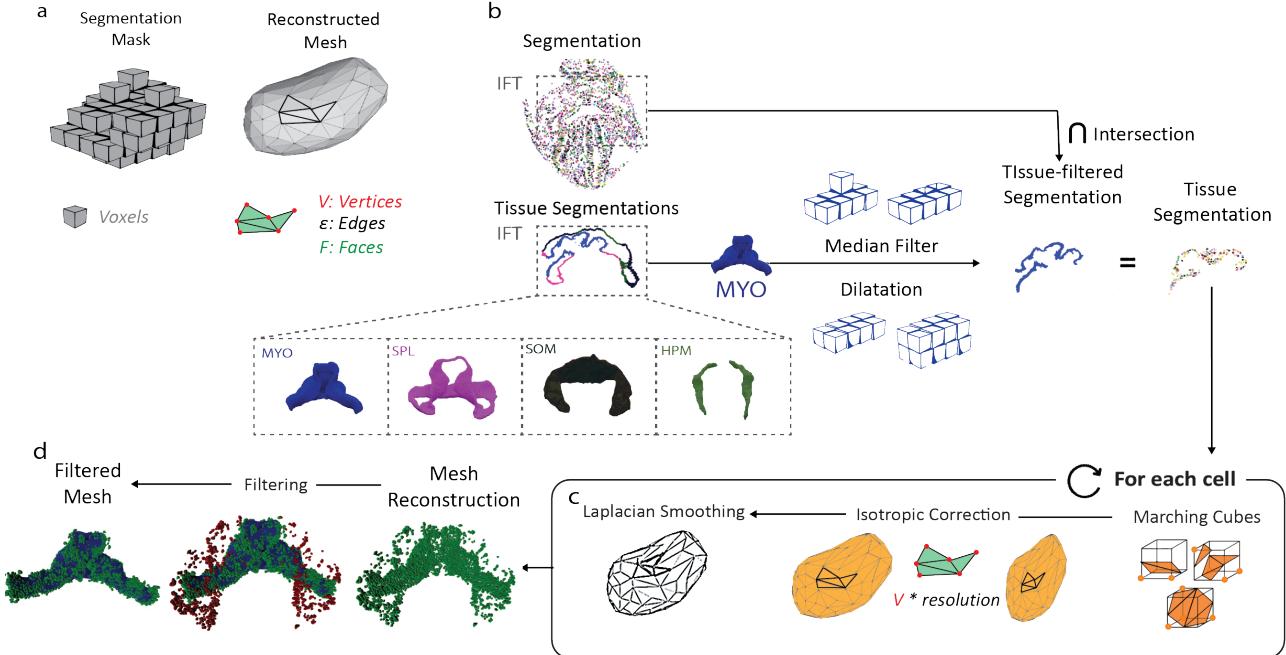


Figure 3.6: Workflow for single-cell 3D mesh reconstruction. **a**| Schematic representation of a mesh as a set of vertices (V), edges (ϵ), and faces (F). **b**| Single-cell segmentation masks and Tissue segmentation masks for key structures of the cardiac inflow tract (IFT) region: myocardium (MYO), splanchnic mesoderm (SPL), somatic mesoderm (SOM), and the proximal heart region (HPM). **c-d**| Mesh reconstruction pipeline. Single-cell segmentation masks are filtered using preprocessed tissue segmentations (median filtering and dilation). Marching Cubes generates raw 3D meshes, which are further refined with isotropic correction, Laplacian smoothing, and distance-based filtering.

As the objective is not to create a 3D reconstruction of all embryo cells but to generate single-cell meshes within the tissues of interest for further analysis, each segmentation mask was filtered using the 3D segmentations of manually segmented tissue. Before intersecting the single-cell segmentation masks with tissue segmentations, two preprocessing steps were applied to the tissue segmentation masks: Median filtering to reduce noise and smooth boundaries, and Dilation to expand the tissue boundaries slightly, ensuring sufficient overlap with adjacent cells (Figure 3.6b).

As shown in Figure 3.6c, the pipeline is then executed for each individual cell and nuclei, beginning with the application of the Marching Cubes algorithm to generate an initial 3D mesh from the segmented cell data. To address the anisotropy in voxel dimensions present in the original images, isotropic correction is applied by scaling the vertices of the reconstructed meshes according to the voxel resolutions in each axis. Subsequently, Laplacian smoothing is employed to refine the mesh surface by iteratively adjusting the position of each vertex based on its neighbors, reducing surface irregularities caused by noise while preserving the overall shape and anatomical accuracy of the cells. Additionally, the cell id information was saved as custom face and vertex metadata for correct feature mapping in posterior analysis.

Once all individual cell meshes are reconstructed, an additional post-processing step is applied to filter out cells that do not correspond to the tissue of interest (Figure 3.6d). This is achieved by excluding meshes located beyond a predefined distance threshold from the tissue, mitigating artifacts and errors introduced by noise in the tissue segmentation masks.

3.3 Feature extraction

Feature extraction is the process of deriving meaningful quantitative information from data to characterize specific properties of the objects or regions of interest. In the context of biological imaging, feature extraction translates complex visual information into measurable metrics that can describe cellular and tissular characteristics. These metrics, referred to as features, provide insights into morphological, textural, or intensity-related properties, enabling systematic analysis of biological structures. This section outlines the methodologies employed to extract radiomics-based and complex features, which together provide a comprehensive dataset for analyzing the cellular and tissular morphology of the developing cardiac regions.

For the characterization of cells in the cardiac region of interest, we followed a similar procedure as described in the mesh reconstruction pipeline (Section 3.2). Cells were filtered based on their association with the specific tissues of interest, using the 3D segmentation masks of manually annotated tissue regions. To ensure accurate cell-tissue assignments, the tissue segmentations were pre-processed, including median filtering to reduce noise and smooth boundaries, and dilation to expand tissue boundaries slightly, facilitating sufficient overlap with adjacent cells as shown in Figure 3.6).

3.3.1 Radiomics features

The extraction of quantitative data from the cells and nuclei was carried out from the segmentation masks combined with the raw images, using radiomics as baseline. Radiomics (Lambin et. al., 2012) [27] is a computational approach involving a large number of quantitative characteristics from medical images to uncover patterns and information that may not be visible through standard analysis. These features, often referred to as radiomic features, quantify properties such as shape, texture, intensity, and spatial relationships within the imaged structures. Radiomics has gained attention in fields like developmental biology due to its ability to link imaging-derived data with phenotypic outcomes, providing a robust framework for feature-based analysis, allowing for systematic characterization of cellular and tissular morphologies in a reproducible and scalable context (Lambin et. al., 2012) [27]. The set of radiomic features were divided into different groups depending on the application. For instance, in this analysis morphological features (shape-based) were extracted for both cells and nuclei and first-order features (intensity-based) for nuclei.

Before extracting features, isotropic reconstruction was applied to the raw images and segmentation masks to ensure uniform voxel dimensions across all axes. This step is particularly critical for shape-based features, as anisotropic voxel resolutions can distort the geometric properties of image objects. For instance, objects may appear stretched or compressed along the Z-axis compared to the XY-plane, leading to inaccuracies in measurements such as volume, surface area, and compactness. By rescaling the voxel dimensions to be uniform, isotropic reconstruction mitigates these distortions, ensuring that the extracted morphological features accurately reflect the true 3D geometry of the objects. This improves the reliability and comparability of shape-based feature measurements across different samples and imaging conditions.

3.3.1.1 Morphological features

The morphological or shape-based features are independent of the intensity level distribution of the raw images and therefore calculated using the segmentation masks. In this group of features

we included the descriptors of the 3D size and shape of the region of interest (ROI). A complete description of the features is listed in Appendix B Table B.1.

3.3.1.2 First-order features

The first-order or intensity-based features describe the distribution of voxel intensities within the image region defined by the segmentation mask through commonly used and basic metrics. Given that cells are segmented based on the membrane boundaries, intensity-based features do not provide relevant information when applied to the membrane channel. However, their application to the nuclei channel, stained with DAPI, could be highly relevant. DAPI staining highlights the nuclei with fluorescent signals, reporting DNA content, related to the state of the chromatin condensation and cell cycle stages, allowing for precise quantification of intensity-related properties. These features enable the characterization of nuclear morphology and chromatin organization, offering insights into cellular states and behaviors. By leveraging the signal of DAPI staining, intensity-based features provide a robust dataset for analyzing nuclear properties in the context of developmental processes. These features are therefore calculated using the nuclei segmentation masks along with the raw DAPI channel images. A complete description of the features is listed in Appendix B Table B.2.

To ensure accurate feature extraction, 2D normalization was applied to the raw images to address the depth-associated signal degradation inherent to confocal microscopy. This normalization step adjusts pixel intensity values to a consistent range across slices, compensating for the loss of signal in deeper tissue regions. This preprocessing step enhances the comparability of intensity-based features across slices and specimens.

3.3.2 Complex features

In addition to radiomic features, a second set of features, termed from now on *complex features*, was derived from the reconstructed 3D cell meshes and their corresponding tissue meshes. These complex features go beyond traditional pixel- or voxel-based metrics by leveraging the geometric and spatial properties of the 3D reconstructions. By integrating information from both the individual cell meshes and their surrounding tissue structures, these features capture intricate morphological and spatial relationships that are critical for understanding developmental processes and structural variations across specimens.

The implementation of the complex feature extraction pipeline involves different computational steps. To establish the relationship between a cell and its surrounding tissue, a graph representation of the tissue mesh is constructed, where nodes represent the face centroids of the mesh, and edges represent the euclidean distance between the adjacent face centroids (Figure 3.7a). This graph enables efficient neighborhood searches and geodesic distance calculations (Figure 3.7b), which are critical for spatial analyses. Geodesic distances, computed using Dijkstra's algorithm (Dijkstra et. al., 1959) [28] on the graph, provide a measure of the shortest path along the tissue surface.

The geodesic distance, as opposed to Euclidean distance, addresses a key challenge in mesh-based analyses: ensuring that identified neighbors are spatially and topologically connected. Using Euclidean distance in a 3D mesh could erroneously include points on opposite surfaces of the tissue, such as basal and apical regions, as close neighbors due to their proximity in 3D space. By leveraging geodesic distance, the pipeline ensures that neighborhood points are accurately restricted to those that are truly connected on the mesh, preserving the spatial integrity of the analysis.

To evaluate the alignment of a cell with its tissue, a plane is fitted to the local neighborhood of the tissue using Principal Component Analysis (PCA) (Pearson et. al., 1901) [29] (Figure 3.7c). The PCA identifies the principal axes of variation in the neighborhood points, with the normal vector to the fitted plane corresponding to the third principal component. This provides an approximation of the local tissue surface orientation. Similarly, the shape and orientation of each cell is approximated using an ellipsoid fitted from its mesh vertices (Figure 3.7c). The ellipsoid is computed by determining the covariance matrix of the cell's vertex coordinates and performing eigenvalue decomposition. The eigenvalues represent the variance along the three principal axes of the cell. The longest axis of the ellipsoid, corresponding to the largest eigenvalue, is used to determine the primary orientation of the cell.

