

École polytechnique de Louvain

Advancing image segmentation of vascularization in contrast-enhanced microCT data

Author: **Sean NACHTRAB**

Supervisors: **Greet KERCKHOFS, Sébastien JODOGNE**

Readers: **Isabelle GENNART, Juliette VANDERHAEGHEN**

Academic year 2025-2026

Master [120] in Computer Science and Engineering

Preface

This thesis was written in the context of a degree in medical informatics, to work on a problem at the crossroads of two fields, and as a result has two supervisors. It made use of data collected during a previous thesis collected before certain methodological improvements were implemented.

I am deeply thankful for the guidance I have received over the course of the writing process as well as the trust placed in my capacity to execute a multidisciplinary project by Prof. Greet Kerckhofs and Prof. Sébastien Jodogne.

This project has seen a wide diversity in tools used which required consultation with members of the MEED team to whom I am thankful for our user interviews and conversations. Finally, I have been able to count on the feedback of my friends and fellow SINF students over the course of writing.

The developed plugin is available at this link under *plugin*, with included instructions and the code required to generate the written report and its figures available on github

The Generative AI tool Piccolo was used during the writing of this thesis for feedback and writing modifications, with details in Appendix A.5. Claude Code was used for support during certain sections of software development, mainly pertaining to the initial setup, 3D Slicer point placement interaction, creation of graphing code and scripting.

Nachtrab Sean

Table of contents

Preface	i
Abstract	iv
1 Introduction	1
2 State of the art	5
2.1 Tissue imaging	5
2.2 Image processing for vessel segmentation	8
2.3 Software for 3D analysis	16
3 Problem statement, goal and objectives	19
3.1 Problem statement	19
3.2 Goal and objectives	19
4 Methodology	21
4.1 CECT datasets	21
4.2 Building a 3D Slicer plugin	25
4.3 User interaction model	26
4.4 The CollaboratiVessel pipeline	28
5 Results and discussion	33
5.1 Hyperparameter sensitivity analysis	33
5.2 Qualitative comparisons	34
5.3 Quantitative comparison	37
6 Conclusion and future perspectives	47
6.1 Limitations	48
6.2 Future work	48
A Appendices	53
A.1 List of software for working with 3D data	53
A.2 Nanotom scan parameters	54
A.3 Compute characteristics	55
A.4 Environmental and CO2 impact of the thesis	56
A.5 Use of LLMs and AIs	57
A.6 Development computer specifications	57
B Supplemental figures	59
B.1 Visualizations of results	59
B.2 Vessel heatmaps	62

B.3 Matching graphs	70
Bibliography	79

Abstract

Tissue biopsy imaging has evolved from 2D microscopy to 3D modalities of ever-increasing resolution, with current contrast-enhanced microCT (CECT) datasets comprising tens of gigabytes and spanning thousands of voxels. The computational tools needed to extract coherent 3D structures from these volumes has lagged behind: existing solutions are either proprietary, only implement simple methods that do not generalize, or rely on deep learning expertise and associated data that is unavailable publicly and difficult to produce. This thesis focuses on CECT datasets of tumor vasculature on which previous standard image segmentation showed limited accuracy, with the main challenges being that vessels can span only a few voxels across, be disconnected, data can contain regions of high signal non vessel structures and contrast staining can be non-uniform.

Therefore, in this thesis an open-source 3D Slicer plugin called CollaboratiVessel is proposed for microvasculature extraction from CECT data, constructed based on CECT datasets from murine tumors collected for the study of the antiangiogenic drug Pazopanib. Rather than requiring annotated data or computer science expertise, the end user is placed in the algorithmic loop by providing sparse input to an extraction pipeline that combines multi-scale vesselness filtering with ridge following and reconnection, ingesting and emitting standard formats compatible with downstream analysis software. CollaboratiVessel is evaluated on CECT datasets with manually annotated ground truths, and is shown to recover vascular structure with improved connectedness than currently used intensity-based thresholding methods, supporting future research on vascular changes and enabling work on structures with more challenging vasculature than was previously possible.

Chapter 1

Introduction

Biological tissue function depends strongly on spatial organization of its constituents, its cells and extracellular matrix. This is particularly true for vascular networks in organs and tumors, whose branching geometry, vessel diameters, and connectivity collectively determine how functional the vascularization is. Interpretation of the effect of diseases or drug treatments on tissue vascularization depends therefore on accurate imaging data extraction and subsequent structural quantification. Vascular networks have higher-order structural properties: vessel densities, diameters, branching frequency and changes in tortuosity which cannot be reliably recovered from 2D images, nor from voxel-level metrics in 3D datasets that ignore network topology. This 3D nature motivates the use of high-resolution 3D imaging able to discern small vessels inside of tissues.

In this work, in order to examine vessels in ex-vivo biopsies of murine tumors, micro-computed tomography (microCT) is used to generate high resolution 3D datasets. It suffers however from low contrast between soft tissues, for which contrast-enhancing agents were utilized to increase vessel signal strength, a technique termed contrast-enhanced microCT (CECT).

High resolution 3D imaging brings with it new challenges when compared to 2D histology: a single CECT dataset can reach into the thousands of slices, with large datasets on the order of 3000^3 at 16 bits per voxel, resulting in orders of magnitude more data than a comparable 2D acquisition. This means that manual analysis as with 2D microscopy is impractical. CECT data itself is also highly variable and challenging to analyze: datasets differ on the same machine due to changes in the imaged tissue, contrast agents and protocols used to make structures of interest visible, and further variation arises across machines and due to researchers' choices of acquisition parameters. Microvasculature presents additional challenges, illustrated in Figure 1.1: vessels are only a few voxels across with low contrast and many discontinuities, the surrounding tissue can contain high-signal non-vessel-like structures, as well as shell artefacts and contrast gradients across the imaged volume.

Analysis techniques for CECT data can broadly be grouped into two, both with significant drawbacks. Data driven techniques such as machine learning exhibit poor out-of-distribution performance, needing retraining for new datasets or target structures. This requires technical expertise with large quantities of annotated training data that is challenging to produce. Classical algorithmic techniques avoid the need for annotated data but are typically delivered

through host software to enable user interaction, which is a fragmented landscape of closed source, paid proprietary commercial software, or open-source software with variable quality. Neither technique has robust out-of-the-box solutions for microvasculature extraction. As a result vessels in datasets like ours are unable to be accurately and reliably extracted, which prevents downstream characterization of tissue vascularization.

Classical algorithmic methods avoid the data-acquisition problem but are typically delivered through host software, and the available host platforms form a fragmented landscape of paid proprietary commercial tools and open-source tools of variable quality

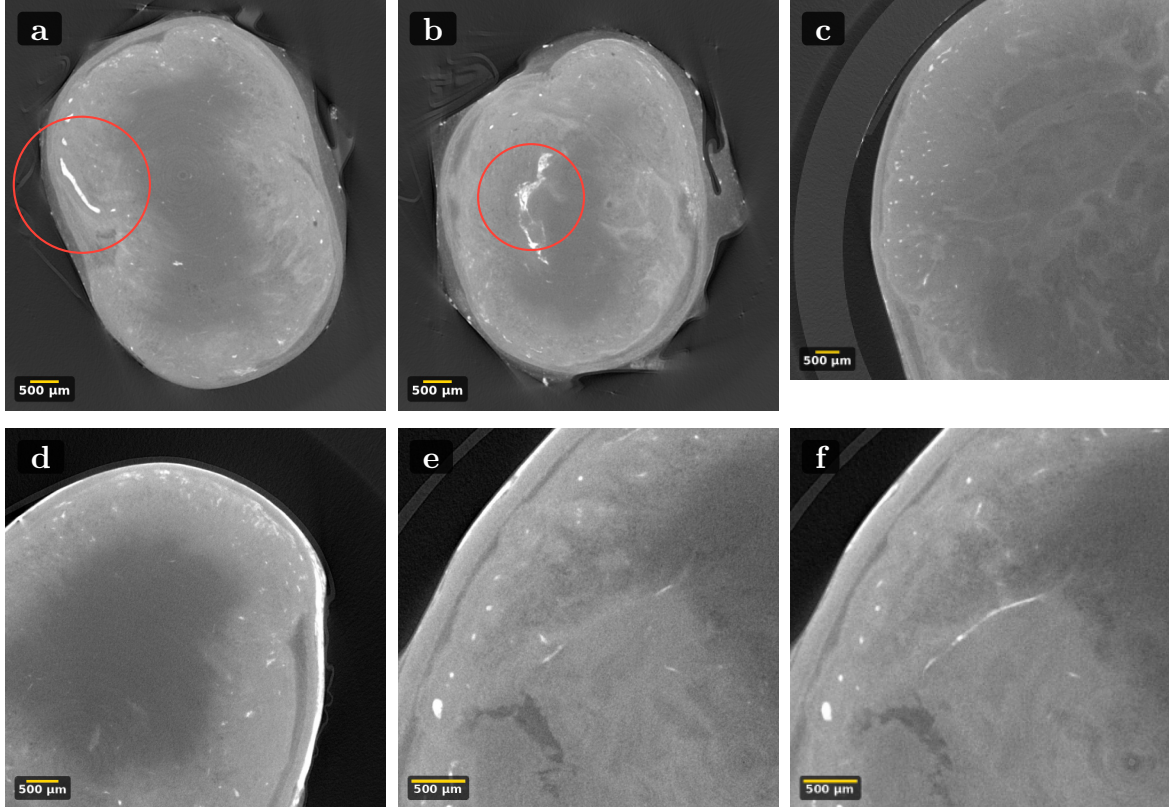


Figure 1.1: Slices illustrating the principal challenges: **(a)** A large high contrast vessel of approximately 14 voxels (indicated by the circle). **(b)** A large high-intensity non vessel-like structure resulting from hemorrhage. **(c)** Upper, left: many small vessels in cross-section, ranging from 2 to 8 voxels in diameter, where the partial-volume effect is present, as well as compression artifacts. **(d)** Outer surface: the “shell effect”, a high-intensity boundary surrounding the tumor caused by the limited diffusion of the contrast agent - also visible: a strong gradient between outside and center **(e, f)** Two slices from the same volume, separated by 5 voxels along the z-axis. The vessel indicated appears discontinuous in (e) but continuous in (f), highlighting the relevance of 3D methods.

Contribution and scope

To address the challenges of microvasculature extraction from CECT data, an open-source pipeline called CollaboratiVessel is created in this thesis to obtain segmented blood vessels,

with the goal of enabling downstream extraction of biologically relevant vasculature characteristics.

CollaboratiVessel is an easily installable plugin for 3D Slicer, an open-source and free base software, that ingests data in DICOM and outputs binary masks, a portable data formats familiar to researchers, to ensure re-usability and compatibility with existing downstream analysis. To segment vessels CollaboratiVessel implements multi-scale vesselness filtering, seed-guided region growing, ridge following and endpoint reconnection. Hyperparameter tuning is handled by providing robust and tested standard parameters while keeping those relevant for common domain shifts in resolution, vessel size and intensity exposed, enabling adaptation to new datasets. User input is leveraged in the segmentation loop with manually placed points guiding segmentation, and ease of use is prioritized to bridge the divide between computer science and researchers. Final performance is measured against a manually annotated baseline and intensity thresholding, the previously used technique that did not succeed in segmenting with sufficient consistency for subsequent analysis. CollaboratiVessel outperforms thresholding on vessel consistency and continuity, extracting vessels more relevant for downstream analysis, and exceeds manual annotation in extrapolation capabilities, while requiring minimal human intervention and maintaining repeatability and consistency across runs.

All code written in the implementation of this thesis is made open-source under the MIT license, and the source of this document is labeled under the creative commons as CC0 - No Rights Reserved.

Chapter 2

State of the art

2.1 Tissue imaging

2.1.1 Classical 2D histology

The process of imaging is a critical step in obtaining an understanding of a tissue biopsy in both clinical and research contexts. Classical 2D histology is considered the gold standard having a long history and producing high resolution 2D images down to 5 μm for light based methods [1]. The diversity in imaging techniques for classical 2D histology has rapidly increased in resolution, with electron microscopy able to reach in the order of sub nanometer scale [2]. The dimensionality of imaging has also evolved with computation enabling rapid progress in 3D methods using techniques like tomography, where multiple images are taken non-destructively of a target from different angles and assembled, being used in-vivo and ex-vivo with methods such as computed tomography (CT) and magnetic resonance (MRI).

For the analysis of inherently 3D structures such as vascular networks in tissues, classical 2D histology is insufficient: physical sectioning of the sample is destructive, is not orientation-agnostic, and introduces deformation artifacts that are difficult to compensate for even with embedding techniques a problem described in Silva *et al* [3]. Additionally, samples may undergo structural changes over time during preparation: they dry out, and certain elements oxidize, although techniques exist to mitigate these disadvantages as explored in de Bournonville *et al* [4].

To achieve 3D reconstruction/visualization from 2D histology, 2D slices are stacked across a virtual axis [5]. The resolution achievable when stacking 2D slices to reconstruct a virtual third dimension is limited by slice thickness and inter-slice registration errors. Certain optical techniques making use of slices allow partial recovery of 3D information from 2D acquisitions, such as confocal microscopy, light sheet microscopy, and optical coherence tomography [6]. However they are limited in sample penetration depth and volume. When the target structure to be imaged and understood does not follow a single preferred axis, as is fundamentally the case for vascular networks, the limitations of slice-based histology

make proper reliable quantification of blood vessel parameters impossible, and the relevance of 3D methods like MRI and CT clear.

