Modelling, Segmenting and Analyzing in Vitro Microcirculatory Networks using a graph and machine learning based framework.

Jesús Ávila Trigueros

Thesis submitted to the University of Ottawa in partial fulfillment of the requirements for the degree of Master of Applied Science, Biomedical Engineering

Department of Mechanical Engineering
Faculty of Engineering
University of Ottawa

uOttawa

Abstract

Microcirculatory phenomena play an important role in physiological transport processes and are increasingly investigated using microfluidic platforms. However, analysis of complex networks is often constrained by fragmented workflows, manual image processing, and single vessel analysis. In particular, cell-free layer (CFL) measurement commonly relies on individual channel studies using manual or semi-automated methods.

This thesis presents an integrated analytical framework for microfluidic network analysis that combines graph theory, image segmentation with deep learning, and a unified software environment. A microfluidic chip resembling the human retina was abstracted into a directed graph, enabling explicit representation of network topology and systematic association of experimental data with individual vessels. This representation supports the application of established network analyses to characterize flow organization and connectivity.

To enable automated CFL image segmentation, a fully automatic red blood cell (RBC) core segmentation method based on an Attention U-Net architecture was developed. The method displayed good performance across a range of experimental conditions (e.g hematocrit level, channel width and lighting conditions) delineating the red blood cells core accurately enabling indirect estimation of CFL thickness, with performance primarily limited by ground-truth quality.

These components were integrated into an integrative analysis environment that supports visualization, data management, network analysis, and comparative studies across experimental conditions. Together, the proposed framework provides a scalable and extensible tool for the quantitative analysis of biomimetic microfluidic networks, with direct applications in microcirculatory research and the study of physiologically relevant vascular systems.

Keywords: Microcirculation, cell-free layer, graph theory, microfluidics, user interface, microfluidic networks, Attention U-NET, machine learning

Declaration of Authorship

I hereby certify that this thesis is entirely my own original work except where otherwise indicated. I am aware of the University's regulations concerning plagiarism, including those concerning consequent disciplinary actions. Any use of the works of any other author, in any form, is properly acknowledged at their point of use.

During the preparation of this work I used ChatGPT in order to check for grammar, spelling and synonyms for some words, and brainstorming ideas for writing and coding. After using this tool, I reviewed and edited the content as needed and take full responsibility for the content of the published material. Copilot was used to review some sentences of this thesis by my thesis supervisor.

Acknowledgements

To Dr. Marianne Fenech, I am grateful for your optimism, encouragement, mentorship, and patience throughout my studies, working with you has been a pleasure and an experience full of lessons beyond academia.

To my mom Martha who always taught me to work hard for what I wanted.

To my friend Zachary Flahaut to be there for me and read one of my first drafts of this thesis.

To my friends Andrés, Adriana, Alonso and Ana to be always there for me and made my experience in Ottawa amazing.

To my lab mates Aude, Afra, Kyle, Tatiana, Maya, Andy and Elena that cheered me up and gave me feedback on my projects through all the stages of my master's.

To Dr. Adrian Chan and Dr. José Morán, I am grateful for beign part of my examination commitee.

I acknowledge the Digital Research Alliance of Canada and the utilization of their Beluga and Rorqual clusters, where the training for the segmentation model, critical for this project, was done.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter outlines the scientific and methodological motivations underlying this thesis and defines the objectives developed in response to current limitations in microcirculation research.

1.1 Research motivation

Biological transport systems, such as the blood circulatory system, plant vasculature, and respiratory systems, share a common structural feature: they are organized as branching networks [1]. For instance, the vascular network resembles a branching tree: the aorta forms the trunk, large arteries act as primary branches, and progressively smaller vessels extend outward to form fine terminal structures. The smallest vessels (capillaries) serve as the interface between arterioles and venules, where essential exchanges of gases, nutrients, and waste products occur [2].

Depending on the organ, capillary diameters can vary widely, ranging from less than 100 μm down to structures smaller than 20 μm [2, 3]. At the microscale, the structural

and dynamical properties of these networks fundamentally influence blood flow behaviour, motivating detailed experimental and computational investigation.

Among all the diverse phenomena happening during microcirculation, one interesting phenomenon is the migration of red blood cells (RBCs) toward the center of the vessel depleting the near-wall region of cells. The near-wall region depleted of RBCs is known as the cell-free layer (CFL) [4].

The CFL plays an important role in microcirculatory physiology and has been studied in the context of various vascular conditions, including Bernard–Soulier syndrome, Glanzmann thrombasthenia, and malaria [5]. CFL study has therefore attracted significant attention, particularly by using microfluidic devices. Microfluidic platforms allow the investigation of physiological conditions in a controlled *in vitro* setting, avoiding invasive *in vivo* procedures for both diagnosis and research [3, 6].

Despite extensive experimental work on CFL characterization [4, 7–10], many existing microfluidic studies rely on manual or semi-automated segmentation techniques. These techniques are time-consuming and introduce variability due to user dependence, especially when large image datasets are involved [7]. Consequently, the lack of a fully automated and reproducible segmentation pipeline remains a major bottleneck in high-throughput microcirculatory analysis.

However, segmentation challenges are only part of a broader limitation in current microcirculatory research, most microfluidic studies of microcirculation rely on simplified geometries, typically single-channel configurations [5, 11, 12]. While useful for controlled

experiments, these setups do not capture the interconnected nature of vascular networks. As a result, they overlook the influence of network topology on key phenomena such as CFL formation, hematocrit (HTC) distribution, and flow dynamics.

Microfluidic networks have emerged as a promising alternative, as they better reproduce the hierarchical and interconnected structure of biological systems. These platforms enable parallel experimentation, the study of network level processes and support applications such as vascular biology, tissue engineering, and disease modelling, including cancer, diabetic retinopathy, and cardiovascular disorders [12–14]. However, their analysis remains fragmented. Researchers often rely on multiple tools (e.g. R, MATLAB, Python, ImageJ) and perform repetitive, manual workflows.

To address this limitation, a mathematical framework capable of representing interconnected systems is required. Graph theory provides an effective approach for modelling and analyzing interconnected systems [15]. In particular, graph-based models facilitate the study of how experimental variables propagate through a network (e.g., flow conservation across bifurcations), the identification of upstream–downstream dependencies between vessels, and the systematic comparison of measurements across hierarchical levels. These capabilities are not accessible using isolated or image-based analyses alone.

While graphs are applied in fields such as artificial intelligence, social network analysis and anatomical modelling, their use in microfluidics remains limited [15–18]. Existing studies have explored graph-based representations for large biological networks, such as the mouse brain or renal systems [19, 20], or focused primarily on visualization and topology [2,

21–23]. However, integrative tools that combine topology with experimental data analysis are still lacking.

In summary, two main gaps motivate this thesis. First, current CFL segmentation workflows remain user-dependent and inefficient for large datasets, limiting scalability and reproducibility. Second, there is a lack of analytical tools capable of representing and analyzing microfluidic systems as interconnected networks. To address these challenges, this project aims to extend prior work on CFL segmentation by developing a fully automated method that reduces variability and processing time. In parallel, it introduces a graph-based framework for modelling and analyzing microfluidic networks, enabling system level investigation of microcirculatory behaviour.

1.2 Research objectives

The main goal of this thesis is to develop an analytical framework for the study of *in vitro* microcirculatory networks. This framework will enable researchers to extract, visualize, and interpret high-dimensional data from these intricate systems, ultimately advancing our understanding of blood microcirculation.

Because existing analyses largely ignore the topological structure of microcirculatory networks, the **first objective** of this thesis is to propose a graph representation of a biomimetic microfluidic chip designed to topologically mimic the human retina and highlight the benefits of the model.

Since manual or semi automatic segmentation of the cell-free layer is time consuming and prone to user bias, the **second objective** is to develop a fully automatic CFL segmentation tool that can be readily integrated into microcirculation research.

Lastly, to overcome the fragmentation of existing analysis pipelines, the **third objective** is to design and implement a graph-based and user-friendly, integrative tool for the manipulation and analysis of experimental data obtained from microcirculatory networks. This tool will provide multiple functionalities within a single environment: it will allow researchers to visualize both network structures and dynamic processes, conduct customized analyses, generate figures and statistical summaries, and support comparative evaluations across different experimental conditions. By bringing these capabilities together, the tool will streamline the transition from raw data to generating new knowledge about microcirculatory dynamics.

In particular, the analyses will focus on identifying flow patterns and characterizing the propagation of some microfluidic features, thereby illustrating how the framework can be used both for methodological validation and for advancing the biological understanding of microcirculation. As these analyses can provide quantitative insight into how network topology influences flow heterogeneity and transport phenomena, thereby linking microfluidic design parameters to biologically relevant microcirculatory behaviours.

1.3 Contributions

The work presented in this thesis is within the scope of a broader research effort conducted within the Fenech Group, which aims to advance the understanding of blood rheology at the microscale, across both simple and complex microvascular geometries. In this context, the thesis contributes methodological and computational tools that support ongoing and future experimental studies of microcirculatory flows.

A retinal-inspired network, together with simpler microchannel geometries routinely used in the Fenech Group, constitute the experimental data set used in this work. The retinal chip was developed in collaboration with Dr. Fullana and Dr Julien (Sorbonne University), Dr. Paques (Hôpital de Paris, clinician) and fabricate by Dr. Charlot (Université de Montpellier). The experimental data set were obtained by graduate students (Maya Salame, Laureline Julien, Vermeerbergen Ferry). The image datasets acquired from these experimental systems served a dual purpose: (i) training and validating machine learning models for the automated detection of the cell-free layer (CFL), and (ii) demonstrating the relevance of graph-based representations for analyzing transport phenomena in complex microvascular networks.

Within this collaborative framework, the main contributions of this thesis are as follows:

- Development of a graph-based framework for representing biomimetic microfluidic networks, enabling network-aware analysis of experimental data.

- Design and implementation of an integrative analysis environment that unifies image processing, graph modelling, and experimental data visualization within a single platform.

- Development and validation of a machine learning approach based on an Attention U-Net architecture for automated RBC core segmentation and CFL estimation.

- Demonstration of network-level analyses, including variable propagation, inference, and topology-aware interpretation, through multiple experimental use cases.

Part of the work presented in this thesis has been disseminated through the following contributions:

- Conference Paper: J. R. . Avila Trigueros and M. . Fenech, "Use of graph theory for biomimetic microchannel network blood flow analysis", CMBES Proc., vol. 47, no. 1, May 2025.

- Poster award: Second place under the category "Enabling technologies for health care and augmented life" at the Graduate Poster Competition organized by the Faculty of Engineering at University of Ottawa. Poster title: "Deep learning image segmentation for microvascular flow analysis".

1.4 Thesis organization

This thesis is structured into six chapters, progressing from conceptual foundations to methodological development and integrative application.

Chapter 1 introduces the research motivation, defines the problem, and outlines the objectives guiding this work.

Chapter 2 discuss the theoretical and methodological background, covering microfluidic networks, graph theory, cell free layer, and deep learning for image segmentation, therefore framing the interdisciplinary context of the thesis.

Chapter 3 details the developed graph model of the retina microfluidic chip. It presents the abstraction process from physical network to directed graph and defines the analytical capabilities enabled by this model.

Chapter 4 develops the machine learning component of this work, introducing an Attention U-Net implementation for RBCs core segmentation. It details dataset preparation, model design, and evaluation, providing the image-derived measurements later integrated into the network framework.

Chapter 5 presents an integrative analysis environment designed to enable network-aware investigation of microfluidic systems. This environment unifies experimental data, image processing outputs (Chapter 4, and graph based analysis (Chapter 3, enabling topology aware investigation of microfluidic networks.

Chapter 6 concludes the thesis by synthesizing the main contributions, discussing their implications, and outlining directions for future research.

Chapter 2

Literature review

This chapter covers the current state of the main research dimensions of this thesis, starting with an overview of microfluidic devices, focusing on microfluidic networks, especially those used for the study of microcirculation.

The second section explains the fundamentals of graph theory, comparing the different tools available for graph creation and analysis, and exposes the different uses of graph theory within biomedical engineering highlighting those currently being used.

Lastly, this chapter shows the different methods for CFL segmentation that exist at the moment and highlights some of their limitations. Most microfluidic studies have focused on isolated channels, most graph-theoretical approaches have not implemented microfluidic network analysis, and CFL segmentation pipelines are currently prone to inexactitudes and performed using heterogeneous tools.

Lastly, the integration of each domain into a unified framework for the study of microcirculation remains limited. This chapter reviews the advancements in each of these domains and highlights the conceptual and methodological gaps that motivate the framework proposed in this thesis.

2.1 Microfluidic networks

Microfluidic devices have become an important tool in biomedical engineering due to their ability to manipulate small volumes of fluids in controlled environments. Their applications include diagnostics, therapeutics, drug discovery, disease modelling (e.g. Parkinson's disease, Alzheimer's disease, and multiple sclerosis), tissue engineering, biosensing, lab-on-a-chip and organ-on-a-chip technologies [21, 24].

The development of microfluidic systems has progressively evolved from simple single channel devices toward more complex architectures designed to replicate more accurately biological environments. Early studies focused on isolated channel geometries to investigate fundamental transport phenomena, such as diffusion, mixing, and laminar flow [24]. More recent advances have led to the emergence of organ-on-a-chip platforms, which aim to reproduce specific physiological functions by incorporating relevant cell types, mechanical stimuli, and biochemical gradients. These platforms enable controlled experiments with small sample volumes and offer fast, low-cost, and portable solutions for diagnostics and research [3, 4, 7, 10].

Despite these advances, many conventional microfluidic devices remain limited in their ability to replicate the interconnected and hierarchical nature of biological vascular systems [12]. In microcirculation, blood flow is distributed through highly branched and interconnected networks, where local hemodynamic conditions are strongly influenced by upstream and downstream connectivity [3]. Phenomena such as flow redistribution at

bifurcations, heterogeneity in vessel diameters, and coupling between neighboring vessels play a critical role in regulating oxygen delivery, shear stress, and cellular behaviour. These effects cannot be fully captured using isolated or weakly connected channel geometries.

However, the main limitation lies not in fabrication but in analysis: current approaches remain localized and do not capture network-level behaviour. Most existing studies focus either on localized measurements within individual channels or on qualitative observations of network behaviour [12–14, 25].

To overcome these limitations, collaborators from Sorbonne University designed an artificial microvascular network inspired by the human retina. The retinal microvasculature is the only human microvascular network that can be observed non-invasively *in vivo*. As a result, it provides valuable insight into the state of the broader vascular system. The design aimed to reproduce the full range of vessel diameters present in the retinal circulation from the central retinal artery down to the capillaries.

The network was generated recursively under the assumption of an ideal configuration following geometrical construction laws and the relationship between mother and daughter vessels [1, 26]. Through collaboration with a clinical research group, morphological data were obtained from retinal images. Thanks to the work performed by Laureline Julien, these different images enabled the adaptation of the construction rules to reproduce the global architecture of a human retinal network without making it patient-specific. The chip was fabricated using the backskater lithography method by Dr. Benoit Charlot [26]. A representation of the final result for the microfluidic chip can be seen in Figure 2.1.

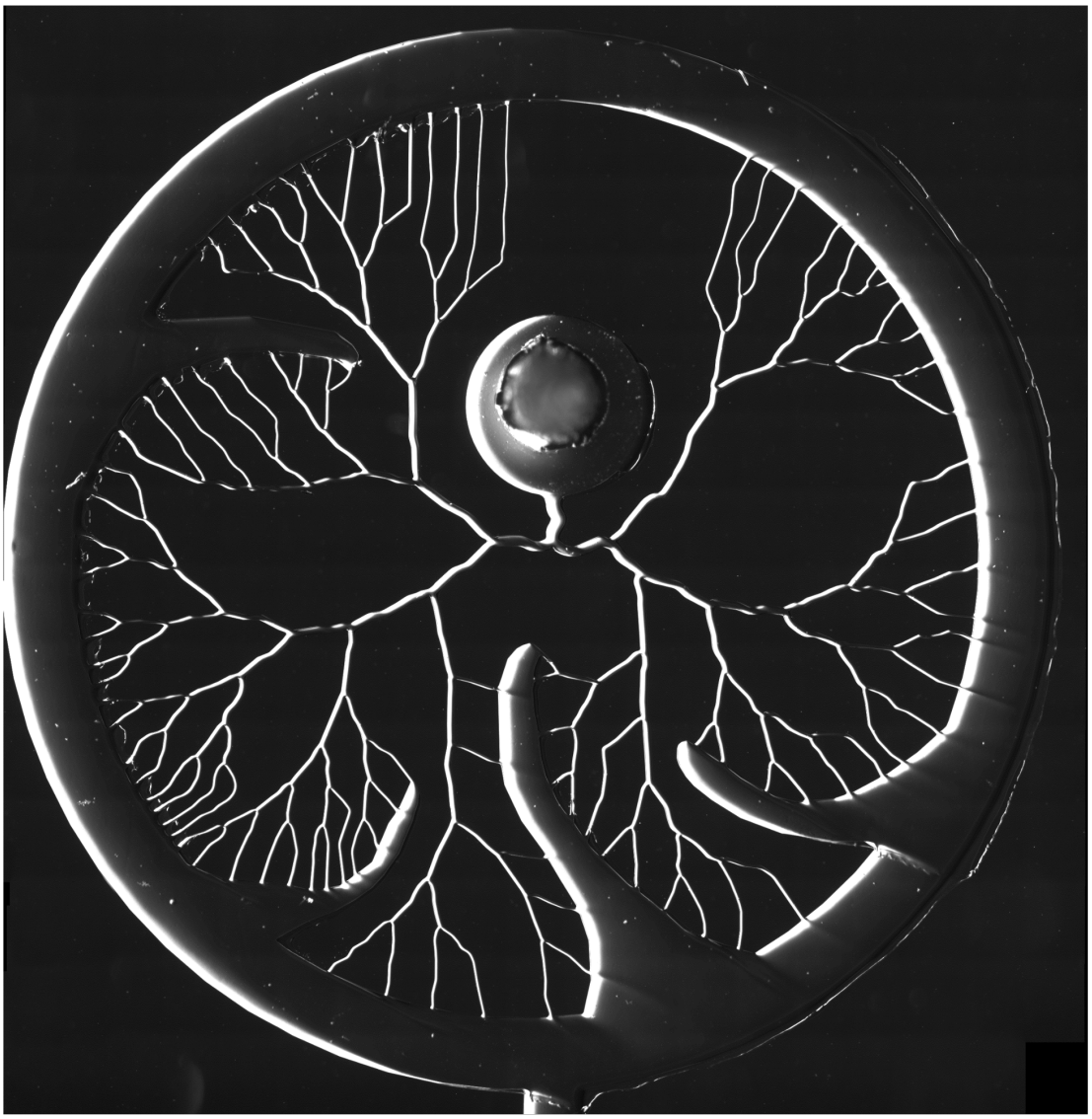

Figure 2.1: Stitched image of the retina chip obtained obtaining through microscopy imaging. Image built by Aude Vermeerbergen. Ottawa 2024. The circle at the center of the image is the inlet whereas the corona where all the vessels end is the outlet.

The resulting biomimetic microfluidic network, consists of a chip with one inlet (seen

as a big circle at the image) where different branches emerge from. At this scale, channel branching is predominantly dichotomous [1, 27]. The thickness of such vessels is reduced by each new bifurcation and all of the vessels connect to the single outlet by a corona. This network provides a controlled yet physiologically relevant platform to study microcirculatory topology.

Retinal-inspired microfluidic networks reflect real microcirculation and exhibit an organized but heterogeneous structure [1, 27]. However, tools for systematically integrating experimental data with such network topology are still limited, and the lack of unified analytical frameworks restricts the ability to perform comprehensive, system-level analyses. As a result, an analysis pipeline is needed. The biomimetic microfluidic retina therefore serves as the central model investigated in this work, motivating the use of topology-aware analytical approaches. In this context, graph-based representations offer a natural and powerful framework to model such systems, enabling the integration of topology, experimental measurements, and derived variables within a unified analytical structure.

2.2 Graph Theory

Graph theory provides a mathematical framework for representing and analyzing systems composed of interconnected elements. Unlike traditional analytical approaches that treat components in isolation, graph-based representations explicitly encode relationships between elements, enabling the study of both local interactions and global system behaviour [15].

In the context of microfluidic networks, this distinction is critical. Conventional analyses of microcirculatory experiments are typically performed at the level of individual channels or localized regions, implicitly neglecting the influence of upstream and downstream connectivity . As a result, they are limited in their ability to capture how flow redistribution, structural hierarchy, and vessel interdependence affect observed phenomena [5, 11, 12]. In contrast, graph theory enables the explicit modelling of connectivity between channels, allowing variables such as flow, hematocrit, or CFL characteristics to be interpreted within the context of the entire network. This makes it possible to analyze path-dependent transport, quantify hierarchical organization, and study how local changes propagate through the system—capabilities that are not accessible through isolated or purely image-based analyses.

Formally, a graph is defined as a set of nodes (or vertices) connected by edges, where nodes represent entities of interest and edges represent relationships between them. The arrangement of these elements defines the graph topology, which governs how information or physical quantities can propagate through the system [15, 20]. Graph-based approaches therefore provide a natural and scalable representation for microfluidic networks, where channels are physically connected and flow behaviour is inherently dependent on network structure.

Beyond representation, graph theory offers a wide range of analytical tools that enable quantitative characterization of complex systems. These include local measures, such as node degree and neighborhood structure, as well as global metrics, such as shortest paths, connectivity, and network organization [20]. Together, these tools allow for a multi-scale

analysis that links local experimental observations to system-level behaviour.

While graph theory has been widely applied in domains such as social network analysis, artificial intelligence, and anatomical modelling, its application to microfluidic systems remains limited [15–18]. Existing approaches in biomedical contexts often focus on large-scale anatomical networks or structural visualization, with limited integration of experimentally derived data [2, 21–23]. This gap highlights the need for frameworks that not only represent microfluidic networks as graphs, but also integrate experimental measurements within these structures to enable topology-aware analysis.

2.2.1 Graph Properties

To model a microfluidic network as a graph, only a subset of graph-theoretical concepts is required. Rather than providing an exhaustive description, this section focuses on the properties that directly support the representation and analysis of microcirculatory systems.

A graph is defined by a set of nodes and edges. Nodes serve as abstract representations of the entities under study and the edges define relationships among them. For example, nodes and edges can represent mutual friends, when used for social media; types of mammals, when used for taxonomy; connected organs, when talking about anatomical systems and so on [2, 20]. An illustrating image of both components is presented on Figure 2.2.

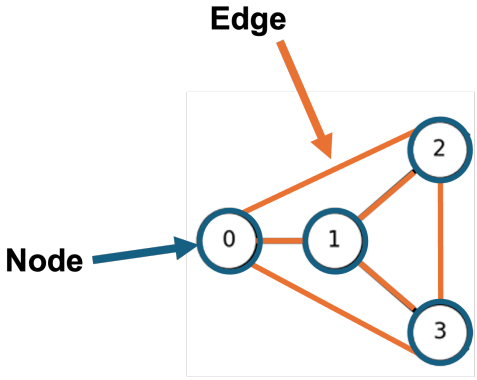

Figure 2.2: A graph showcasing 4 nodes fully connected by edges.

The number of edges reflects the degree of interaction that each node maintains within the system. A practical example is the blood circulation through the heart structures. The process starts when de-oxygenated blood arrives in the right atrium through the vena Cava and then is redirected to the lungs thanks to the pulmonary arteries where it receives oxygen thanks to the capillaries. Finally, the oxygenated blood is received by the left atrium and then the aorta distributes it to all the body where the nutrients and oxygen will be removed and the cycle begins again. This process is illustrated in Figure 2.3. Because blood flow has a defined direction, the resulting structure is naturally modeled as a *directed cyclic graph*, where edges encode the direction of transport while nodes represent each vascular structure in which blood flows through [2].