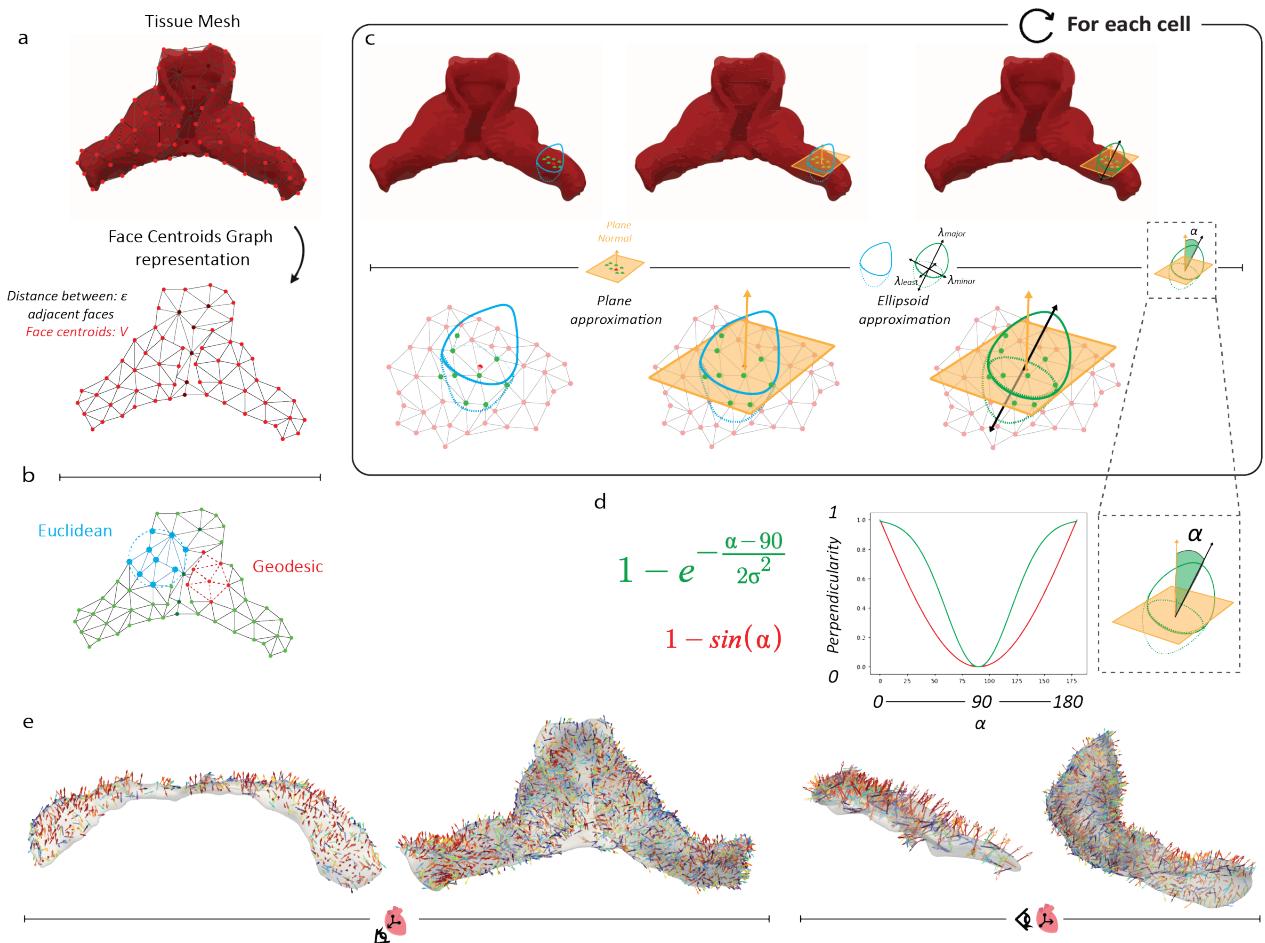


Figure 3.7: Complex feature extraction perpendicularly from 3D meshes overview. **a|** Face centroids (red) are used to construct a graph representation of the tissue mesh. **b|** Euclidean (blue) and geodesic (red) distances. **c-d|** Geodesic distances are calculated between neighboring centroids. A local plane is fitted to the neighborhood with the normal vector (orange arrow). An ellipsoid is approximated from the cell meshes. Perpendicularity between ellipsoid longest axis and the plane normal is quantified using angular deviation metrics, with mathematical adjustments shown to evaluate curvature and surface alignment. **e|** Examples of perpendicularity vector maps of two different embryos from group 1 and group 7 respectively.

Sphericity, a measure of how closely the shape of the cell approximates a sphere, is calculated using the ratios of the three eigenvalues of the covariance matrix: $S = \sqrt[3]{\frac{\lambda_{\text{minor}} \lambda_{\text{least}}}{\lambda_{\text{major}} \lambda_{\text{major}}}}$. Unlike simpler approaches that use only two eigenvalues, incorporating all three eigenvalues provides a more accurate representation of the cell's 3D shape. Higher sphericity values indicate a more compact

and spherical structure, while lower values suggest elongation or flattening.

Perpendicularity (\perp) is then calculated as the angular difference between the normal vector of the tissue plane and the longest axis of the cell ellipsoid. To address potential noise and inaccuracies arising from the approximations in plane fitting and cell axis assessment, a mapping function smooths the perpendicularity values when the angle approaches 0 or 180 degrees. This mapping, based on a Gaussian-like transformation (Figure 3.7d) (green), reduces the abrupt changes of the sinusoidal approaches (Figure 3.7d) (red) and ensures stability in the computed values, providing an smoother measure of alignment (Figure 3.7d).

Finally, **Columnarity**, a higher-order feature, combines sphericity and perpendicularity to characterize the columnar organization of cells relative to their surrounding tissues. Columnarity is defined as $(1 - \text{Sphericity})\text{Perpendicularity}$. This formulation emphasizes elongated cells (low sphericity) that are aligned perpendicularly to the tissue plane. The feature highlights cells that deviate from spherical shapes while maintaining a perpendicular orientation.

In addition, **Parallel Columnarity**, a complementary feature, combines sphericity and parallel alignment to characterize cells organized parallel to the tissue plane. Parallel Columnarity is defined as $(1 - \text{Sphericity})(1 - \text{Perpendicularity})$. This formulation highlights elongated cells that exhibit a parallel orientation relative to the tissue surface.

By integrating these features, derived from the geometric and spatial relationships between cells and tissues, the pipeline offers a robust framework for investigating structural and organizational patterns in developing cardiac regions. The use of graph-based methods, PCA, and ellipsoid approximations ensures computational efficiency while maintaining high accuracy in feature extraction.

3.4 Feature-based application: Cell division

Capturing cell proliferation events is a critical task in understanding cellular dynamics. This study is focused on classifying cell division stages by predicting three distinct classes of cell division: Interphase (no division), Prophase/Metaphase, and Anaphase/Telophase (Figure 3.8a). Integrating accurate classification of these stages within the analysis pipeline could provide insights of spatial and temporal cell behaviour, particularly in the context of heart morphogenesis where accurate classification can reveal critical insights into how cellular behaviors drive tissue development and contribute to organ formation. To achieve this, a feature-based approach was used by leveraging a 2D CNN architecture (Figure 3.8b-c), extended to 3D inference (Figure 3.8d), optimized for robustness and interpretability.

3.4.1 Model architecture

To identify the optimal model architecture, grid searches were used. These searches assessed various configurations, including different base models, loss functions and hyperparameter configurations, such as learning rates (LR) and pooling strategies. Two different searches were carried out, where in the first one the base models were mainly tested. Another search was used to, having the base model, assess the best-performing LR, loss function, pooling strategy and top (last) layers setups for the selected base model.

3.4.1.1 Base model architecture selection

Base models tested included VGG16, DenseNet121, EfficientNetV2M, EfficientNetV2L, ResNet50, MobileNetV2, NASNetMobile, and InceptionV3 [30, 31, 32, 33, 34, 35, 36]. Among these configurations, VGG-16 and DenseNet121 base models showed the best-performing architectures. VGG-16 was finally selected due to its significantly lower computational complexity and alignment with the relatively simple cell shapes, where nuclei are almost circular with not very complex morphologies. Given the way CNNs and convolutions operate, using very dense models such as DenseNet121 leads to unnecessarily deep operations for this task.

3.4.1.2 Discriminative Feature Localization: Class Activation Mapping

Class Activation Mapping (CAM) (Zhou et. al., 2016)[37] is a technique that provides interpretability by highlighting the regions of the input image that contribute most to the model’s predictions. By leveraging CAM techniques, it can be identified which information is used by the network for classifying the images, addressing the so called “black box” for neural networks. To assess the importance of the membrane channel, it was employed CAM-based top layer configurations. These layers include a transition convolutional layer followed by pooling and the final prediction (dense) layer to enable feature localization (Figure 3.8b). A combination of the activations of the prediction layer and the weights of the prediction layers enables the computation of colormaps, allowing to visualize whether the model focuses on nuclei or extended to the membrane regions (Figure 3.8c), providing insights of the impact of membrane channel.

CAM-based top layers setup were compared to a “standard” configuration, which consists of fully connected layers with activation functions. This comparison was essential to ensure that including CAM layers for interpretability did not result in model performance decay.

3.4.1.3 Top layers adaptations

The final model consists on a CNN based on the VGG16 architecture, which served as feature extractor. To adapt this model to the cell division task, the standard top (last) layers (fully-connected layers and final classification layer) were removed and replaced by a transition layer, a global pooling layer, a prediction layer at the end, serving as backbone for the CAM, enhancing interpretability by visually identifying regions of interest critical for classification quality assessment. Additionally, batch normalization and dropout layers were introduced to improve training stability and prevent overfitting.

Batch normalization normalizes the activations of the layers to a standard distribution, ideally accelerating convergence. Dropout layers randomly deactivate a fraction of neurons during training, adding regularization to improve generalization and reducing overfitting.

Transition Layer

The inserted transition layer, required to transform the activations from previous layers into a uniform dimension, consisted on a 2D convolution with a kernel size of one. This layer helps to pass down the weights from the pre-trained base model in a standard form, critical for using this layer’s activations to further generate the heatmap in the CAM step.

Global Pooling Layer & Prediction Layer

Pooling layers are used for reducing the spatial dimensions of the feature maps while preserving the most significant features, thereby lowering computational requirements and ideally mitigating

overfitting. These layers play a crucial role in summarizing the presence of features within regions of the image. Pooling strategies compared during the grid search included Global Average Pooling, which computes the average value for each feature map, and the finally selected Extended Log-Sum-Exp (LSE) pooling. Prediction layers are responsible of outputting the prediction probabilities. The prediction layer for this architecture use a softmax activation function to transform the outputs into a probability distribution across the classes, ensuring that the predicted probabilities sums one.

Both Global Pooling and Prediction Layers are designed not only to be part of the multi-class classification network but also to generate the CAM localization heatmaps. The localization with a peak in the heatmap generally corresponds to the presence of learned cell division patterns with a high probability. By performing a global pooling after the transition layer, the weights learned in the prediction layer can function as weights of spatial maps from the transition layer. The pooling layer plays a crucial role that chooses what information is passed down.

Rather than conventional max or average for Global Pooling, Log-Sum-Exp (LSE) Pooling was used. Since the LSE generally suffers from overflow/underflow problems, it was modified as follows, proposed by (Pinheiro et. al., 2015) [38]:

$$x_p = x^* + \frac{1}{r} \log \left[\frac{1}{S} \sum_{(i,j) \in S} \exp(r(x_{ij} - x^*)) \right],$$

Where x_{ij} is the activation value at (i,j) , being one location in the pooling region S , $S = sxs$ is the total number of locations in S , and the adapted $x^* = \max\{|x_{ij}|, (i,j) \in S\}$. By controlling the hyper-parameter r , the pooled value ranges from the minimum in S (When $r \rightarrow \infty$) to average ($r \rightarrow 0$), serving as an adjustable option between Max and Avearge Pooling.

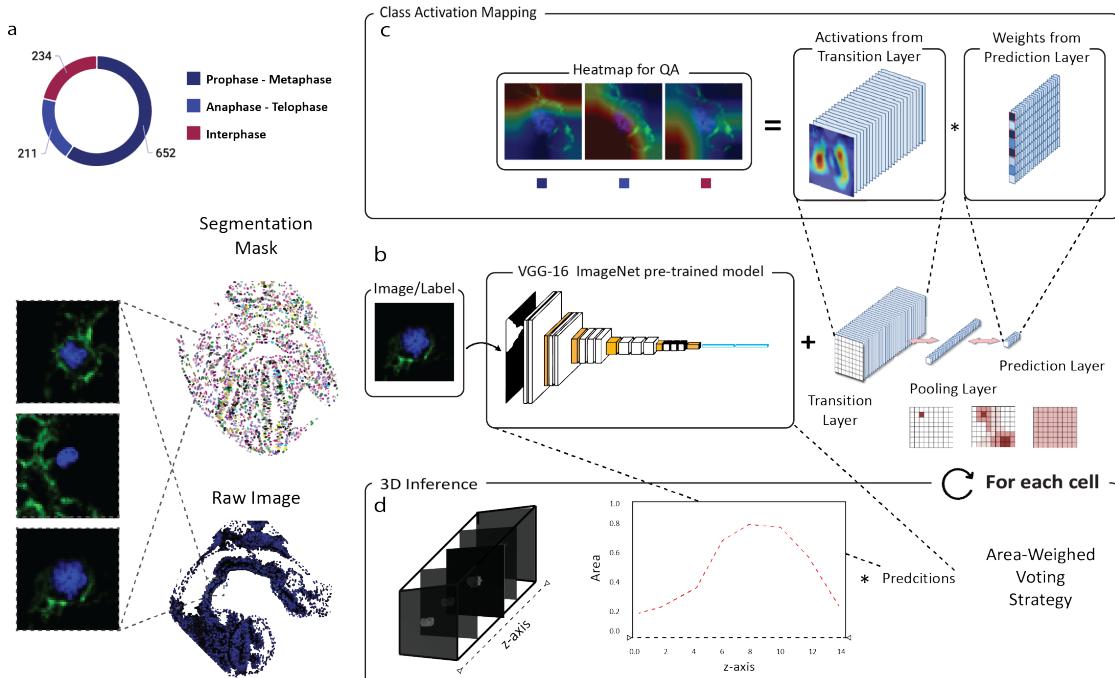


Figure 3.8: Illustration of cell division classification processes. **a|** Annotated class distribution for cell division stages: Prophase/Metaphase, Anaphase/Telophase and Interphase. Single-cell 2D training image examples. **b|** Model architecture: VGG-16 base model with the additional layers. **c|** Class Activation Mapping (CAM) showing heatmaps for classification discriminative feature localization and quality assessment. **d|** Illustration of 3D inference extension with weighted voting stategy for feature extraction.