2.1.2 3D histology

Non-destructive 3D volumetric imaging methods overcome many of the limitations of 2D slice based histology by collecting data across dimensions, enabling virtual slicing across any plane without requiring physically sectioning the sample. Methods for 3D imaging range from the aforementioned microscopy using visible or near infrared light, to those like MRI that utilize magnetic fields or X-ray imaging in the form of X-ray computed tomography (CT). These techniques allow the collection of qualitative and quantitative 3D microstructural data of tissues and their constituents: analysis is not restricted to a single orientation and does not require sample destruction. They also have associated high resolution variants allowing micrometer level imaging: MicroMRI has an inherently high-contrast for soft-tissue and is able to reach 20 μm resolution as used in Chen *et al* [7]; Microfocus CT is commonly used at micron-scale, and resolutions down to 1 μm are possible, leveraged in Balcaen *et al* [1]. Although microCT suffers from low soft tissue contrast, it still stands out as most relevant for ex-vivo tissue histology due to the high resolutions achievable, relevant at the scale of tissue biopsies, and relatively short capture times when compared to microMRI.

2.1.3 Contrast-enhanced microCT (CECT)

To improve the contrast between soft-tissue in microCT, two common approaches exist: changes to imaging methodology, and modifications to the sample. Imaging techniques such as phase-contrast microCT (PC-CT) utilize the refractive properties of the X-rays rather than its attenuation alone, enhancing soft tissue edge detection at the cost of higher complexity and long acquisition times [8]. Tissue modifying techniques that aim to increase contrast include the use of various Contrast Agents referred to as contrast-enhanced microCT (CECT): Casting contrast agents (CCA) are perfused into vasculature but pressure must be controlled carefully as used in Leyssens *et al* [9], this is not applicable to microvasculature due to the size of vessels to be observed. Contrast-enhancing staining agents (CESAs) diffuse through the tissue to reach the structures of interest, and bind to it. For the tissue vascularization studied in this thesis, ex-vivo tissue binding CESAs are used, namely Hafnium Wells-Dawson Polyoxometalates (Hf-WD POM) which bind to the red blood cells in the vessels, thereby identifying the vessel lumen or interior. These are selected due to their nondestructive nature and low tissue shrinkage during incubation after excision. Techniques exist that can be used in combination with CESAs: contrast can be increased using freezing in Cryogenic CECT (cryo-CECT) which preserves tissue microstructure with reduced deformation, proposed in Maes *et al* [6].

However, in case the vessels are partially or entirely devoid of residual red blood cells, it prevents or reduces the action of CESAs and thereby reduces contrast and introduces discontinuities. CESAs can potentially bring additional disadvantages as seen in the data provided for this thesis, visible in Figure 2.1: the diffusion of CESAs throughout the tissue happens from the outside in, resulting here in a gradient of the amount of CESA and consequently a gradient in contrast, as opposed to CCAs that perfuse the blood vessels. Additionally, non-vessel-like structures such as hemorrhage or handling artifacts also generate high signal areas outside of vessels.

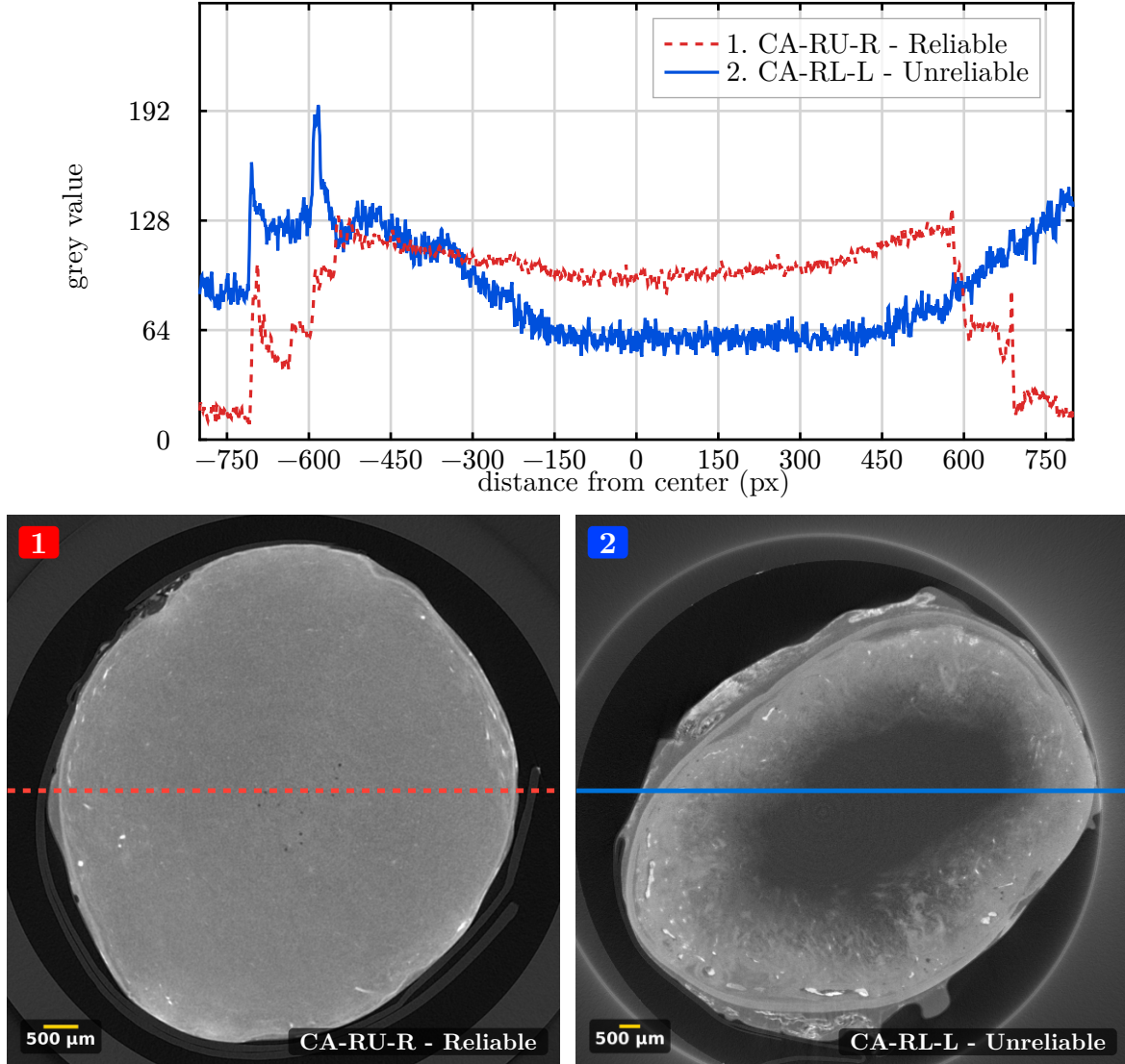


Figure 2.1: Cross section of CECT datasets:

Top: Illustrative grey values of a reliable dataset (Red) and unreliable (Blue). An ideal dataset would be have an approximately even greyscale value across the tumor.

Bottom Left: CA-RU-R, considered reliable. **Bottom Right:** CA-RL-L, considered previously unreliable.

Problem 1.

The CECT data in this thesis presents a variety of challenges in the form of non-uniform CESA diffusion causing grey-value gradients, varying vessel contrast levels, disconnections and non-vessel-like structures. The subsequent imaging processing and analysis pipeline must therefore be robust to these elements and work on all dataset samples.

2.2 Image processing for vessel segmentation

The challenges introduced by CECT shape what each paradigm of segmentation method can achieve. *Image acquisition* and *information extraction* have evolved together: contrast agents make structures more salient and increased resolution makes them more granular. Information extraction methods have progressed from manual inspection through the three paradigms of classical methods: thresholding and other histogram based methods, to hand crafted geometrically motivated filters and path-based methods, and finally data-driven learning approaches.

Vasculature extraction from 3D tomographic data is a longstanding challenge in computer science, predating deep learning as reviewed in Lesage *et al* [10]. The 3D CECT data of this thesis carries a unique set of challenges specific to modern high resolution microCT: the vessel diversity ranges in the supplied data from singular voxels (6 μ m) to tens of voxels (100+ μ m) across, corroborated by Ehling *et al* [11]. The vasculature is diverse in shape and branching structure, with variable grey levels and most critically disconnections of variable size. Despite this, vessels share a distinctive and consistent set of geometric priors: they are tubular, connected, branching structures and contain blood, providing a distinct imaging profile against surrounding tissue when combined with a CESA. This geometry is what subsequent segmentation methods either *exploit explicitly* through hand crafted classical filters or *learn implicitly* in data-driven methods, illustrated in Figure 2.2.

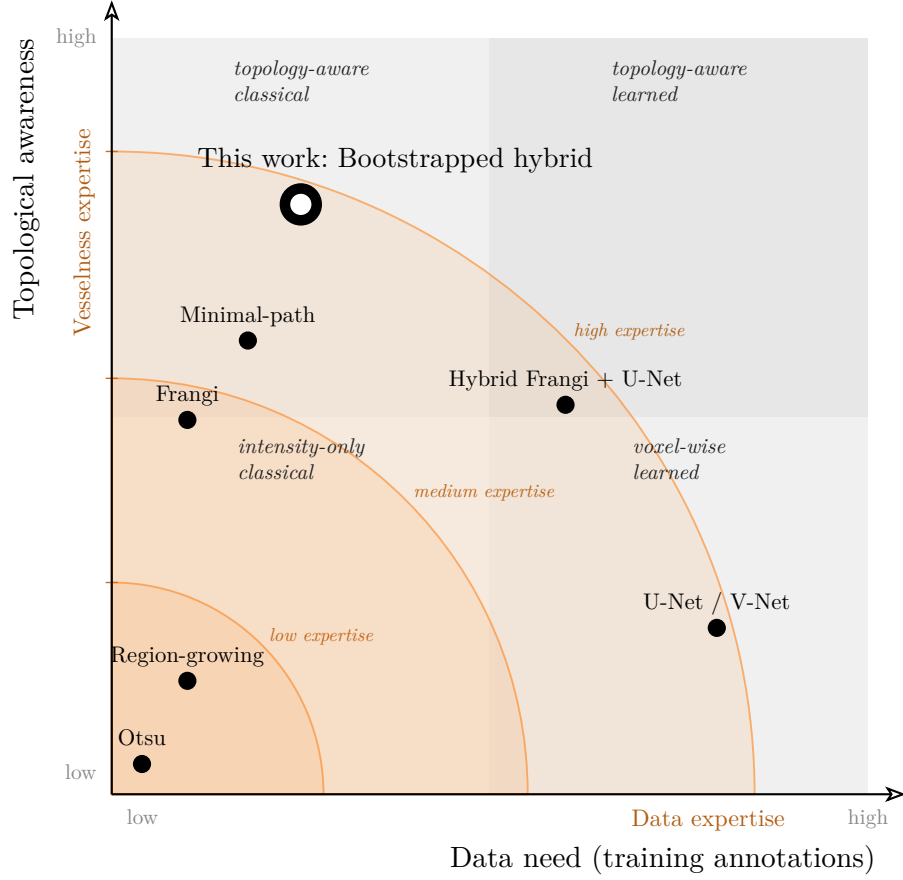


Figure 2.2: **Vascular segmentation methods positioned by data need (x) and topological awareness (y).** Orange arcs indicate the required level of user expertise: methods within the innermost arc require minimal expertise in either discipline (simple threshold selection), those within the middle arc require moderate familiarity (vessel specific parameter tuning), and those in the outer arc require significant expertise (combined vessel and annotation knowledge, and model training for the deep learning approaches). Classical methods cluster on the left with low data needs; data-driven methods cluster on the right. Our hybrid approach combines vesselness-based topological awareness with limited user input while still requiring the ability to identify vessels, sitting in the low-data, high-topology quadrant, and with a relatively higher vesselness expertise needed than classical techniques alone.

2.2.1 Classical segmentation

Intensity-based methods

In its simplest form, intensity-based thresholding methods such as Otsu [12] select an optimal global intensity threshold by maximizing inter-class variance across the image histogram. Such methods are computationally inexpensive and interpretable, but are sensitive to noise, imaging artifacts, and intensity inhomogeneities, as they optimize for a numerical goal.

Gradient-based and global-threshold methods are a natural first approach to CECT data, since contrast agents are intended precisely to increase the contrast of the structures of interest. However, methods that reason globally over the image can fail in practice due to diffusion gradients and acquisition noise.

Region-growing methods [13] extend thresholding by incorporating spatial connectivity: starting from one or more seed points, labeled regions are iteratively expanded according to local intensity criteria, such as in flood-fill, and generalized further by methods such as watershed [14], which treats the image as a topographic surface. Incorporating this spatial connectivity prior makes them more robust to global intensity variation, but remain susceptible to over- or under-segmentation in complex structures: they fail to encode the more specific vascular priors, producing fragmented segmentations in the low-CESA, weak-signal regions characteristic of diffusion-CECT data.