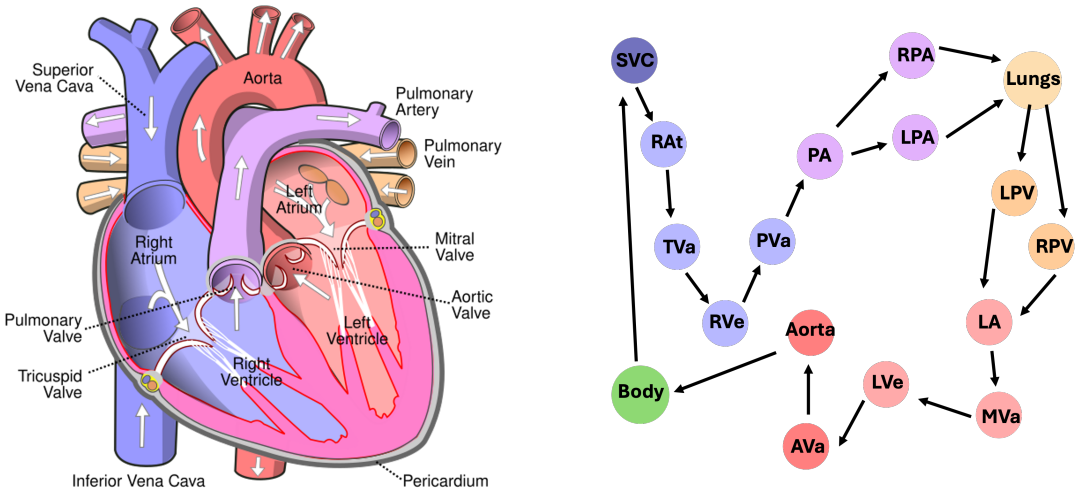

Figure 2.3: Graph representation of the cardiac structures involved in blood circulation inspired from [2]. Node names correspond to anatomical elements, and edges represent the direction of blood flow between them. The abbreviations on the right image match the names described on the left image.

In the context of microfluidic networks and this work, edges can represent specific spatial locations within the device (e.g., bifurcations, junctions, or measurement points), while nodes can represent the channels through which flow occurs. This representation enables the incorporation of both topology and experimental data. Each node can store local information such as image-derived features, flow measurements, or geometrical properties, while edges capture connectivity and enable the propagation of variables across the network.

Several graph properties are particularly relevant for the analyses performed in this work. The *degree* of a node reflects the number of connected channels and is directly related to branching behaviour at junctions. A *path* defines a sequence of connected nodes and edges, enabling the study of how quantities such as flow or hematocrit propagate

through the network. The *distance* between nodes, defined as the number of edges along a path, provides a simple way to quantify spatial relationships within the topology [20].

At a global level, the notion of *connectivity* is essential. Microfluidic networks are inherently connected systems, where local behaviour is influenced by upstream and downstream conditions. This property cannot be captured by isolated channel analyses and motivates the use of a graph representation. Based on these properties, graphs can be classified into various types. *Connected* graphs are those in which a path exists between every pair of nodes, whereas *disconnected* graphs contain nodes or clusters without mutual connections. Similarly, a *cyclic* graph contains at least one closed loop, while an *acyclic* graph has no such cycles. For its better understanding, Figure 2.4 showcases the different types of graphs.

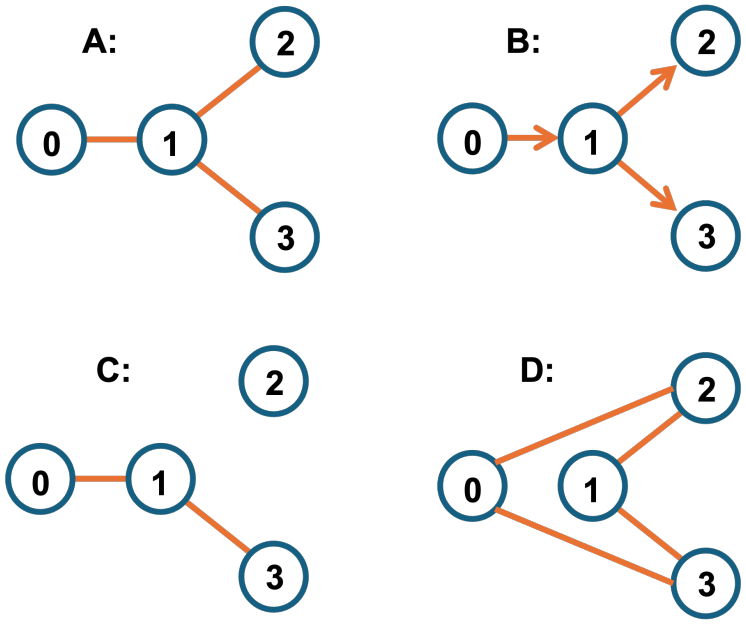

Figure 2.4: Graph types. A: a fully connected, acyclic, undirected graph; B: fully connected, acyclic, directed graph; C: disconnected graph; D: fully connected cyclic graph.

Finally, subgraphs provide a mechanism to isolate and analyze specific regions of the network (e.g., local branching structures or regions of interest) while preserving their relationship to the overall topology [20]. This is particularly useful for comparing localized phenomena with system-level behaviour.

Together, these properties form the minimal set of graph concepts required to represent microfluidic networks in a way that supports topology-aware analysis. More advanced metrics exist, but they are not required for the scope of this work and are therefore not considered here. Additionally, these definitions form the basis for the graph construction and analysis procedures described in Chapter 3.

2.2.2 Applications

Graph theory has been widely implemented to analyze complex interconnected systems across multiple domains. In geosciences, for instance, Earth systems such as river networks and fault structures can be modeled as graphs to study their spatial organization and dynamics [28]. In the social sciences, graph-based representations are commonly used to describe relationships between individuals and to analyze patterns of interaction within groups [17]. In biomedical engineering, its applications have been particularly found in the study of complex biological networks, such as brain connectivity and vascular systems, where graph-based models enable the characterization of structural organization and functional relationships [2, 15].

Advances in imaging techniques, have further supported these applications by enabling the reconstruction of biological networks, allowing graph representations to be used for visualization, structural analysis, and, in some cases, the study of disease progression [22]. Similarly, graph-based approaches have also been applied to large scale biological systems, such as mouse brain networks and renal vascular structures, where they facilitate the analysis and visualization of highly complex and dense networks [19, 20]. However, these applications are primarily focused on structural representation and visualization, with limited integration of experimentally derived variables and functional analyses at the level required for microfluidic studies [2, 21–23].

This limitation becomes more pronounced in the context of microfluidic networks. While these systems are explicitly designed to replicate interconnected microvascular struc-

tures , their analysis remains largely localized, and graph-based representations are rarely used to study the interaction between topology and experimentally measured quantities such as flow behaviour or cell distribution [12, 24].

To the best of our knowledge, only a limited number of studies have explored the use of graph theory in microfluidics, and these are primarily restricted to structural descriptions or simplified representations [20, 21]. The absence of frameworks that integrate topology with experimental measurements highlights a critical gap between graph-theoretical methods and microfluidic data analysis.

The present work addresses this gap by developing a graph-based framework that not only represents the topology of microfluidic networks but also integrates experimental data, image-derived variables, and analytical tools within a unified environment. This approach enables system-level, topology-aware analyses that are not supported by existing methods. To apply graph theory in practice, several software tools have been developed for graph construction and analysis.

2.2.3 Available tools

The main objective of a graph software is to enable visual exploration of structures, patterns, and interactions. Multiple software tools are available for these tasks. Among the most widely used options are NetworkX, Gephi, and Neo4j [2, 16, 19]. The first two are open-source, whereas Neo4j operates under a commercial license. Additional tools identified through documentation review include Cytoscape, Graphviz, and iGraph [29–34].

Across these platforms, some of their functionalities include the display of basic metrics such as number of nodes and edges, graph diameter, and graph type.

Additionally, some of them support evaluation of shortest paths, neighborhood relationships, degrees, and reachability. Lastly, only a few of them provide the ability to associate attributes or metadata with nodes and edges. A summary of the tools and their distinctive features is presented in Table 2.1. To reduce redundancy, only features that meaningfully differentiate the software are included.

Table 2.1: Summary of software for graph creation and analysis.

Tool	**Main strengths**	**Data types**
NetworkX	Comprehensive analysis tools; highly flexible attribute storage; extensive Python integration	Numbers, strings, lists, dictionaries, arrays, vectors
Gephi	Strong interactive visualization; suitable for exploratory analysis of large networks	Numbers, strings
Neo4j	Designed for large-scale graph databases; efficient querying using Cypher language	Numbers, strings, lists, vectors
Cytoscape	Focused on biological and molecular networks; strong plugin ecosystem	Numbers, strings
Graphviz	High-quality static layout and rendering tools	Numbers, strings
iGraph	Fast and memory-efficient; strong support for large-scale graph analysis	Numbers, strings

Although many of these tools provide robust visualization and descriptive capabilities, most are primarily oriented toward applications such as biological interaction networks,

drug–target relationships, social networks, or large scale data structures [2, 16, 19, 21], however, a common limitation among them is the reduced options for data storage at the nodes.

Experiments in microfluidics generate diverse data types (eg. images, vectors, arrays, text, etc.), therefore, beyond structural representation, a suitable tool must support heterogeneous data integration at the node level. Additionally, the selected tool must perform customizable queries, and integrate with Python for the purposes of this thesis regarding network connectivity and integration. Among the reviewed tools, NetworkX (version 3.2.1) provides the most flexibility for handling heterogeneous data and integrating with Python workflows.

It is also worth noting that, aside from the first three software options discussed and the examples presented in [2, 19], the use of graph-based representations in microfluidics, particularly for modelling microfluidic networks, is technically not found as the current approaches focus on a macro representation of the circulatory system or models of vascular networks not present in microfluidic chips. This reveals a clear gap and an opportunity to expand the use of graph-based methods in microfluidic research. While graph-based tools support structural analysis, microcirculatory research also relies heavily on image-derived measurements. One of the most important of these is the characterization of the CFL.

2.3 Cell free layer analysis

CFL formation is a fundamental feature of blood flow in vivo, where it contributes to reducing hydrodynamic resistance. While CFL formation has been extensively studied in individual channels [4, 7–10], its behaviour within interconnected microfluidic networks remains underexplored, particularly in relation to network topology and flow redistribution. The CFL arises from RBC migration toward the flow centerline, creating a near-wall depleted region [4]. It has been studied in the context of various pathologies like malaria, Bernard–Soulier syndrome, and Glanzmann thrombasthenia [5].

Beyond its physiological relevance, quantifying the width of the CFL has been a long-standing interest because of its role in regulating microcirculatory function and other biomedical applications, including plasma separation, gas transport mechanisms, vascular homeostasis and microfluidic diagnostic platforms [7, 8]. A representation of the CFL is depicted on Figure 2.5 below.

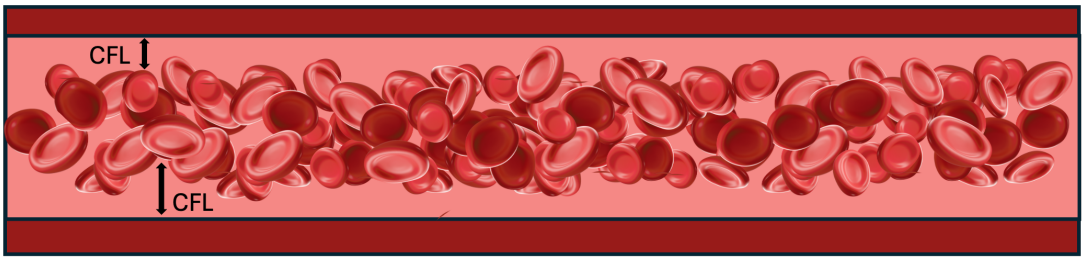

Figure 2.5: Representation of the CFL inside a vessel showcasing RBCs migration towards the flow centerline. The zone with no RBCs is pointed by arrows for both the upper and lower sections.

2.3.1 Current analysis methods

While there have been some studies focusing on CFL measurement using *in vivo* approaches [8, 9]. Microfluidic devices have been helping to analyze the CFL either by studying it through manual single channel analysis [4, 10] or by using semi automatic methods [1, 7] in a less invasive way as other methods involve *in vivo* measurements which also limit the analyses performed [6]. In all the cases, the task is the same: to segment the core full of RBCs to then compare it with the distance from the vessel's walls and determine the width of the CFL.

Besides the work presented in [7] where they presented a semi automated approach, other methods to segment the CFL still rely on handcrafted solutions [4, 8–10] that need to be customized case by case and deal with large datasets that can lead to inter and intra user errors. Some examples of the manual methods include grayscale analysis [4], image thresholding paired with blue filters during image acquisition [8], edge detection [9] and image stacking followed by statistical analysis and histogram analysis [10]. The results obtained in all of them are satisfactory, but their process can be improved and homogenized by creating a fully automated process to facilitate this task by using new segmentation tools, like those based on convolutional neural networks (CNNs), that have shown promising performance in other biomedical applications.

2.3.2 CNNs Based methods

CNNs have become a dominant approach for medical image segmentation and they have outperformed the state of the art in many visual recognition tasks, however, their success is often limited due to the size of the available training sets and the size of the considered networks [35]. CNNs are a class of feedforward neural networks inspired by the human visual system and a subgroup of Artificial Intelligence under machine learning techniques. They are designed to automatically extract spatial hierarchies of features from data through the use of convolutional operations [36, 37]. A CNN typically consists of three main types of layers: convolutional, pooling, and fully connected. Each contributing to feature learning and classification.

The *convolutional layer* serves as the feature extractor of the network. It applies a set of kernels (also referred to as filters) that act as receptive fields, responding to various spatial features such as edges, textures, or gradients. Through successive convolutions, the network learns increasingly abstract representations of the input data [36]. Since the convolution operation can reduce the spatial dimensions of the input, *zero-padding* is often employed to preserve the original dimensions and minimize information loss.

Following feature extraction, a *pooling layer* performs downsampling to reduce the dimensionality of feature maps. This step lowers computational complexity and mitigates the risk of overfitting by retaining only the most salient features [36, 37]. Common pooling strategies include max pooling and average pooling, which summarize the activations within local neighborhoods.

Finally, the *fully connected layer* integrates the learned features to perform high-level reasoning and classification. Each neuron in this layer connects to all outputs of the preceding layer, aggregating the extracted information into a final representation. The processed outputs are then passed through an activation function to introduce nonlinearity and enable the network to model complex relationships [36, 37].

A powerful CNN, U-Net (or UNet) is commonly used for image segmentation tasks because of its good performance and efficient use of GPU memory [38]. A U-Net operates fully with CNNs, but its layers are not fully connected. To predict the pixels in the border region of the image, the missing context is extrapolated by mirroring the input image. This tiling strategy is important to apply the network to large images, since otherwise the resolution would be limited by the GPU memory.

This process of skipping connections have proved effective in recovering fine-grained details of the target objects; generating segmentation masks with fine details even on complex background [39]. Traditional U-Net treats all features equally during feature fusion, failing to emphasize the importance of key features, which may result in interference from irrelevant features. To address this issue, Attention U-Net introduced an attention gate (AG) into U-Net's skip connection part, allowing it to adaptively adjust the weight of feature maps [37].

AGs were used commonly for natural image analysis, knowledge graphs, and language processing for image captioning, machine translation, and classification tasks [37, 38]. Trainable attention can be divided as hard and soft attention with the first one relying on

reinforcement learning for parameter updates and it is often non differentiable. Conversely, soft attention is probabilistic and uses back propagation without need of sampling. Figure 2.6 illustrates the performed operations inside an AG.

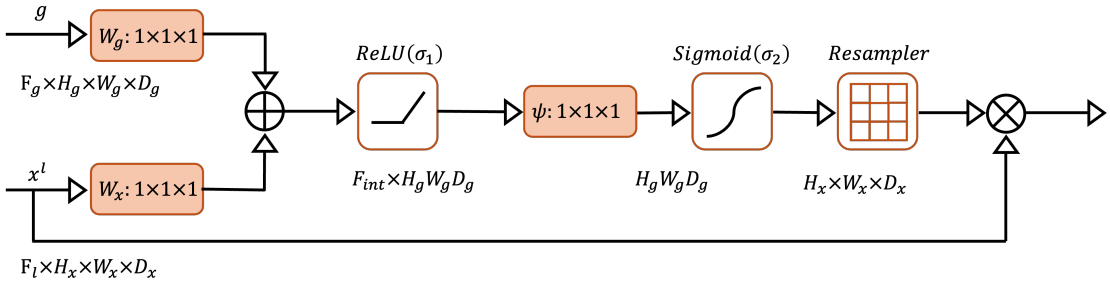

Figure 2.6: Flow diagram of an AG inspired from [38]. Representing the two feature vectors g and x entering into the AG and then being summed element wise before getting flat by a rectified linear unit (ReLU) function. After getting flattened the result gets evaluated using a sigmoid function and lastly the result get scaled to the original dimensions of x.

Models with AGs can be trained from scratch in a standard way similar to the training of a Fully Convolutional Network (FCN) model, and AGs automatically learn to focus on target structures without additional supervision. These gates generate soft region proposals implicitly on-the-fly and highlight salient features useful for a specific task. Figure 2.6 illustrates this process by showing how two feature vectors g and x, coming from an initial layer and another one level below, x and g respectively, are summed element wise. This sum results in the growth of aligned weights and a reduction of non aligned weights. Then, a rectified linear unit (ReLU) function flattens it before being evaluated using a sigmoid function to scale those values in a range between 0 and 1 (representing the attention coefficients) and lastly, those coefficients are re-scaled to the same dimensions of the original vector x [38].

Its application do not introduce significant computational overhead and do not require a large number of model parameters, thanks to the aforementioned attention coefficients, AGs progressively suppress feature responses in irrelevant background regions without the requirement to crop a region of interest (ROI) between networks. To the best of my knowledge, limited work has explored a machine learning approach, the implementation of an U-NET Network with an AG pose a powerful tool to identify the RBCs core and therefore characterize the CFL as it does not require a large dataset and its versatility to extract diverse features, paying special interest in the most essential ones thanks to the AGs.

As in other areas of medical imaging, segmentation performance must be rigorously evaluated to assess the overall effectiveness of a model. The most commonly used metrics in medical image segmentation for tasks similar to the ones in this project, according to the findings in [37], are summarized below:

1. **Intersection over Union (IoU):** IoU measures the degree of overlap between the predicted segmentation and the ground truth. It ranges from 0 to 1 and incorporates false positives (FP), false negatives (FN) and true positives (TP). IoU is widely used in both object detection and segmentation, including organ and lesion delineation in medical imaging.

$$IoU = \frac{TP}{TP + FP + FN} \tag{2.1}$$

2. **Dice Coefficient:** The Dice coefficient is another similarity measure ranging from 0 to 1 and is closely related to IoU but more sensitive to TP. It is particularly robust

to class imbalance, making it highly suitable for medical image segmentation, where the target region often occupies only a small portion of the image.

$$Dice = \frac{2TP}{2TP + FP + FN} \tag{2.2}$$

Both metrics align with the desired robustness when evaluating the effectiveness of a segmentation task, mostly when the segmentation is relatively smaller than its background. Although IoU offers a good starting point to prioritize the segmentation in medical imaging, the Dice Coefficient is more sensitive for cases where the background is predominant as the case of the RBCs core compared with the rest of the microfluidic channel.

2.4 Discussion

Although existing segmentation approaches to segment the RBCs core generate good results, their variability depending on the user, method chosen and the high volume of images to analyze make the implementation of an Attention U-Net, a promising approach to segment CFL in a faster and fully automated way.

Additionally, while existing graph tools offer robust visualization and metric computation, they are not designed to integrate heterogeneous experimental data usually found during microfluidic experiments such as images, time series, or spatial measurements at the node level. Moreover, their lack of native support for microfluidic workflows limits their direct applicability. The present work addresses these limitations by developing a graph-

based framework that integrates experimental measurements, image-derived variables, and network topology within a unified analytical environment.

In summary, the literature review reveals three persistent gaps: **i.** A lack of topology-aware analysis tools for microfluidic networks, **ii.** the absence of fully automated CFL segmentation pipelines robust to overcome experimental variability, and **iii.** the fragmentation of analytical workflows across diverse software tools. Together, these limitations reveal a lack of unified, topology-aware analytical frameworks. Chapter 3 introduces such a framework by formalizing a graph representation of a retinal-inspired microfluidic network.

Chapter 3

Graph model

This chapter presents the methodological framework developed to transform a biomimetic microfluidic network into a graph-based representation. This chapter details that process, including vessel labeling, graph construction, data integration, and the definition of analysis modes enabled by the resulting representation.

3.1 Materials and methods

3.1.1 Labeling of the retina chip

As described earlier in Chapter 2, the retina chip consists of a series of consecutive bifurcations resembling the arterial tree of the human retina. Starting from a single inlet, the network splits progressively into smaller branches until the outlets are reached [1]. Thanks to this topology and the usefulness of numerical identifiers for queries, a binary labeling system was adopted, inspired by the approach in [19] as it provides a compact, mathematically tractable representation of hierarchical branching and enables efficient comparisons

including generation-based analyses. The root vessel (inlet) was assigned the binary value `0`. At every bifurcation, a digit was appended to the label depending on the branch taken: a `0` for the port (left) branch and a `1` for the starboard (right) branch. This labeling scheme is not only a naming convention but a structural encoding of the network topology, enabling efficient traversal, hierarchical analysis, and direct mapping between physical vessels and graph nodes.

For example, a parent vessel labeled `1010` will generate daughters `10100` (left) and `10101` (right), which correspond to decimal identifiers 20 and 21 respectively. Figure 3.1 illustrates this system that works well for tree-like networks but would require adaptation for networks containing loops or anastomoses.

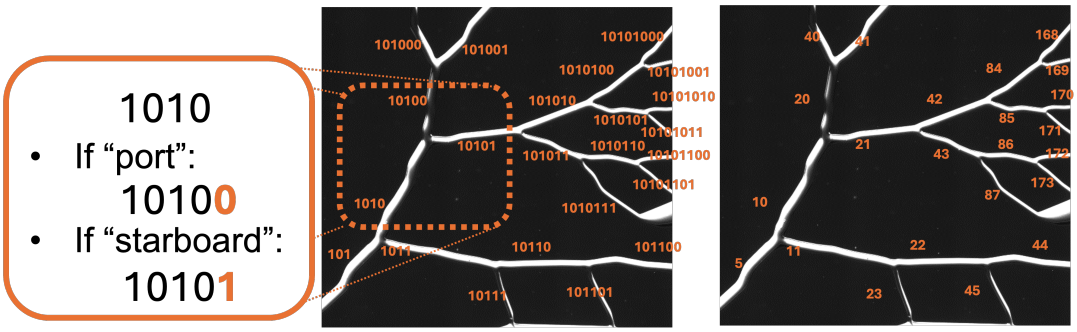

Figure 3.1: Example of the proposed labeling system applied at each bifurcation. (Left) Appending process depending on the direction of the bifurcation for a vessel. (Center) binary representation of the labels. (Right) Labels in decimal numbers.