3.4.2 Training strategy

Loss functions are mathematical formulas used to quantify the difference between the predicted outputs of the model and the actual target values. They guide the iterative optimization or "learning" process by providing feedback on how well the model is performing. For instance, during training, the loss function calculates the error of each prediction, and the optimizer uses this error to adjust the model's parameters to minimize that loss. Loss functions tested included Categorical Cross-Entropy Loss, Focal Loss and Weighted Cross-Entropy Loss (W-CEL).

Due to dataset unbalance, the model may have difficulties learning division events with lower number of instances. To address this imbalance, an extension of the Cross-Entropy Loss was used, which assigns dynamic weights to underrepresented classes, by adding a positive and negative balancing factors β_P, β_N for each class individually in multi-class classification problems. These factors are adapted into the Weighted Cross-Entropy Loss (W-CEL) as follows:

$$L_{w-CEL}(f(x), y) = \beta_P \sum_{y_c=1} -\ln(f(x_c)) + \beta_N \sum_{y_c=0} -\ln(1 - f(x_c)),$$

Where $\beta_P = \frac{|P|+|N|}{|P|}$ and $\beta_N = \frac{|P|+|N|}{|N|}$. Here, $|P|$ and $|N|$ represent the total number of positive and negative instances, respectively, for each class. This weighted formulation helps mitigate the adverse effects of class imbalance by amplifying the contribution of underrepresented classes.

Fine-tuning was applied to help the model learn task-specific features while leveraging pre-trained weights from the base model. These pre-trained weights, derived from training on the ImageNet dataset (Deng et. al., 2009) [39], provide a strong initialization by embedding general visual features such as edges, textures and shapes. This approach accelerates training and improves performance, specially for tasks where labeled data is limited.

During fine-tuning, the initial layers of the VGG16 base model were frozen, meaning the weights of these layers were kept unchanged to retain the general feature extraction characteristics. Freezing layers allows the network to focus on training the new added layers while preventing the disruption of the pre-trained knowledge.

The training process involved random downsampling to address class imbalance, with Prophase/ Metaphase being the most abundant class. Instead of relying on simple accuracy, the Area Under the ROC Curve (AUC) was used as a target metric. In this setup, TensorFlow computes one-vs-rest ROC curves for each class and then averages these curves to provide a balanced view of sensitivity and specificity, which is particularly suitable for imbalanced datasets. Early stopping and learning rate reduction techniques were implemented to prevent overfitting and ensure convergence.

3.4.3 Model calibration

Deep learning classifiers often produce output class probabilities that may be misaligned with the statistical true likelihood of each class. This mismatch, also known as misscalibration, can complicate decision-making processes, specially in biomedical applications, where high confidence estimates are essential (Guo et. al., 2017) [40]. To address this, a range of calibration techniques were used to attempt to refine the model's predicted probabilities while preserving its discriminative performance.

For instance, calibration aims to ensure that the model's predicted probabilities align with the true likelihood of each class, in other words, when the model estimates class A with 80% confidence, class A should be indeed correct about 80% of the time, but with these models, this may not be true.

Four calibration strategies were explored, Temperature scaling, Vector scaling, Matrix scaling (Kull et. al., 2019) [40] and Dirichlet calibration (Kull et. al., 2019) [41]:

Temperature scaling Temperature scaling divides the raw logits (final Softmax inputs) by a scalar trainable parameter $T > 0$. For multi-class classification problems involving $k > 2$ classes, the network logits z_i are vectors, where $y'_i = \text{argmax}_k z_i^{(k)}$, and p'_i is derived using the softmax function σ_{SM} :

$$\sigma_{SM}(z_i)^k = \frac{\exp(z_i^{(k)}/T)}{\sum_{j=0}^K \exp(z_i^{(j)}/T)}$$

Vector scaling Vector scaling extends temperatures scaling using a logistic-based calibration approach by learning a distinct linear and trainable scale (α_k) and bias (β_k) per class (k), leading to per-class adjustments of the logits before the final softmax.

$$z'_k = \alpha_k z_k + \beta_k$$

Matrix scaling Matrix scaling further generalizes vector scaling by introducing a learnable weight matrix W and a bias vector b applied to the logits. In difference to the per-class vector approach, this method models the "interaction" between classes, often yielding better results in multi-class problems at the cost of more parameters to fit.

$$z' = Wz + b$$

Dirichlet calibration Unlike the above methos, which work on logits, DIrichlet calibration directly operates over the probabiliy distributions. It applies learnable exponents $\alpha = (\alpha_1..,\alpha_K)$ to each class probability p_k from the original model's softmax outputs:

$$p'_k = \frac{p_k^{\alpha_k}}{\sum_{j=0}^K p_j^{\alpha_k}}$$

Each calibration approach was implemented as a new custom TensorFlow Layer and appended at the top (last layers) of the CNN after the initial training. For Temperature, Vector and Matrix scaling, the corresponding layer was inserted after the final CNN's logits layer but before the final softmax activation. For Dirichlet calibration, the custom layer is inserted at the very top, taking the softmax outputs from the CNN. Each probability vector (p_1, p_2, p_3) is transformed into $(p_1^{\alpha_1}, p_2^{\alpha_2}, p_3^{\alpha_3})$. Calibration refines the confidence estimates while leaving the core learned discriminative capacity intact. Hence, the layers learns their respective trainable parameters through back-propagation by freezing all the rest of the model weights, making only trainable the weights of the calibration procedure.

3.4.3.1 Evaluation Metrics

In order to evaluate the calibration quality, the following metrics were used: The Expected Calibration Error (ECE) summarizes the difference between the model’s predicted confidences and its true accuracy over multiple bins: $ECE = \sum_{j=1}^M \frac{|B_j|}{N} | acc(B_j) - conf(B_j) |$. Smaller ECE values indicates the model’s confidence is closer to its observed accuracy. Negative Log-Likelihood (NLL) measures how well the predicted probability distribution aligns with the true labels: $NLL = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^K y_{i,k} \log(p_{i,k})$. Smaller NLL values indicate that the model assigns higher probabilities to the correct classes. Brier Score is defined as the mean squared error (MSE) between the predicted probabilities and the actual outcomes. For a single class: $BrierScore = \frac{1}{N} \sum_{i=1}^N (p_i - y_i)^2$. For multi-class problems, the score is computed per class and averaged.

3.4.4 Model 3D inference extension

To extend the 2D classification architecture to account the 3D volumetric information of the cells, a weighted majority-voting strategy was implemented across the different 2D slices from the same nucleus (Figure 3.8d).

As weights, it is first computed the area of the cell for each slice using the segmentation masks, further normalized to sum one, ensuring that slices with larger nuclear regions have greater influence on the final prediction. This approach was implemented to reduce noise from the slices near the boundaries of the nuclei, which may exhibit partial and more noisy signals.

Then, each slice is passed to the network, yielding the probability distribution p_z . Finally, given each Z slices, each with its associated weight w_z , the final 3D classification probability and class prediction are defined as

$$p_{3D} = \sum_{z=1}^Z w_z p_z \quad \text{and} \quad y_{3D} = \text{argmax}(p_{3D})$$

Overall, this weighted voting strategy aims to help including all the 3D cell information while making border slices do not disproportionately affect the final 3D classification.

3.5 Feature Mapping

In order to understand and compare morphological patterns across and within the embryonic samples and developmental stages, the following pipeline is proposed and highlighted in Figure 3.9, where the different extracted cellular characteristics are mapped onto the 3D tissular surfaces of the embryos as feature heatmaps. In particular, the heatmaps are mapped into what will be called the atlas shape of each group: a reference model serving as average shape of all embryos within each group of the 10 developmental stages, as illustrated in Figure 3.9a.

The creation of these group-level atlas is critical for understanding and comparing how the features vary across the embryos and developmental stages. By wrapping each individual embryo to a shared reference shape—an atlas—we can unify disparate geometries into a single coordinate space, ensuring that any local differences do not compromise underlying biological patterns. This consistency allows us to average, compare, and analyze feature distributions across large collections of embryos in a coherent manner, ideally revealing location-specific trends that might otherwise remain hidden due to geometrical variability.

The proposed pipeline will be split into three main mapping stages; Cell-Tissue mapping (Figure 3.9b), where each cell will be mapped onto its corresponding tissue surface. Tissue surface map computation (Figure 3.9c), where each atlas will be registered into the tissue surfaces of each embryo within that group. Finally, previous mapping will serve to create the final feature mapping for creating the 3D feature-specific heatmaps of each atlas (Figure 3.9c ii).