Geometrically motivated methods

Beyond intensity-based methods, algorithms integrating local geometric priors exist such as Gabor filters or Hessian-based filters, established by Sato *et al* [15], where the local second-order structure is analyzed to detect tubular shapes, characterized for vessel detection by Frangi’s multiscale vessel enhancement filter proposed in Frangi *et al* [16], with extensions such as the Jerman filter by Jerman *et al* [17] and many other methods being created since. Despite attempts to improve Frangi, the reference implementation remains highly competitive, being shown to consistently offer the highest true positive to false positive rates when compared to 6 methods as shown by Lamy *et al* [18].

These second-order methods utilize the eigenvalues of the Hessian matrix of local image intensities (capturing how rapidly intensity changes are themselves changing - the local curvature of the intensity surface) at multiple Gaussian scales (corresponding to multiple candidate “tube” sizes) to produce a vesselness score for each voxel, modeling the prior of blood vessels by responding to tubular structures while suppressing blob- and plate-like ones. Frangi and related methods offer greater robustness than simple thresholding but depend on the user fine tuning algorithm hyperparameters to optimize performance for a given domain or usecase when working with a tool such as for imageJ [19]. Additionally the filter’s response degrades at vessel bifurcations, at the endpoints of vessels, and in regions of low contrast. Jerman *et al* [17] proposes extensions to the vesselness formulation that improve responses at junctions and at low-contrast boundaries to increase robustness of the filter.

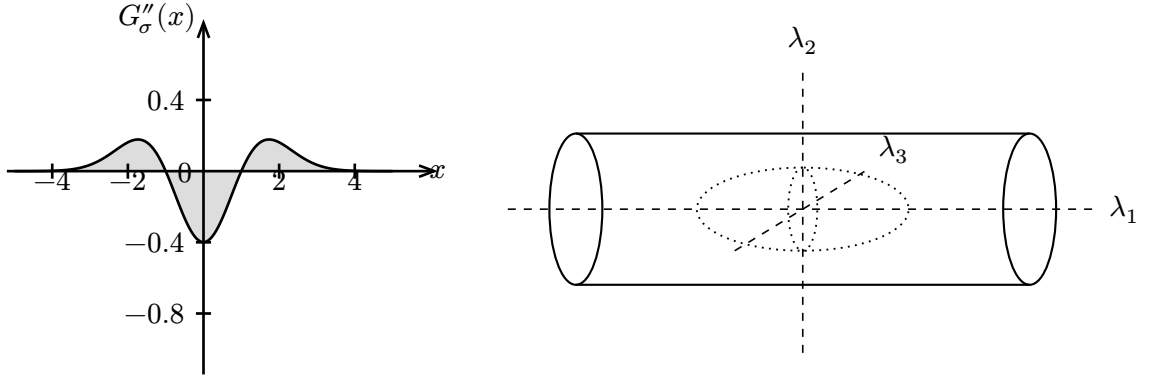


Figure 2.3: The Frangi vesselness filter: Figure adapted from [16].

Left: The second-order derivative of a Gaussian kernel G''_σ probes inside-versus-outside contrast over the spatial range $(-\sigma, \sigma)$; here $\sigma = 1$. Convolving an image with this kernel measures for each point how strongly intensity curves over the scale *sigma*: high response means a detected peak or valley of width comparable to *sigma* - “matching” vessels.

Right: At each voxel, the eigen-decomposition of the Hessian matrix yields three principal directions of curvature $\lambda_1, \lambda_2, \lambda_3$. For a tubular structure, the eigenvalue along the vessel axis (λ_1) has small magnitude (intensity is roughly constant along the vessel) while the two perpendicular eigenvalues (λ_2, λ_3) have large magnitude and the same sign (reflecting the drop in intensity in either direction perpendicular to the vessel) this constitutes the geometric signature Frangi’s filter exploits and are combined to form the Frangi vesselness score.

Although offering high performance, second-order methods are limited to working on local image structure at multiple scales without accounting for global vessel topology, failing to incorporate the overarching goal of vasculature extraction: in Figure 2.2 they fall in the center left. A different class of methods reformulate extraction as a global optimization rather than a local filter response, such as minimal path methods which formulate vessel centerline extraction as an optimization problem. The path of least cost is taken between user-defined endpoints, with cost derived from vesselness or image intensity. Benmansour *et al* [20] integrate the geometry of vessels, adding radius as an additional dimension of the path space. This class of methods is particularly well-suited to expert-in-the-loop approaches where small amounts of data need to be annotated: a user can place seed points that guide the extraction of a complete vascular tree, requiring no training data, and enabling iterative improvement. These methods increase the topological awareness while also marginally increasing data requirements, illustrated by a move up and to the right Figure 2.2. This method can be seen used for large arteries and airways available in tools such as the Vascular Modeling Toolkit (VMTK) [21]. This strength is also a weakness: the low scalability means that for a densely vascularized tissue, or a tissue with non-uniform characteristics, many seed points can be required which is impractical. Automatic seed point placement is a potential solution although introducing its own tradeoff, moving the optimization and work into the point placement.

2.2.2 Unreliability of ground truth

Manual annotation of vascular networks in CECT images is time intensive and requires expert knowledge. When annotation is carried out, the resulting annotations exhibit significant inter-annotator variability, shown by Lin *et al* [22]. This variability represents an irreducible uncertainty in the ground truth that places an upper bound on the performance achievable by any supervised method, especially those training on voxel-level data. Work-arounds exist: structural extraction (e.g., skeletonization) may be used on the data first, with the resulting structures used to guide subsequent voxel-level re-annotation, reducing inter-annotator noise. However this means the biases of the skeletonization algorithm are now inherited by the annotations and introduced into models trained on them.

An alternative to real annotated data is the use of simulated and automatically annotated data, using either geometrically driven techniques such as L-Systems, to generate branching tree structures grounded in the physiological laws of arterial branching, as reviewed by Zamir *et al* [23], or neural-network driven systems such as generative adversarial networks (GANs), reviewed for medical applications by Yi *et al* [24].

When data is available it may be used directly, although models trained on one imaging protocol, contrast agent, sample batch, or tissue type exhibit degraded performance when applied to data from a different distribution: this is termed domain shift, and is particularly acute in imaging methods containing many adjustable parameters and machine hardware specificities as in CECT imaging. This can require the use of techniques like transfer learning, where a model is first trained on a similar task with large data availability then fine-tuned on a small amount of in-domain data as used for vessels by Tetteh *et al* [25]. Annotated data can also be *augmented* to artificially increase the available amount and diversity by applying random degradations, elastic deformations and intensity perturbation while preserving the original annotations.

Problem 2.

Data from different datasets is not necessarily directly usable as a proxy without careful consideration. In-domain data manually annotated for performance measurement, especially by a non domain expert, cannot be considered an absolute ground truth. For the purpose of evaluating this work, data will be annotated in distribution, and calculated errors should take into account the potential variability from annotations.

2.2.3 Data-driven learning methods

Data-driven methods approach extraction from a different direction: rather than hand-engineering features and priors, parameters are learned from labeled examples. Supervised learning algorithms map labeled inputs to outputs with classical machine-learning approaches such as kNN and gradient-boosted trees by Chen *et al* [26] being used for classification, but these do not scale to our data. With the deep learning paradigm exploding in popularity since being introduced by Krizhevsky *et al* [27], higher dimensionality inputs became analyzable without hand crafted features, and led to fully convolutional encoder-decoder architectures being used in the medical domain such as with U-Net proposed by Ronneberger *et al* [28], a leap in performance for image segmentation.

U-Net specifically constituted a breakthrough in medical imaging due to its ability to segment large images with high compute and data efficiency, and across multiple scales by introducing skip connections between encoder and decoder pathways. These allow the network to combine low-level spatial detail with high-level semantic context, enabling the segmentation of thin structures that require context such as cells and vasculature. The structure of U-Net can be extended into 3D with different approaches such as 3D U-Net proposed by Çiçek *et al* [29]. Recently, transformer-based architectures have superseded convolutional networks as a more generalist approach to machine learning, used for object detection in DETR by Carion *et al* [30], and being extended to segmentation with paradigms such as Segment Anything by Kirillov *et al* [31], offering higher performance thanks to a more general computation model. Expert crafted feature extraction is removed, such as in the locality prior of convolutions, enabling the capture of more diverse features and long-range spatial dependencies, at the cost of an increase in required training data, an issue for CECT.

As algorithm complexity and data dimensionality increase, so must the amount of training or example data increase, an issue for convolutional networks and even more so for transformers. This limits the application of deep learning to CECT data, where annotated data is expensive and where inter dataset variance is large, even though attempts exist to palliate the high data requirements by offering self-configuring training frameworks with model weights as explored in nnUnet by Isensee *et al* [32] or by crafting more data-efficient models. Examples also exist of trying to make use of so called foundation models, based on a large transformer and able to be “prompted” to customize the segmentation to the usecase at hand, such as TomoSAM for 3D Slicer by Semeraro *et al* [33].

Deep learning has been applied to vascular segmentation for blood vessel extraction in low noise environments directly with notable success: DeepVesselNet by Tetteh *et al* [25] introduced a family of architectures designed for vessel segmentation, centerline prediction, and bifurcation detection in 3D angiographic data by integrating the structural priors inherent to blood vessels as secondary learning targets. The model was trained on murine brain data, presenting different vessel structures, sizes and greyscale values to our tissues, with vessels tens to hundreds of voxels across, few discontinuities, and vessel intensities with markedly low or non existent overlap with the background. Re-use of such a model would require at a minimum a fine-tuning step, needing an undefined amount of annotated data, with the authors admitting that vessel sizes outside of their training distribution suffer from reduced performance.

Finally, hybrid approaches offer the interesting property of combining classical extraction methods with deep learning by leveraging input filters on the image to obtain richer features for learning, such as applying vesselness maps to the image before processing, to integrate the priors of the vesselness filtering explicitly into the algorithm and reduce data needs explored for u-Net by Chen *et al* [34]. This decouples the structural prior, encoded by the filter, from the learning, allowing the learning element to act as a correction or enhancement stage for the shortcomings of classical methods. This use of classical priors as a feature stage with a form of learning as a correction sets the direction of the present work, with the novel addition of sparse user input as a method for achieving training data acquisition.

Problem 3.

Segmentation has branched from classical methods encoding geometric priors toward data-driven methods, requiring annotated training data that is difficult to acquire and high variance for CECT. The work should need no training data while leveraging the user to provide simple annotations, and present minimal relevant and pre-tuned hyperparameters to enable portability across diverse data.

2.2.4 Error calculation

Evaluating vascular segmentation is challenging because the metrics that are easy to compute and commonly used for segmentation do not reflect the failures that matter for downstream analysis. Three families of metrics are surveyed: voxel-level losses, topology-aware losses, and object-level matching, with each addressing a limitation of the previous.

Voxel-level metrics are commonly used for segmentation as in UNet [28]. Vessel segmentation has a severe class-imbalance with vessels being a small minority of voxels, and as a result the choice of loss function matters: voxel-level losses like cross-entropy treat each prediction as independent, meaning there is a bias toward predicting background and producing fragmented or incomplete vessel predictions. Milletari *et al* extended U-Net into 3D with V-Net [35] and improved performance by using DICE, a method better suited for class-imbalanced settings.

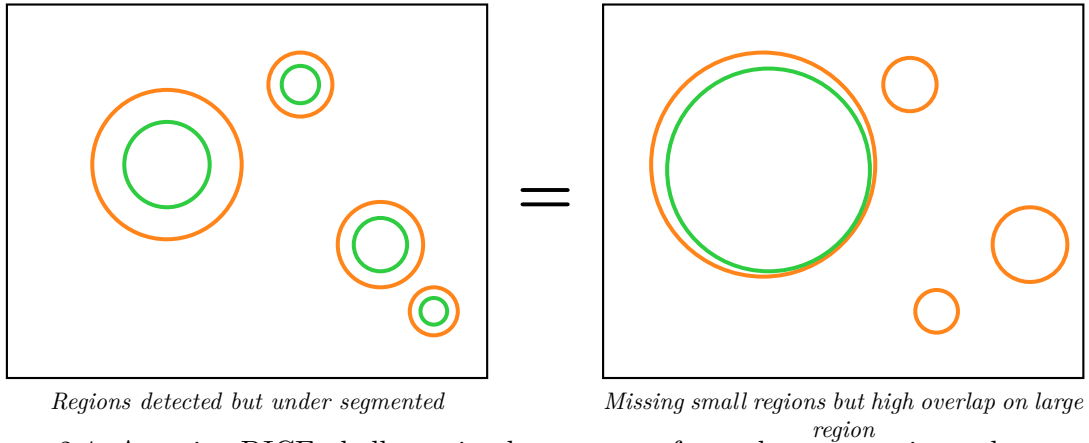


Figure 2.4: A major DICE challenge in the context of vessel segmentation, where many small vessels (orange) are present but detections (green) in one case totally fail to pick up on certain vessels. These situations would have a similar DICE score, showing the bias towards large overlapping regions, as discussed by Taha *et al* [36].

DICE has some specific known downsides for 3D medical image segmentation discussed by Taha *et al* [36], and relevant here: the equal treatment of false positives and false negatives, the unawareness of spatial differences with voxel differences being treated as equal between large and small vessels, and the detection gap where small completely undetected areas are treated the same as small mismatches in large coverage areas.