Once the identifiers were generated, they were manually written onto the stitched image. Then the full chip was labeled as depicted on Figure 3.2. Spanning 191 labels with 4 uninterrupted generations from the inlet (vessels 1-31) and having a maximum of 9 bi-

furcations along one path (vessels 712 and 713) starting from the inlet all the way to the middle far right part of the chip

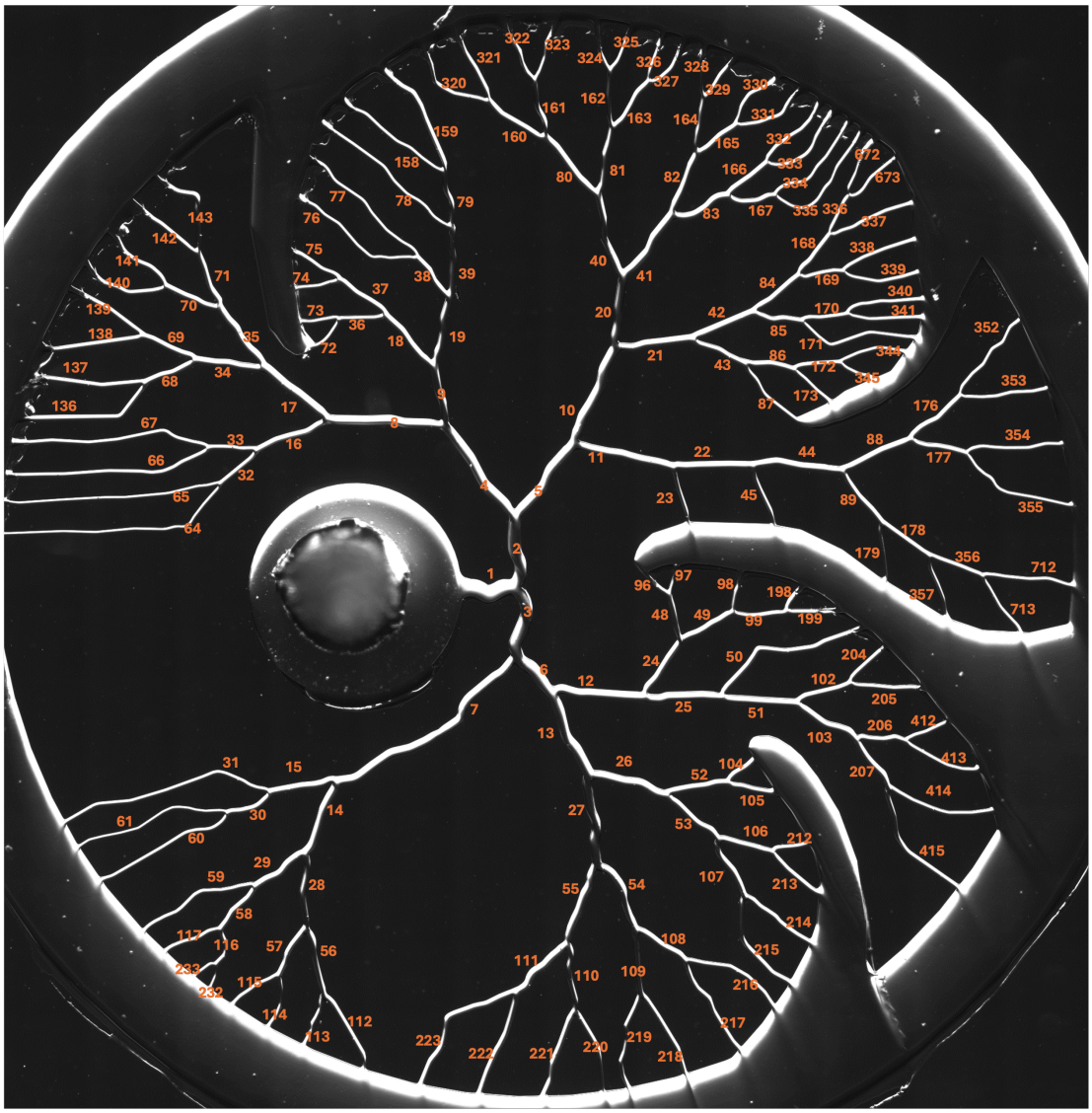

Figure 3.2: Final stitched map of the chip with all vessel labels superimposed. Label 0 and 1024 correspond to the inlet (circle in the middle of the image) and the outlet (corona where all the vessels are connected to) respectively.

3.1.2 Graph creation

After having all the labels set for each vessel, the graph can be built. Its design relies on a couple of assumptions reflecting the design of the retina chip: **i.** vessels bifurcate strictly in a dichotomous manner, **ii.** the network is fully connected between a single inlet and a single outlet, and **iii.** flow direction is assumed to be known and consistent across experiments. Additionally, in this work each node represents a complete vessel rather than a junction, while edges encode parent-daughter relationships between vessels. This choice reflects the experimental nature of the data, where measurements (e.g., flow, CFL thickness, velocity) are associated with vessel segments rather than discrete junction points. This representation simplifies data integration and enables direct mapping between experimental observations and graph elements.

The selected tool to create the graph was NetworkX, thanks to its open source nature, flexibility for attribute storage (required for the diverse experimental data), and integration with Python scripts. The construction process began with generating a complete binary tree of nine generations (based on the highest vessel number), which served as a flexible structural scaffold as shown in section A of Figure 3.3. Subsequently, two trimming functions were implemented to selectively remove irrelevant branches and retain only the nodes and pathways present in the physical retina chip. These functions first deleted all successors of a selected node and their corresponding edges the trimmed result is presented in section B of Figure 3.3.

To improve interpretability, a custom layout algorithm was developed to organize the

graph by generations, improving visual separation and interpretability. By leveraging the hierarchical structure encoded in the labeling system, an additional sorting algorithm was developed. Nodes are positioned along the horizontal axis based on their topological generation, while vertical spacing is adjusted to evenly distribute nodes within each generation. Lastly, to further improve visualization, some layout parameters are added with assistance of a helper function, the result of this sorting algorithm is presented in section C of Figure 3.3.

Finally, as all the vessels are connected to a single outlet in the real retina chip, all terminal nodes are detected and linked to a single outlet node. A custom algorithm identified nodes without successors, organize them into an array and linked them to the outlet, returning a complete graph connecting the inlet and outlet with all the spanning nodes representing the vessels of the retina chip. Figure 3.3 illustrates the evolution of the graph through these stages, incorporating an intermediate layout step to emphasize structural clarity prior to final connectivity. All helper functions involved in the graph creation are shown in Appendix A.1 .

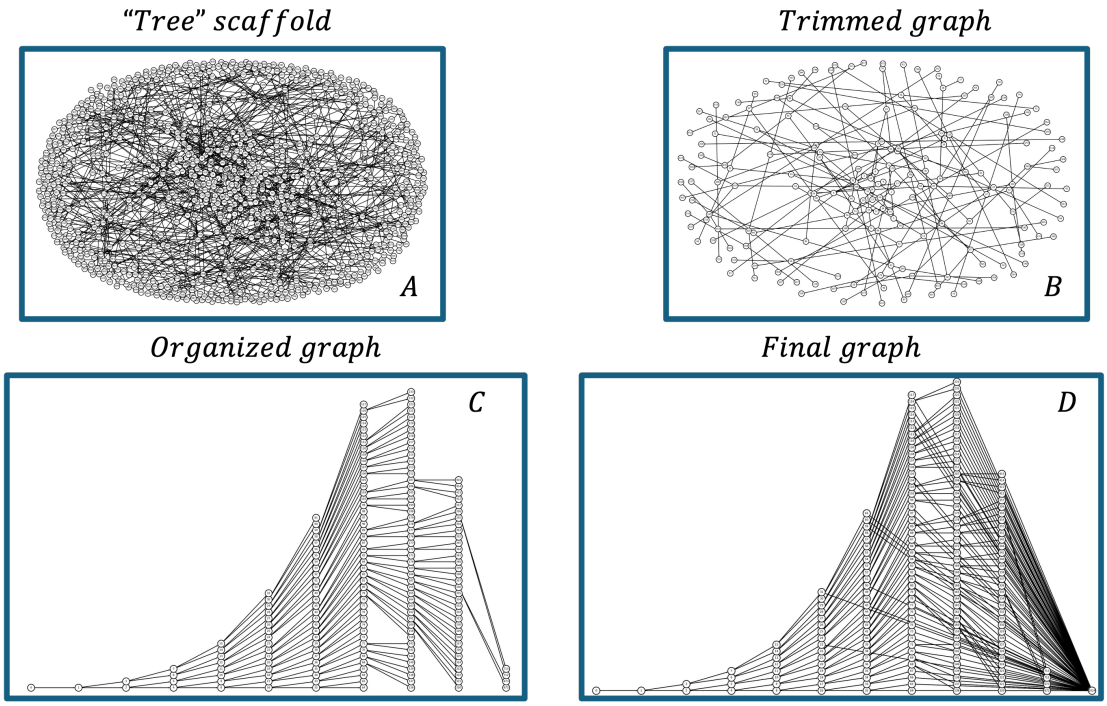

Figure 3.3: Graph construction pipeline showing intermediate stages, including scaffold generation, trimming, generational organization, and final outlet connectivity. Left to right: (A) complete scaffold (binary tree), (B) trimmed graph matching physical chip, (C) organized graph prior to outlet linking, and (D) final directed graph with outlet connections.

3.1.3 Storing and accessing information

Information storage in the graph is performed using the attribute system of NetworkX. A variable can be assigned to any node using the syntax shown in Figure 3.4 . Node attributes are flexible and can include any hashable object e.g., a text string, an image, an XML object, another Graph, and a customized node object [29]. Accessing the stored information can be done by requesting either the entire dictionary of attributes of a node,

for example: `G.nodes[node]`, or a specific value using `G.nodes[node]["Variable"]`.

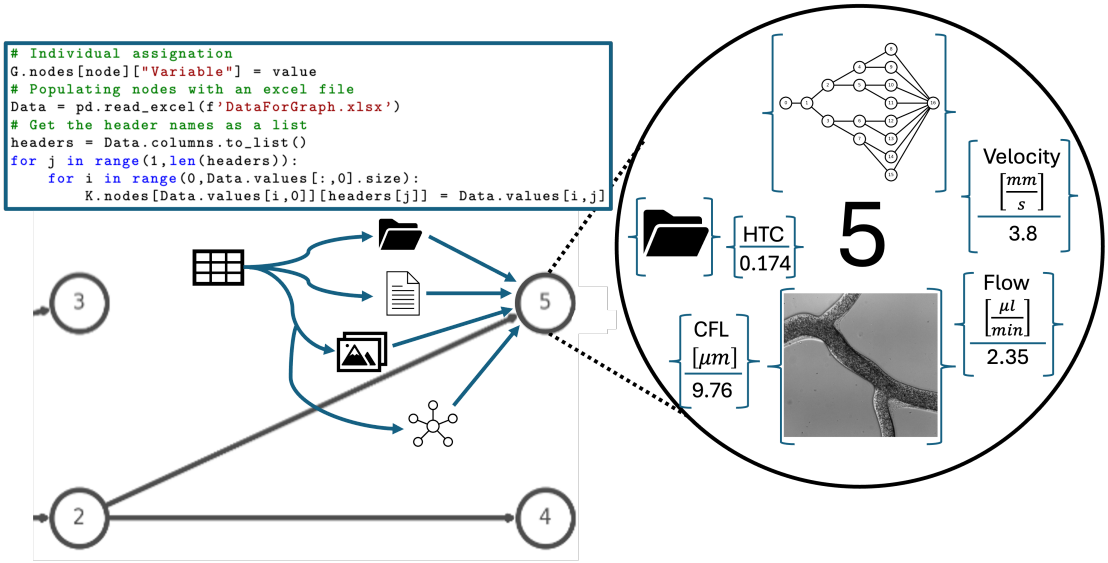

Figure 3.4: Visual representation of data assignation to a node coming from an excel file assigning different data (Velocity, HTC, CFL thickness, flow, and a folder path) to the node number 5. Similarly, it shows the other data types it can store as graphs or text documents.

3.1.4 Implemented graph analyses

The combination of flexibility of data for node attributes and graph connectivity enables some analysis modes. A **progressive analysis** identifies all descendants of a selected node. This analysis enables investigation in conservative systems where a quantity at the parent vessel is expected to correspond to the sum or distribution of its downstream components.

Conversely, a **regressive analysis** traces the path from a given node back to its most distant ancestor. This analysis enables investigation of how upstream conditions influence local measurements, supporting hypothesis testing about flow dependencies.

Finally, a **generational** analysis groups vessels according to their topological depth. This analysis allows systematic comparison of microcirculatory behaviour across hierarchical vessel levels.

Besides those analyses, NetworkX provides a powerful feature to transform the current graph into an **adjacency matrix** which is a square matrix with order equal to the number of nodes in the graph. It assigns a value of 1 if there's an edge connecting two nodes and 0 otherwise. For example, in the graph segment in Figure 3.4, the node 2 is connected with nodes 4 and 5 so the element (2,4) and (2,5) in the matrix will assign 1 since there's an edge connecting them and it is a directed graph. When dealing with an undirected graph, the resulting matrix will be symmetrical having then the elements (4,2) and (5,2) assigned with 1 instead of a 0.

3.2 Results

The resulting graph preserves the hierarchical and fully connected structure of the physical microfluidic network, ensuring consistency between the abstract representation and the experimental system. The final directed graph contains 193 nodes and 287 edges and it is illustrated in Figure 3.3 D. With the exception of the root and outlet nodes, each vessel has exactly one parent and two daughters. The root node is connected only to its first branch, whereas the outlet receives connections from all terminal vessels (96 in total).

Using data from [1] showcasing blood flow, a subgraph is created from the main graph

to represent the vessels with information and exclude those no depicted on it. Both the subgraph and the blood flow graph are used to illustrate the different analysis modes. Figure 3.5 shows a case where a progressive analysis is performed in the subgraph resulting in a tree graph. Those nodes that were not used remained uncolored while the resulting nodes for the analysis are highlighted on green in both the subgraph and the blood flow graph.

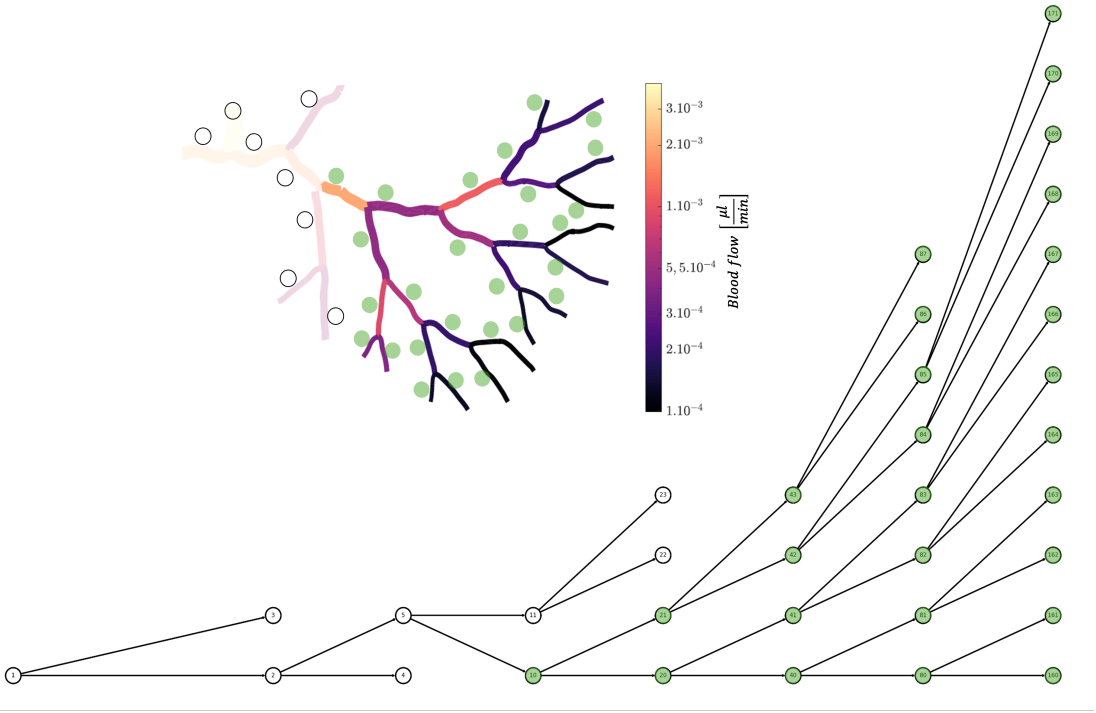

Figure 3.5: Example of a progressive analysis corresponding to vessels depicted on the graph above representing blood flow in the retina chip; this subgraph highlights on green those nodes containing according to the graph for the blood flow starting at node 10. Nodes without info can be seen next to their vessels on the faded portion of the original graph.

Figure 3.6 illustrated a regressive analysis where the main focus was to trace back the

blood flow from node 22 up to the origin.

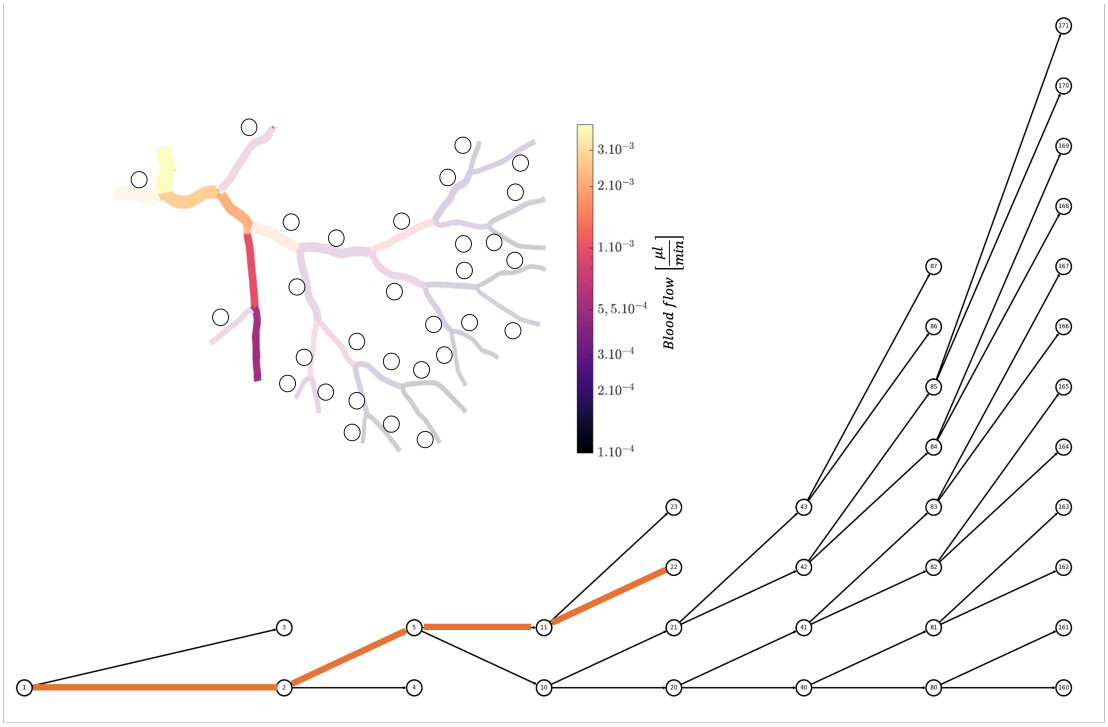

Figure 3.6: Example of a regressive analysis showing up the matching blood flow graph. The starting node, node 22 is traced back to its origin at node 1, the path is highlighted on the figure with the orange lines on the edges. Nodes without info can be seen next to their vessels on the faded portion of the original graph.

The example is shown in Figure 3.7 shows the result for a generational analysis. It is worth mentioning that there are cases where a generational analysis can represent the same nodes as a progressive. The key difference between the graph in Figure 3.5 and 3.7 is the branch present for the 4th and 5th generation depicted on blue and red respectively below.

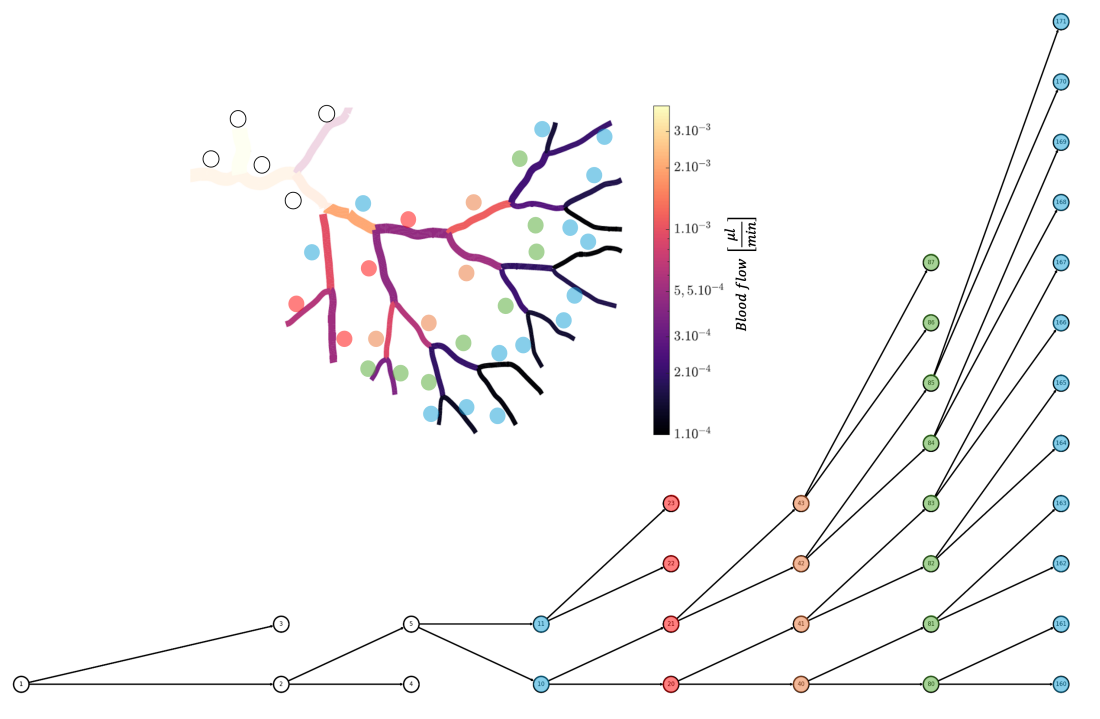

Figure 3.7: Example of a generational analysis showcasing the different generations for blood flow in the retina chip starting at the 4th generation and ending at the 8th. Each generation is depicted with a different colour for its representation, repeating only blue for the 8th generation. Nodes without info can be seen next to their vessels on the faded portion of the original graph.

Additionally, as the generated adjacency matrix for the obtained graph with 193 nodes would result in a 193x193 matrix, the results cannot be appreciated due to scaling problems. To make it simpler, a subgraph of the first portion of the graph is presented in Figure 3.8 along with the matrix representation as an adjacency matrix.

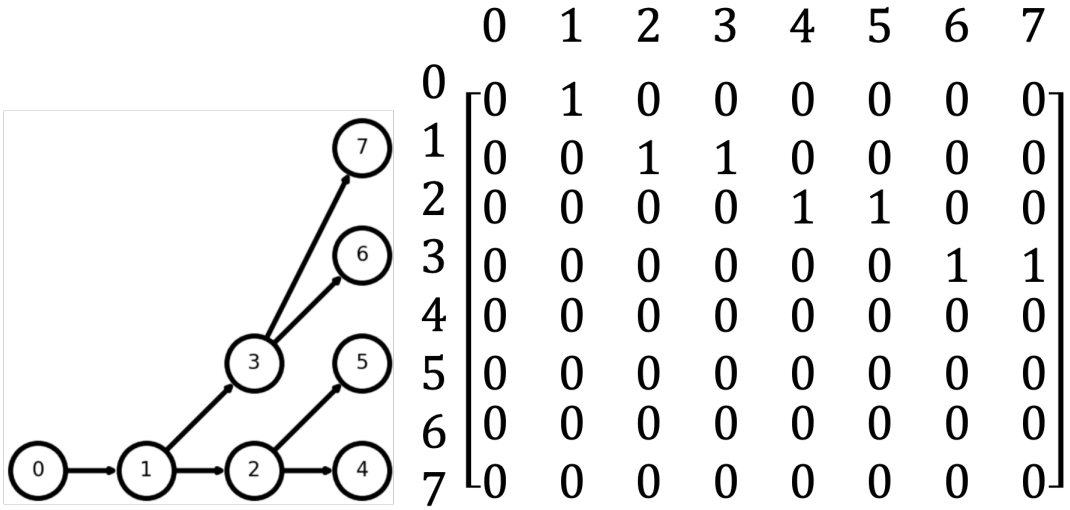

Figure 3.8: Example of an adjacency matrix (8x8) from the first 8 nodes of the directed graph obtained in this work.