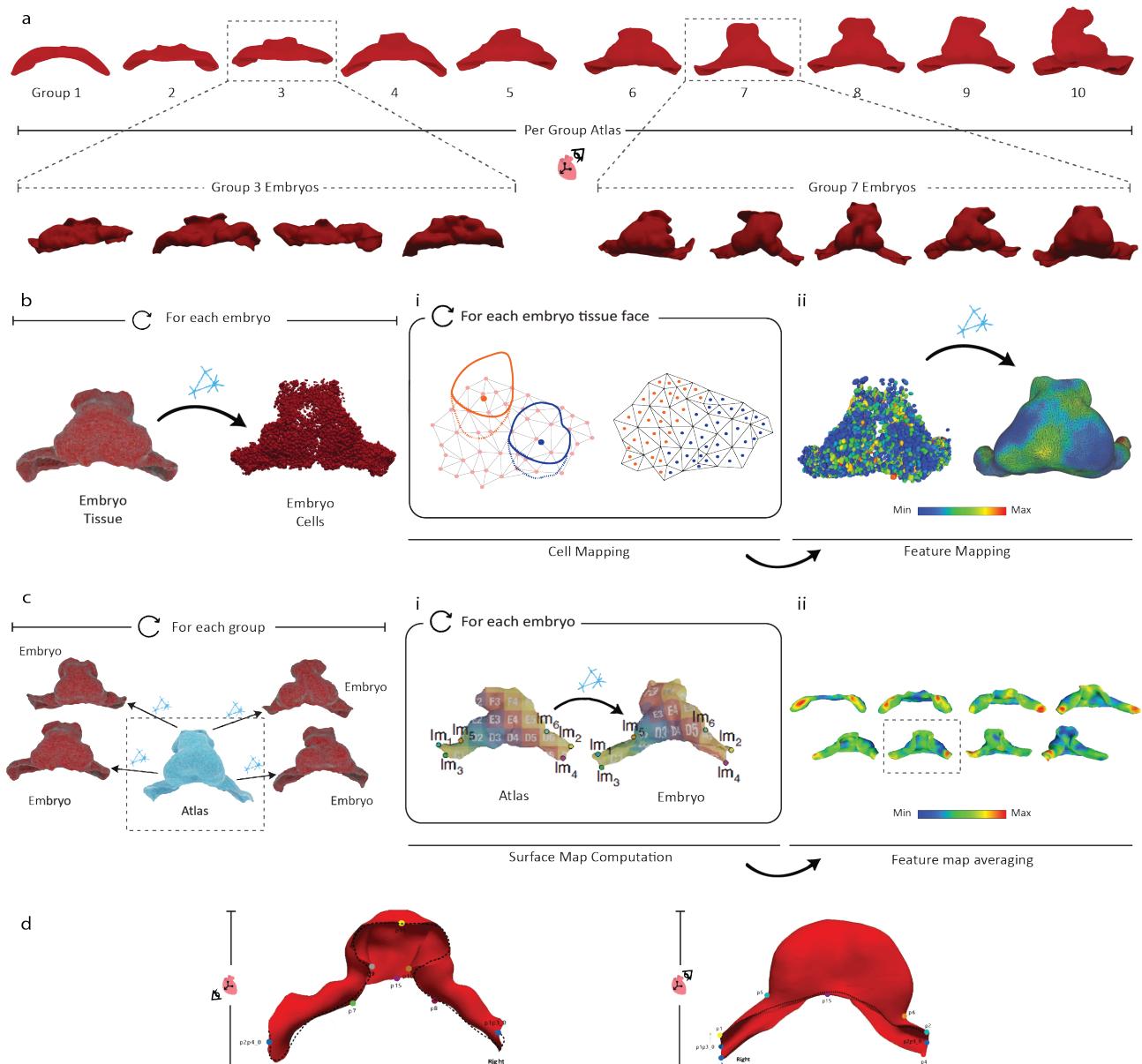


Figure 3.9: Overview of the feature mapping pipeline across the different embryonic samples for yielding 3D heatmaps highlighting feature gradients within and across developmental stages. **a|** The 10 group-level atlas mesh representing the average tissue shape of each group embryos. Some embryo examples are illustrated for groups tree and seven. **b|** Tissue-Cell level analysis. **i|** Each tissue face is associated with its closest cell mesh. **ii|** The previous mapping used for mapping the cellular features to the tissue mesh and colored accordingly. **c|** Group level analysis. **i|** Surface Map Computation of the atlas into each embryo of the group for mapping each embryo 3D heatmap in the atlas. **ii|** 3D Heatmaps over the groups atlas averaged within groups. **d|** Ventral and Dorsal representation example of the landmarks used for the myocardium.

3.5.1 Cell-Tissue Mesh Mapping

This pipeline’s section uses the 3D surface meshes representing the tissues of each embryo, along with the reconstructed 3D cells and nuclei meshes. The objective is to project cells onto the tissue surfaces so that cell-level features can be displayed as heatmaps at specific locations (faces) of the embryonic tissues (Figure 3.9b).

This procedure is applied for each embryo on the dataset, where each face in the cell mesh is labeled by a unique identifier corresponding to the cell it belongs (segmentation mask id), so that it can be correctly assigned its corresponding cell-level feature.

The algorithm computes a centroid for each cell by averaging the 3D coordinates of all face centroid sharing the same cell identifier. Meanwhile, the centroid of each tissue triangular face is calculated and stored in a face-based graph, where each node is the centroid of the face and edges are created within each direct neighbor face centroid alongside its 3D euclidean distance. This way, spatial geodesic queries can be optimized and accessed by using Dijkstra’s graph search algorithm, avoiding the creation an adjacency matrix for each query.

To establish the mapping, each tissue face is assigned to the nearest cell centroid. Consequently, each tissue face is associated with a single cell identifier, thereby allowing every point on the tissue surface to inherit that cell’s features. Additionally, for smoothing the color gradients of the features, each face is associated with neighbor faces within a specified geodesic distance, in this case $50 \mu m$, representing approximately a radius of two cells. Neighbors are identified through Dijkstra’s algorithm on the tissue adjacency graph, simulating queries with the geodesic distance for avoiding 3D inconsistencies.

Finally, the mapped features can be visualized per-embryo by coloring each face of the tissue mesh according to its associated cell feature. Overall, this step creates unified 3D representations where cell-level attributes are placed on the tissue surface, laying the groundwork for subsequent atlas-level comparisons.

3.5.2 Tissue Surface Map Computation

Each group of embryonic samples is associated with a group-level atlas representing the average morphology of the tissues of its corresponding group. The goal is to place the atlas mesh into the coordinate space of the individual embryo’s shape so that features can be directly averaged per groups and compared across developmental stages (Figure 3.9c). To achieve this alignment, also referred as registration, the method uses a specialized and optimized non-rigid registration tool (Schmidt et. al., 2022) [42], performing a surface map between a pair of 0-genus closed surfaces, which is used to warp the source (atlas) mesh to the target (embryo) mesh, guided by landmark correspondences.

Non-rigid registration methods are designed to align shapes that may differ not only by rotation or scaling but also by local morphological variations, common in biological structures that deform in complex ways. In contrast to rigid techniques, which allow only uniform transformations like translation, rotation or uniform scaling, non-rigid methods introduce localized flexibility.

By iteratively deforming the source mesh and constraining the process to match anatomical landmarks, the registration produces a smooth, anatomically consistent mapping from the atlas to the embryo’s surface. This approach preserves important local structures while adjusting the non-uniform differences that arise among embryos of the same developmental stage (Figure 3.9ci).

The software uses landmarks—anatomically meaningful points—on both source and target meshes as anchors to guide the registration (Figure 3.9). For the myocardium, it was used a total of 12 manually annotated landmarks. Each landmark is first transformed from a 3D coordinate to a vertex index in the mesh. During non-rigid registration, the algorithm tries to minimize the distance between corresponding landmarks while optimizing other constraints, like surface smoothness or local area preservation.

Before alignments, it is mandatory to check whether both source and target mesh are 0-genus closed surfaces and shares the same number of faces and vertex, as a software specific restriction. If the meshes does not share the same number of faces, the embryo mesh is subdivided or decimated until the number of faces coincide. Any changes to the mesh topology are followed by landmark updates to ensure vertex indices remain accurate.

Finally, the tool generates a file representing the map from A to B as a tuple (ϕ, h, ι) , where $\phi : V_A \rightarrow T_B \times \mathbb{R}^2$ (vertex-to-surface map), where each vertex of A is assigned to a point in a triangle of B , expressed in a barycentric coordinate system. Repeating this process for each embryo in a group yields a consistent non-rigid alignment of the atlas, enabling direct comparison of per-vertex or per-face across specimens within the same 3D reference framework.

3.5.3 Atlas Feature Mapping

Once the group-level atlas has been aligned to the individual embryonic shapes, each embryo’s per-face features can be projected into the atlas faces. This section of the pipeline then proceeds to lift these features onto the shared atlas by leveraging the vertex-to-surface map generated during the non-rigid registration.

The key step in this process is the barycentric projection, where each vertex of the atlas is assigned a barycentric coordinate (α, β) of two dimensions in a face of the embryo’s tissue mesh. Barycentric coordinates—center of mass—express a point within a triangle as a weighted combination of the triangle’s three vertices, with the property that $\alpha + \beta + \gamma = 1$. Because the mapping file stores only two coordinates (α and β), it is necessary to compute the third coordinate γ by applying the aforementioned property.

Once the barycentric weights have been established for each vertex, it becomes possible to interpolate any per-face feature from the embryo’s mesh onto the atlas mesh (Figure 3.9cii). Any atlas vertices that cannot be mapped directly are handled by examining their nearest direct neighbours and averaging their values. This neighbour-based averaging addresses additional vertices introduced as part of the surface map computation software’s smoothing for reliable-region mapping, a pre-processing step that may extends the original mesh vertices beyond the original geometry. Because these vertices have no exact counterpart on the original meshes, values in such vertices must be inferred by averaging adjacent information, thus preventing small gaps in the final heatmaps.

Once all features are consolidated in the atlas, group-level averages are computed, yielding a single consensus heatmap for the entire developmental group. The resulting atlas provides a framework for visually comparing how each measured feature varies across the embryonic tissue surface.

Additionally, a new set of heatmaps is performed by normalizing the feature values across all groups for time-variation comparison. By normalizing feature distributions on a global scale, fluctuations due to different measurement ranges can be reduced, enabling direct group comparisons. Consequently, a single consensus heatmap is generated for each stage, facilitating a view of how each feature varies within and across developmental stages.

Results

Due to the high computational resources required by several stages of the proposed framework, analyses have focused primarily on the myocardium as the main region of interest. Although the pipeline can address other embryonic mesodermal tissues introduced in Section 1.3.2, these have been excluded from the present study to maintain computational viability limited to time availability. Consequently, the results reported in this section emphasize findings related to the myocardium.

4.1 Cell Segmentation and Reconstruction

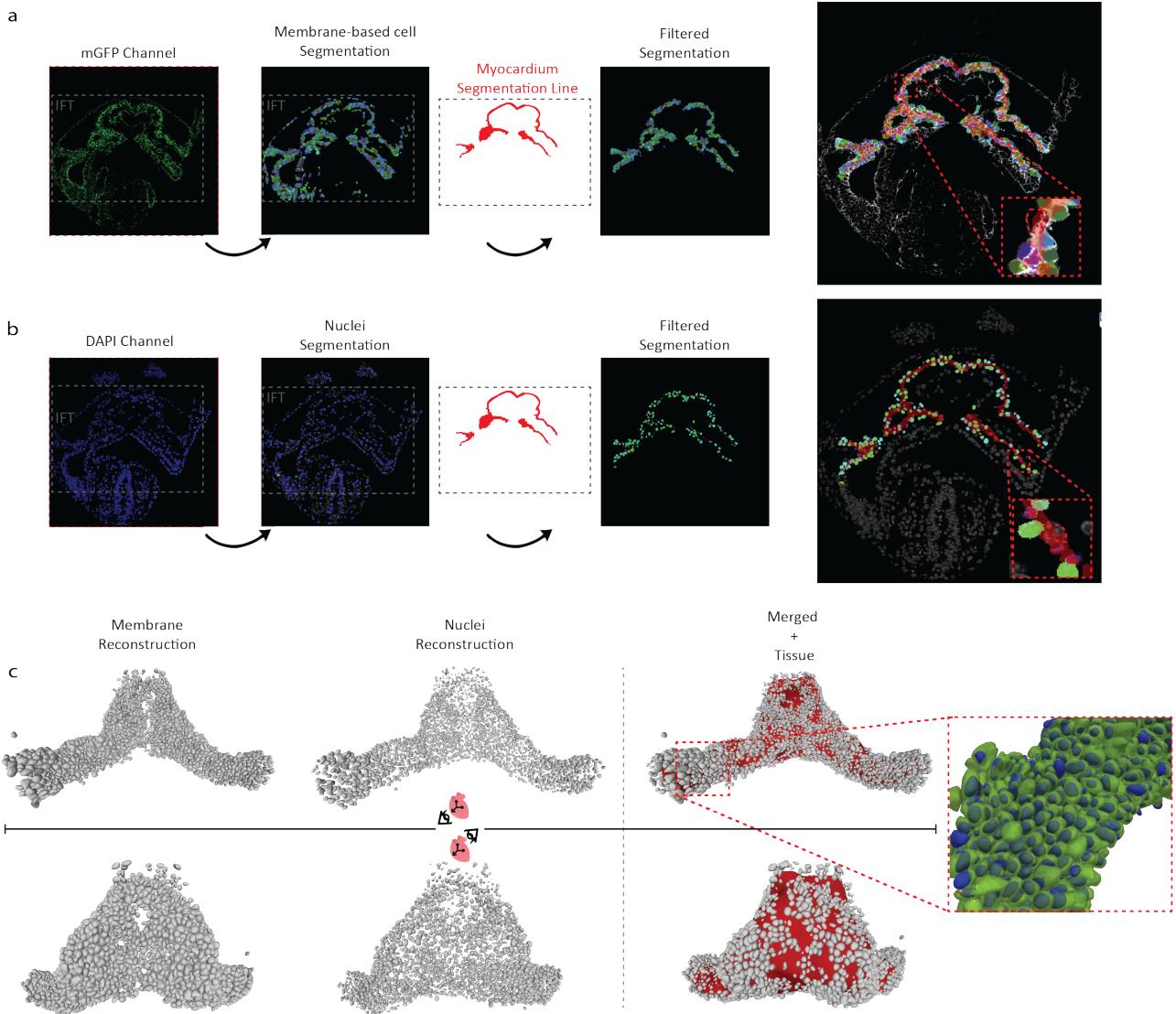


Figure 4.1: Segmentation example over a z-slice and mesh reconstruction. **a|** Membrane-based cell segmentation over the mGFP channel in the heart tube region. **b|** Nuclei segmentation over the DAPI channel. The red boxes over the composite image (Right) shows examples of miss-segmentation. **c|** Dorsal and Ventral views of the reconstructed cells and nuclei with the tissue segmentation on red. Red box (Right) shows both cells (Green) and nuclei (Blue) merged.