To integrate some of these issues into the loss, Yukun Zhou *et al.* [37] propose CFLoss to

encode clinically relevant vascular features into the loss function. A collection of simpler topology-aware loss functions is available as a python package [38], able to place emphasis on different topological features. For vasculature specifically, breaks in segmentation can be difficult to reconnect downstream and differences in size less important than the matching of the vessel itself. This motivated the creation of a purpose-built loss function called *centerlineDice* (clDice) by Shit *et al* [39] which evaluates segmentations on their skeletons rather than their masks, moving emphasis to shape and connectivity.

$$\begin{aligned} \text{Dice} &= \frac{2 \text{ TP}}{2 \text{ TP} + \text{FP} + \text{FN}} & \text{Dice}(A, B) &= \frac{2 |A \cap B|}{|A| + |B|} \\ \text{Tprec}(S_P, V_L) &= \frac{|S_P \cap V_L|}{|S_P|} & \text{clDice} &= 2 \cdot \frac{\text{Tprec} \cdot \text{Tsens}}{\text{Tprec} + \text{Tsens}} \\ \text{Tsens}(S_L, V_P) &= \frac{|S_L \cap V_P|}{|S_L|} \end{aligned}$$

Figure 2.5: **Top:** Dice for segmentation A and B , in confusion-matrix form (left) and as set intersection (right). Dice is agnostic to true negatives (TN), avoiding their outsized weight in imbalanced settings. **Bottom:** clDice is computed in two steps: Topology precision (Tprec) and topology sensitivity (Tsens) to measure how much of the predicted skeleton S_P lies inside the ground-truth mask V_L , and how much of the ground-truth skeleton S_L lies inside the predicted mask V_P . clDice is their harmonic mean, similar to how standard Dice is the harmonic mean of precision and recall, enabling a simple drop in replacement.

Beyond loss functions, the evaluation itself can integrate vessel structure: graph-matching compares predicted and reference vascular trees at the level of branches and bifurcations rather than voxels, enabling metrics on the branch-level as explored for vessel level predictions by Suprosanna *et al* [40]. In panoptic segmentation proposed by Kirillov *et al* [41], the goal is to not only segment a structure but also individually identify it. Graph inspired methods are used to calculate a score based on the matching between the ground truth and predictions on a per object basis, combatting the issue highlighted by Figure 2.4 where missed vessels are not weighed appropriately. This comes at the cost of needing predictions to be individual objects, a problem for vessels that are *by nature* continuous. To combat this, when comparing matching, our metrics will also account for size and allow one to many or many to one matches. Finally, on top of comparing predicted segmentations to a reference, the user placed seed points are also considered informative: if marked a voxel as vessel, the final segmentation should preserve that classification and provide feedback on these points.

Evaluation criteria

Following the analysis above, outputs will be evaluated based on three complementary criteria: **(1)** simple user placed annotation points during the *vessel* and *background* point placement step, **(2)** on a voxel level with connectivity aware clDice, measured on a manually annotated binary ground truth and **(3)** on a matching between predicted and ground truth on an individual vessel level, to combat the biases of voxel-based methods.

Problem 4.

Evaluation of a segmentation performance requires integrating the structure of the problem. Our evaluation methodology should make use of topology-aware losses in the form of cDice when calculating voxel error rates, and per-vessel object-level matching to avoid voxel level bias.

2.3 Software for 3D analysis

The choice of host platform shapes the extraction pipeline and who can use it. A host software platform suitable for distribution of a microvascular extraction pipeline should therefore **(1)** import 3D scans from open and standard slice formats, **(2)** export to non-proprietary formats for downstream analysis, and **(3)** support coded extensions that can be redistributed and modified. Platforms were evaluated against these criteria, with additionally the practical considerations of documentation, active development and ease of installation.

Candidate platforms used in the CECT community are summarized in Table 2.1, with UCLouvain’s IMMC specifically using closed-source commercial tools: Avizo, Dragonfly3D and CTan. These offer limited extension development and redistribution, as well as requiring commercial paid licenses in the case of Avizo and CTan. Dragonfly3D presents the distinct advantage of currently having a “free-for-academics” *FreED* license enabling widespread use in research, but brings with it a licensing challenge: despite being free to use, *FreED* restricts the redistribution of extensions, preventing the open-source release of derivative work.¹

The lab has prior experience with the extensible and open-source ImageJ/FIJI, but it lacks the established 3D segmentation infrastructure for medical imaging. Standalone vessel-specific tools DeepVesselNet by Tetteh *et al* [25] and SPROUT by He *et al* [43] are not platforms intended for extension but instead use case specific implementations. 3D Slicer, developed from work by Fedorov *et al* [44] is an open-source platform under a permissive BSD-style license, able to import and export a variety of formats and is widely adopted in the medical imaging field. It also supports 3D segmentation workflows with established plugins relevant to vascular analysis such as the Vascular Modeling Toolkit (VMTK) by Izzo *et al* [21] and R-Vessel-X by Affane *et al* [45]. Its Extension Manager allows non-technical users to install plugins through a one-click interface without compiling code or managing dependencies.

¹“[The user] shall not distribute or transfer the Software or Improvements [...], without prior written permission [from Dragonfly3D]” [42]

Problem 5.

Existing tools for 3D analysis are split between proprietary paid software, often restricting plugin redistribution, and open-source platforms, neither having bespoke microvascular support. The vasculature extraction pipeline must therefore be built on an open-source host software for 3D analysis, intended for use in research, and enabling redistribution: 3D Slicer.

2.3.1 Existing pipelines and tools

Several existing tools target vascular segmentation, but few focus on microvasculature at high resolution within an integrated environment. Segmentation-focused tools operate on the image itself: the Vascular Modeling Toolkit (VMTK) [21] integrates in 3D Slicer and provides a comprehensive suite for vascular segmentation, centerline extraction, and surface reconstruction, but its methods are oriented toward low resolution, large vessels: arteries, aortas, and major branches. It also relies on seed-based interaction that does not scale to the dense, branching geometry of microvasculature.

ITKTubeTK developed by Aylward *et al* [46] offers a library of algorithms for tubular-structure segmentation and graph extraction built on the ITK framework, but is packaged as a programming library meaning it requires implementation into a tool or use in code, challenging for non-technical researchers. VesselKnife by Szczypinski *et al* [47] provides an integrated pipeline specifically targeting vessel segmentation, skeletonization, and graph extraction from CECT data but is a standalone application, requiring researchers to leave their existing analysis environment, and is not intended for small vasculature. SimVascular proposed by Updegrave *et al* [48] operates on the outputs of segmentation and extends vascular extraction into simulation, supporting downstream modeling of blood flow, a step laying downstream of this work.

The landscape of vascular segmentation tools reflects a tension in biomedical imaging: general-purpose deep learning segmentation frameworks offer good performance on the datasets they are trained on, but require large annotated datasets, offer limited interpretability and controllability, and are typically not integrated into the tools researchers commonly use. Classical model-based methods such as Frangi filtering are more transparent and adjustable, built into existing tools and easy to use, but require manual parameter tuning and struggle with complex vascular geometries. Neither pole serves the specific need addressed by this thesis: microvasculature-scale extraction, integrated into a widely-used segmentation environment, with parameters set by principled means rather than expert tuning. This combination is the niche the present work occupies.

Problem 6.

Existing pipelines for vasculature extraction either target large vessels (VMTK) or require programming expertise to deploy (ITKTubeTK). The solution should be tailored to the unique challenges of micro vasculature and integrated into a platform.

Chapter 3

Problem statement, goal and objectives

3.1 Problem statement

In a diverse software landscape for analysis of 3D data, this work aims to answer the question “*How can an open-source microvasculature extraction pipeline be developed, able to be used by non-computer scientists, leveraging classical segmentation methods and sparse user-driven input to produce improved segmentations when compared to thresholding on CECT data across a diverse dataset?*” Answering this question requires abiding by three principal constraints:

(i) Data: CECT tissue microvasculature with compression artifacts, intensity gradients and discontinuities, and a highly imbalanced data distribution with small target structures only a few voxels across.

(ii) Segmentation and its two paradigms: *data-driven* methods (machine learning) with potentially high performance but transferring poorly across datasets and requires annotated training data that is difficult to generate for CECT, and *classical* methods that encode vessel priors explicitly but require hyperparameter tuning, and reason locally with poor extrapolation resulting in fragmented segmentations and weaker performance in low-contrast regions.

(iii) Tooling: existing software is focused on low resolution data, human arteries and organs with large vessels and few disconnections. Microvasculature extraction methods are sparse, rarely tailor-made or integrated into available tooling

3.2 Goal and objectives

This thesis aims to develop a 3D Slicer extension that produces connected microvasculature segmentations on CECT through a hybrid pipeline combining classical algorithms with sparse user input, achieving higher vessel retrieval rates than thresholding while also

improving connectivity, measured using a bipartite matching and cIDice, operating across data with strong and weak contrast gradients.

To do so, the following objects must be defined:

1. Deliver a user-friendly 3D Slicer extension able to export in formats useful for downstream analysis.
2. Leverage a user-in-the-loop approach for point placement & basic vessel-size context to drive automated parameter selection, replacing manual error-prone hyperparameter tuning.
3. Build the segmentation core on multiple algorithms, combined through a framework that enables per component tuning, evaluation and allows for future extension.

To make the segmentation gap concrete, the simple and previously used approach of intensity thresholding is evaluated. The shortcomings are expressed in Figure 3.1: a sweep is run across all possible thresholding values, and measure against a manually annotated ground truth on an illustrative subvolume that contains small disconnected vessels, a light gradient, and some noisy elements. The precision-recall curve ignores true negatives and avoids giving them an outsized weight, but still allows us to understand the problem at hand: classical thresholding, no matter how high, will include false positives (*all high valued voxels are not necessarily vessels, the structural gap (1)*) such as in the shell region in Figure 3.19 and at low values needed to capture faint vessels, the false positives are very high (*vessel intensities overlap with background, a challenging extraction situation, making thresholding a compromise (2)*).

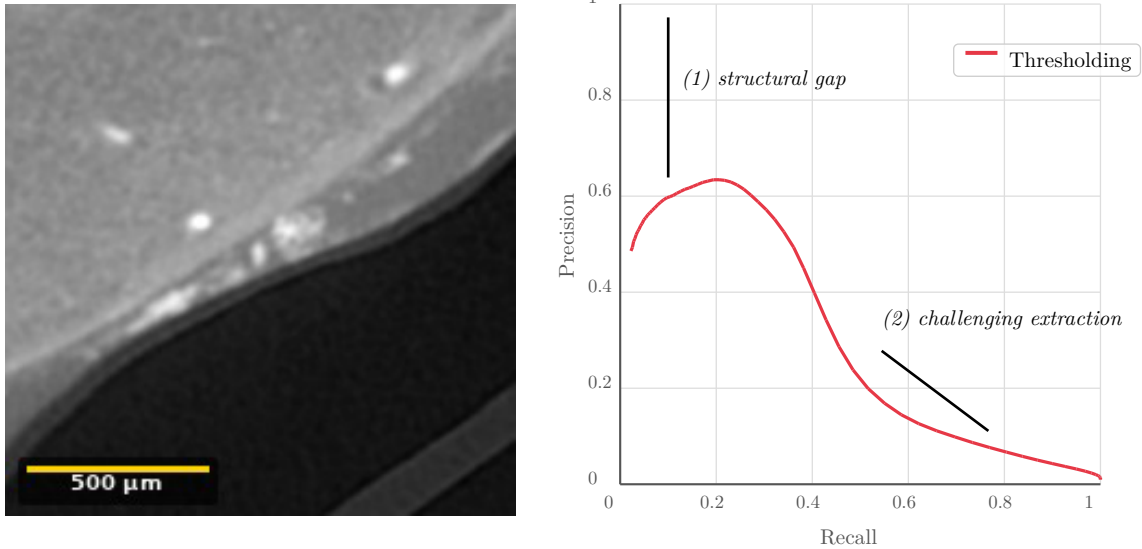


Figure 3.1: **(Left)** Typical CECT cross section, with vessels (high valued, inside the sample) and non-vessel-like structures (high signal areas on the outside) **(Right)** Thresholding precision-recall curve showing (1) *structural gap*: precision never reaches 1.0 because there are false positives in high valued areas and (2) *challenging extraction*: precision falls off sharply as recall increases due to the rapid increase in false positives; high recall cannot be achieved without sacrificing precision

Chapter 4

Methodology

4.1 CECT datasets

The CECT datasets in this work are all *ex vivo* murine tumours, imaged using a Phoenix Nanotom with isotropic 6 μm voxels. Further acquisition parameters are summarized in Table 4.3. CECT data management is challenging: data is received from the machine as 16-bit TIFF files, with a 2000x2300 slice weighing 9.2MB and a whole dataset with 2400 slices at 22.1GB. The delivered scans were windowed to 8-bit and already compressed to JPEG: this windowing process was unprincipled, with the window chosen based on the researcher’s best judgment, and the original uncompressed data discarded. Windowing and compression were both motivated by data storage and cost concerns, with the compressed datasets of this work ranging from 0.103 to 13.2GB (597x698x854 to 3000x3000x2653). Additionally, metadata was limited with datasets stored as folders of slices with no associated acquisition metadata.

Dataset size shaped the development process: to run algorithms on a volume, 3D Slicer uncompresses the entire volume into working RAM. The largest datasets would fill the development machine memory and leave no space for algorithms. As a result, datasets were subsampled to sizes large enough to contain representative vascular structure but small enough to fit in RAM alongside intermediate results.