3.3 Discussion

The directed graph developed in this chapter successfully captures the topological characteristics of the physical retina chip, enabling the storage from diverse data types allowing each node to operate as an independent container of heterogeneous microfluidic data, and analyses like connectivity with descendants (progressive), path tracing (regressive) and generational. Together, progressive, regressive, and generational analyses provide complementary perspectives on flow propagation, upstream dependency, and hierarchical organization within the microfluidic network. Additionally, the adjacency matrix offer a method to solve equation systems as information within the nodes can be treated as known

variables.

Despite these advantages, one limitation identified is the absence of an image based graph construction algorithm. Current graph modelling tools require manual or script based definition of nodes and edges, but do not provide direct, intuitive conversion from a single image of a microfluidic network into a fully queryable graph object. Given that only a single complete network geometry was available, prioritizing manual, controlled abstraction over generalized image-to-graph automation reduced the risk of introducing unvalidated geometric assumptions and extended considerably the time frame in which this project was done. Under these constraints, manual labeling and graph refinement offered a faster and more reliable path for representing the chip topology, while minimizing risks of introducing unverified geometric bias into the graph structure.

Additionally, the limited number, to the best of our knowledge, of published implementations of graph-based models for microfluidic networks highlights a research opportunity. Most existing graph studies in biomedical engineering focus on domains such as neural connections, genetic signatures involved in metabolism [15], and drug interaction networks [23], while comparable applications in microfluidic systems are rarely present or related to other aspects as fabrication [40]. This scarcity in the literature suggests a methodological gap that this thesis begins to address. The results presented here indicate that graph theory can provide actionable structural insight for microfluidic network analysis, particularly when combined with systematic vessel identifiers and database style attribute propagation.

Overall, this chapter demonstrates that microfluidic networks can be systematically

transformed into graph-based representations that preserve both topology and experimental relevance. By enabling structured data integration and topology-aware analyses, this framework addresses key limitations identified in Chapter 1 and Chapter 2, particularly the lack of tools capable of analyzing microcirculatory systems as interconnected networks.

These findings also informed later developments in this thesis, where an interactive user interface to assist research in microfluidic networks is presented in Chapter 4, expanding the practicality and accessibility of this methodological approach. Through these contributions, this work moves toward bridging graph-based modelling with biomimetic microfluidic networks, while encouraging future adaptation by other research groups working with similar platforms. This graph model serves as the backbone for the interactive analysis environment presented in Chapter 5, where experimental data, visualization, and network-based queries are integrated within a single user-oriented framework.

Chapter 4

Cell free layer segmentation

This chapter develops a machine learning–based framework for segmenting the RBCs core as a prerequisite for estimating the CFL thickness. Because CFL estimation depends on boundary localization rather than region coverage alone, segmentation must be both robust to experimental variability and precise at the vessel interfaces. To address this requirement, an Attention U-Net architecture is implemented and evaluated. The chapter first defines the dataset and preprocessing steps required to standardize heterogeneous experimental conditions, then presents the model design and training strategy, and finally evaluates performance using both conventional segmentation metrics and boundary-based measurements aligned with the physiological objective.

4.1 Materials and Methods

After evaluating different segmentation approaches, discussed in Chapter 2, an Attention U-Net architecture was selected due to its ability to perform accurate segmentation with limited training data and its flexibility for both single-class and multi-class tasks. In CFL

segmentation, the region of interest occupies only a small fraction of the image, while background structures and illumination artifacts dominate the field of view. Attention gates help suppress irrelevant spatial features and emphasize the RBCs core boundaries, improving robustness to contrast variability.

To train the attention U-Net, two primary types of data are required: images and ground truth masks segmenting the intended object/region. The images were obtained from multiple experimental setups of raw channel images acquired using a high speed camera. For the ground truth masks, each of those experiments had their corresponding binary masks delineating the spatial extent of the RBC core, with foreground pixels corresponding to the core region and background pixels corresponding to everything else. The description of the complete dataset used, as well as the steps to obtain the ground truth masks are explained in the next subsections.

4.1.1 Dataset

The image dataset comprises 35 experimental groups. Each of those experiments was conducted within the laboratory for two different channel types, straight channels [41–44] and vessels from the retina chip [1]. The first 18 groups correspond to retina-chip experiments, with an average of 40 images per group, while the remaining 17 groups correspond to straight channel experiments approximately 200 images per group. In total, 4194 images were organized into 35 groups. Images within each group share similar experimental conditions, lighting, acquisition parameters and sizes. Representative examples of the image

dataset are shown in Figure 4.1. A deeper overview of the entire dataset as well as the complete details for the dataset is presented in Appendix A.2.

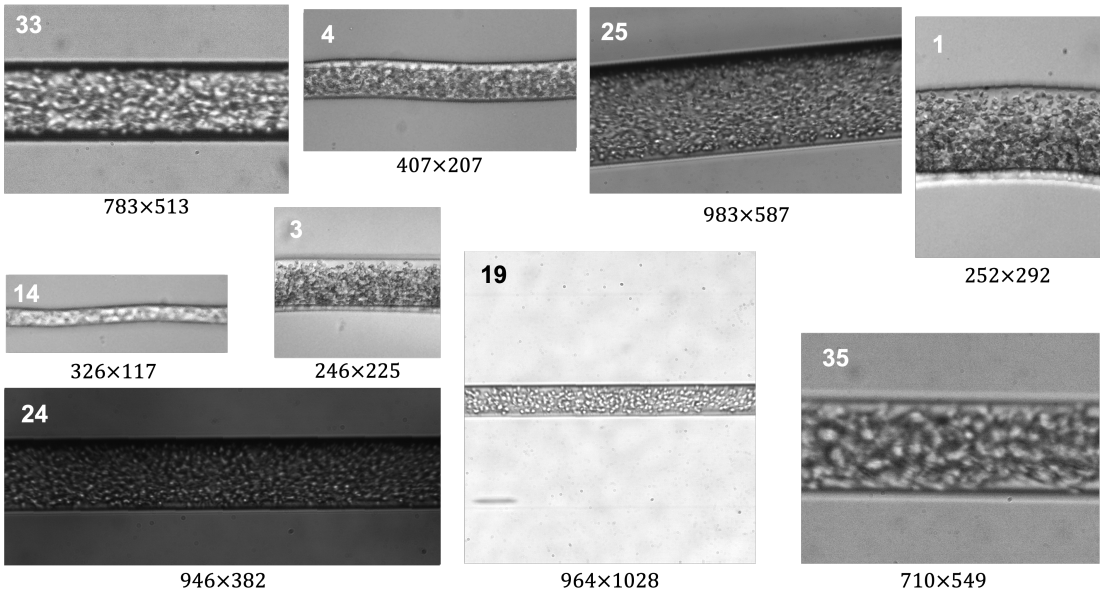

Figure 4.1: Representative images from 9 different groups of the dataset used for CFL segmentation, illustrating variations in size, lighting and acquisition conditions. The group where they came from is indicated on each image.

The dataset exhibits noticeable variability in image size, geometry, orientation, and experimental conditions per group, with a range for hematocrit between 5 to 20 %, diameters of 25 to 50 μm for straight channels and 30 to 180 μm for the retina chip channels, other conditions like illumination and frames per second were different having a maximum of 1000 frames per second for their acquisition. Based on these variations, four main acquisition scenarios were identified: bright cells on a bright background, bright cells on a dark background, dark cells on a bright background, and dark cells on a dark background.

These variations introduce significant challenges for model generalization. As a result, they motivate the use of preprocessing choices specifically designed to improve robustness to heterogeneous imaging conditions.

Additionally, the original file naming convention was based on experimental metadata and was not well suited for automated processing. A preprocessing pipeline was implemented to rename all images using a consistent and user-friendly convention. The original masks were generated using the semi automated method (SAM) described in [7]. As many of those masks extended beyond the RBCs core boundaries, an additional enhancing stage was applied to improve mask quality prior model training.

4.1.2 Dataset preparation

The quality of ground truth masks is a critical factor in supervised segmentation, particularly for tasks where boundary accuracy directly affects downstream measurements. In this dataset, the available masks were generated using the SAM [7]. While these masks reliably identify the region of interest (ROI), they tend to overestimate the extent of the RBC core by including surrounding structures and intensity artifacts. As a result, they are not directly suitable for training a model aimed at precise boundary localization.

To address this limitation, a post-processing strategy was developed to refine the masks by combining complementary sources of information. The key insight is that gradient images provide strong localization of RBC boundaries due to their sensitivity to intensity transitions, whereas the original SAM masks provide robust spatial constraints for the

ROI. By integrating these two representations, it becomes possible to obtain masks that are both spatially consistent and boundary-accurate.

The refinement process proceeds as follows. First, gradient images derived from the original data are thresholded to produce a binary representation (G1) that emphasizes high-frequency boundary information. In parallel, an average mask is computed across each experimental group and binarized (A1), then slightly dilated (A2) to ensure full coverage of the ROI. The intersection of G1 and A2 isolates boundary-relevant structures within the valid spatial region.

Because this intermediate result may still include fragmented or noisy regions, morphological erosion is applied to remove spurious structures and improve boundary coherence. The resulting binary image is then inverted to match the desired foreground representation. Finally, this refined boundary mask is combined with the original ROI mask (A1), producing a final ground truth mask that preserves both accurate boundaries and consistent spatial coverage. Figure 4.2 describes the process described above while Appendix A.3 dives deeper in the parameters used for the process and shows an example code.

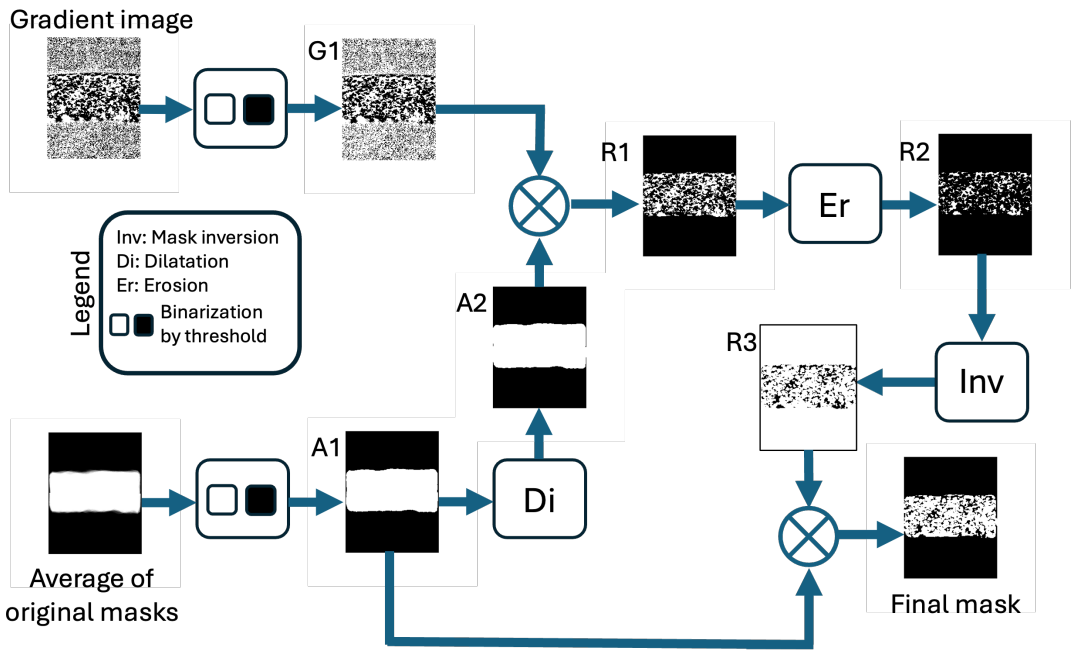

Figure 4.2: Flow chart of the post processing stage to obtain masks delimiting better the RBCs core from the results of the SAM presented in [7]. Labeling the resulting images with their names on the upper left corner.

This process transforms previous overestimating masks into precise boundary-aware masks, improving the quality of supervision provided to the model. As demonstrated in Figure 4.3, the refined masks more closely follow the extent of the RBC core, which is essential for reliable CFL thickness estimation.

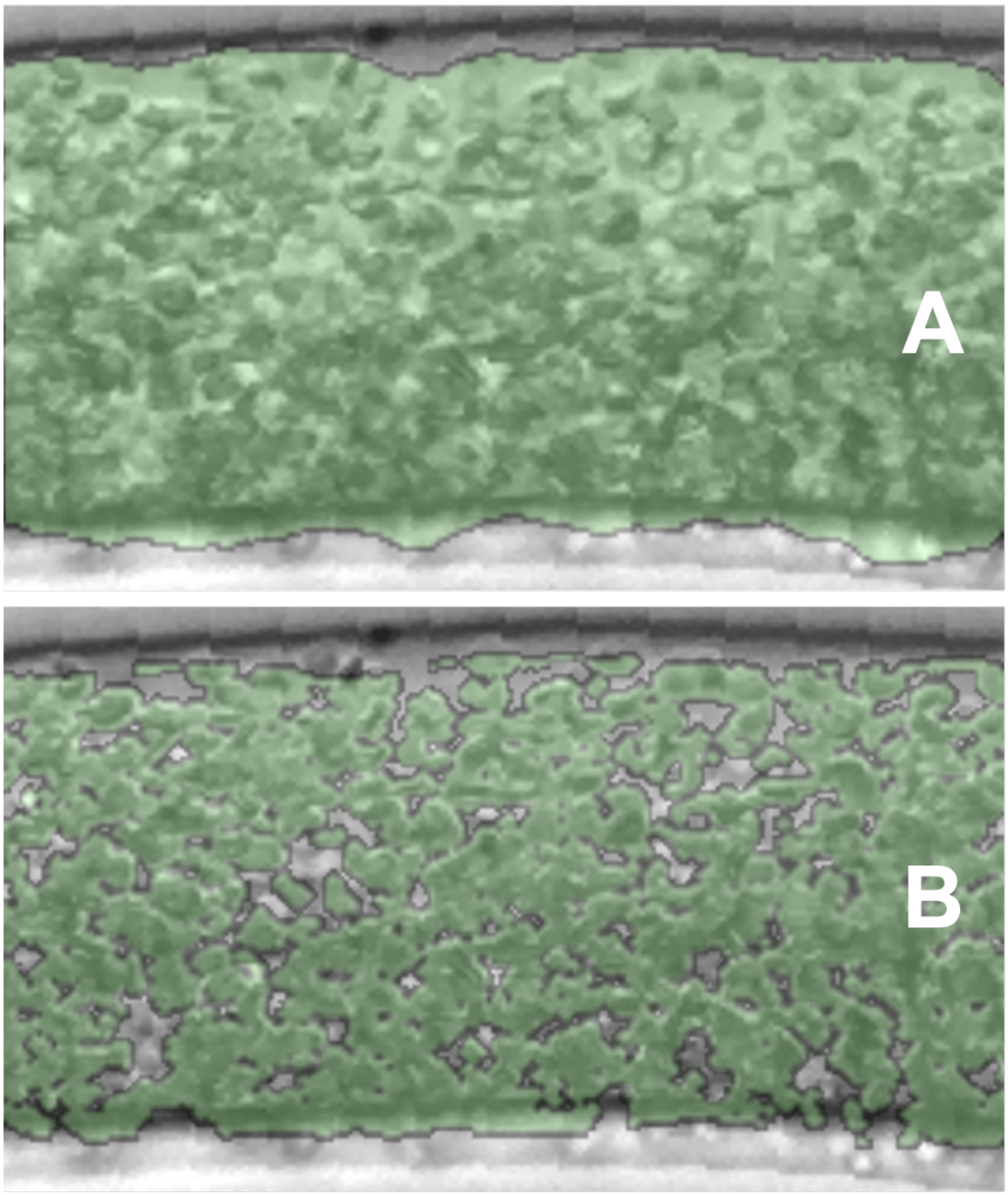

Figure 4.3: A: Original mask generated by the semi-automated method [7] overlaid on its corresponding image. B: Final mask generated by the post processing stage.

Finally, as the different groups contained names with metadata associated, a renaming process was performed to standardize the names and to facilitate batch processing, iteration, and comparison during training and evaluation. For example, an image originally named "Vessel#D75H20_0185" was renamed as "Group0024_0185". Both processes, the renaming and mask enhancement were performed for each group resulting in a dataset with the two required types of data: channel images and their corresponding binary masks segmenting the RBC core. Figure 4.4 describes graphically the process described above.

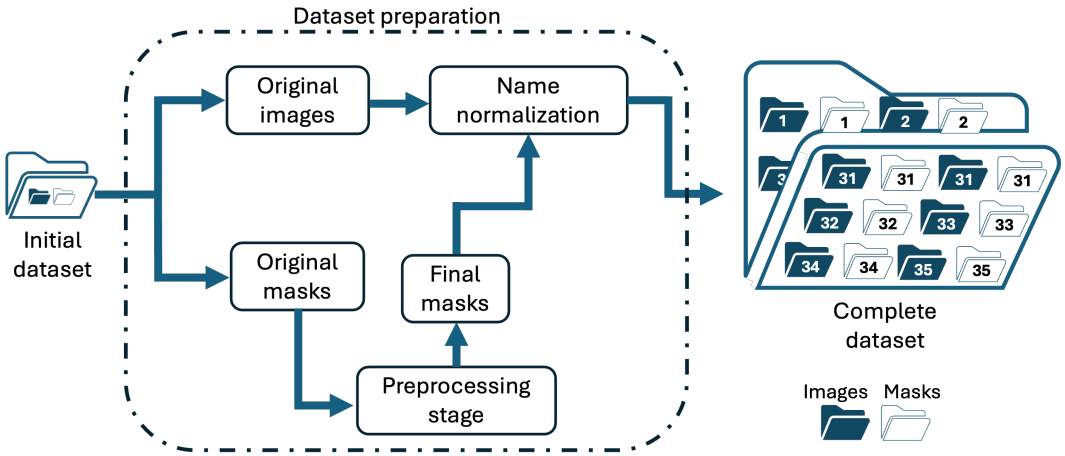

Figure 4.4: Workflow for the data to be integrated into the final dataset and subsequently split to be used to train the model.

4.1.3 Preprocessing input data

Prior to training, various preprocessing operations were applied to reduce variability across experimental conditions while preserving structural features relevant for segmentation, ensuring that the model learns consistent representations across heterogeneous inputs. To

mitigate variability in illumination and contrast from the varying experimental conditions from each group, Contrast Limited Adaptive Histogram Equalization (CLAHE) was applied. This method enhances local contrast by dividing the image into tiles, equalizing each tile independently, and limiting contrast amplification to reduce noise [45]. As depicted in Figure 4.5 , this operation preserves the overall intensity distribution while reducing the dominance of dominant pixel intensities.

To handle variability in image dimensions, two strategies were implemented: zero padding for smaller images and cropping for larger images. To ensure compatibility with the repeated downsampling and upsampling operations, U-Net models generally use input image dimensions that are multiples of 16 or 32, in this case, a fixed input size of 512×512 pixels was selected as a compromise between computational efficiency and spatial resolution since another size would have resulted in requiring more computational resources or losing important features due to resizing the images [37, 38]. This resizing step applied to the input images was identically applied to their corresponding masks to preserve pixel-wise correspondence. Figure 4.5 illustrates the preprocessing workflow for one image.

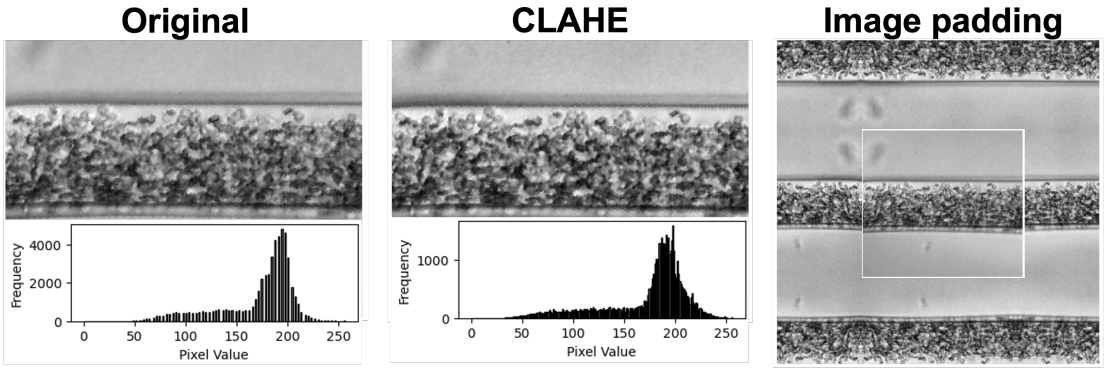

Figure 4.5: Preprocessing steps applied to the dataset: original image (left), CLAHE-enhanced image (middle), and padded image resized to 512×512 pixels (right). The original image is highlighted within the white border.

Lastly, to evaluate generalization across experimental conditions, the dataset was split at the group level rather than randomly at the image level. Approximately 80% of the groups (Groups 1–29) were used for training, while the remaining 20% (Groups 30–35) were reserved for validation. This strategy prevents data leakage between training and validation sets and ensures that the model is tested on unseen experimental conditions, providing a more realistic assessment of its robustness. To facilitate the understanding of the workflow of the datasets before training the model Figure 4.6 summarize it.

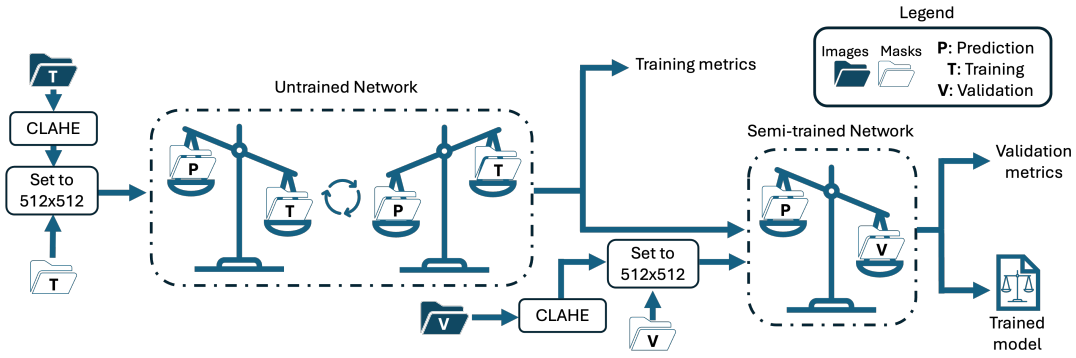

Figure 4.6: Workflow of the training and validation datasets within the training process of the network to obtain the different outputs including the final files containing the best weights and the fully trained model.