4.1.1 Membrane-based cell instance segmentation

The membrane-based cell instance segmentation was executed over a 3D crop covering the entire heart tube, excluding the regions out of the interest region apart for saving computational resources (Figure 4.1a). Segmentations were evaluated by visual inspection due to the absence of ground truth (GT) segmentation labels. It can be observed that image quality over the mGFP channel decays considerably when increasing depth, affecting precision in deeper tissue layers. Despite these limitations, the combined pre- and post-processing strategies for membrane segmentation yielded reliable distinction of individual cell instances.

Occasional under-segmentation arose, when two or more cells are segmented as a big one, usually removed by volume filtering. This error results from the semantic segmentation step, due to the lack of GFP-labelled membranes. These labelling lack lead to incomplete segmentations of the membrane edges. As a result, the GASP algorithm misinterprets the data and incorrectly links multiple membrane edges together. Additionally, some cell regions were not segmented at all, as shown in the red dotted box on Figure 4.1a, leading to a slight decrease in number accurately segmented cells. Nevertheless, the segmentation results proved enough accurate for subsequent analysis.

4.1.2 Nuclei instance segmentation

The nuclei instance segmentation was conducted on the HPC cluster described on Section 2.4 due to the high computational resources needed to run the model over the 3D images (Figure 4.1b). The presence of densely packed nuclei contributed to complexity in establishing clear, cell-by-cell boundaries, leading to under-segmented instances. Additionally, the intensity over the depth of the volumes decreases considerably on the DAPI channel.

Visual inspections of preliminary segmentations suggested suboptimal results, indicating that a different approach could better exploit DAPI information to achieve a more robust segmentation and subsequent analysis. This was confirmed through a qualitative evaluation: A small 3D crop (≈ 190 cells) was manually annotated. It was then used to conduct a grid search of model hyper-parameters along pre- and post-processing steps, enabling a quantitative assessment of segmentation accuracy. The selected metrics, including Dice coefficients (Dice et. al., 1945)[43], volume-weighted Jaccard indices (Jaccard et. al., 1901)[44] and a graph-based evaluation approach proposed by Kar et. al [45], where quality estimations go further aforementioned indices. This graph-based approach allows to better assess whether segmentation errors are due to under-segmentation, where two or more real nuclei are segmented as one, over-segmentation, where a single nucleus is labeled as different instances or even if there are more complex relationships as multiple-to-multiple associations, in which a group of real nuclei is labeled as a group of predicted segmentations, making it more challenging to assess one-to-one or one-to-many mismatches. A pseudocode of the graph-based implementation is shown in Appendix C.

After testing over ≈ 1020 different permutations of pre- and post-processing strategies and model parameters: flow threshold, cellprob threshold and prob threshold for 2D and 3D models, the optimal combination was found, yielding good quality segmentations. Similar as for the membrane-based cell segmentation, occasional nuclei are missing, under- or over-segmented. Having the small crop as reference, there are $\approx 1.96\%$ of missing nuclei, $\approx 3.1\%$ over-segmented and $\approx 13.8\%$ under-segmented.

4.1.3 Single-cell 3D mesh reconstruction

The three-dimensional reconstruction of the single-cell/nuclei volumes (Figure 4.1c) enabled a closer inspection of the cell shapes. The surface meshes further validate that the segmentation results seem to align with realistic cellular geometry. In particular, visually inspecting surfaces of each cell and properly enclosed nuclei provided greater confidence in the method's capacity to distinguish individual cells over the region of interest, where no cell-mesh discontinuities are found, suggesting that the segmentations were accurate enough to capture cell boundaries. However, it is notable that the ventral section of the heart tube is the region where the pipeline struggles most to capture cells (Figure 4.1c). Additionally, there are visible regions where cells lack segmented nuclei and vice versa.

4.1.4 Exploratory data analysis of cell counts: myocardium

An additional count-based simple analysis was conducted by aggregating the total number of segmented and filtered cells on the myocardium (membrane-based) and nuclei for each experimental group and then computing group-specific averages. Figure 4.2a shows the mean counts of membranes and nuclei per group along with standard deviations, showing the slight under-estimation of the membrane-based cell segmentation in comparison with nuclei segmentation. Additionally, we found a notable difference in counts for some specimens in groups eight and nine, where we found embryos with very low number of cell counts despite its great volumetric surface, which may suggest suboptimal segmentations for some of the embryos in these groups acting as outliers. For this reason, we considered these specimens outliers and excluded them from further analysis.

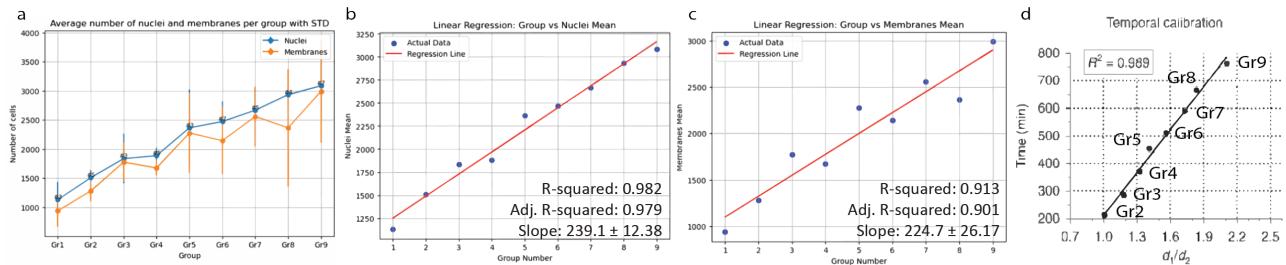


Figure 4.2: Linear analysis. **a**|Average cell and nuclei segmented and filtered counts per group. **b**| Linear Regression (LR) nuclei mean count per group. **c**| LR cells mean count per group. **d**| Temporal calibration of the progression of d_1/d_2 values derived from Esteban et. al. [3].

Figures 4.2c-d illustrate Linear Regressions (LR) between groups and mean cell/nuclei numbers. The slopes for nuclei and membranes ($\approx 239.1 \pm 12.4$ and $\approx 224.7 \pm 26.2$, respectively) highlight a steady increase in detected cell counts as developmental stages progress. From these growth trends and from the temporal calibration of the same embryos proposed by Isaac et. al. [3] in Figure 4.2d, it can be inferred that the myocardium approximately doubles its cell number over a period of about 6-7.5 hours, aligning with previously reported division rates for cardiac mesodermal cells (Sendra et al., 2023 [46]). High coefficients of determination ($R^2 = 0.982$ and $Adj.R^2 = 0.979$ for nuclei; $R^2 = 0.913$ and $Adj.R^2 = 0.901$ for membranes) suggest the regression model fits these data well. Although biological variability and potential imaging artifacts introduce scatter, the strong correlation and observed growth in cell counts reinforce that the segmentation pipeline scales reliably enough with the embryonic developmental stage.

4.2 Feature-based application: Cell division

Figure 4.3 summarizes the cell division classification results under different experimental configurations and calibration approaches. Figure 4.3a presents a grid search over different pre-trained base model architectures and the different configurations tested. These initial search revealed VGG-16 and DenseNet121 as the best-performing architectures with the weighted-cross entropy loss function, CAM top and LR of 0.001 (Configuration 5), finally selected VGG-16 as base model due to its high simplicity compared to DenseNet121.

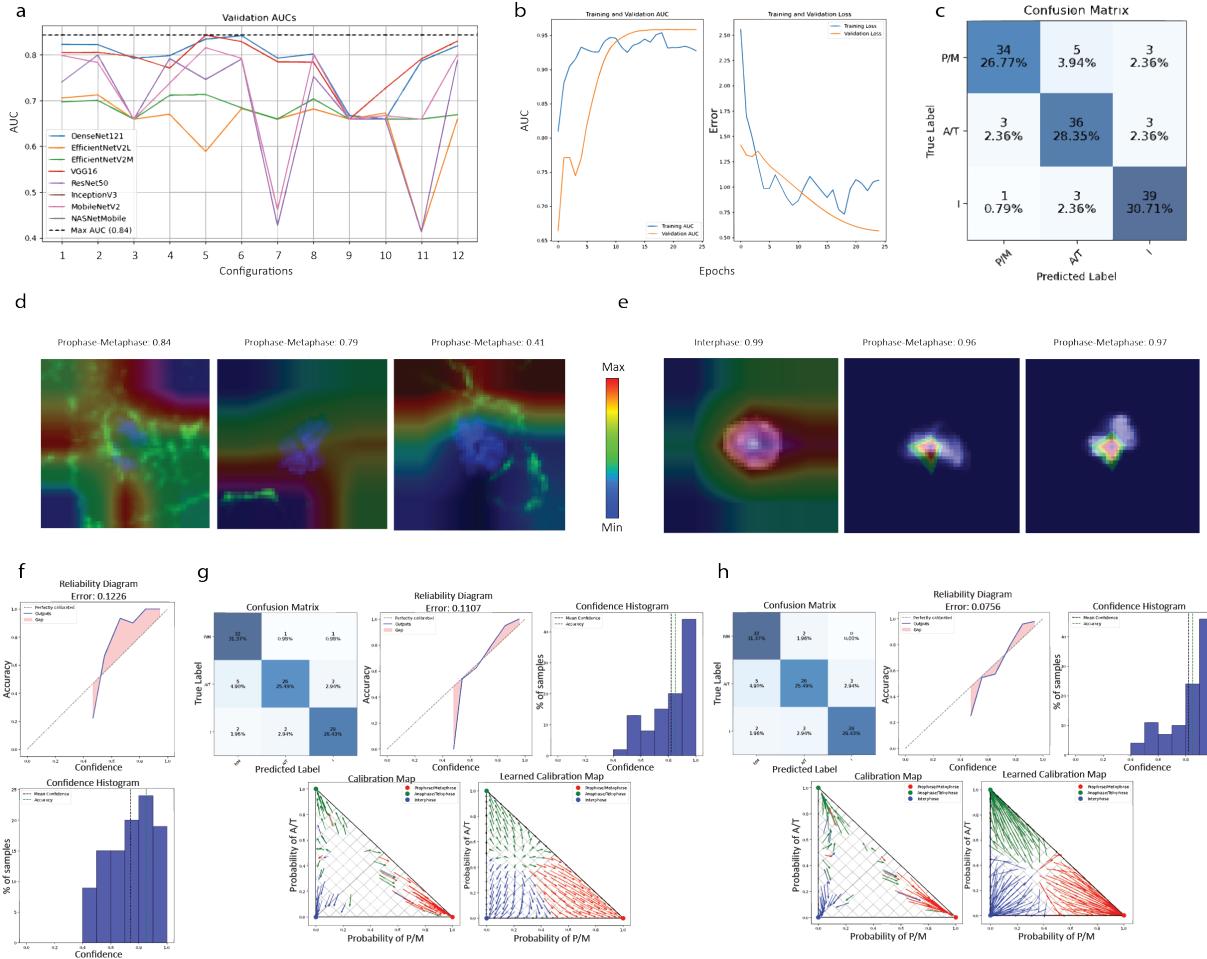


Figure 4.3: Cell division model results overview for the three classes: Prophase/Metaphase (P/M) Anaphase/Telophase (A/T) and Interphase (I). **a**| Grid search over different base pre-trained models and configurations. **b**| Training results over epochs of final model displaying AUC metric and weighted cross entropy loss. **c**| Confusion Matrix of final model. **d**| CAM heatmaps for nuclei + membrane channels as model training inputs. **e**| CAM heatmaps for just nuclei channel as model training inputs. **f**| Reliability Diagram (Top) and Confidence Histogram (Bottom) final model. **g**| Dirichlet calibration results **h**| Matrix scaling calibration results.