4.1.1 Subsampling

To enable development and inference on regions that fit into memory, as well as allow the creation of voxel level annotations to evaluate CollaboratiVessel performance, training and validation sets were created: a validation set was made of four datasets subdivided into a 6x6x6 grid for subsampling. For each, three subregions were selected: one at each distance step from the center as visualized in Figure 4.1. This subdivision method was chosen as it guarantees a sample at each distance band, while also enabling annotation in a reasonable amount of time with enough context for evaluating vessels: samples are obtained from the center of datasets as well as edges, both small and large vessels are represented, gradients are captured and samples present a diversity of noise. The training set was generated in a similar fashion, and discarded once development complete.

4.1.2 Ground truths

Two complementary forms of ground truth are used in this work: user placed vessel points, and a binary voxel level annotation.

The user placed annotation points are those that guide the algorithm hyperparameters, and performance on these is also shown to the end user during use to provide direct in-the-loop feedback by reporting how many vessel points are correctly classified after a run. This method is fast and accessible but is limited when used for development: evaluation is sparse and pointwise, connectivity is ignored entirely, vessel size is not captured, and the inherent 2D nature of point placement makes coverage of small vessels uneven. As a result, binary voxel level annotations were created to enable more granular evaluation of the performance of CollaboratiVessel, and give a common reference to compare against thresholding.

4.1.3 Limitations of ground truth

Annotations were produced by a non-domain expert using 3D Slicer’s built-in 2D brush tool, with two known sources of bias:

- **Visibility:** Vessels that are not clearly visible in a slice or by moving between slices are not annotated, meaning that gaps are not filled. This results in a performance hit for algorithms that extrapolate across low signal gaps.
- **Boundary uncertainty:** The brush has a fixed size and is moved by the user, meaning vessels do not necessarily always stop at the same intensity gradient. Vessel edges in the annotation therefore have some noise, which affects Dice-based metrics.

Annotation took approximately 60 hours across two rounds, a first blind annotation round done before development and a second after reviewing the outputs of thresholding, the application of a Gaussian smoothing kernel to reduce visual noise and the reviewing of literature. The second round helped identify vessels that were missed due to noise, a lack of context during the first round, and the small sizes and low contrast of vessels.

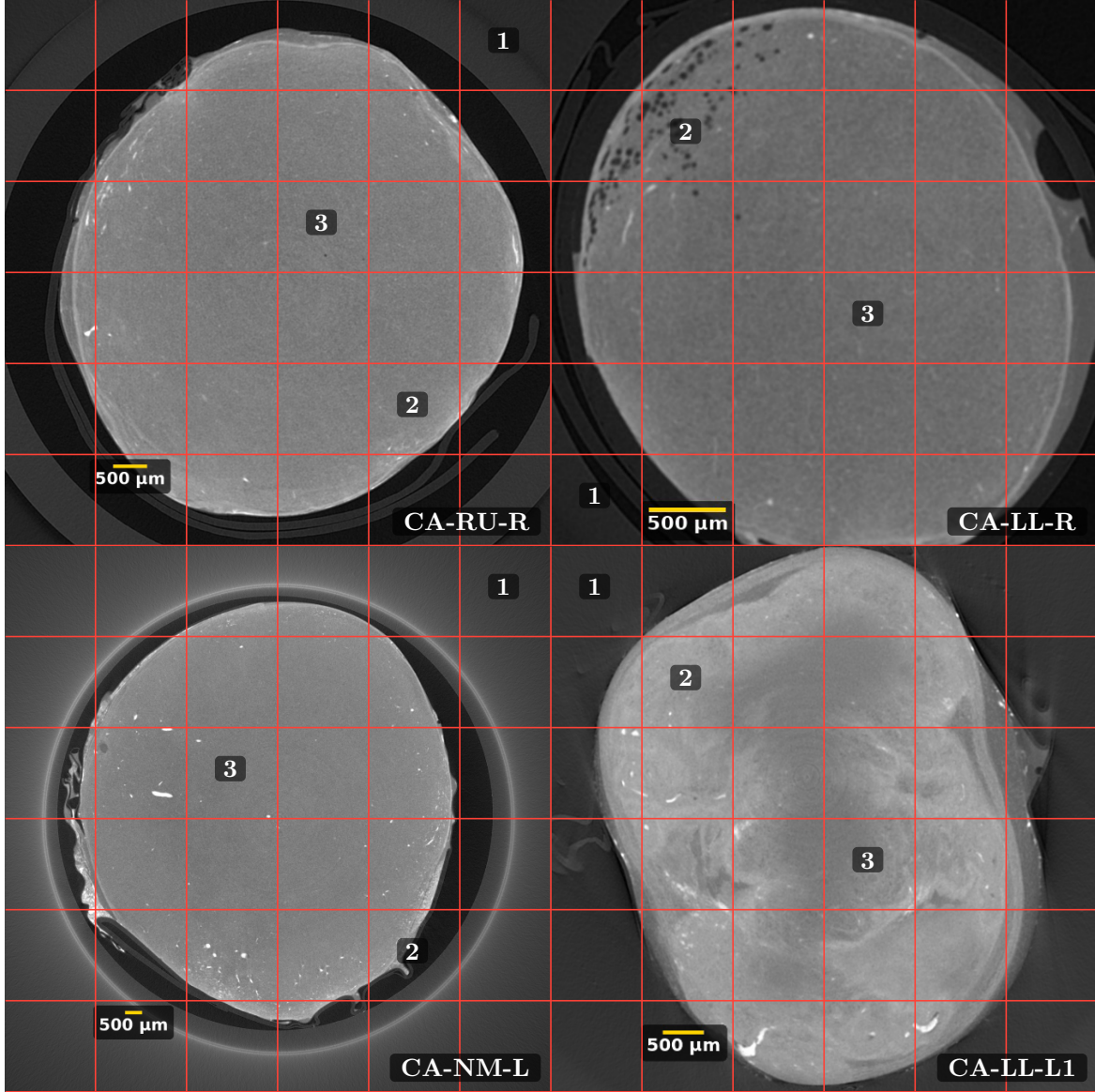


Figure 4.1: 6x6x6 Grid subsample overlaid on the central 2d slice, showing the locations of the three subregions randomly selected for annotation at three distances from the center. The diversity in tumor staining and vessel contrast can be seen, as well as coverage of central and edge regions and size diversity.

Of the 4 datasets, 6 of the 9 subvolumes were retained for annotation and will henceforth be called:

- CA-RU-R volumes 2 and 3, referred to as samples 1 and 2
- CA-LL-R volume 3, referred to as sample 3
- CA-NM-L volumes 2 and 3, referred to as samples 4 and 5
- CA-LL-L1 volume 3, referred to as sample 6

Discarded volumes correspond to the three volumes that fell entirely outside of the target tumor.

4.2 Building a 3D Slicer plugin

CollaboratiVessel development effort can be split into two main sections: the 3D Slicer plugin, and data processing logic. With 3D Slicer as the host, the development is explored first. To guide plugin development, plugins with similar stated goals were explored, and their shortcomings analyzed:

- 3D Slicer hangs when executing algorithms, as execution blocks the main thread. The OS will ask the user if they wish to stop the program; the user must wait until any logic completes before being able to interact again. This cannot be mitigated by the plugin.
- User interface (UI) development is restrictive: UI must be in the left corner, and few context menus are available. All UI development was done within these constraints.
- Execution in Python is single core. In order to better leverage available compute, external libraries implemented in C++ were used.
- Machine learning extensions did not come pre-packaged with the model weights and often required a GPU and installation of external software. This work requires no special hardware or external data and works on 3D Slicer cross platform.
- Errors during installation were not user-friendly enough for a non computer scientist to be able to debug an installation problem. All our required libraries are stated in error messages with the required command to install the library.
- Many available plugins were unmaintained, throwing errors during installation or use, or failed to produce outputs. To mitigate this, the plugin is packaged with a known good version of 3D Slicer.

4.2.1 Development process

With these issues in mind, the publicly available 3D Slicer programming guide by Perkins [49] was consulted. Plugins can be implemented in one of three forms: as command line interface (CLI) tools where they may be called as an encapsulated piece of code, the most abstracted way of working with external code or a C++/Scripted (Python) loadable module. The Scripted loadable approach was selected for its use of Python, lowering the barrier to entry, ability to use the UI features of 3D Slicer, and having access to the full slicer application interface. To enable inter plugin compatibility and fit the 3D Slicer structure, communication uses the Medical Reality Markup Language (MRML) Figure 4.2, where modules read and write to MRML nodes.

The extension wizard² was used to generate the required extension boilerplate, and the ScriptedLoadableModule was used as a starting point for the implementation code. 3D Slicer implements a user interface development module in Qt³, enabling drag and drop creation of UI elements, making it the natural method of developing a seamless UI experience. Although basic, this radically simplifies the process of creating a cross platform UI.

²https://slicer.readthedocs.io/en/latest/user_guide/modules/extensionwizard.html#extension-wizard

³<https://doc.qt.io/QtInterfaceFramework/>

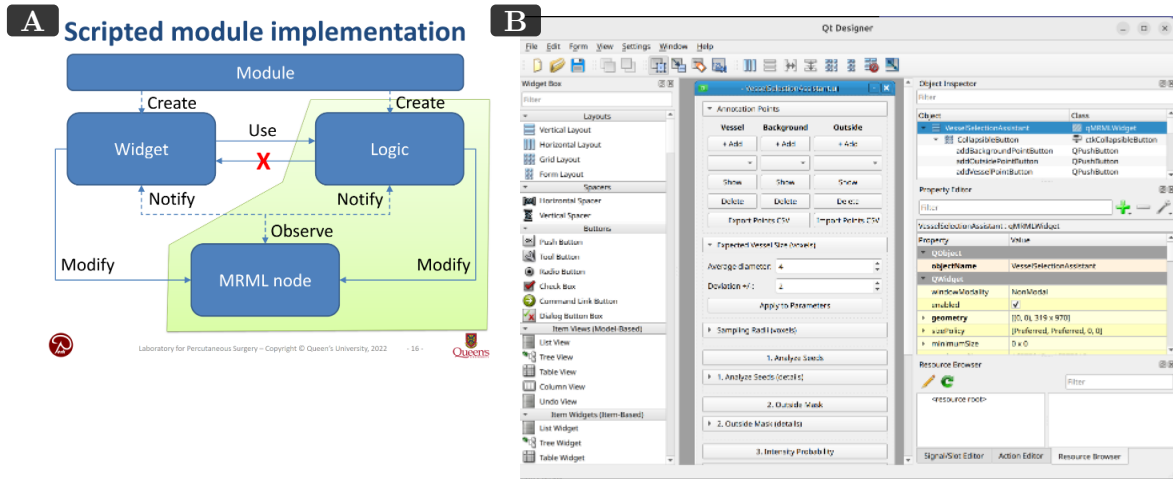


Figure 4.2: **A:** Structure for module implementation from [49]. The module being implemented must create a widget and a piece of logic that communicate via an MRML node. This motivates our separation of the code into two: a main plugin and a library of implemented logic. **B:** The Qt user interface designer, as used during the development of the plugin. Key advantages of this UI creation method are the drag and drop nature, integration into 3DSlicer and limited styling options.

4.2.2 Unifying data representations

The provided datasets were in JPEG, a lossy format that carries no information about physical voxel spacing between slices and introduces spatial frequency noise along the axes of compression. All volumes were converted to DICOM with explicit isotropic spacing of 6 μm per voxel, matching the original acquisition. Co-ordinate systems were also unified: 3D Slicer uses two coexisting coordinate systems⁴, one for code called IJK that corresponds to voxel indices and one for physical-world coordinates in millimetres (RAS: right, anterior, superior), used by all geometry-aware primitives such as markup points and segmentation transforms. User-placed seeds are stored in RAS and converted to IJK at the point of use, so annotations survive any intermediate resampling or coordinate-space transform. This also enables subsampling of volumes without losing volume location in the original sample.

It is important to note that the datasets used were isotropic: anisotropic data would require axis-dependent scaling at every IJK $\leftrightarrow$ RAS conversion, and some modifications to the code flow to handle this difference.

4.3 User interaction model

Multiple interviews were carried out with different research team members, from these it was noted that decisions such as hyperparameter selection, intensity windowing and grey value were often made without a principled, replicable and data driven decision method. This poses three problems: **1.** As the selected values are not data driven, decisions and pipelines cannot easily be replicated, and are more fragile; **2.** As the user acts as the evaluator, they

⁴https://www.slicer.org/wiki/Coordinate_systems

induce bias in the subsequent steps **3**. Any user without prior knowledge of the downstream steps will not be able to make a fully informed decision for the hyperparameters to select, requiring iteration and potentially falling into local optima.

These issues are addressed by reducing user input to two structured forms. **Anchor points** where the user places “vessel”, “background”, and (optionally) “outside” seed points by clicking *add* in the relevant category and clicking the corresponding location in any of the 2D viewers. These seeds drive the vessel extraction logic and seed segmentation performance results are provided in-the-loop as feedback after CollaboratiVessel runs. **Vessel size and standard deviation** in voxels, which estimated using the 3D Slicer data probe, and provides the scale context needed for Frangi vesselness and the local intensity probability map.