4.1.4 Training parameters

The Attention U-Net was implemented in Python using TensorFlow, a software library for machine learning and artificial intelligence. The model performance was evaluated using the Dice coefficient, with a combined Dice and binary cross-entropy loss function. Model training and an initial inference were performed on the Beluga and Rorqual clusters provided by Digital Research Alliance Canada, using a Bash-based job submission workflow. The implementation follows the core principles described in the original work on [38], with additional preprocessing and adaptation steps tailored to the characteristics of the present dataset.

The network was trained for 180 epochs with a batch size of 16. Training was performed using one NVidia H100 SXM5 (80GB), one CPU core AMD EPYC 9654, 4GB in RAM memory and three hours for allocated time for its training were requested from the Rorqual

cluster (Digital Research Alliance of Canada).

4.1.5 Performance evaluation

While overlap-based metrics such as the Dice coefficient provide a measure of segmentation accuracy, they do not directly reflect the accuracy of CFL thickness estimation, which depends on boundary localization. Therefore, an additional evaluation step was introduced based on boundary extraction.

By reducing both predicted and reference masks to their upper and lower boundaries, the average distance between these contours can be computed. This metric directly reflects the accuracy of RBCs core thickness estimation and provides a physiologically meaningful assessment of model performance. An illustration of this additional step is presented in Figure 4.7.

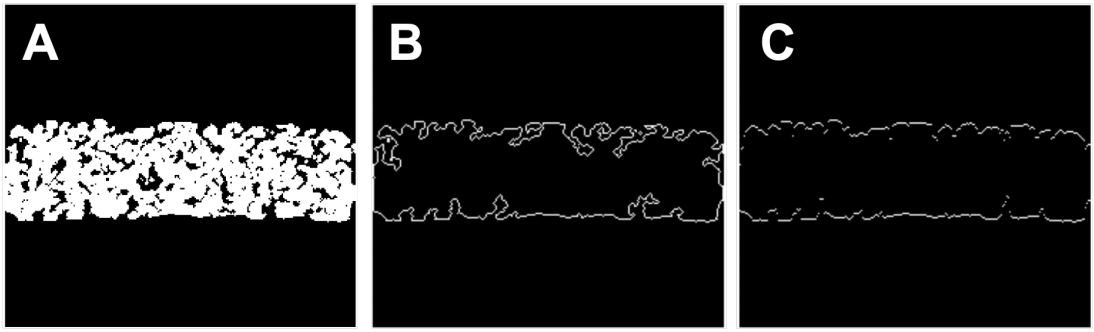

Figure 4.7: Border extraction process from a binary mask. A: input mask. B: mask contour. C: extracted upper and lower boundaries.

4.2 Results

The trained model achieved a Dice coefficient of 97.04% on the training set. Qualitatively, the predicted masks closely matched the reference masks in both width and boundary location. In addition, the extracted borders and their widths for both masks are close in values. Figure 4.8 illustrates this evaluation process for an example from group 3, where the predicted width (60.57 pixels) closely matches the reference mask (61.13 pixels).

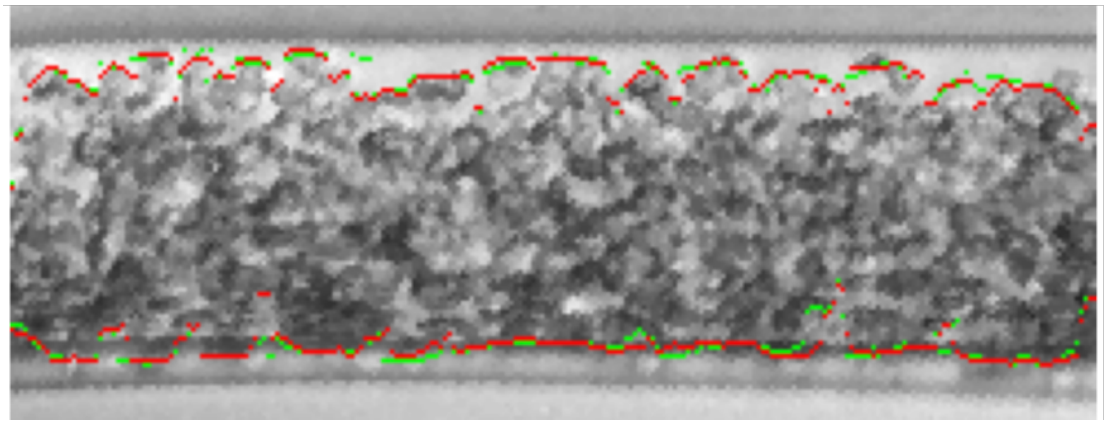

Figure 4.8: Illustration of extracted predicted (red) and reference (green) borders overlaid with the original image.

An inter group analysis was performed to assess model robustness. The Dice coefficient for each group is shown in the top panel of Figure 4.9. Variability in performance is observed across groups, with the lowest Dice coefficients and highest variability corresponding to the last three groups. Following border extraction, the difference in estimated core width between the predicted and reference masks was computed for each group. The

corresponding border-based distance analysis is shown in the bottom panel of Figure 4.9, exhibiting a trend consistent with the Dice coefficient results.

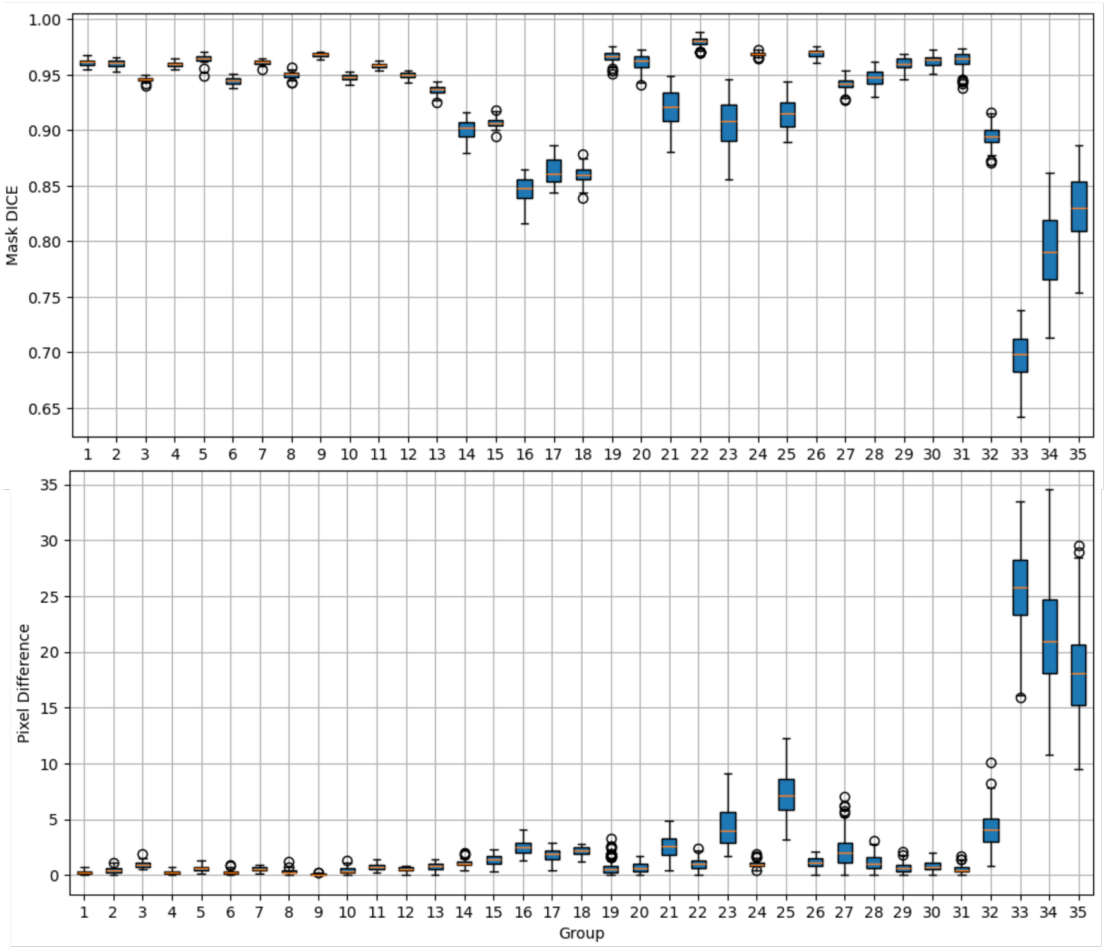

Figure 4.9: Group variability of Dice coefficient showcasing the analysis done for the original masks and predictions (top) and the analysis for their border differences (bottom).

To further examine the groups with the lowest performance, representative samples from Groups 33 to 35 are presented in Figure 4.10, along with their corresponding masks

obtained from the dataset preparation stage. Visual inspection of these samples reveals common artifacts, including shadowing, motion blur, segmentation error as well as increased variability in mask geometry.

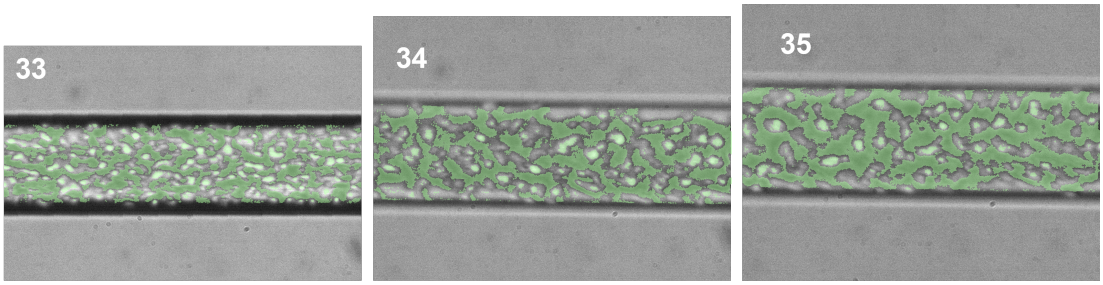

Figure 4.10: Samples of the three groups with lowest performance and their corresponding masks (highlighted on green). Images were scaled to fit in the document but their original dimensions are displayed below each group.

Moreover, to test the segmentation capabilities of the trained model, six new experimental groups without masks were fed into the algorithm. The predictions of the RBCs core from those new images can be seen in Figure 4.11 where representative images from each group are overlaid with the border highlighted in yellow showing a well delimited RBCs core across the six new experimental groups.

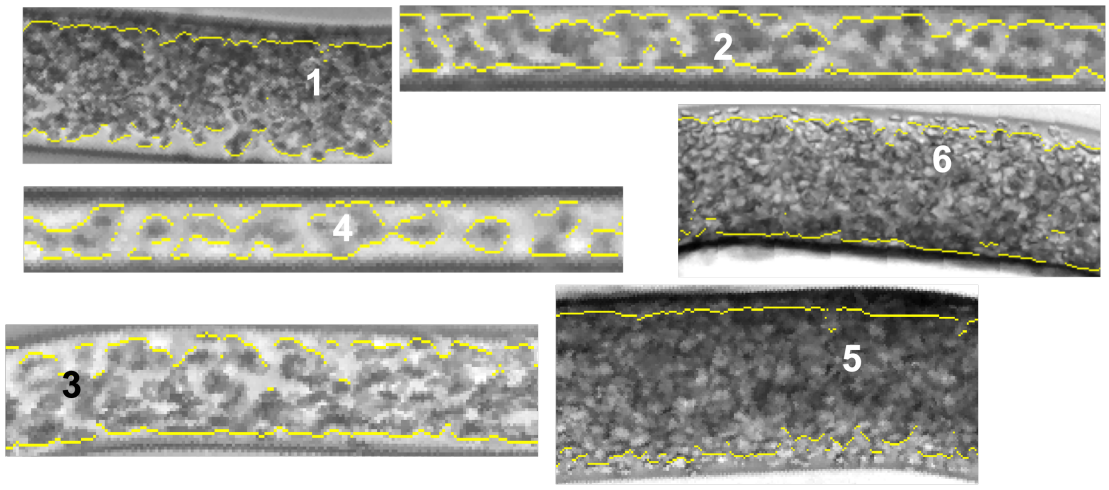

Figure 4.11: Predicted CFL borders obtained using the trained attention U-Net (yellow) for six new groups of images with no masks.

4.3 Discussion

The results presented in this chapter highlight the critical importance of high-quality validation data in supervised machine learning. Both Dice coefficients and border-based metrics remained relatively stable across most of the groups showing consistent behaviour across them. The results indicate that segmentation performance is influenced strongly by mask quality. This effect is particularly evident in groups 33 to 35 illustrated in Figure 4.10 where inconsistent annotations coming from the stage prior (not the SAM), influenced the performance of the proposed segmentation method as opposed to the mask presented in Figure 4.3. In contrast, groups 24 and 25 exhibit visually challenging conditions (see Appendix A.2) but their masks were well annotated and their predictions are more stable

than the ones from groups 33 to 35. Rather than representing isolated failures, these cases define the operational limits of the proposed approach.

Additionally, the new predictions across the six new experimental groups, showcases the ability of the algorithm to generalize to new data with varying image conditions. However, the images with good contrast between RBCs and the vessel surroundings had a clearer segmentation. This suggests that improvements in acquisition standardization would likely enhance performance.

This chapter corroborates that attention U-Net can replicate and automate a validated CFL analysis workflow while eliminating user dependency. By performing a boundary comparison instead of area-based metric, the proposed evaluation strategy aligns model performance with the physiological objective of CFL thickness estimation. These segmentation results are integrated into the graph-based environment described in the next chapter, enabling network-aware analysis of CFL-related variables.

Chapter 5

Integrative Analysis Environment

This chapter presents an integrative analysis environment designed to enable network-aware investigation of microfluidic systems. While previous chapters introduced a graph-based representation of the retina-inspired chip (Chapter 3) and an automated method for extracting cell-free layer (CFL) measurements (Chapter 4), these components remain limited when used independently. In conventional workflows, image processing, quantitative measurement, and structural analysis are performed using separate tools, resulting in fragmented pipelines where experimental data are disconnected from the network topology that governs their interactions.

This separation limits analysis to individual channels, preventing reasoning at a system level in microvascular networks where local variables (such as velocity, HTC, or CFL thickness) are inherently coupled through branching structure. To address this limitation, the environment developed in this work embeds experimental data directly within a graph-based representation of the microfluidic network. By unifying spatial data, experimental measurements, and topology into a single model, it enables analyses that treat the system as an interconnected network rather than a collection of independent segments.

The environment is designed to support key classes of scientific questions that cannot be addressed through conventional pipelines, including the propagation of variables across connected vessels, the influence of hierarchical structure on local measurements, and the inference of missing data based on network relationships. Through this integration, the environment establishes a framework for system-level interpretation of microfluidic experiments.

5.1 Materials and Methods

5.1.1 Graph Data Representation, Structure and Analysis

The core of the environment relies on the graph model presented in Chapter 3 implemented using NetworkX library. The microfluidic network is represented as a directed graph, where nodes correspond to complete vessels and edges represent connectivity relationships. Each node supports flexible attribute storage, allowing experimental variables, images, metadata, and derived quantities to be associated directly with their corresponding network locations. This approach mirrors the structure of vascular systems and provides a natural foundation for integrating experimental data with topological analysis.

Most of the previous experimental data is acquired through MATLAB workflows, therefore, MATLAB files with variables coming from experiments are expected as part of the input data. All data within the environment are stored in a global dictionary structure inspired by NetworkX's handling of graph data. The data are organized hierarchically into

two categories:

- Main data, which define the global context of the experiment: the graph structure, the full network image, the labeled network representation and Local data.
- Local data, which depend on the selected region of interest (ROI): image dataset, MATLAB variables coming from experiments (MATLAB 2024b), ROI image.

This separation enforces a workflow in which topology is defined once and reused across multiple localized experiments, ensuring consistency and reproducibility. Entire analysis sessions can be saved and restored, preserving both data and graph state. Sessions can be saved and restored using compressed archive files (.zip) with all required components, ensuring reproducibility and continuity across sessions. Figure 5.1 illustrates the different data types and the aforementioned options for the sessions.

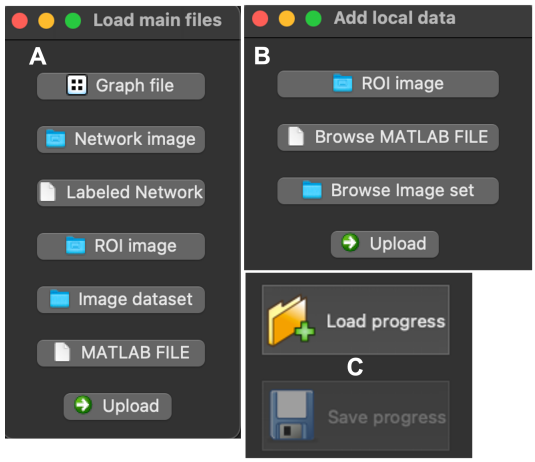

Figure 5.1: The windows corresponding to the two categories for data. A: Main data and B: local data. Both showcasing the different files they accept. C: Options to save or load a session.

5.1.2 Environment Workflow

Based on the regular workflow to process data with other analysis tools and designing the environment around the user, the overall data flow follows a structured pipeline as depicted in Figure 5.2 having two possible starting points to initialize a project, both options have a green contour. The following steps follow a loop that enable the user to do as many iterations as needed and finishing by saving the progress to continue working on it later. This iterative pipeline ensures consistent propagation of data from raw images to graph-based analyses while preserving user flexibility.

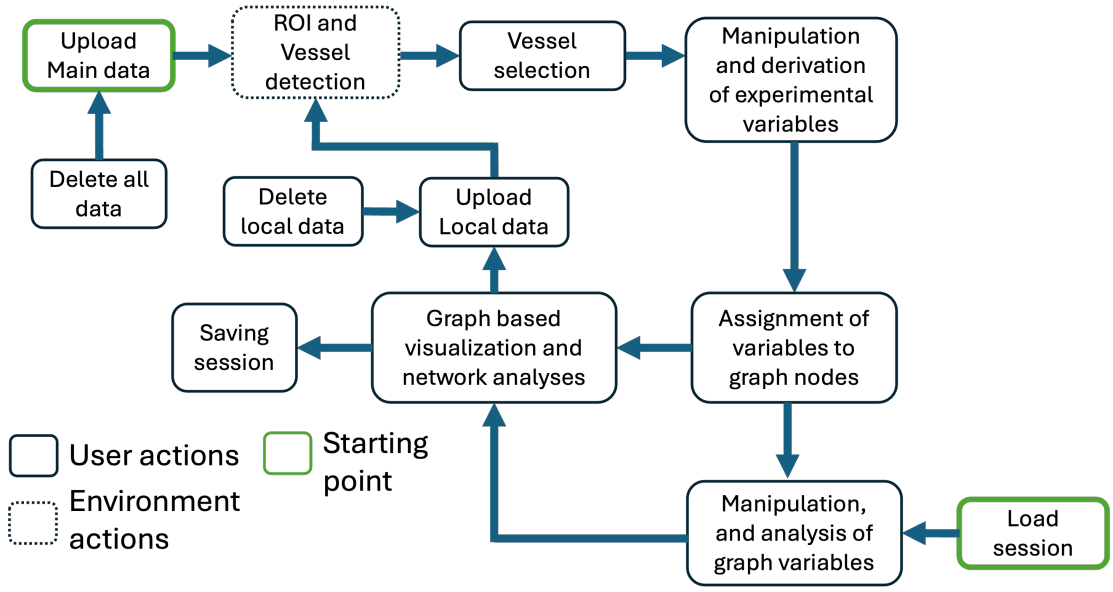

Figure 5.2: Environment workflow simplified from uploading data to saving the session. Showing the different data-driven options.

5.1.3 ROI and Vessel Detection

To reliably map experimental measurements to graph nodes, vessels must first be accurately segmented, labeled, and spatially aligned with the reference network. The first step involving vessel segmentation is performed using a multi step image processing pipeline, it is done previously to use the environment as it is case dependent. For the current network under study, first, a Sato filter [46] is applied to enhance tubular structures, followed by Otsu thresholding [47] to generate a binary vessel mask. Morphological erosion is then used to refine vessel boundaries and reduce noise. Figure 5.3 shows the described process.

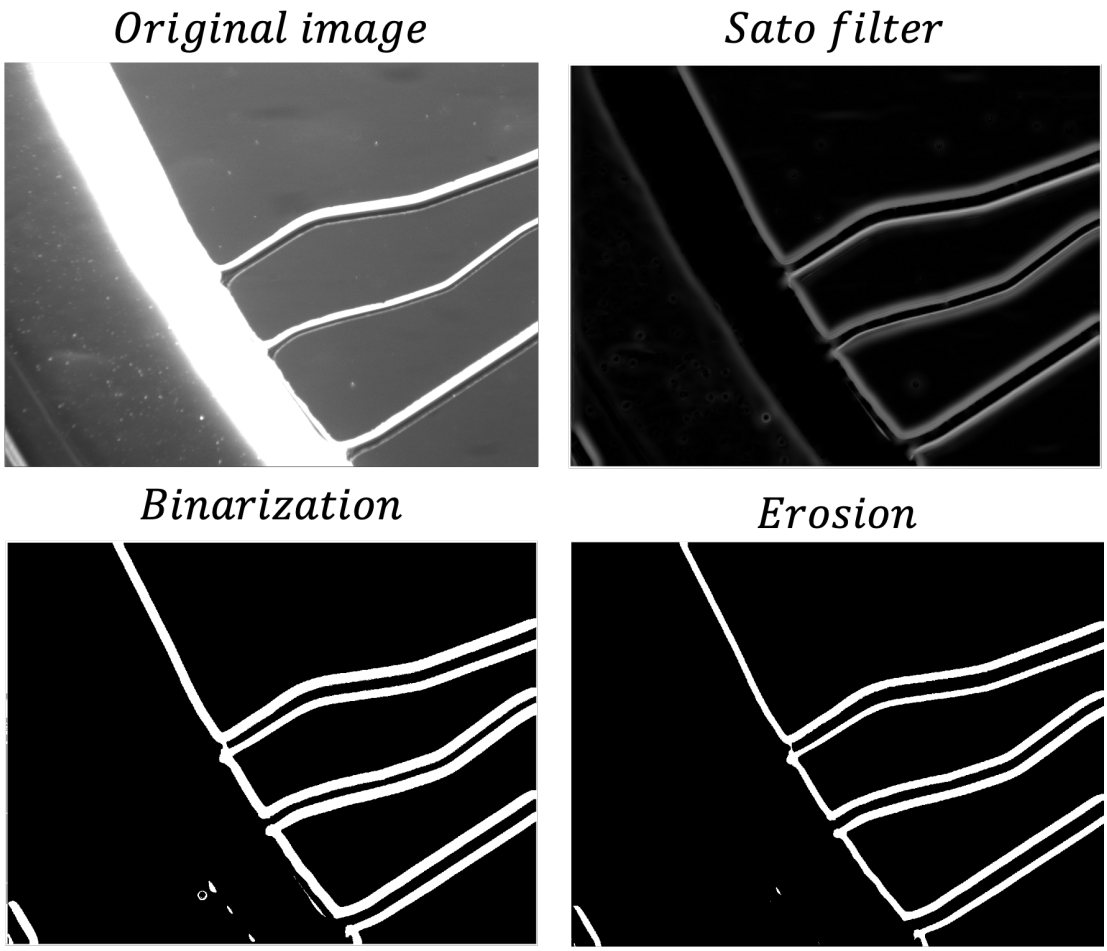

Figure 5.3: Stages of the vessel segmentation pipeline: filtering using Sato filter, binarization through Otsu's threshold, and morphological refinement through erosion.