Figure 4.3b shows the training and validation metrics of the final chosen model over the training epochs. A steady increase in the validation AUC, along with corresponding decrease in validation loss, indicates network convergence to a stable steady state without significant overfitting. The final confusion matrix (Figure 4.3c) demonstrates a balanced performance across the three cell-division classes, with most test set classification falling along the main diagonal, with a final test AUC of ≈ 0.94 . Although some missclassifications arise, notably between Prophase/Metaphase (P/M) and Anaphase/Telophase (A/T), the 2D model overall maintains robust discrimination among classes.

Figure 4.3d-e shows class activation maps (CAMs) for different validation example inputs, with and without the membrane channel respectively, offering insights into which regions of the image contributed most strongly to the model’s classification decision. When trained with both nuclei and membrane channels (Figure 4.3d), the model tends to focus on regions outside the nuclei area, effectively treating membrane signals as additional and important features but capturing noise from the extracellular space. By contrast, when only using nuclei channel for training (Figure 4.3e), heatmaps tighten around the nuclear region, showing more focus directly on the regions of interest. This shift in attention is followed by a notable increase in prediction confidences, suggesting that omitting membrane signals can lead to increased model’s performance.

Calibration results from Matrix scaling and Dirichlet calibration are displayed in Figures 4.3f-h. Figure 4.3f shows a reliability diagram (RD) and confidence histogram (CH) of the uncalibrated model. The RD reveals that, although remains fairly well-aligned to the diagonal of perfect calibration, probabilities tend to be slightly overconfident. Temperature and vector scaling (not shown) helped but did not eliminate this effect. Dirichlet calibration (4.3g) and Matrix scaling (4.3h) notably helped mitigating the effect of misscalibration and increasing prediction confidences illustrated in both CHs, with matrix scaling achieving the lowest calibration error, therefore being selected as calibration strategy.

Additionally, learned calibration maps (LCM) show a grid of possible uncalibrated probabilities (start of the arrows) and their corresponding calibrated probabilities (head of the arrows). For Dirichlet calibration, LCM seems to unexpectedly shift the predictions towards Prophase/Metaphase and Interphase.

4.3 Feature mapping

Figure 4.4 shows the pipeline for mapping per-cell features into the group-level atlas for the columnarity feature over the first developmental group as an example, illustrating both intermediate and final outcomes. Figure 4.4a shows the Cell-Tissue mesh mapping, where each segmented cell is projected into the surrounding tissue surface, creating embryo-specific cellular feature heatmaps.

Figure 4.4b shows the Embryo-Atlas Surface Map Computation (SMC), in which the group’s atlas is aligned with its corresponding embryo surfaces via non-rigid registration. Purple lines indicate a path of concatenated vertices on the atlas and their corresponding faces on the embryo’s surface, reflecting how local deformations aligns between the two surfaces. Through this registration step, each point in the atlas can be associated with the features matched on the individual embryo surfaces. However, as the SMC software was not developed with this purpose, it can fail to produce complete transformations for certain embryos. Consequently, those embryos lacked fully reliable maps and were excluded from the final group feature analysis.

The final feature map is shown in Figure 4.4c, where per-vertex data from the different embryos within the group are then averaged, highlighting local spots and feature gradients across the atlas surface. In this example, the columnarity feature is represented, with Figure 4.4d extending the feature heatmaps to all study groups, enabling direct visual comparison of the spatial patterns. Although columnarity is shown in the figure as a representative example, the pipeline fits any cellular feature extracted from the segmentation or the mesh outputs.

This framework opens the door for interactive software that allows researchers to explore multiple cellular features in a single atlas coordinate system, potentially comparing morphological

features across large cohorts of embryos. By integrating user-friendly visualization tools along with robust calibration steps, the pipeline can serve as an exceptional platform for deepen insights into embryonic heart morphogenesis.

The complete code repository, including all scripts used in this project, is available in the following [GitHub repository](#).

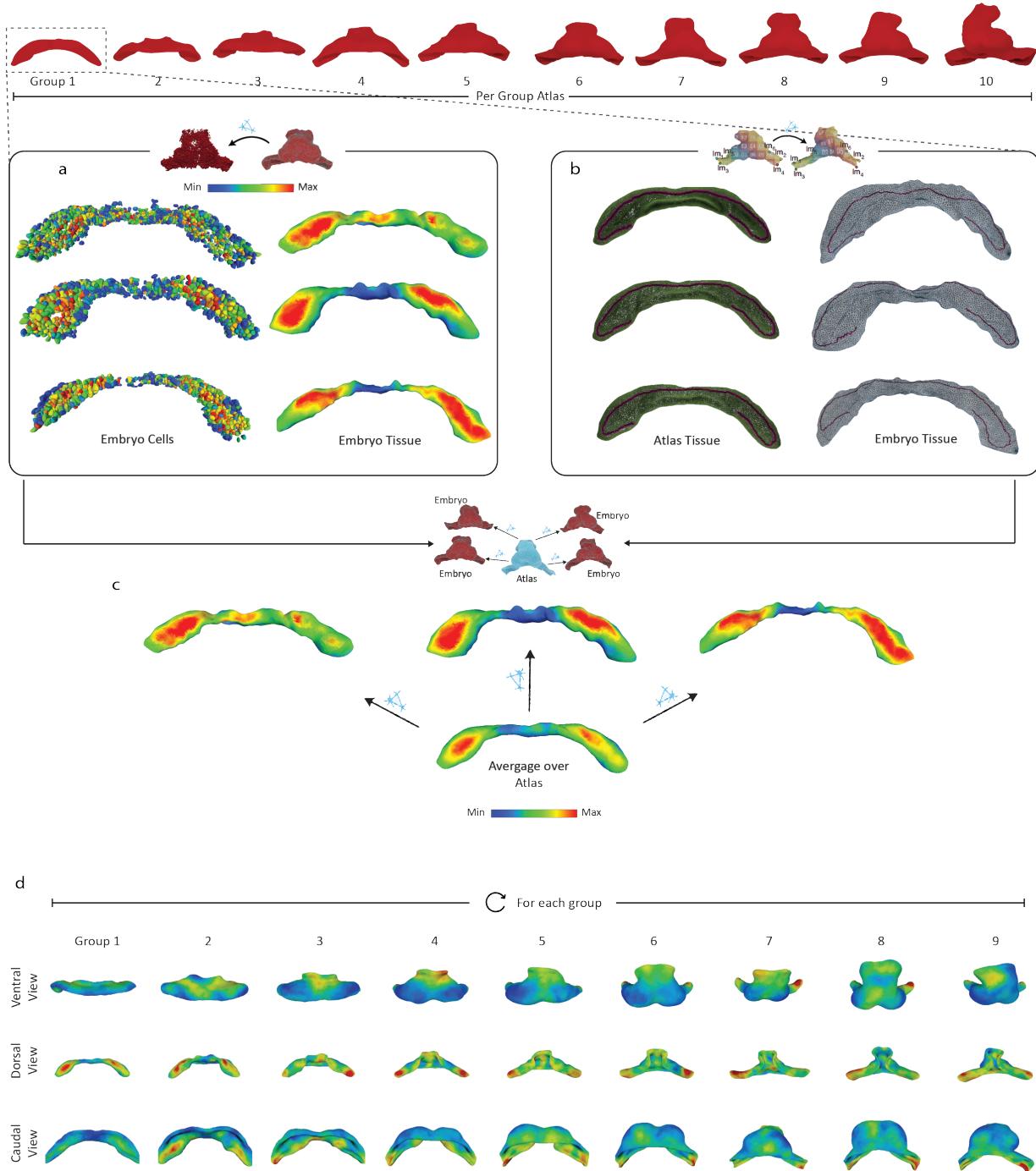


Figure 4.4: Pipeline example for group one columnarity. **a**| Cell-Tissue mesh mapping, displaying the columnarity feature for each cell, and the corresponding tissue surface colored after the mapping. **b**| Embryo-Atlas Surface Map Computation. Purple lines represent the atlas vertices mapped into their corresponding embryo faces. **c**| Final feature map averaged over the group atlas. **d**| Columnarity feature heatmaps over all study groups. First row shows the ventral view, second row show dorsal view and third row shows caudal view of the myocardium.

Discussion

The current study developed and evaluated a multi-stage computational framework aiming to understand embryonic development through image-based cellular analysis. This pipeline was centered around three key stages: Instance segmentation and 3D reconstruction of cellular membranes and nuclei along with feature extraction, cell division classification with calibration and feature mapping onto 3D group-level atlas (developmental-stage-level) for subsequent comparative analysis. Results from each step underscored the complexity of the biological samples alongside the inherit challenges of large-scale image processing and segmentation. The developed computational framework helps tackle these challenges by enhancing the efficiency and accuracy of the process.

5.1 Single-cell segmentation and 3D reconstruction

Both membrane-based cell and nuclei segmentations showed enough reliable performance under visual conditions, although occasional miss-, under- and over-segmentation was detected, specially over embryos in more advanced developmental stages. These errors were further emphasized in deeper tissue layers, where signal quality degrades due to inherit microscopy limitations. The reconstructions of single-cell 3D meshes served as an additional confidence check, confirming that the implemented algorithms reasonably captured individual cells. In particular, the 3D reconstructions illustrated boundary defects that might have been difficult to detect uniquely trhough 2D slices, reinforcing the importance of combining 3D implementations and different channels. Despite this, the developed pipeline successfully segments and reconstructs cells and nuclei, allowing the precise quantification of single-cell features in subsequent steps, which will contribute to achieve the goal of a detailed understanding of embryonic development.

5.2 Cell divison characterization

The cell division classifier achieved robust accuracy over the provided dataset, even though integrating membrane signals with nuclear data failed to enhance classification and in some cases slightly decreased performance and confidence. This suggests that the additional morphological information from membranes can obscure key nuclear features and introduce noise. Nevertheless, overall miss-classifications between closely related phases remained manageable, while the inclusion of confidence calibration helped to mitigate potential misinterpretations by refining the probability estimates. It is worthy mentioning that this phase classification also proved challenging for human experts, where initial manual annotations were decided to merge the four different phases into the two major categories—Prophase/Metaphase and Anaphase/Telophase—plus Interphase, to solve disagreements in labeling some instances. After calibration, matrix scaling proved effective at aligning the model’s output probabilities with true likelihoods, highlighting the value of multi-parametric calibration in multi-class problems. Future work may focus on further optimizing the classification process, potentially by tailoring the training set or incorporating additional data to improve the classifier’s ability to distinguish closely-related cell division stages, and maybe increasing the granularity of the classification into more specific cell-division stages. Nevertheless, the current 2D classifier allows for accurate detection of major cell division stages and events, enabling the study of how cellular behaviors drive tissue development and contribute to organ formation during heart morphogenesis.

Although achieving an accurate classification for the 2D model, visual inspections of the 3D classification extension strategy over the embryos cells suggested a significant number of false negatives. 3D cells predicted as a proliferation stage (M/P or A/T) were indeed dividing cells, while there were some dividing cells missclassified as Interphase (no division). Therefore, it is necessary to further revise the 3D extension in order to decrease that amount of false negatives and improve the 3D generalization.

5.3 Feature mapping and atlas-level comparisons

By integrating segmentation and 3D reconstruction outputs with non-rigid registrations, the framework provided a shared coordinate system in which per-cell features were aggregated across entire developmental groups and potentially visualized for different tissues. This step effectively addressed the cross-embryo variability in 3D surfaces, enabling direct comparison of location-specific trends. The generation of feature heatmaps on group-level atlas allows highlighting patterns and developmental shifts across different stages, to be further analyze in future work. However, the feature mapping software was not initially designed for vertex-to-face mapping of this nature, forcing the exclusion of several embryos whose meshes could not be aligned properly. Although all groups had at least two embryos for averaging, groups eight and nine finally contained just one embryo, making their resulting heatmaps less robust. This constraint will be reexamined in future work to expand the sampling size for each group and improve the overall reliability of the analysis. Although only the myocardium was analyzed due to computational constraints alongside time limitations, the methodology remains applicable to other tissues.

5.4 Limitations and future work

A significant limitation of this work arises from the high computational cost, specially when processing large 3D image stacks and dataset size. Restricting analysis to the myocardium was a choice for maintaining feasibility, but the integration other tissues may underscore additional patterns that remain unexplored. Furthermore, ground-truth labels were not generated for full volumes, so smaller manually annotated crops and visual inspections served as main segmentation quality estimations.