To preserve this user-provided information across sessions, the structured points are able to be exported and imported to a universal CSV format using RAS coordinates. To avoid file naming and versioning challenges, filenames are automatically constructed from the current volume’s series name, description, and timestamp, removing the need for manual organization. Vessel size and standard deviation setting are also saved in the user facing feedback, enabling future replication.

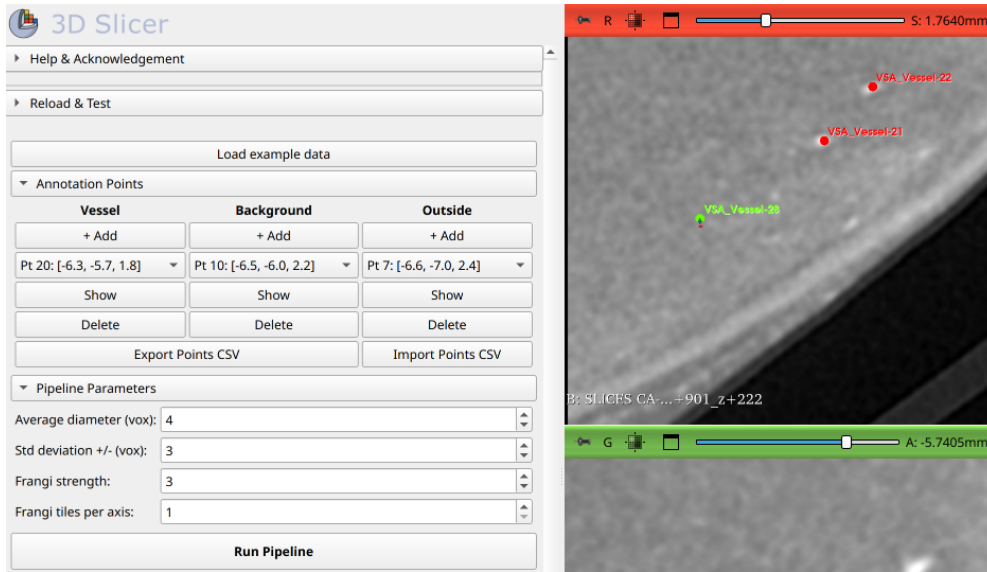


Figure 4.3: View of the seed annotation and vessel size definition panes. The user may press *add* and click on the locations in any of the right hand panes to place one or more points. Annotations can be imported or exported in a CSV format with RAS co-ordinates.

4.4 The CollaboratiVessel pipeline

Using CollaboratiVessel

To use CollaboratiVessel, points for the three categories are placed as detailed previously in the user interaction model. Once points are placed, hyperparameters must be adjusted: the base parameters are relevant for use by most users are vessel average diameter and standard deviation. Additionally, the user may adjust the Frangi strength, which defines the required intensity for a tube to be detected, as well as the Frangi tiling: implemented to reduce peak memory use, it cuts Frangi inference into substeps across the volume. Finally, the user can open and adjust the advanced parameters, such as ridge following strength, seeding passes, seed rounds and seeds per round. Advanced parameters do not need to be adjusted for the data in this thesis, but are exposed as they can be relevant to better fit different domains.

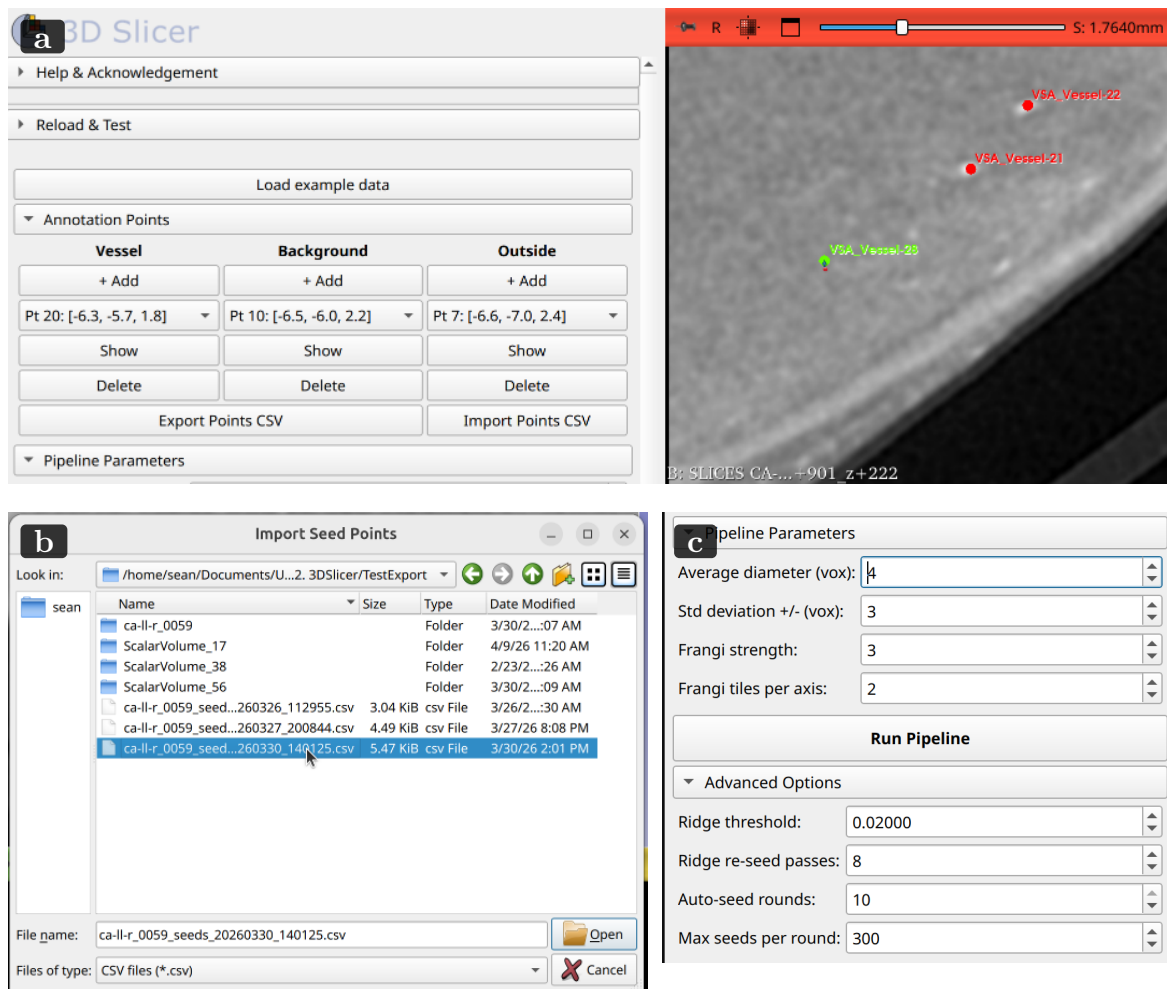
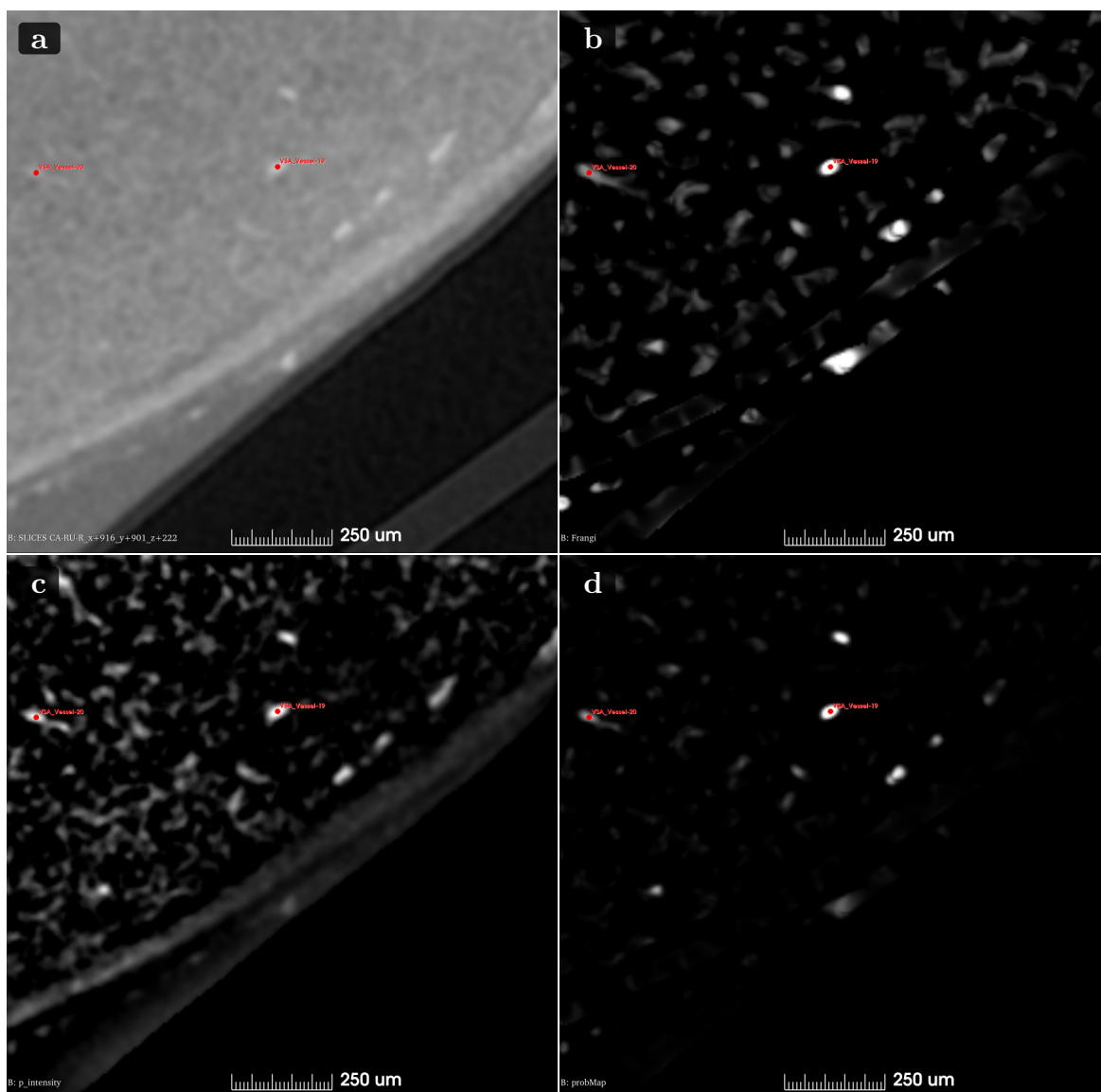


Figure 4.4: CollaboratiVessel use: (a) point placement, (b) loading of previously annotated points (c) parameter adjustment with defaults, and advanced parameter pane contents.

4.4.1 The CollaboratiVessel logic

No single classical method offers sufficient performance to extract challenging vasculature, as other vessel extraction tools such as vesselknife [47] have shown. As a result, to process data, CollaboratiVessel leverages a combination of algorithms to extract and combine vessel signal across steps, relying on the underlying assumptions that vessels have a particular shape and have a higher grey value intensity than surrounding tissue.

These steps are presented bellow, with the predicted probability that each voxel is a vessel visualized as a greyscale value.



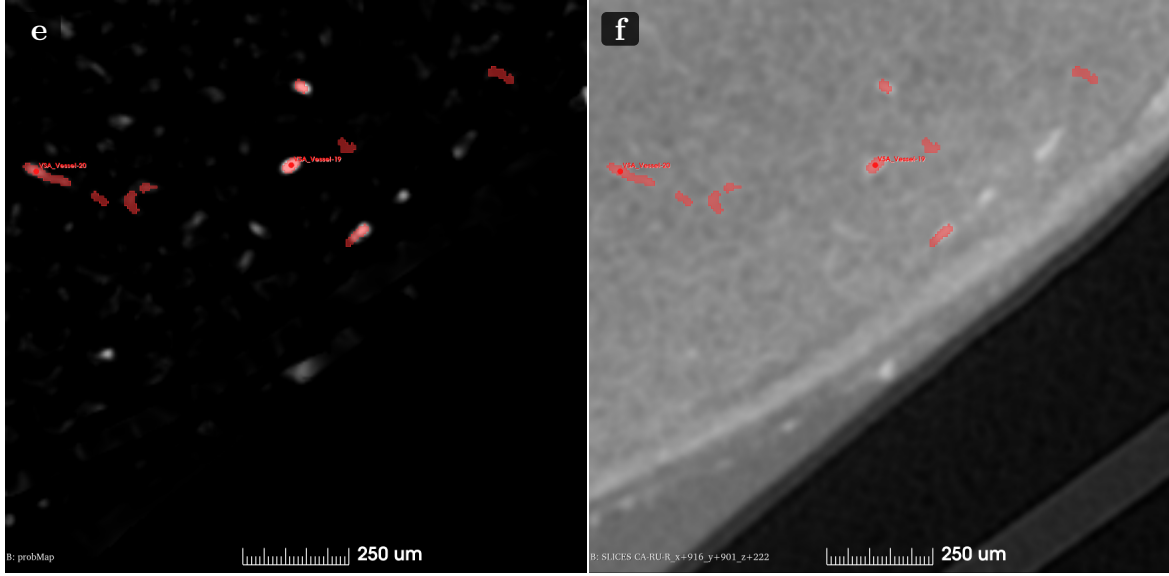


Figure 4.5: CollaboratiVessel vessel probability at each step, illustrating the iterative refinement leading to the final segmentation. (a) slice after user point placement (b) Frangi, (c) Intensity probability, (d) reconnection and reconstruction, (e and f) segmentation, overlaid on step d and on step a.