Secondly, to enable a visual aid for each vessel and to perform a coherent analysis using the graph nodes and its numbers, a vessel isolation was performed as required by MATLAB's *bwlabel()* function. This algorithm labels non-connected elements in a binary image using the pixel values. Therefore, each vessel of the binary representation of the

network got manually isolated from its mother and daughter vessels by removing some pixels at branching points. Once isolated, the algorithm labels non-connected elements in a binary image assigning the labels based on a left-to-right, top-to-bottom scan. The resulting labeled network is illustrated in panel A of Figure 5.4 showing the lowest pixel values (label numbers) on the left while displaying the highest to the right. However, those labels did not represent the intended label system discussed in Chapter 3. Neither by number nor location as highlighted on the colour bars on Figure 5.4.

To enforce consistency with the predefined vessel numbering scheme, those initial labels needed to be manually changed. As result, the lowest values are then set at the center of the chip, and bigger values can be found around the chip's periphery. Panel B of Figure 5.4 illustrates the resulting image, showing better accordance with the actual range of vessel numbers (1-713). Lastly, as a standard 8-bit image is insufficient to store those pixel values (greater than 256). A 16-bit image format was therefore adopted to ensure accurate label storage and visualization.

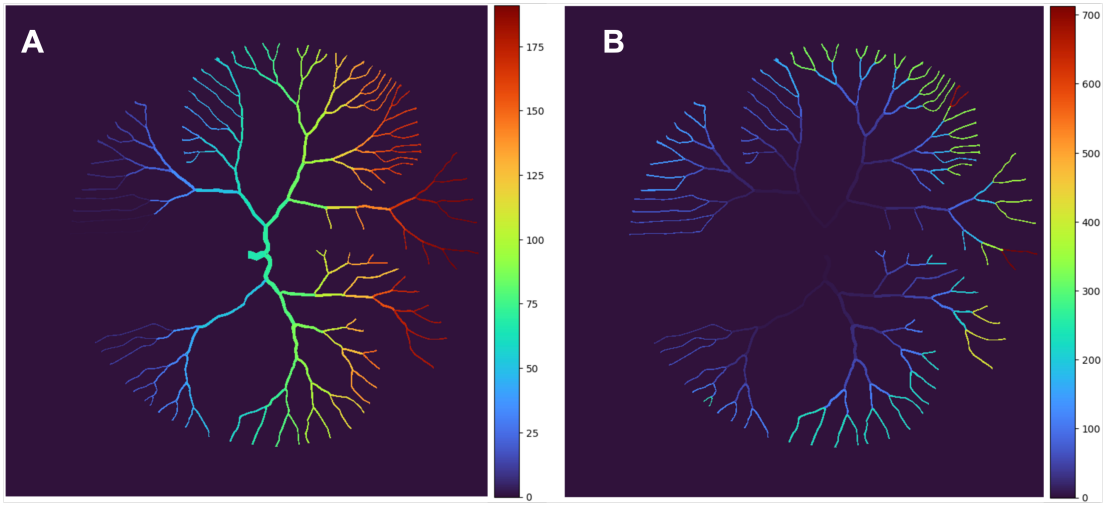

Figure 5.4: Vessel labeling results. A: Labeling produced by MATLAB showing the lowest numbers (initial) to the left and the highest (final) to the right. B: Corrected, consistently assigned vessel identifiers showcasing the starting number at the center and the highest around the chip.

Spatial alignment between experimental images and the network reference is performed through a template matching process. A filtered ROI template is convolved with the filtered network image using a sliding-window algorithm using correlation as similarity metric. The resulting similarity map is analyzed to identify optimal alignment coordinates, which define the ROI location within the full network. Additionally, the resulting area is then extracted from the labeled network image to obtain the precise vessels depicted in such ROI. This process ensures that each experimental image is accurately mapped to its corresponding physical location on the chip. Figure 5.5 shows the process.

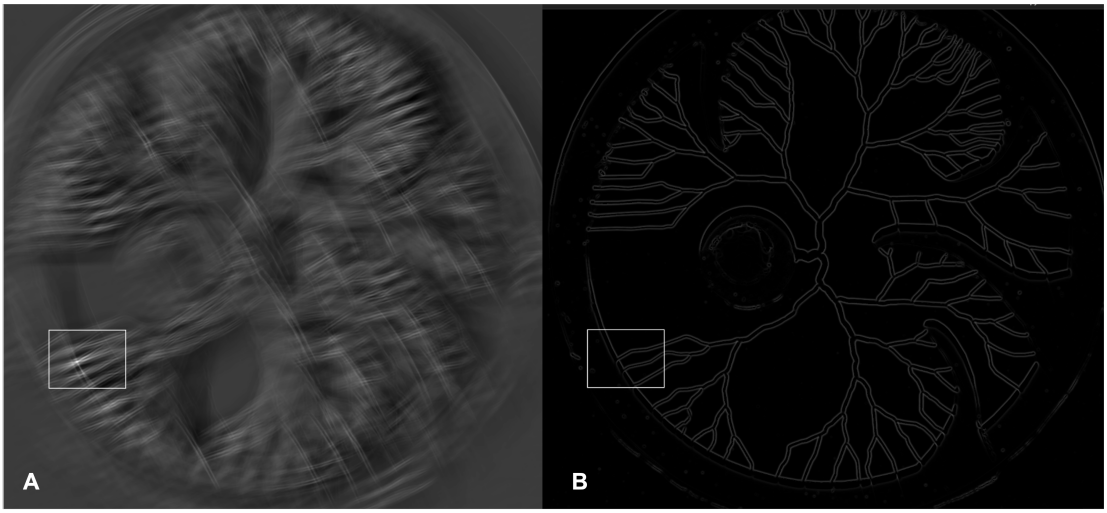

Figure 5.5: ROI detection process. A: Similarity map from sliding template matching illustrating the sliding template over the chip map with the white values showing similarity. B: Detected ROI overlaid on the filtered network image alligning the template where the highest similarity was found on the microfluidic network.

5.1.4 Data Assignment and Node Interaction

Once the mapping between ROI vessels and graph nodes is established, experimental data can be assigned directly to the corresponding nodes. This includes scalar variables (e.g., velocity, hematocrit), segmentation outputs, and associated metadata. The environment supports common scientific Python libraries such as `NumPy`, `math`, or `stats`. Each generated variable is stored automatically in the experiment dictionary, added to the visualization tree, and logged in a history panel for reuse. An example is shown in Figure 5.6.

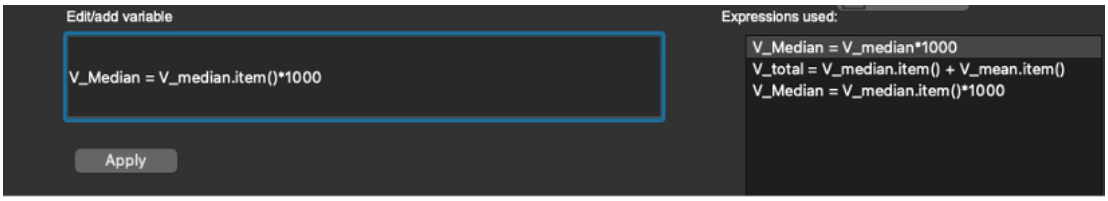

Figure 5.6: Variable creation interface, with expression input (left) and history of previously used formulas (right).

To support reliable data integration across multiple experiments, the environment provides visual feedback by highlighting selected vessels within the ROI. This reduces ambiguity during data assignment and ensures consistency when data are incrementally added across the network.This process is illustrated in Figure 5.7.

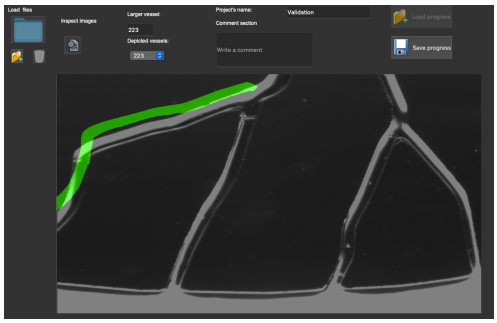

Figure 5.7: ROI with selected vessel highlighted on green.

Additionally, the environment allows direct interaction with node attributes, including manual editing of values. This capability is particularly important for exploratory analysis, where missing or uncertain measurements may need to be estimated or corrected based on contextual information. Figure 5.8 shows an example where missing nodes display a question mark, while nodes with assigned values display their numerical data and the two

options for variable manipulation.

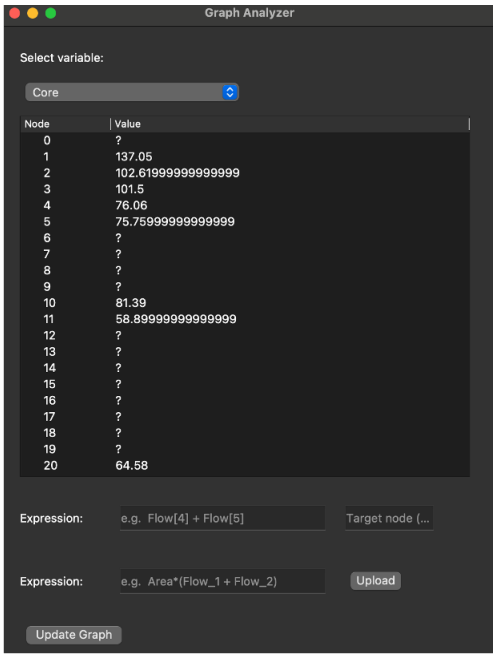

Figure 5.8: Manual node-editing interface for a selected variable, highlighting missing and assigned values.

5.1.5 Graph-Based Visualization and Network Analyses

Three visualization modes support network-level analysis and facilitate interpretation of experimental variables. Each visualization creates a pop-up window. Once a variable is selected, its values are normalized and mapped to a colour scale, allowing users to visualize how the variable distributes across the network as depicted in Figure 5.9.

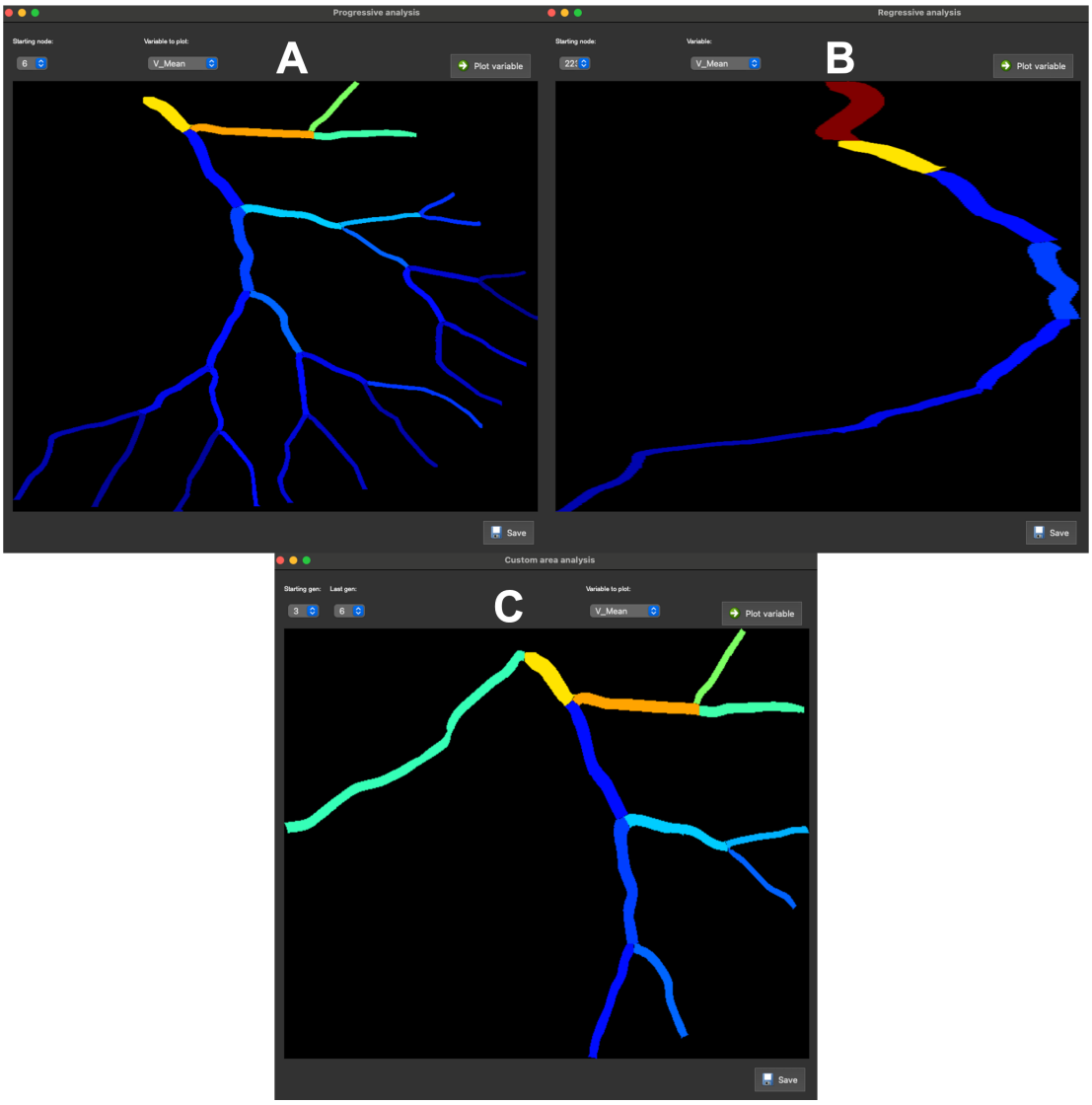

Figure 5.9: Graph-based visualization options available in the environment. A: Progressive analysis. B: Regressive analysis. C: Generational analysis

The environment supports visualization of variables across network generations, allowing users to investigate how measurements evolve from inlet to outlet as portrayed in

Figure 5.9 C. Progressive analyses examine how quantities propagate through the network from upstream to downstream vessels, supporting studies such as mass conservation or cumulative effects. Regressive analyses explore relationships between variables by tracing relationships backward from distal vessels toward their sources, enabling exploration of experimental data for processes where research questions align with the evolution of certain variable. A visual example for both analyses is presented in Figure 5.9 A and B respectively.

All network visualizations can be exported as images or subgraphs. These exports facilitate further analysis, reporting, and integration into publications, ensuring that results generated within the environment are easily transferable. Figure 5.10 illustrates these export features. An example of the code for the postprocessing modification can be found in Appendix A.4

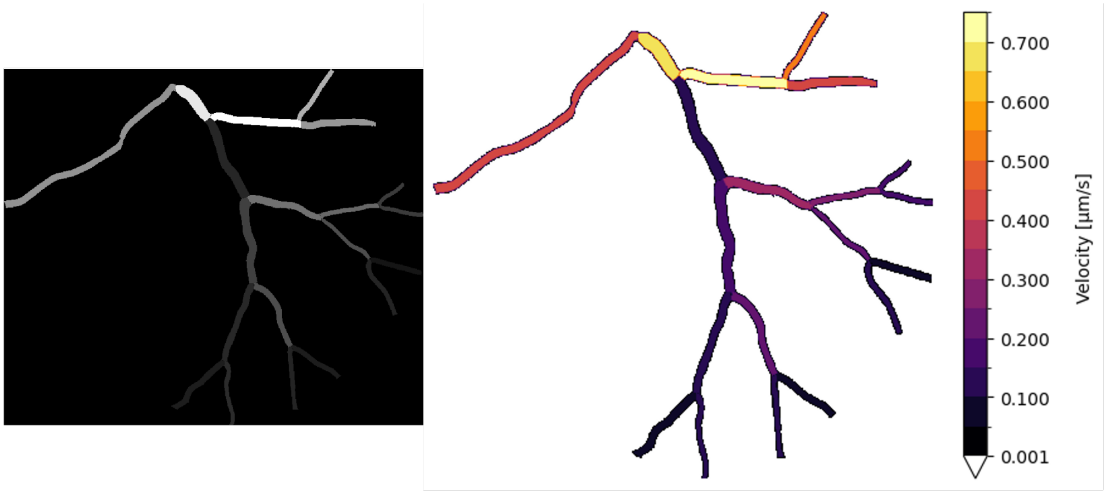

Figure 5.10: Example of exported network visualization (left) and subgraph corresponding to a generational analysis with post processing modifications (right).

5.1.6 Integration of RBCs Core Segmentation

To eliminate the need for external preprocessing pipelines, the CFL segmentation method developed in Chapter 4 was integrated directly into the environment facilitating its practical use in microfluidic research. This module reconstructs the trained Attention U-Net model and performs inference on user-provided image datasets by loading the weights obtained during the training stage. The resulting segmentation masks are processed to extract RBC core boundaries and estimate CFL thickness. Outputs are visualized directly on the input images and can be exported alongside numerical results.

This integration ensures that segmentation outputs are immediately available for graph-based analysis, completing the connection between image data and network-level interpretation. Figure 5.11 illustrates the segmentation interface, highlighting the predicted RBC core border (shown in blue) overlaid on an example input image. The figure also depicts the corresponding controls for data loading, weight selection, and output configuration.

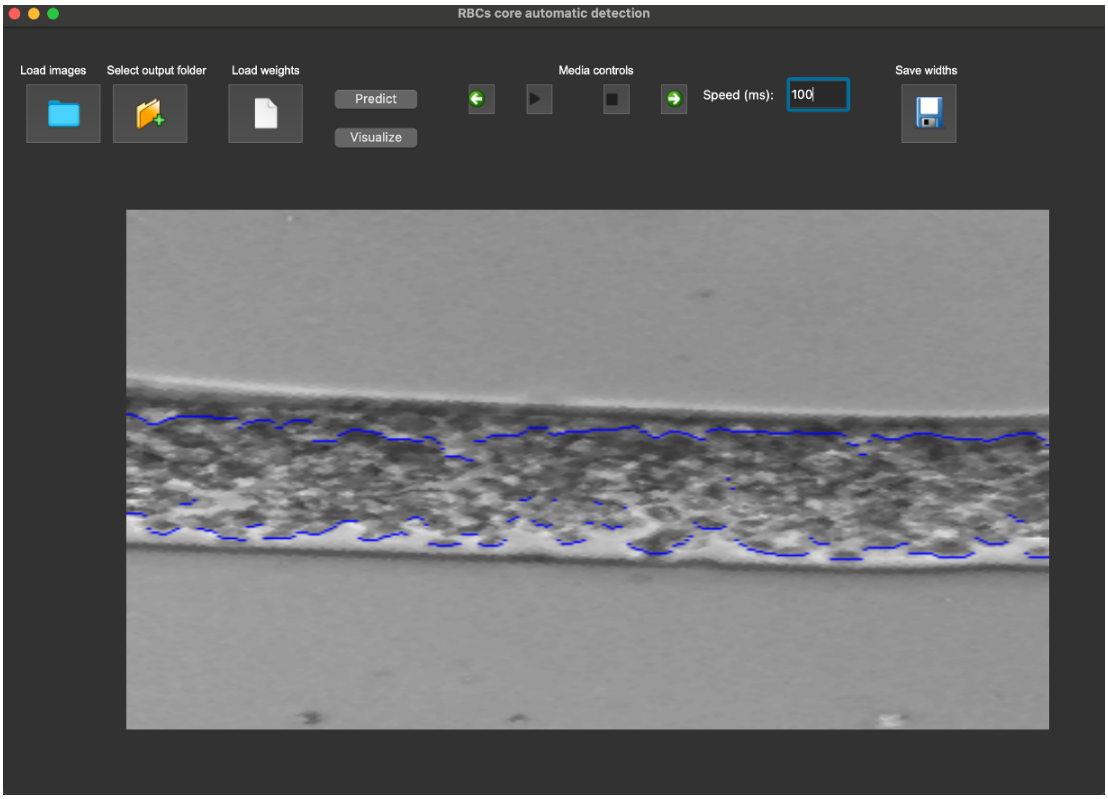

Figure 5.11: RBC core segmentation tool integrated into the analysis environment, showing the predicted border overlaid on the original image.

5.1.7 Performance and Workflow Evaluation

The environment was evaluated using experimental datasets acquired in the retina chip. Validation focused on the application of the different features offered in this environment to answer examples of the scientific questions presented on the motivation for the realization of this project highlighting the ability to perform network-aware analyses. Three cases are presented using data obtained from two main sources.

For the first source, data were obtained from preliminary micro particle image velocimetry (micro-PIV) analyses for the lower section of the retina aiming to perform a 3D mapping the velocity fields for the vessels . These experiments were performed using micro particles suspended in a solution of deionized water acquiring images at three different depths on the focal plane of the microscope by Aude Vermeerbergen Ferry at the University of Ottawa. Meanwhile, the second source for the second and third cases rely on experimental measurements using blood performed by Laureline Julien at the CBS laboratory in Montpellier [1]. Figure 5.12 highlights the ROI where the data was obtained within the chip for these cases.

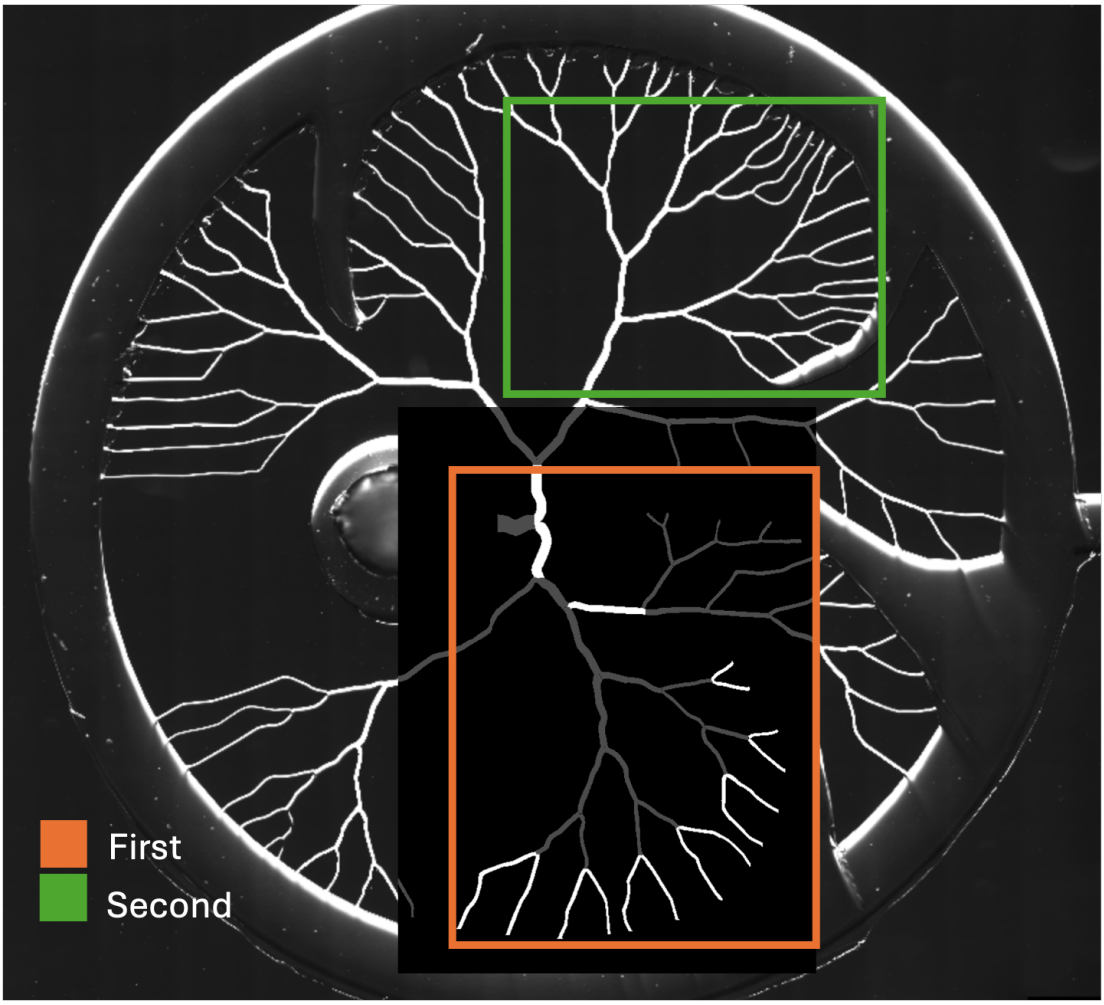

Figure 5.12: Retinal network with the two regions of interest used for workflow evaluation. The first dataset (orange square) corresponds to vessels with data obtained by Aude V. Ferry, highlighted in white, while vessels without data are faded. The second dataset (green square) corresponds to data obtained and processed by Laureline Julien.