Another notable challenge involves feature map computation, where the non-rigid mapping software was originally designed for different alignments objectives. This limitation forced the exclusion of multiple embryos due to unreliable mesh alignments. In particular, only one embryo remained in groups eight and nine, restricting statistical power of atlas-level comparisons in these groups. Readapting the approach would increase sampling sizes and strengthen the robustness of the feature mapping.

Regarding the computation of single-cell features, the perpendicularity feature showed noise at the borders of the meshes due to the curved nature of surface meshes representing the different tissues. Future work will aim to use zero-thickness meshes to minimize the effect of boundary regions, although the computational cost for generating these meshes has so far limited their generation for this work. Intensity-based features also exhibit technical rather than biological gradients, caused by signal decay along the volumes depth. 2D stack-by-stack normalization did not fully solve this artifacts, requiring further refinements involving advanced intensity calibration. Moreover, the addition of new features can further enhance cellular analysis for more advanced understanding of embryonic

development.

Another promising approach involves combining the static data from this work with dynamic time-lapse live imaging data that captures embryonic growth in real time. By integrating this data, each cell could be tracked across developmental stages, enabling measurements of cell progression and morphological transformations. Such a spatiotemporal approach would deepen biological understanding by linking measured features to actual developmental trajectories. Future research will focus on these integrations, aiming to create a comprehensive framework that unifies spatial and temporal analysis of embryonic development.

In summary, the pipeline demonstrated a robust framework for large-scale embryonic analysis, but future improvements are vital to fully capture spatiotemporal complexity of embryonic development by decreasing technical noise and improving statistical robustness.

Conclusions

The development and evaluation of this multi-stage framework represents a step toward understanding complex cellular level embryonic processes, such as organogenesis, through 3D image-driven analysis. By integrating segmentation algorithms, deep-learning methodologies and an atlas-based feature mapping approach, the framework provides a comprehensive way of examining how cells organize and change across developmental stages.

In particular, membrane-based cell and nuclei segmentations demonstrated the feasibility of tracking cells in 3D, even under challenging imaging conditions. These segmentations and further reconstructions enable the extraction of features, that can be easily extended, which can be further analyzed in 3D. Some features being more complex, allowing the quantification of the relationships between cells and the different embryonic tissues. The cell division classifier performed well in discriminating key mitotic stages. Mapping extracted features onto unified 3D atlases further enables the visualization and quantification of spatial patterns across multiple specimens overcoming local morphological variability.

Nevertheless, multiple challenges remain. High computational demands and software adaptations limited sampling size of tissues analyzed. In addition, intensity-based gradients reflects technical artifacts more than true biological variation. Addressing these issues will expand the pipeline's usefulness for large and complex datasets. These advances promise to deepen our understanding of embryonic development and further validate computational strategies as visual tools in developmental biology.

Bibliography

- [1] A. Guimier. “MMP21 is mutated in human heterotaxy and is required for normal left-right asymmetry in vertebrates”. In: *Nature Genetics* 47.11 (2015), pp. 1260–1263. DOI: [10.1038/ng.3376](https://doi.org/10.1038/ng.3376).
- [2] R. R. Markwald and M. V. de la Cruz. *Living Morphogenesis of the Heart*. Birkhäuser Boston, 1998. DOI: [10.1007/978-1-4612-1788-6](https://doi.org/10.1007/978-1-4612-1788-6).
- [3] I. Esteban, P. Schmidt, and A. Desgrange. “Pseudodynamic analysis of heart tube formation in the mouse reveals strong regional variability and early left-right asymmetry”. In: *Nature Cardiovascular Research* 1 (2022), pp. 504–517. DOI: [10.1038/s44161-022-00065-1](https://doi.org/10.1038/s44161-022-00065-1).
- [4] Mark Buckingham, Stéphane Meilhac, and Sophie Zaffran. “Building the mammalian heart from two sources of myocardial cells”. In: *Nature Reviews Genetics* 6.9 (2005), pp. 826–835. DOI: [10.1038/nrg1710](https://doi.org/10.1038/nrg1710).
- [5] Audrey Guimier, Gabriel Gabriel, and Tania Attie-Bitach. “Genetic basis of congenital heart diseases”. In: *Experimental Cell Research* 335.1 (2015), pp. 114–123. DOI: [10.1101/cshperspect.a036749](https://doi.org/10.1101/cshperspect.a036749).
- [6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241. DOI: [10.1007/978-3-319-24574-4_28](https://doi.org/10.1007/978-3-319-24574-4_28).
- [7] Kaiming He et al. “Mask R-CNN”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2961–2969. DOI: [10.1109/ICCV.2017.322](https://doi.org/10.1109/ICCV.2017.322).
- [8] Adrian Wolny and Lorenzo Cerrone. “Accurate and versatile 3D segmentation of plant tissues at cellular resolution”. In: *eLife* 9 (2020), e57613. DOI: [10.7554/eLife.57613](https://doi.org/10.7554/eLife.57613).
- [9] Carsen Stringer and Marius Pachitariu. “Cellpose3: one-click image restoration for improved cellular segmentation”. In: *bioRxiv* (2024). DOI: [10.1101/2024.02.10.579780](https://doi.org/10.1101/2024.02.10.579780).
- [10] Martin Weigert and Uwe Schmidt. “Nuclei instance segmentation and classification in histopathology images with stardist”. In: *IEEE International Symposium on Biomedical Imaging Challenges (ISBIC)*. 2022, pp. 1–4. DOI: [10.1109/ISBIC56247.2022.9854534](https://doi.org/10.1109/ISBIC56247.2022.9854534).
- [11] Manuel E. González-Benito and etc. Others. “Assessment of Deep Learning Algorithms for 3D Instance Segmentation of Confocal Image Datasets”. In: *bioRxiv* (2021). DOI: [10.1101/2021.06.09.447748](https://doi.org/10.1101/2021.06.09.447748).
- [12] David A. Van Valen et al. “DeepCell: Automated Analysis of Live-Cell Imaging Data by Deep Learning”. In: *PLoS Computational Biology* 12.11 (2016), e1005177. DOI: [10.1371/journal.pcbi.1005177](https://doi.org/10.1371/journal.pcbi.1005177).
- [13] Özgün Çiçek et al. “3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation”. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2016, pp. 424–432. DOI: [10.1007/978-3-319-46723-8_49](https://doi.org/10.1007/978-3-319-46723-8_49).
- [14] Francesco Cutrale et al. “MARS: A Multi-Scale, Object-Oriented Approach to Automatic Image Segmentation in Developmental Biology”. In: *Developmental Biology* 428.2 (2017), pp. 192–203. DOI: [10.1016/j.ydbio.2017.06.005](https://doi.org/10.1016/j.ydbio.2017.06.005).
- [15] Johnson A Alberts B and Lewis J. *Molecular Biology of the Cell*. 4th. New York, USA: Garland Science, 2002. URL: <https://www.ncbi.nlm.nih.gov/books/NBK21054/>.
- [16] Martin Hailstone Dominic Waithe Tamsin J Samuels Lu Yang et al. “CytoCensus, mapping cell identity and division in tissues and organs using machine learning”. In: *eLife* 9 (2020), e51085. DOI: [10.7554/eLife.51085](https://doi.org/10.7554/eLife.51085).
- [17] Théo Aspert, Didier Hentsch, and Gilles Charvin. “DetecDiv, a generalist deep-learning platform for automated cell division tracking and survival analysis”. In: *eLife* 11 (Aug. 2022). Ed. by Ihor Smal, Carlos Isales, and Pascal Hersen, e79519. ISSN: 2050-084X. DOI: [10.7554/eLife.79519](https://doi.org/10.7554/eLife.79519). URL: <https://doi.org/10.7554/eLife.79519>.
- [18] Christian Szegedy et al. “Rethinking the Inception Architecture for Computer Vision”. In: (2016), pp. 2818–2826. DOI: [10.1109/CVPR.2016.308](https://doi.org/10.1109/CVPR.2016.308).
- [19] Johannes Schindelin et al. “Fiji: an open-source platform for biological-image analysis”. In: *Nature Methods* 9.7 (2012), pp. 676–682. DOI: [10.1038/nmeth.2019](https://doi.org/10.1038/nmeth.2019).
- [20] Stéfan van der Walt et al. “scikit-image: image processing in Python”. In: *PeerJ* 2 (2014), e453. DOI: [10.7717/peerj.453](https://doi.org/10.7717/peerj.453). URL: <https://doi.org/10.7717/peerj.453>.
- [21] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17.3 (2020), pp. 261–272. DOI: [10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2). URL: <https://doi.org/10.1038/s41592-019-0686-2>.
- [22] Özgün Çiçek et al. “3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016* 9901 (2016), pp. 424–432. DOI: [10.1007/978-3-319-46723-8_49](https://doi.org/10.1007/978-3-319-46723-8_49).

- [23] Serge Beucher and Fernand Meyer. “The watershed transformation applied to image segmentation”. In: *Scanning Microscopy* 1992.6 (1992), p. 28. URL: <https://digitalcommons.usu.edu/microscopy/vol1992/iss6/28>.
- [24] Alberto Bailoni et al. “GASP, a generalized framework for agglomerative clustering of signed graphs and its application to Instance Segmentation”. In: *CoRR* abs/1906.11713 (2019). URL: <http://arxiv.org/abs/1906.11713>.
- [25] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- [26] William E. Lorensen and Harvey E. Cline. “Marching Cubes: A High Resolution 3D Surface Construction Algorithm”. In: *ACM SIGGRAPH Computer Graphics* 21.4 (1987), pp. 163–169. DOI: [10.1145/37401.37422](https://doi.org/10.1145/37401.37422).
- [27] Philippe Lambin et al. “Radiomics: extracting more information from medical images using advanced feature analysis”. In: *European Journal of Cancer* 48.4 (2012), pp. 441–446. DOI: [10.1016/j.ejca.2011.11.036](https://doi.org/10.1016/j.ejca.2011.11.036).
- [28] Dijkstra and Edsger W. “A Note on Two Problems in Connexion with Graphs”. In: *Numerische Mathematik* 1.1 (1959), pp. 269–271.
- [29] Karl Pearson. “LIII. On Lines and Planes of Closest Fit to Systems of Points in Space”. In: *Philosophical Magazine* 2.11 (1901), pp. 559–572. DOI: [10.1080/14786440109462720](https://doi.org/10.1080/14786440109462720).
- [30] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *arXiv preprint arXiv:1409.1556* (2014). URL: <https://arxiv.org/abs/1409.1556>.
- [31] Gao Huang et al. “Densely Connected Convolutional Networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 4700–4708. DOI: [10.1109/CVPR.2017.243](https://doi.org/10.1109/CVPR.2017.243).
- [32] Mingxing Tan and Quoc Le. “EfficientNetV2: Smaller Models and Faster Training”. In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2021, pp. 10096–10106. URL: <https://arxiv.org/abs/2104.00298>.
- [33] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- [34] Mark Sandler et al. “MobileNetV2: Inverted Residuals and Linear Bottlenecks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 4510–4520. DOI: [10.48550/arXiv.1801.04381](https://doi.org/10.48550/arXiv.1801.04381).
- [35] Barret Zoph et al. “Learning Transferable Architectures for Scalable Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 8697–8710. DOI: [10.48550/arXiv.1707.07012](https://doi.org/10.48550/arXiv.1707.07012).
- [36] Christian Szegedy et al. “Rethinking the Inception Architecture for Computer Vision”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2818–2826. DOI: [10.1109/CVPR.2016.308](https://doi.org/10.1109/CVPR.2016.308).
- [37] Bolei Zhou et al. “Learning Deep Features for Discriminative Localization”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2921–2929. DOI: [10.48550/arXiv.1512.04150](https://doi.org/10.48550/arXiv.1512.04150).
- [38] Pedro O. Pinheiro and Ronan Collobert. “From image-level to pixel-level labeling with convolutional networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1713–1721. DOI: [10.1109/CVPR.1997.609286](https://doi.org/10.1109/CVPR.1997.609286).
- [39] Jia Deng et al. “ImageNet: A large-scale hierarchical image database”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009), pp. 248–255. DOI: [10.1109/CVPR.2009.5206848](https://doi.org/10.1109/CVPR.2009.5206848).
- [40] Chuan Guo et al. “On Calibration of Modern Neural Networks”. In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*. 2017, pp. 1321–1330. DOI: [10.48550/arXiv.1706.04599](https://doi.org/10.48550/arXiv.1706.04599).
- [41] Meelis Kull et al. “Beyond Temperature Scaling: Obtaining Well-Calibrated Multi-Class Probabilities with Dirichlet Calibration”. In: *Advances in Neural Information Processing Systems* 32 (2019). DOI: [10.48550/arXiv.1910.12656](https://doi.org/10.48550/arXiv.1910.12656).
- [42] Patrick Schmidt et al. *SurfaceMapComputation and ViewMap software (1.0)*. Zenodo. Version 1.0. 2022. DOI: [10.5281/zenodo.6390818](https://doi.org/10.5281/zenodo.6390818).
- [43] Lee R. Dice. “Measures of the Amount of Ecologic Association Between Species”. In: *Ecology* 26.3 (1945), pp. 297–302. DOI: [10.2307/1932409](https://doi.org/10.2307/1932409).
- [44] Paul Jaccard. “Etude de la distribution florale dans une portion des Alpes et du Jura”. In: *Bulletin de la Societe Vaudoise des Sciences Naturelles* 37 (Jan. 1901), pp. 547–579. DOI: [10.5169/seals-266450](https://doi.org/10.5169/seals-266450).
- [45] Anuradha Kar et al. “Assessment of deep learning algorithms for 3D instance segmentation of confocal image datasets”. In: *bioRxiv* (2021). DOI: [10.1101/2021.06.09.447748](https://doi.org/10.1101/2021.06.09.447748).
- [46] Sandra M et al. “Cre recombinase microinjection for single-cell tracing and localised gene targeting”. In: *Development* 150.3 (2023). DOI: [10.1242/dev.201206](https://doi.org/10.1242/dev.201206).