Step 1: Background removal and ROI extraction

When looking at a full volume, the sampled tumor is generally surrounded by air or empty space. To prevent the surrounding region from interfering and to reduce computation, if outside points are provided by the user, an initial background filter is applied and the midpoint between the highest outside intensity and the lowest background-seed intensity is used as an intensity cutoff. The voxels below the cutoff are zeroed providing a mask of the data aligned to the sample rather than the bounding box of the scan.

This removal introduces a high-gradient artificial boundary between zeroed and preserved voxels, which is suppressed using region growing and a mask to avoid Frangi, the gradient sensitive method, from detecting the edge as a tube.

Step 2: Frangi vesselness

The Frangi vesselness filter [16] identifies tubular structures on a local scale using the intensity and its change. The implementation of SimpleITK written in C++ is used for efficiency. Frangi is memory hungry due to its multiscale nature, with each scale requiring a full-volume Hessian, resulting in the need for multiples of the original volume to be kept in memory. To address this, the volume is optionally cut into subregions and processed in halo-padded tiles whose halo equals four times the largest vessel standard deviation, ensuring no information is lost at tile edges. The tile outputs are reassembled into a single volume before normalizing and scaling the response into $[0, 1]$ to make thresholds comparable across scans.

Step 3: Localised intensity probability

Frangi captures shapes but abstracts away the intensity signal and is sensitive to noise. To re-introduce intensity evidence, a local z-score is computed: for each voxel, the value is compared to the local mean and standard deviation over a window ten times the vessel diameter. Voxels with a value above their local neighbourhood are flagged as vessel candidates. The score is calibrated against vessel-seed intensities so that the user's annotations score approximately 0.8 in the local-probability map. This makes the output directly comparable to Frangi: both maps are normalised to $[0, 1]$ and represent independent evidence that a voxel is part of a vessel.

A known limitation of this localized signal approach is that window regions with high signal vessels can hide legitimate signal from smaller nearby vessels or weaker branches due to the mean being raised. This is one of the reasons the later reconstruction stage needs to be robust to broken or fragmented input, and for the use of both frangi and vesselness combined.

Step 4: Combined evidence map

The Frangi map and the local intensity probability are multiplied elementwise to produce the combined evidence map: `probMap = frangi x P_intensity`

The product is high where both signals agree, providing evidence for tubularity **and** a local intensity difference. It suppresses voxels supported by only one signal. This filters out two common false-positive sources: (i) noise that happens to look tubular to Frangi but has no intensity support, and (ii) bright artefacts that have no tubular structure, or that are disconnected by the aforementioned high local signal from large vessels. This `probMap` is used as the input to the structural reconstruction stage.

Step 5: Vessel reconstruction

provides evidence to the existence of vessels but does not follow them to fill small gaps. To convert the evidence into connected vessel structures, two methods are used: Ridge extraction and reconnection through skeletonization and morphology.

TubeTK's ridge extraction [46] is used to follow ridges in the input volume starting from a seed point. This fits a tube model with a smoothly varying radius along the centerline, producing more consistently shaped vessels. Initial ridge extraction starts from the user-placed vessel seeds. To capture vessels not directly seeded by the user, an iterative re-seeding loop proposes new seeds at high-probability voxels not yet visited, then re-runs ridge extraction from them. This is done iteratively for a configurable number of rounds.

To improve connectivity further, the output of the ridge following step, which identified the continuous vessels that are based on intensity, is followed-up with a more powerful reconnection process: firstly, **Morphological closing** (dilation followed by erosion) merges fragments separated by gaps smaller than 2 voxels, without inflating vessel diameters. This handles the cheap, geometry-only case where two fragments are physically adjacent. Secondly, a powerful **Skeleton based bridging** step identifies disconnected vessel tips via skeletonisation, finds pairs within a maximum gap distance, and routes a minimum-cost path through `1 - probMap`. A bridge is accepted only if its cost is below a threshold, enforcing a meaningful probability to support connection. This handles larger gaps and avoids spurious reconnections by requiring evidence from the underlying signal.

Output

The binary mask vesselness segmentation is displayed in the 3D viewer. User feedback is provided in the form of the amount of correctly classified elements, and user points vessel intensity mean and standard deviation. Additionally for evaluation purposes there is the possibility of loading a manually segmented DICOM mask for detailed evaluation based on multiple metrics such as DICE, cDice, vessel length, precision, recall and volume fraction.

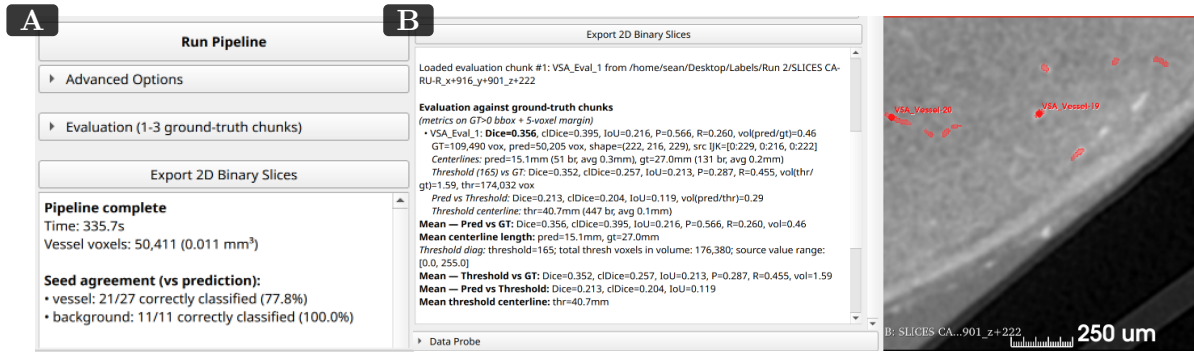


Figure 4.6: **A:** View of the feedback box as seen after running CollaboratiVessel, with user placed point segmentation performance. **B:** Feedback box when utilizing the evaluation feature to compare predictions with a binary annotated ground truth, providing more advanced metrics than point wise evaluation available without ground truth.

Chapter 5

Results and discussion

Vessel extraction performance of CollaboratiVessel is evaluated both *quantitatively* and *qualitatively* against intensity thresholding on the subvolumes previously laid out in Figure 5.1 and their ground truths.

CollaboratiVessel is evaluated as it would be used in practice: the default parameters are used, which were obtained following hyperparameter sweep on the training data. These hyperparameters are only adjusted when qualitative inspection, done before collecting the quantitative results, reveals a clear failure in the form of a segmentation failing to identify more than half the user annotated vessel points. In order to evaluate thresholding, the value yielding the peak cIDice was selected after running a sweep across all possible intensity values. This value is used as it represents the best achievable performance on a key evaluation metric relevant to users, holding CollaboratiVessel to a high bar: other values such as the mean value of the user placed vessel points introduced a bias towards the placement of points.

5.1 Hyperparameter sensitivity analysis

A key disadvantage of thresholding is its sensitivity towards the selection of the exact right threshold: if incorrectly configured, the resulting 3D map is unusable, and optimal thresholds vary by location within a scan as well as between scans. To highlight a key advantage of the pipeline based approach, we analyze the cIDice result sensitivity of the parameter of vessel size and deviation against thresholding on subsamples 1 and 2, representing a sample from the edge and one from the center of a volume. Frangi intensity, the third relevant parameter set by reference implementations to 5 and controlling the intensity difference used to identify tubular shapes was evaluated separately due to its narrow useful range of 2-6: values were evaluated at 2, 3, 4, 5 and 6 with cIDice performance of .3081, .3086, .2984, .3085, .2984, resulting in our default of 3.

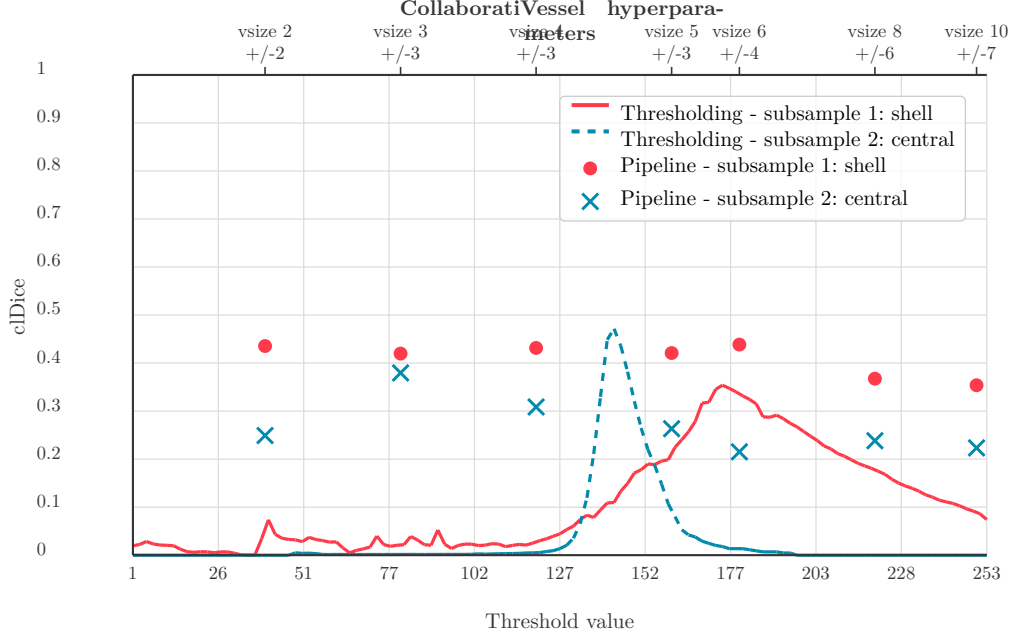


Figure 5.1: Sensitivity analysis of CollaboratiVessel vessel size and standard deviation when compared to thresholding values, showing: (1) dashed blue, thresholding on a central subsample, with a narrow peak indicative of a narrow distribution of vessel intensities, (2) continuous red, performance on a subsample that contains the outer edge of a dataset, with a much wider spread of vessel intensities, and (3) the stability of CollaboratiVessel when adjusting the vessel size and standard deviation.

From our sensitivity analysis, we can observe that there may potentially exist parameters that further optimize the performance of CollaboratiVessel for these two samples, but also that CollaboratiVessel performance is relatively flat: the quantitative metrics extracted next, done using default parameters, are thus likely close to the local optima for each run.

5.2 Qualitative comparisons

To obtain an understanding of the outputs of CollaboratiVessel when compared to thresholding, we can observe the segmentation output from the 2D and 3D perspectives: 2D enables the visualization of the slice and its intensities, with the segmentation method overlaid. 3D provides an understanding of the structure of the outputs and how they relate to each other, without giving a good sense of scale due to depth reprojection. Sample 1 Figure 5.2 is a challenging outer boundary example, and 5 Figure 5.3 a cleaner central sample with large, high contrast vessels, the only example that required vessel size adjustment to 12 vox $\pm$ 6.

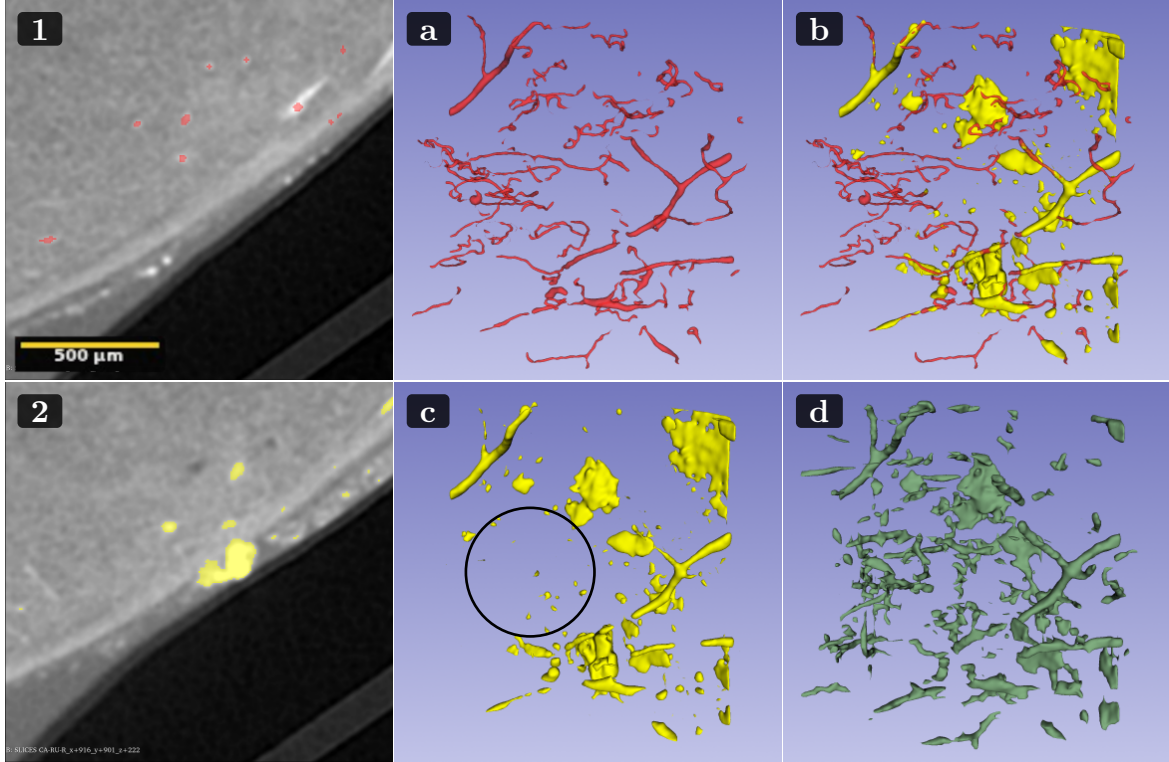


Figure 5.2: **Sample 1**, (1) (2) slice views: in red CollaboratiVessel, in yellow: thresholding, highlighting a large false positive. (a-d) Comparison of 3D Views: (a) CollaboratiVessel output, (b) CollaboratiVessel overlaid with thresholding, (c) thresholding alone, with black circle highlighting a large area of false negatives, (d) ground truth.