For the **first case**, data obtained from the micro-PIV analyses for one of the three depths were used to infer the velocity in the mother vessels before acquiring the remaining

data for the other vessels. As the acquisition protocol for micro-PIV requires a separation time parameter (dt) for each ROI, gaining a first insight of the expected velocities accelerates the acquisition protocol by estimating the dt parameter sooner for the remaining ROIs. Such insights gained from the inference can help identify both abnormal trends and expected values before committing time consuming or resource intensive experimental protocols. For instance, each scan can take up to 12 hours and they might result in wasting micro-particles. By performing an inverse progressive analysis, the data from the mother vessels were be inferred by adding the values of ending vessels (daughters) using the manual node editing feature of the environment paired with the vessel visualization aid. Subsequently, the inferred values were compared with the experimental results to evaluate the predicted behaviour.

The **second case**, with experimental data containing HTC concentration and vessel's width, the user wants to compare the relationship between the vessels topology and the HTC, to see how the width affects the propagation of RBCs through the network. Similarly, this comparison wants to see if the HTC drops at each bifurcation in a predictable way. HTC concentration per vessel was estimated by Laureline Julien using an image intensity-based method for measuring relative values of red blood cell quantity/volume in the channels. In this case, a progressive analysis was conducted by selecting a starting vessel and propagating information across downstream vessels with available data. Results were then exported for further analysis outside the environment on a separate manner.

Lastly, for the **third case**, the user aimed to estimate the volumetric flow rate in each vessel of the retina chip by combining experimentally measured velocities with geometri-

cal information from the fabricated channels. The fabrication of the chip described in 2 resulted in rounded, non-rectangular channel cross-sections. Geometrical analysis of the microchannels was performed by Laureline Julien based on electron microscopy images of the retina-on-chip devices.

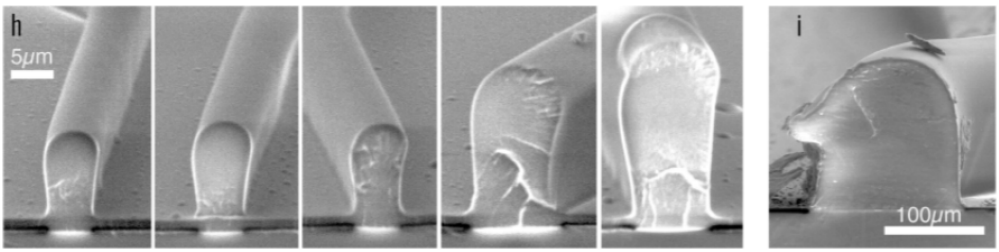

Figure 5.13: Microfabricated channel cross-section during printing and collapse. Panels (h-i) extracted from Fenech et al (2019) [26]. The images illustrate the non-rectangular channel geometry that motivated the parabolic height interpolation used for this case study.

To account for the channel's geometry, the vessel cross-section was interpolated using an experimentally derived parabolic height profile. The selection of a parabolic profile alligns with the overall shape observed through the channels as illustrated in Figure 5.13. Thanks to a profilometry analysis, the height h_m depending on the channel width w can be expressed as follows [1]:

$$h_m(w) = 0.7604w + 15.58 \tag{5.1}$$

with the reference channel height h_m and width w expressed in micro meters. Subsequently, the parabolic profile lead to a parabolic equation for the height of the channel h

along x oriented perpendicular to the channel:

$$h(x) = -\frac{4h_m}{w^2}x^2 + h_m \tag{5.2}$$

By combining the measured average velocity with the interpolated cross sectional area, the volumetric flow rate in each vessel can be computed. The derivation of experimental variables and flow rate calculation are implemented using the analysis functions of the environment, allowing systematic and consistent estimation of flow distribution throughout the network. The cross-sectional area A is obtained thanks to the parabolic height interpolation by integrating the height profile h across the channel width x by:

$$A(w) = \int_{-\frac{w}{2}}^{\frac{w}{2}} h(x)\,dx \tag{5.3}$$

$$A(w) = \frac{2}{3}h_m w \tag{5.4}$$

Assuming a known average velocity $\bar{V}$, the volumetric flow rate Q is then given by:

$$Q = \bar{V}A(w) \tag{5.5}$$

Thus, the flow rate can be expressed as:

$$Q = (0.5069w^2 + 10.387w)\bar{V} \tag{5.6}$$

The average velocity in each branch were estimated by Laureline Julien using particle image velocimetry (PIV), based on a cross-correlation algorithm implemented in the open-source software PIVlab (MATLAB-based). Using blood cell, this approach provides an estimate average velocity [48].

Once the analytical framework was established, the workflow within the environment consisted of defining new global variables: first, the cross-sectional area was computed using the vessel width, followed by the calculation of flow rate using both area and velocity verifying units to obtain flow expressed as micro liters per minute. These operations were performed through the environment's variable editing interface, enabling consistent and reproducible flow estimation across the network.

5.2 Results

5.2.1 Use Case 1: Data inference for experimental comparison

Figure 5.14 illustrates the inference process applied to upstream (mother) vessels based on known values from downstream (daughter) vessels and compares the inferred results with experimental measurements. The inferred values show good agreement with experimental data at the terminal vessels.

The largest discrepancies are observed in the initial vessel segment, where inferred values deviate more noticeably from experimental measurements, as shown in panels C and D with the bar plots. Panel E quantitatively compares experimental and inferred

velocities across the analyzed vessels, revealing that the maximum relative error occurs in an intermediate vessel located between the first and terminal segments. For this vessel, the relative error reaches approximately 140% with respect to the experimentally measured velocity as depicted on panel F.

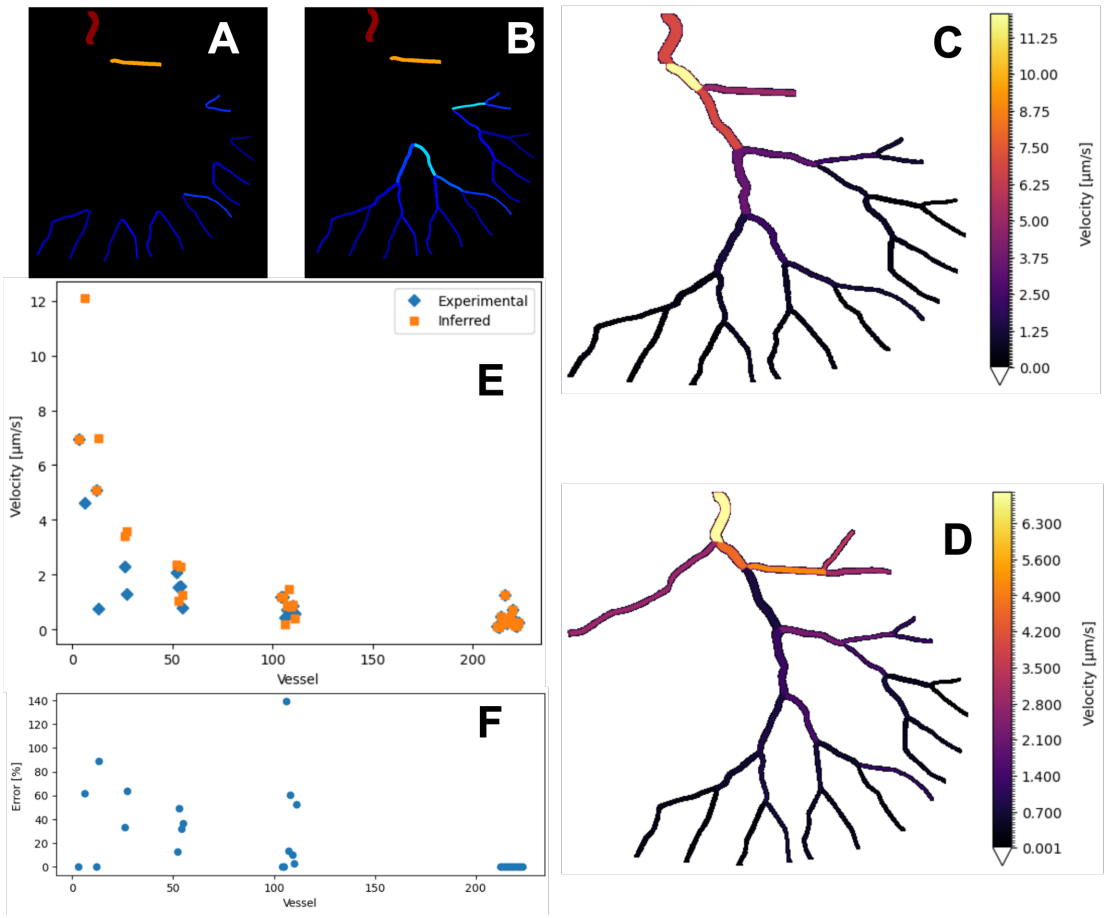

Figure 5.14: Results of the inference of mother vessel velocities based on daughter vessel information using the integrative environment. A: Initial state of the analysis, B: inference across two generations of upstream vessels, C: final inferred values displayed in the environment, D: corresponding experimental measurements, E: comparison between inferred and experimental velocity values, F: relative error between experimental and inferred velocities.

5.2.2 Use Case 2: Influence of Vessel Topology on Hematocrit

Figure 5.15 presents the spatial distribution of HTC and vessel width within the microfluidic network, displayed in panels A and B, respectively. Both variables are then plotted having HTC as funcion of vessel width to see the influence of the width. The resulting plot is shown in panel C.

Panel D illustrates the regressive analysis used to identify the path associated with the highest HTC values across the network. This path was extracted using the saving options of the environment. It can be seen that the enhanced plot comes from the data in panel A. However, thanks to an external Python script, elements such as a different colour map and the colour bar were added. An example of this code can be found in Appendix A.4. Together, these results illustrate how the integration of topological information with experimentally derived variables enables network-level analyses that extend beyond local or individual channel observations.

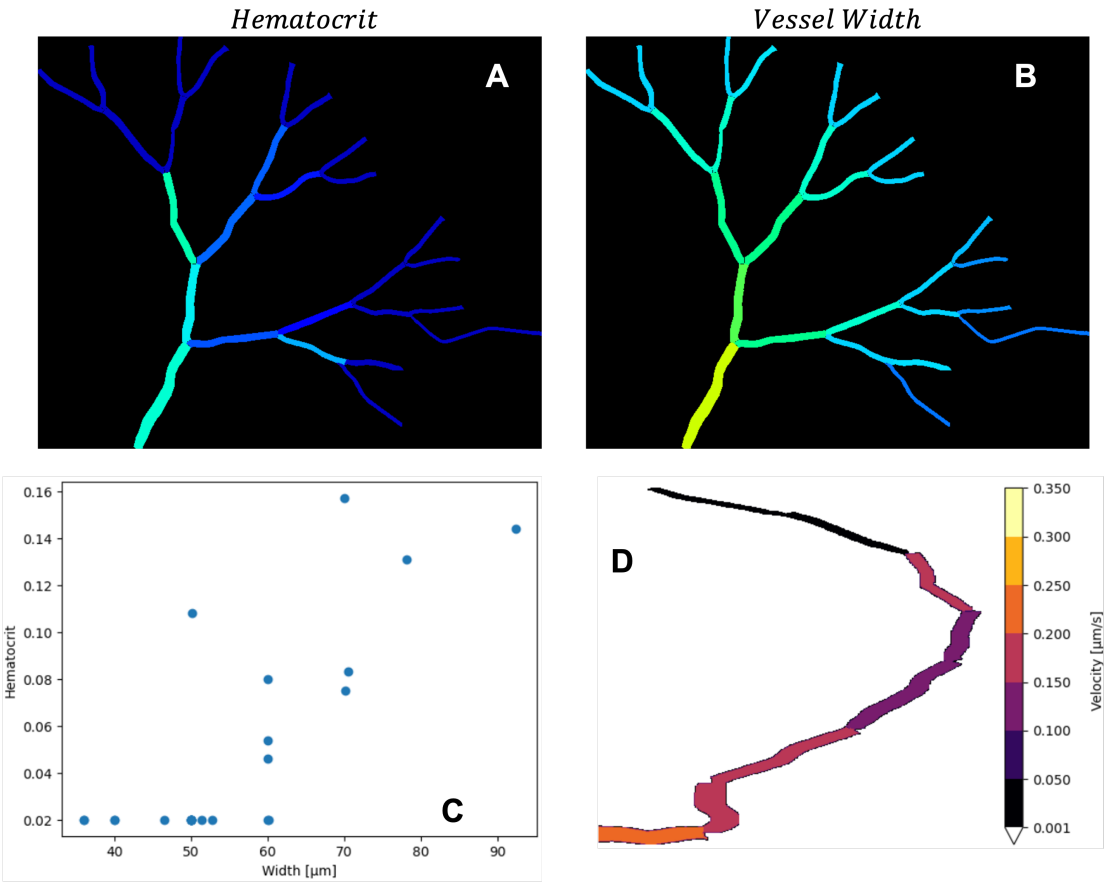

Figure 5.15: Results of the comparison of the width and HTC. A: HTC values displayed in the environment, B: Vessel width values displayed in the environment, C: HTC values compared with width values for their matching nodes, D: Regressive analysis output enhanced outside the environment.

5.2.3 Use Case 3: Variable derivation for flow rate estimation

To estimate the flow rate in each vessel by combining experimental velocities with an approximation of their cross sectional area. Panel A in Figure 5.16 presents the initial

velocity data. In parallel, the calculation of the area as function of the vessel widths is plotted on panel B along with the expression entered into the environment. This verifies the environment's ability to perform single variable operations. Subsequently, panel C shows the resulting flow rate data expressed as a function of velocity and cross-sectional area, including unit conversion to microliters per minute. Similarly, this calculation highlights the environment's feature for interaction with global variables to calculate new variables dependent on acquired data.

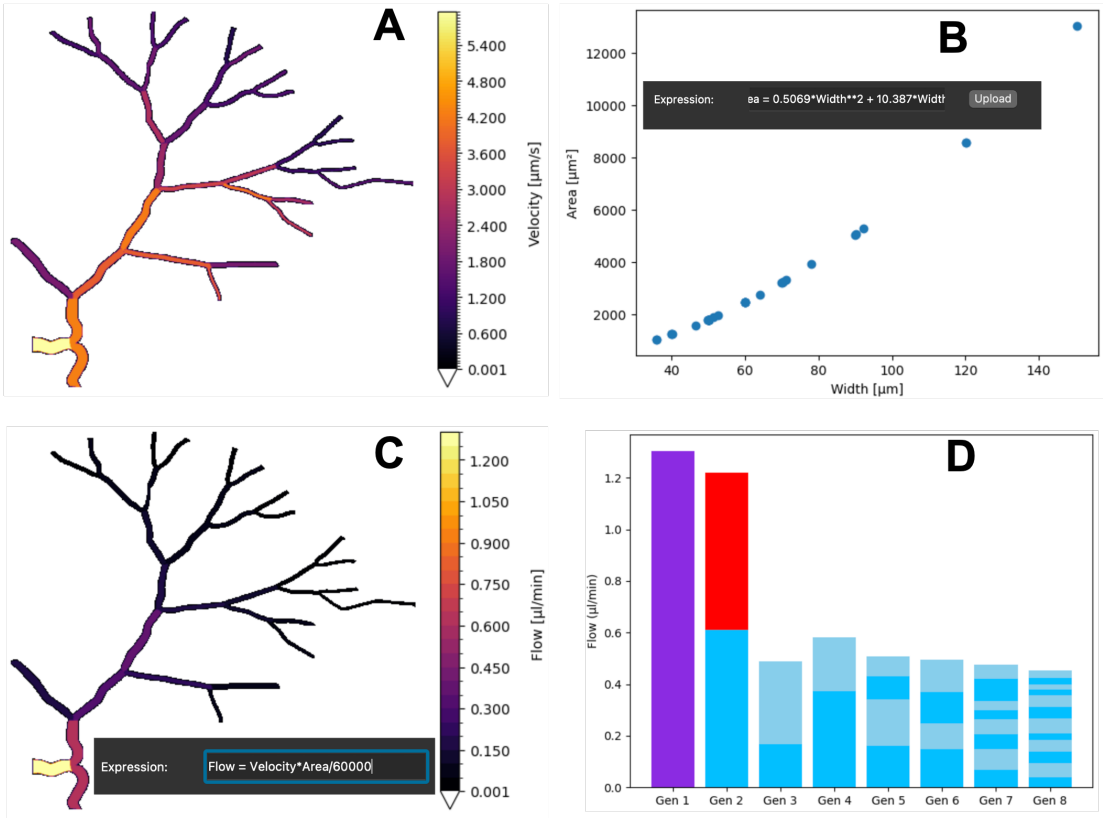

Figure 5.16: Results of the flow calculation from the vessel widths and velocities. A: Velocity at the ROI, B: plot of the area on function of the width, result from the expression in the screenshot, C: calculated flow result of the expression accompanying the plot, D: Conservation mass analysis performed for all the nodes in the subgraph.

Lastly, a mass conservation analysis was performed comparing the sum of the flow values along all the nodes for each generation, in this case, the first two generations differ from the other 6 as highlighted on Figure 5.16 panel D. However, the rest of the vessels exhibit a consistent flow when compared with vessel 2 (blue bottom of the column for the second generation).

5.3 Discussion

The integrative environment developed in this work addresses a fundamental limitation in microfluidic analysis: the inability to relate experimental measurements to the network structures that govern them. By embedding data within a graph model, the proposed framework shifts analysis from a single channel perspective to a network oriented paradigm, enabling variables to be interpreted in terms of connectivity, hierarchy, and propagation.

This shift introduces several new capabilities. First, it enables inference of missing or unmeasured variables using topological relationships. Second, it supports relational analyses that combine multiple variables across the network. Third, it provides a consistent framework for deriving and validating new quantities, such as flow rate, within a unified representation.

Despite these advantages, the current implementation relies on partially manual steps for graph construction and spatial alignment. A persistent limitation is the absence of automated image conversion into graphs, primarily due to the lack of large, standardized datasets for microfluidic networks. Future work could address this through the integration of computer vision methods for automated topology extraction.

The use cases helped to visualize some applications for the developed environment. For the **first case**, it helped to accelerate the acquisition protocol for the micro PIV by calculating faster the parameter dt, additionally the presented variations between the inferred values and the experimental ones lead to identify an issue with the experimental

setup. For the **second case**, visualizing the HTC paths along the network highlighted the influence of the width and vessel angles, streamlining the process for multi variable analysis. Finally, for the **third case**, the mass conservation analysis confirmed the accuracy of the geometric estimations for the cross sectional area calculation and subsequent flow estimation function of the velocities and areas. While the first two generations exhibit a higher flow than the subsequent generations, this can be easily explain as the depicted vessels only make up for the flow in that ROI. If a complete dataset was analyzed, all the generations will exhibit a similar flow as the third generation is about half of the size of the second generation, this is further illustrated with the rest of the generations were they are compared with the bottom value of the plot for the second generation.

Beyond the individual use cases presented, the combination of inference, variable manipulation, and graph-based visualization enables more advanced exploratory analyses of microfluidic systems. Potential developments include the incorporation of physics-based models, automated inference algorithms, and comparative analysis across multiple network designs.

By unifying experimental data and topology within a single analytical framework, this work demonstrates that graph-based methods can provide meaningful insight into microfluidic networks, opening new directions for both experimental design and data interpretation.

Chapter 6

Summary and Conclusions

This chapter summarizes the main outcomes of this thesis, evaluates them in relation to the original research objectives, and highlights future research directions. By doing so, it provides a coherent closure that links the initial motivation of this work with its methodological developments and obtained results.

6.1 Summary

In this thesis, a biomimetic microfluidic chip inspired by the human retina was used to develop and validate a graph-based framework for the analysis of microfluidic networks. By representing the chip as a directed graph, this work enabled a structured and connected view of the network, better reflecting the intrinsic topology of vascular systems than traditional, isolated channel-based analyses. This approach addresses limitations in the analysis of microfluidic networks, where both the lack of suitable tools and the scarcity of complex network designs have slowed down research in this field [49].

The resulting model consists of a directed graph with 193 nodes and 267 edges, where

each node supports flexible data storage, such as data coming from microfluidic experiments, including images, file paths, numerical arrays, scalars, vectors, lists, and text-based metadata. This structure allows experimental data to be directly associated with specific locations in the network, facilitating integrated topological, spatial, and flow-related analyses. Furthermore, the graph-based representation enables the application of established network metrics, providing new perspectives for studying blood flow and connectivity patterns in microfluidic systems using graph theory.

Building upon this representation, an integrative environment was developed to make graph theory accessible for microfluidic research, and to create a centralized tool that integrate different steps for analyzing data (e.g. organization, visualization, and correlation). Although validated using the retina chip, the environment was designed with generality in mind, allowing it to be adapted to other microfluidic network topologies. Preliminary feedback from users highlighted its potential to organize experimental data, simplify analysis workflows, and support exploratory investigations of microcirculatory behaviour.

Simultaneously, this work addressed the automation of RBC core segmentation using an Attention U-Net. The proposed approach achieved strong performance across diverse experimental conditions and displayed robustness to variability in image characteristics. Observed limitations were primarily linked to inconsistencies in ground truth mask quality, highlighting the critical role of reliable annotation pipelines in supervised learning.

6.2 Future work

Different ideas for future work emerge from this project. First, further automation of vessel labeling and graph construction is recommended. In this thesis, these steps were performed in a manual way, which was appropriate given the availability of a single network topology. However, extending the framework to diverse microfluidic designs would benefit from robust, generalized algorithms capable of extracting graph structures directly from images.

Second, standardization of image acquisition protocols is essential to reduce inter experimental variability. Improvements could include the use of controlled and uniform illumination sources or the establishment of acquisition guidelines or protocols that minimize shadows and contrast inconsistencies. Such standardization would directly enhance the reliability of automated image analysis and segmentation methods. Addressing these limitations would also enable the development of more advanced algorithms, such as fully automated measurements of spatial features (e.g., the distance between the RBCs core and channel walls), with reduced sensitivity to experimental noise.

Finally, feedback from early users indicates that the developed integrative environment represents a solid foundation for future tools targeting microcirculatory research. Suggested improvements include enhanced variable manipulation, richer visualization options, and more interactive analysis capabilities including wall detection, on-site velocity measurements, and intensity-image analysis for local HTC estimation in diffusion studies, all

of which would further strengthen the usability and impact of the framework, and they will address both, the short and medium term needs of our lab.

6.3 Conclusion

This thesis addressed a key limitation in the study of biomimetic microfluidic systems: the lack of analytical frameworks capable of representing and analyzing these systems as fully interconnected networks. While microfluidic devices have been widely used to study microcirculatory phenomena, the majority of those analyses remain limited to isolated channels, overlooking the influence of network topology on experimental observations.