Appendix A: Supplementary model parameters

A.1 Cell segmentation models parameters

	Membrane segmentation	
	3D U-Net	GASP
<i>model</i>	Pre-trained model name to load the different model parameters. In this case <code>confo-cal_3D_unet_sa_meristem_cells</code>	$\beta = 0.5$ Balance under-/over-segmentation; 0 - aim for under-segmentation; 1 - aim for over-segmentation.
<i>stride – ratio = 0.75</i>	Only if patches used due to computational limitations. Stride between patches will be computed as $stride - ratio * patch$ recommended values range (0.5, 0.75) to make sure the patches have enough overlap to get smooth prediction maps.	$threshold = 0.4$ Probability maps threshold. $minsize = 30$ Minimum supervoxels size. $\sigma = 0.8$ Sigma for the gaussian smoothing of the distance transform. $w\sigma = 0$ $minsize = post = 40$ Sigma for the smoothing of boundary Minimum segment size in the final segmentation (Cell minimum size).
	Nuclei segmentation	
	Cellpose	
<i>channels = [0, 0]</i>	0=grayscale, 1=red, 2=green, 3=blu. Set channels to a list with each of these elements, e.g. <code>channels = [0,0]</code> if you want to segment cells in grayscale or for single channel images, or <code>channels = [2,3]</code> if you green cells with blue nuclei.	
<i>diameter = 17</i>	Cellpose needs a user-defined cell diameter (in pixels) as input, or to estimate the object size of an image-by-image basis.	
<i>flow – threshold = 0.4</i>	The flow_threshold parameter is the maximum allowed error of the flows for each mask.	
<i>cellprob – threshold = 0.4</i>	The network predicts 3 outputs: flows in X, flows in Y, and cell “probability”. The predictions the network makes of the probability are the inputs to a sigmoid centered at zero ($1/(1 + e^{-x})$), so they vary from around -6 to +6. The pixels greater than the cellprob_threshold are used to run dynamics and determine ROIs.	
<i>do – 3D = True</i>	3D Cellpose dynamics implementation.	

Further details [here](#).

Appendix B: Supplementary Tables

B.1 Morphological radiomics features description

Table B.1: Shape-based features description. Based on [pyradiomics](#).

Feature	Formula	Description
Mesh Volume	$V = \sum_{i=1}^{N_f} \frac{\mathbf{O}\mathbf{a}_i \cdot (\mathbf{O}\mathbf{b}_i \times \mathbf{O}\mathbf{c}_i)}{6}$	The volume of the ROI is computed using the mesh triangles. For each face i , defined by points a_i, b_i, c_i , the signed volume of the tetrahedron formed with the origin (O) is calculated. Summing these values gives the total volume.
Voxel Volume	$V_{\text{voxel}} = \sum_{k=1}^{N_v} V_k$	Approximates the volume by summing up the volumes of the voxels included in the ROI. Does not use the mesh and is less precise.
Surface Area	$A = \sum_{i=1}^{N_f} \frac{1}{2} \mathbf{a}_i \mathbf{b}_i \times \mathbf{a}_i \mathbf{c}_i $	The surface area is computed by summing the areas of all triangles in the mesh, where each triangle is formed by vertices a_i, b_i, c_i .
Surface to Volume Ratio	$\frac{A}{V}$	Measures the compactness of the shape, where lower values indicate more compact (sphere-like) shapes.
Sphericity	$\frac{36\pi V^2}{A^3}$	A dimensionless measure of shape roundness. Values range from 0 to 1, where 1 represents a perfect sphere.
Maximum 3D Diameter	—	The largest pairwise Euclidean distance between vertices in the mesh. Also known as Feret Diameter.
Maximum 2D Diameter (Slice)	—	The largest pairwise Euclidean distance between vertices in the axial plane.
Maximum 2D Diameter (Column)	—	The largest pairwise Euclidean distance between vertices in the coronal plane.
Maximum 2D Diameter (Row)	—	The largest pairwise Euclidean distance between vertices in the sagittal plane.
Major Axis Length	$4\sqrt{\lambda_{\text{major}}}$	The largest axis length of the ROI-enclosing ellipsoid, computed using the largest principal component.
Minor Axis Length	$4\sqrt{\lambda_{\text{minor}}}$	The second-largest axis length of the ROI-enclosing ellipsoid, computed using the second principal component.
Least Axis Length	$4\sqrt{\lambda_{\text{least}}}$	The smallest axis length of the ROI-enclosing ellipsoid, computed using the smallest principal component.
Elongation	$\sqrt{\frac{\lambda_{\text{minor}}}{\lambda_{\text{major}}}}$	Shows the relationship between the largest and second-largest principal components. Values range between 0 (elongated) and 1 (circle-like).
Flatness	$\sqrt{\frac{\lambda_{\text{least}}}{\lambda_{\text{major}}}}$	Describes the relationship between the largest and smallest principal components, indicating flatness. Values range from 0 (flat) to 1 (sphere-like).

B.2 First-order radiomics features description

Table B.2: Intensity-based features description. Based on [pyradiomics](#).

Feature	Formula	Description
Energy	$\text{Energy} = \sum_{i=1}^{N_p} (X(i) + c)^2$	Measures the magnitude of voxel values in the image. A larger value implies a greater sum of squares of the values. c is a shift value to prevent negative intensities.
Total Energy	$\text{Total Energy} = V_{\text{voxel}} \sum_{i=1}^{N_p} (X(i) + c)^2$	Similar to Energy, but scaled by the volume of the voxel in cubic millimeters.
Entropy	$\text{Entropy} = - \sum_{i=1}^{N_g} p(i) \log_2(p(i) + \epsilon)$	Measures the randomness or uncertainty of the image values. It quantifies the average amount of information needed to encode the image values.
Minimum	$\text{Minimum} = \min(X)$	The smallest intensity value within the ROI.
10th Percentile	—	The intensity value below which 10% of the voxel intensities in the ROI are found.
90th Percentile	—	The intensity value below which 90% of the voxel intensities in the ROI are found.
Maximum	$\text{Maximum} = \max(X)$	The largest intensity value within the ROI.
Mean	$\text{Mean} = \frac{1}{N_p} \sum_{i=1}^{N_p} X(i)$	The average intensity value within the ROI.
Median	—	The median intensity value within the ROI.
Interquartile Range	$\text{IQR} = P_{75} - P_{25}$	The difference between the 75th percentile and the 25th percentile of voxel intensities.
Range	$\text{Range} = \max(X) - \min(X)$	The difference between the maximum and minimum voxel intensities in the ROI.
Mean Absolute Deviation (MAD)	$\text{MAD} = \frac{1}{N_p} \sum_{i=1}^{N_p} X(i) - \bar{X} $	The mean of the absolute differences between the voxel intensities and the mean value of the image.
Robust Mean Absolute Deviation (rMAD)	$\text{rMAD} = \frac{1}{N_{10-90}} \sum_{i=1}^{N_{10-90}} X_{10-90}(i) - \bar{X}_{10-90} $	Mean absolute deviation of voxel intensities, calculated only on values between the 10th and 90th percentiles.
Root Mean Squared (RMS)	$\text{RMS} = \sqrt{\frac{1}{N_p} \sum_{i=1}^{N_p} (X(i) + c)^2}$	The square root of the mean of the squared voxel intensities. A measure of magnitude.
Standard Deviation	$\text{SD} = \sqrt{\frac{1}{N_p} \sum_{i=1}^{N_p} (X(i) - \bar{X})^2}$	Quantifies the variation or dispersion of voxel intensities from the mean.
Skewness	$\text{Skewness} = \frac{\mu_3}{\sigma^3}$	Measures the asymmetry of the intensity distribution in the ROI. Positive or negative values indicate skewness direction.
Kurtosis	$\text{Kurtosis} = \frac{\mu_4}{\sigma^4}$	Quantifies the "peakedness" of the intensity distribution. High kurtosis indicates more intensities in the tails.
Variance	$\text{Variance} = \frac{1}{N_p} \sum_{i=1}^{N_p} (X(i) - \bar{X})^2$	The mean of the squared deviations of voxel intensities from the mean. Measures spread.
Uniformity	$\text{Uniformity} = \sum_{i=1}^{N_g} p(i)^2$	A measure of the sum of squared intensities. Higher values indicate greater homogeneity.

Appendix C: Supplementary algorithms

C.1 Graph-based segmentation quality assessment approach

Algorithm 1: Pseudocode for extracting high-overlap cell associations in both target_in_reference and reference_in_target.

Data: $Pred$ (predicted segmentation) and Ref (ground truth segmentation)

Result: $target_in_reference$ and $reference_in_target$ sets

Let $T \leftarrow$ list of cell labels in $Pred$ (excluding background);

Let $R \leftarrow$ list of cell labels in Ref (excluding background);

Initialize empty dictionaries $target_in_reference$ and $reference_in_target$;

foreach cell label t in T **do**

foreach cell label r in R **do**

Compute overlap $o_{t,r}$ between t and r ;

if $o_{t,r}$ exceeds threshold θ **then**

$target_in_reference[(t, r)] \leftarrow o_{t,r}$;

$reference_in_target[(t, r)] \leftarrow o_{t,r}$;

Remove entries from $target_in_reference$ if $o_{t,r} < \theta$ or if t or r is invalid;

Remove entries from $reference_in_target$ if $o_{t,r} < \theta$ or if t or r is invalid;

Algorithm 2: Pseudo-code for the graph-based segmentation quality approach

Data: $target_in_reference$ and $reference_in_target$ associations between predicted (target) and ground-truth (reference) cells

Result: Connected subgraphs, each labeled by a segmentation state (e.g., *correct*, *undersegmented*, *oversegmented*, *missing*, *confused*)

Initialize bipartite graph $G = (V_t, V_r, E)$ where V_t are target cells and V_r are reference cells ;

foreach (cell pair (t, r) in both target_in_reference and reference_in_target) **do**

Add edge (t, r) to E ;

Compute the connected components $\{C_1, \dots, C_k\}$ of G ;

foreach component C_i **do**

Let $T_i \subseteq V_t$ be the target cells in C_i ;

Let $R_i \subseteq V_r$ be the reference cells in C_i ;

if $|T_i| = 1$ **and** $|R_i| = 1$ **then**

$State \leftarrow \text{correct}$

else if $|T_i| = 1$ **and** $|R_i| > 1$ **then**

$State \leftarrow \text{undersegmented}$

else if $|T_i| > 1$ **and** $|R_i| = 1$ **then**

$State \leftarrow \text{oversegmented}$

else if $|T_i| = 0$ **or** $|R_i| = 0$ **then**

$State \leftarrow \text{missing or background}$

else

$State \leftarrow \text{confused}$