To address this gap, this work introduced a graph-based framework for modelling biomimetic microfluidic networks, using a retina-inspired microfluidic chip as a representative case study. This approach enables the integration of structural connectivity with experimental data, performing analyses as a fully connected system that are difficult to perform using conventional approaches as well as rarely performed. This representation also made graph theory accessible to researchers and served as the backbone for a dedicated software environment, by integrating data visualization, network analysis, and experimental variable manipulation within a single environment, eliminating user's dependency of scattered tools for different stages of their research. Additionally, the environment facilitates more efficient and reproducible experimentation. The results obtained from the presented use cases indicate that the framework can support inference, variable exploration, and topology-aware analysis, highlighting its practical applicability in experimental settings.

Additionally, this thesis contributed to the automation of RBCs core segmentation through the implementation of an Attention U-Net. The results confirm the suitability of this method for automated analysis of microfluidic blood flow images, particularly under well-controlled imaging conditions and proved capable of adapting to variations in experimental setups.

Taken together, these contributions establish a foundation for integrating graph theory, experimental data, and machine learning in the study of microfluidic networks. While the proposed framework does not fully automate all aspects of network construction or analysis, it significantly advances the ability to perform system-level investigations and highlights the importance of topology-aware approaches in microcirculatory research.

Overall, this work contributes a new combination of graph-based modelling, an integrative environment, and machine learning techniques to the study of microfluidic networks. By enabling network-aware interpretation of experimental data, it opens new directions for studying microcirculatory dynamics and provides a basis for future developments in biomimetic vascular systems and microfluidic research.

References

1. Julien, L. *Microvascular network of the retina 1D model, microfluidic device and high resolutional data for the detection of vascular anomalies* PhD thesis (Sorbonne université, 2023).
2. Abeyrathne, R. & Lanel, G. A Study On Graph Theory Properties In Human Blood Circulatory System. *IJSRP* **11,** 444–455. ISSN: 22503153 (Oct. 24, 2021).
3. Hendricks, N. E. The microcirculation. en. *S Afr J Anaesth Analg* **26,** S62–65. ISSN: 2220-1173 (Nov. 2020).
4. Gracka, M. *et al.* Red blood cells tracking and cell-free layer formation in a microchannel with hyperbolic contraction: A CFD model validation. *Comput Methods Programs Biomed* **226,** 107117. ISSN: 01692607 (Nov. 2022).
5. Omori, T., Imai, Y., Kikuchi, K., Ishikawa, T. & Yamaguchi, T. Hemodynamics in the Microcirculation and in Microfluidics. en. *Ann Biomed Eng* **43,** 238–257. ISSN: 1573-9686 (Jan. 2015).
6. Yeo, L. Y., Chang, H.-C., Chan, P. P. Y. & Friend, J. R. Microfluidic Devices for Bioapplications. en. *Small* **7.** _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/smll.201000946, 12–48. ISSN: 1613-6829 (2011).
7. Fenech, M., Le, A. V., Salame, M., Gliah, O. & Chartrand, C. Semi-automated red blood cell core detection in blood micro-flow. *Microvasc Res* **147,** 104496. ISSN: 00262862 (May 2023).
8. Ng, Y. C., Fisher, L. K., Salim, V., Kim, S. & Namgung, B. Visualization and Quantification of the Cell-free Layer in Arterioles of the Rat Cremaster Muscle. *JoVE,* 54550. ISSN: 1940-087X (Oct. 2016).
9. Ong, P. K. *et al.* An automated method for cell-free layer width determination in small arterioles. en. *Physiol Meas* **32,** N1. ISSN: 0967-3334 (Jan. 2011).
10. Rashidi, Y. *et al. Cell-free layer development and spatial organization of healthy and rigid red blood cells in a microfluidic bifurcation* Apr. 17, 2023. arXiv: `2304.08203[physics]`.
11. Van der Meer, A. D., Poot, A. A., Duits, M. H. G., Feijen, J. & Vermes, I. Microfluidic technology in vascular research. eng. *J Biomed Biotechnol* **2009,** 823148. ISSN: 1110-7251 (2009).
12. Li, H., Shang, Y., Zeng, J. & Matsusaki, M. Technology for the formation of engineered microvascular network models and their biomedical applications. *Nano Converg* **11,** 10. ISSN: 2196-5404 (Mar. 2024).
13. Wang, X., Liu, Z. & Pang, Y. Concentration gradient generation methods based on microfluidic systems. en. *RSC Advances* **7,** 29966–29984. ISSN: 2046-2069 (2017).
14. Wu, Z. *et al.* A review on the development and application of microfluidic concentration gradient generators. *Phys Fluids* **36,** 072014. ISSN: 1070-6631 (July 2024).

15. Sporns, O. Graph theory methods: applications in brain networks. *Dialogues Clin Neurosci* **20,** 111–121. ISSN: 1958-5969 (June 30, 2018).

16. Riaz, F. & Ali, K. M. *Applications of Graph Theory in Computer Science* in *2011 Third International Conference on Computational Intelligence, Communication Systems and Networks* 2011 Third International Conference on Computational Intelligence, Communication Systems and Networks (July 2011), 142–145.

17. Harary, F. & Norman, R. Z. Graph Theory as a Mathematical Model in Social Science. en.

18. Tanaka, Y., Eldar, Y. C., Ortega, A. & Cheung, G. Sampling Signals on Graphs: From Theory to Applications. *IEEE Signal Process Mag* **37,** 14–30. ISSN: 1558-0792 (Nov. 2020).

19. Nowak, M. R., Lee, J. & Choe, Y. *A Queryable Graph Representation of Vascular Connectivity in the Whole Mouse Brain* in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* ISSN: 1558-4615 (July 2019), 256–260.

20. Wahl, E. M., Quintas, L. V., Lurie, L. L. & Gargano, M. L. A graph theory analysis of renal glomerular microvascular networks. *Microvasc Res* **67,** 223–230. ISSN: 0026-2862 (May 1, 2004).

21. Almasi, S. *et al.* A novel method for identifying a graph-based representation of 3-D microvascular networks from fluorescence microscopy image stacks. *Med Image Anal* **20,** 208–223. ISSN: 1361-8415 (Feb. 1, 2015).

22. Govyadinov, P., Womack, T., Eriksen, J., Mayerich, D. & Chen, G. *Graph-Assisted Visualization of Microvascular Networks* in *2019 IEEE Visualization Conference (VIS)* 2019 IEEE Visualization Conference (VIS) (Oct. 2019), 1–5.

23. Bossard, A., Kato, T. & Masuda, K. *Supporting reconstruction of the blood vessel network using graph theory: An abstraction method* in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. ISSN: 1558-4615 (Aug. 2012), 5470–5473.

24. Gharib, G. *et al.* Biomedical Applications of Microfluidic Devices: A Review. en. *Biosensors* **12.** Publisher: Multidisciplinary Digital Publishing Institute, 1023. ISSN: 2079-6374 (Nov. 2022).

25. Chartrand, C., Le, A. V. & Fenech, M. Micro-particle image velocimetry for blood flow in thick round glass micro-channels: Channel fabrication and velocity profile characterization. *MethodsX* **10,** 102110 (2023).

26. Fenech, M. *et al.* Microfluidic blood vasculature replicas using backside lithography. *Lab Chip* **19,** 2096–2106 (2019).

27. Bassingthwaighte, J. B. FRACTAL VASCULAR GROWTH PATTERNS. *Acta stereologica* **11,** 305–319. ISSN: 0351-580X (1992).

28. Phillips, J. D., Schwanghart, W. & Heckmann, T. Graph theory in the geosciences. *Earth-Sci Rev* **143,** 147–160. ISSN: 0012-8252 (Apr. 2015).

29. Hagberg, A., Schult, D. & Swart, P. *Reference — NetworkX 3.5 documentation*

30. Eifrem, E. *Neo4j documentation - Neo4j Documentation* en.

31. Bastian, M., Heymann, S. & Jacomy, M. *Introduction | Gephi Desktop Documentation* en.
32. Csardi, G. & Nepusz, T. *python-igraph stable — igraph stable documentation*
33. Shannon, P., Markiel, A. & Ideker, T. *1. Introduction — Cytoscape User Manual 3.10.4 documentation*
34. North, S. *Documentation* en.
35. Ronneberger, O., Fischer, P. & Brox, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation* May 18, 2015. arXiv: `1505.04597[cs]`.
36. Li, Z., Liu, F., Yang, W., Peng, S. & Zhou, J. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Trans Neural Netw Learn Syst* **33,** 6999–7019. ISSN: 2162-2388 (Dec. 2022).
37. Gao, Y. *et al.* Medical Image Segmentation: A Comprehensive Review of Deep Learning-Based Methods. *Tomography* **11,** 52. ISSN: 2379-1381 (Apr. 2025).
38. Oktay, O. *et al. Attention U-Net: Learning Where to Look for the Pancreas* May 20, 2018. arXiv: `1804.03999[cs]`.
39. Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N. & Liang, J. *UNet++: A Nested U-Net Architecture for Medical Image Segmentation* July 18, 2018. arXiv: `1807.10165[cs]`.
40. Soika, J. *et al.* Designing lab-on-a-chip systems with attribute dependency graphs. en. *Proceedings of the Design Society* **4,** 785–794. ISSN: 2732-527X (May 2024).
41. Salame, M. *Blood Flow in Microcirculation: Investigating the Cell-Free Layer with Capillary Microchannels* PhD thesis (University of Ottawa, 2025).
42. Salame, M. & Fenech, M. *High-Resolution Micro-PIV Velocity Profiles and Cell-Free Layer Imaging of Blood Flow in Capillary Microchannels* 2025.
43. Salame, M. & Fenech, M. Key contributors to cell-free layer formation: An experimental investigation of hematocrit and shear rate gradient. *Microvasc Res,* 104859 (2025).
44. Salame, M. & Fenech, M. In vitro blood flow dataset in circular microchannels: High-speed videomicroscopy, micro-piv, and cell-free layer measurements. *Data Brief,* 112367 (2025).
45. Pizer, S., Johnston, R., Ericksen, J., Yankaskas, B. & Muller, K. *Contrast-limited adaptive histogram equalization: speed and effectiveness* en. in *[1990] Proceedings of the First Conference on Visualization in Biomedical Computing* (IEEE Comput. Soc. Press, Atlanta, GA, USA, 1990), 337–345. ISBN: 978-0-8186-2039-3.
46. Sato, Y. *et al.* Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Med Image Anal* **2,** 143–168. ISSN: 1361-8415 (June 1998).
47. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern. Syst.* **9,** 62–66. ISSN: 2168-2909 (Jan. 1979).
48. Chayer, B., L Pitts, K., Cloutier, G. & Fenech, M. Velocity measurement accuracy in optical microhemodynamics: experiment and simulation. en. *Physiol Meas* **33,** 1585–1602. ISSN: 0967-3334, 1361-6579 (Oct. 2012).

49. Trigueros, J. R. A. & Fenech, M. Use of graph theory for biomimetic microchannel network blood flow analysis. en. *CMBES Proceedings* **47.** ISSN: 2371-9516 (May 2025).

APPENDICES

Chapter A
Supporting information

A.1 Graph creation code

Custom functions to help graph creation. The first function **trim** deletes all successors of a selected node. The second function **trim_from** applies this trimming process over a range of nodes. After trimming, the outlet node was connected to all terminal vessels. A helper function **add_last_edge** identified nodes without successors, organize them on an array and linked them to the outlet, returning a complete graph connecting the inlet and outlet with all the spanning nodes representing the vessels of the retina chip.The implementation is shown in Table A.1.

Table A.1: Python functions for graph construction starting with trimming functions and node linking.

```
def trim(graph,node,max_degree):
    #Get maximum generation
    degree = int(np.floor(np.log2(node)))
    # Calculate difference between the current node and max
    diff_degree = int(max_degree - degree)
    base = node
    # Transform the node to binary number
    base = bin(base)[2:]
    # Create a list with all edges connecting the subgraph
    edges_num = list(range(1,np.power(2,diff_degree + 1)))
    # Sum of the partial coordinates and coordinates
    edge_partcoor = [bin(num)[3:] for num in edges_num][1:]
    edge_coor = [base + binary for binary in edge_partcoor]
    # Transformation to decimal
    edge_realnum = [int(binary,2) for binary in edge_coor]
    graph.remove_nodes_from(edge_realnum)
    return graph

def trim_from(graph, first_node, last_node, max_degree):
    for i in range (first_node, last_node + 1):
        graph = trim(graph,i,max_degree)
    return graph

def add_last_edge(graph, last_node):
    # Identify nodes with no successors
    nodes_without_successors = [node for node in graph.nodes() if not any
       (graph.successors(node))]
    # Add an edge between them and the last node
    for node in nodes_without_successors:
        graph.add_edge(node,last_node)
    return graph
```

Lastly, to organize the complete network in an easier and user friendly manner, the helper function **organize_nodes** provides a layout based on the node's generation. The code is presented on Table A.2 below.

Table A.2: Python function for organizing the nodes in a readable manner with a generation-based layout.

```
def organize_nodes(graph):
    nNodes = list(graph.nodes()) # List of nodes
    x_coor = [ int (np.floor(np.log2(node))) for node in nNodes[1:]]
    x_coor.insert(0,-1) # Insertion of 0 at the begining of the x values
    num_order = np.array([])
    for i in range(-1,max(x_coor) + 1):
        idx = [ num == i for num in x_coor]
        num_order = np.append(num_order, sum(idx))
    y_coor = np.array([])
    for i in num_order:
        y_coor = np.append(y_coor, np.array(range(0,int(i))))
    y_coor = y_coor.tolist()
    Y_coor = [int(y) for y in y_coor]
    pos = {node: (x_coor[i], Y_coor[i]) for i, node in enumerate(nNodes)}

    options = {
        "font_size": 20,
        "node_size": 3000,
        "node_color": "white",
        "edgecolors": "black",
        "linewidths": 5,
        "width": 5,
    }
    return pos, options
```

A.2 Extensive description of dataset

The data set is formed by two main differences on the experimental setup, having the first half composed by images from vessels within the retina chip and the second half composed by images coming from straight channels. Figures A.1 and A.2 illustrate the differences of both experimental setups.

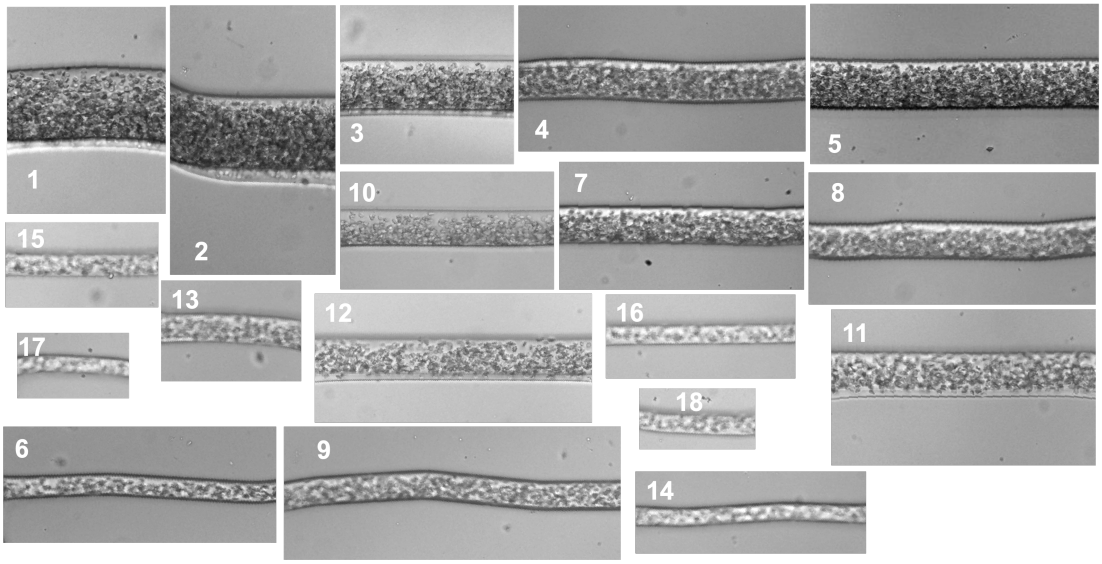

Figure A.1: Representative images from the first 18 groups of the dataset used for CFL segmentation coming from the vessels within the retina chip, illustrating variations in size, lighting and acquisition conditions. The group where they came from is indicated on each image.

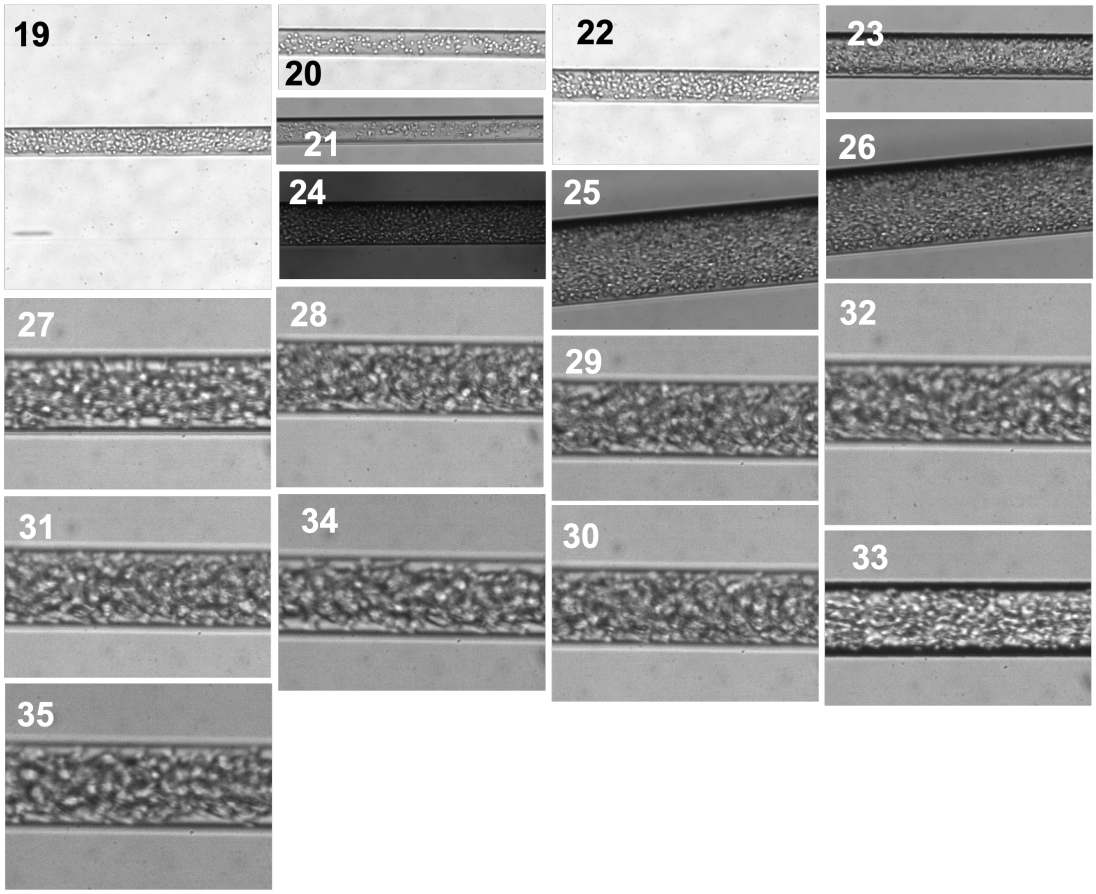

Figure A.2: Representative images from the last 17 groups of the dataset used for CFL segmentation coming from straight channels, illustrating variations in size, lighting and acquisition conditions. The group where they came from is indicated on each image.

Additionally, to better illustrate the diverse dataset, the following table shows the image size along with the number of images within that group and the category it falls in out of the 4 identified scenarios.

Table A.3: Table Showing the diversity of sizes and number in the different groups and their category. 1: bright cells with bright background, 2: bright cells with dark background, 3: dark cells with bright background, 4: dark cells and dark background.

Group	Size	Number	Category
1	225×292	40	4
2	234×380	40	4
3	246×225	20	3
4	407×207	40	4
5	409×226	40	4
6	387×165	40	4
7	346×180	40	4
8	406×186	40	2
9	476×188	40	4
10	302×167	40	2
11	375×220	40	3
12	393×182	40	3
13	198×143	40	3
14	326×117	40	2
15	217×121	40	3
16	268×118	40	3
17	158×92	40	3
18	164×86	40	3
19	964×1028	200	1
20	1013×323	200	1
21	1001×251	200	2
22	858×515	200	1
23	898×359	200	2
24	946×382	200	2
25	983×587	200	2
26	983×587	200	2
27	776×555	294	2
28	826×620	200	2
29	793×489	200	4
30	788×579	200	4
31	773×525	200	2
32	741×668	200	4
33	783×513	200	2

34	761×561	200	4
35	710×549	200	4

A.3 Mask enhancement code

Custom code to enhance the binary masks segmenting the RBCs core by using both the orignal masks coming from the SAM and the gradient images. The implementation is shown in Table A.4.

Table A.4: MATLAB code to mask enhancement.

```
% Obtaining mask to delimit the ROI
Avg_Mask = mean(SAM_Masks,3); % Average from the SAM masks
Avg_Mask = Avg_Mask > 0.2; % Binarization by threshold
disk_d = 3; % Disk diameter MODIFY ACCORDINGLY
sem = strel("disk",3); % Creation of structural element
sem2 = strel("disk",3)
Avg_Mask = imdilate(Avg_Mask,sem); % Image dilation
M1 = imerode(Avg_Mask,sem2); % image erosion

% Modifying gradient image to delimit better RBCs
im_number = 1;
up_th = 120; % Upper threshold
lw_th = 0; % Lower threshold
% Thresholding the gradient images
M = SAM_Grad(:,:,im_number) > lw_th & SAM_Grad(:,:,im_number) < up_th;
M = M.*Avg_Mask; % Combination of average mask and enhaced mask
M = M > 0; % Binarizing the result
se1 = strel("disk",1); % Creation of structural element
Iden = imerode(M,se1); % Image erosion to remove small elements
Ref_Mask = ~Iden.*M1;
Ref_Mask = Ref_Mask*255; % Resulting reference mask
```

A.4 Figure Enhancement Code

Since the environment have an option to export the displayed data from the different network analyses, an external script was used to enhance the visualization for the different cases of study presented in this work. An example of such code is presented in Table A.5.

Table A.5: Python script for plotting the node values of a progressive analysis to enhance the output.

```
path =r"path/FlowRegressiveAnalysis.zip" #Path containing the zip from
    the analysis

with zipfile.ZipFile(path, "r") as zf:
    data = np.load(io.BytesIO(zf.read("ProgressiveAnalysis.npy")))
    G = pickle.loads(zf.read("SubGraph.pkl"))

Graph = G
#--------- Extract the values of the intended variable-------
nodes = list(sorted(Graph.nodes()))
# Collect variable names
Var_Names = []
for node in nodes: # Iterate through all the nodes within the subgraph
    node_data = Graph.nodes[node]
    for name, value in node_data.items():
        # Check if the variable is scalar
        if np.isscalar(value) or isinstance(value, (int, float)):
            if name not in Var_Names:  # Avoid duplicates
                Var_Names.append(name)
print(Var_Names) #Print available variables within the subgraph

palette = copy(plt.get_cmap('inferno')) # Select colormap for colorbar
palette.set_under('white', 1.0) # Get a white blackground for those
    values under a value
v_max = np.max(Var_Val)

# Get divisions for colorbar
levels = np.arange(0, v_max, 0.05)
levels[0] = 1e-3
norm = colors.BoundaryNorm(levels, ncolors=palette.N)

im = plt.imshow(data*v_max, cmap=palette, norm=norm, aspect='auto')
cbar = plt.colorbar(im, extend='min', shrink=1)
cbar.set_label('Flow [\u00B5l/min] ', rotation=90, labelpad=15)
plt.axis('off')
plt.show()
```