In Figure 5.2, thresholding captures large false positive plates, has gaps and holes, and entirely misses the vessels towards the central region (black circle). CollaboratiVessel output is more continuous, although conservative on vessel size as can be seen in the overlapped visualization. Ground truth shows the variance introduced by non expert manual annotations using the 3D Slicer tools, highlighting its limitations as a comparison point: vessels are larger, size is somewhat inconsistent and some blob-like shapes are captured.

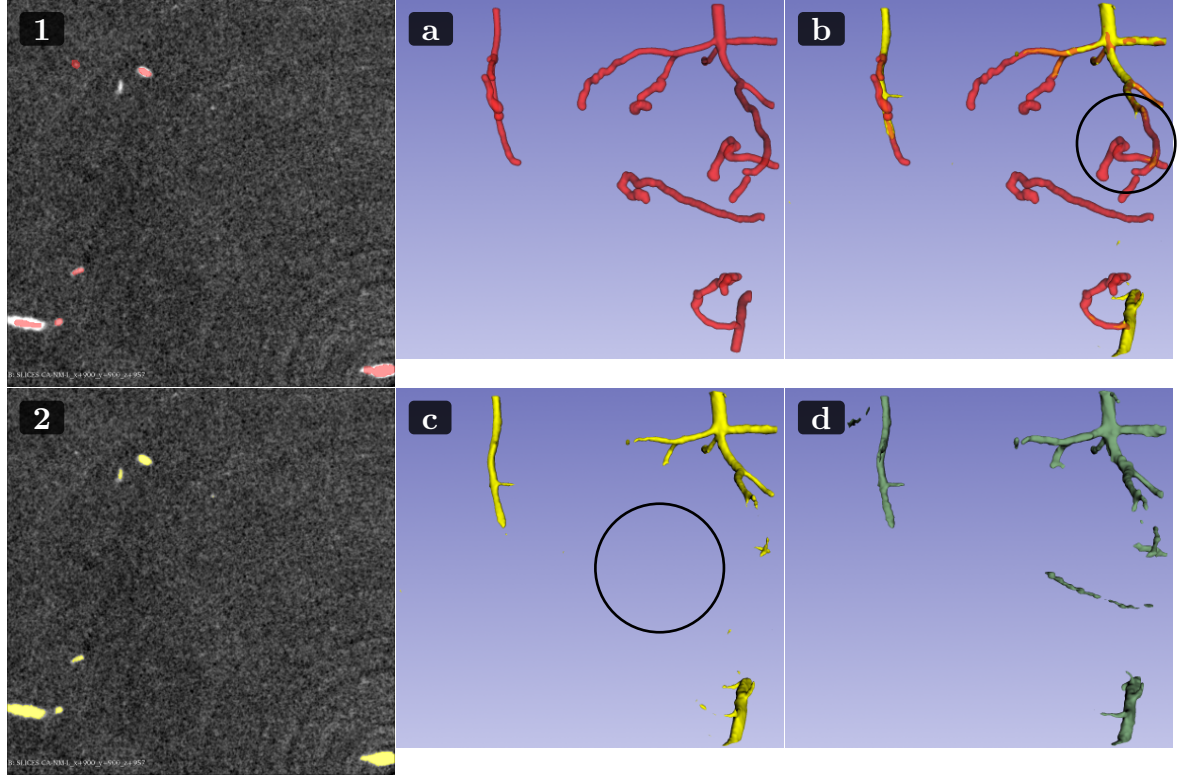


Figure 5.3: **Sample 5:** (1) (2) Slice views, in red CollaboratiVessel, in yellow thresholding, showing the under segmentation of CollaboratiVessel (a-d) Comparison of 3D Views: (a) CollaboratiVessel output, (b) CollaboratiVessel overlaid with thresholding, black circle highlighting a region with a successful reconnection (c) thresholding alone, black circle highlighting a missed vessel (d) ground truth.

In Figure 5.3, a high contrast scenario with large vessels, thresholding misses low contrast vessels and fails to fully connect the large main vessel. The ground truth also shows that CollaboratiVessel is extending vessels beyond where they stop being annotated.

5.3 Quantitative comparison

The three quantitative metrics laid out in Chapter 2.2.4.1 evaluation criteria are examined below on the 6 subvolumes: user placed annotation points, voxel level metrics and connectivity metrics.

5.3.1 User placed points

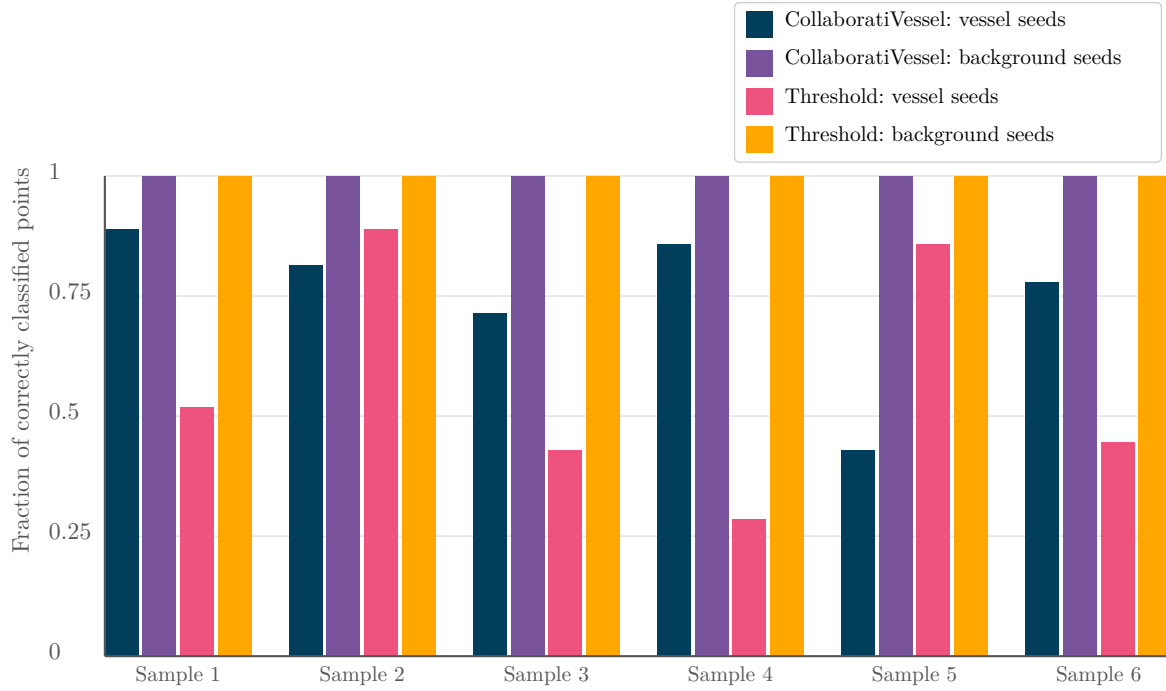


Figure 5.4: Fraction of user-placed vessel and background seed points correctly classified by CollaboratiVessel and intensity thresholding.

As can be seen, background points are always fully classified correctly. Vessel points are more variable: with CollaboratiVessel higher in most cases, however always missing some predictions. By observing the data on a more granular level, we can learn an interesting first fact that encourages our move from low representative power user placed points towards cIDice: the threshold value giving the highest proportion of correctly classified vessel points does not correspond to that with the highest cIDice.

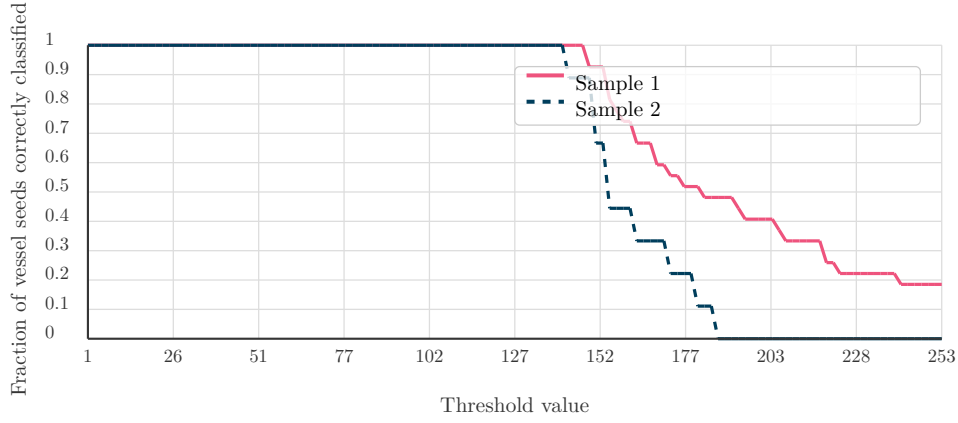


Figure 5.5: Fraction of user-placed vessel seed points classified as vessel by intensity thresholding, as a function of the threshold value, on two subsamples.

This is obviously a tradeoff between false-positives and false negatives, difficult to balance in a useful way at such a low level.

5.3.2 clDice

To analyze performance quantitatively, we explore clDice, the voxel level metric chosen for its better representation of connectivity and the ratio of correctly classified vessel points (annotation points placed by the user).

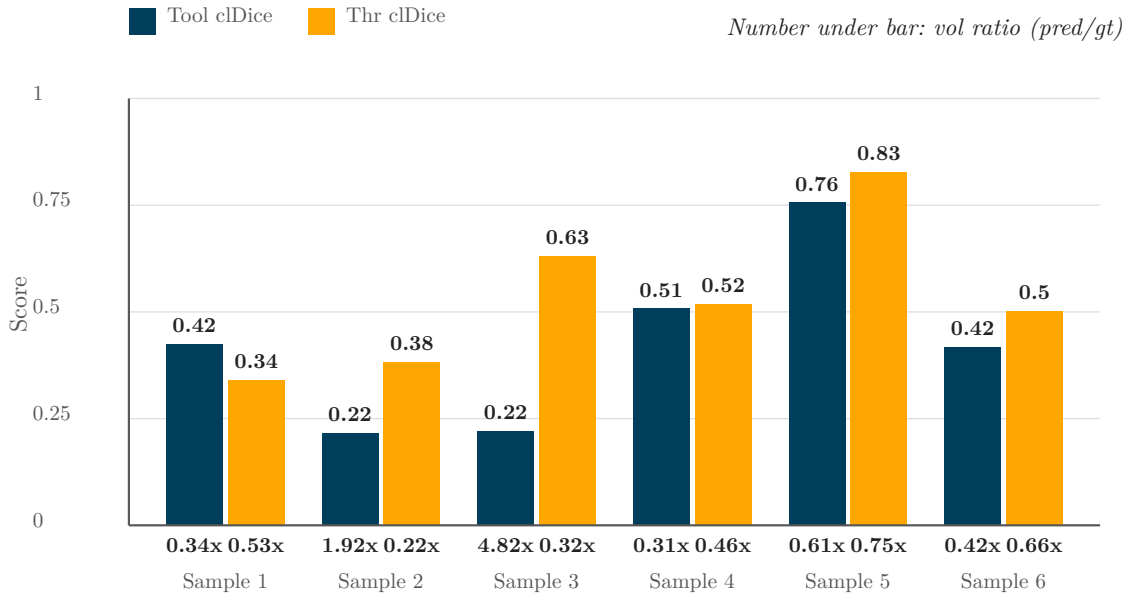


Figure 5.6: clDice comparison of CollaboratiVessel against thresholding, using the optimal threshold for the highest clDice. Volume ratio (prediction/ground truth) presented numerically underneath. The results highlight two volumes for which thresholding substantially outperforms the model: these volumes have marked oversegmentation with regard to the ground truth, predicting 1.92 and 4.82 times more voxels respectively. These highlight two volumes for which there is substantial extrapolation, Figure 5.2 and Figure 5.3. Other volumes show closely matched clDice values, and under segment.

These cIDice scores reveal are interesting from a segmentation perspective, but fail to inform us about the vessels themselves. In order to better understand the vessels predicted by both techniques when compared to the ground truth, we analyze the distributions of vessel length and size, initially visualizing all predictions (false positives and true positives) in Figure 5.7, then focusing only on true positives in Figure 5.8.

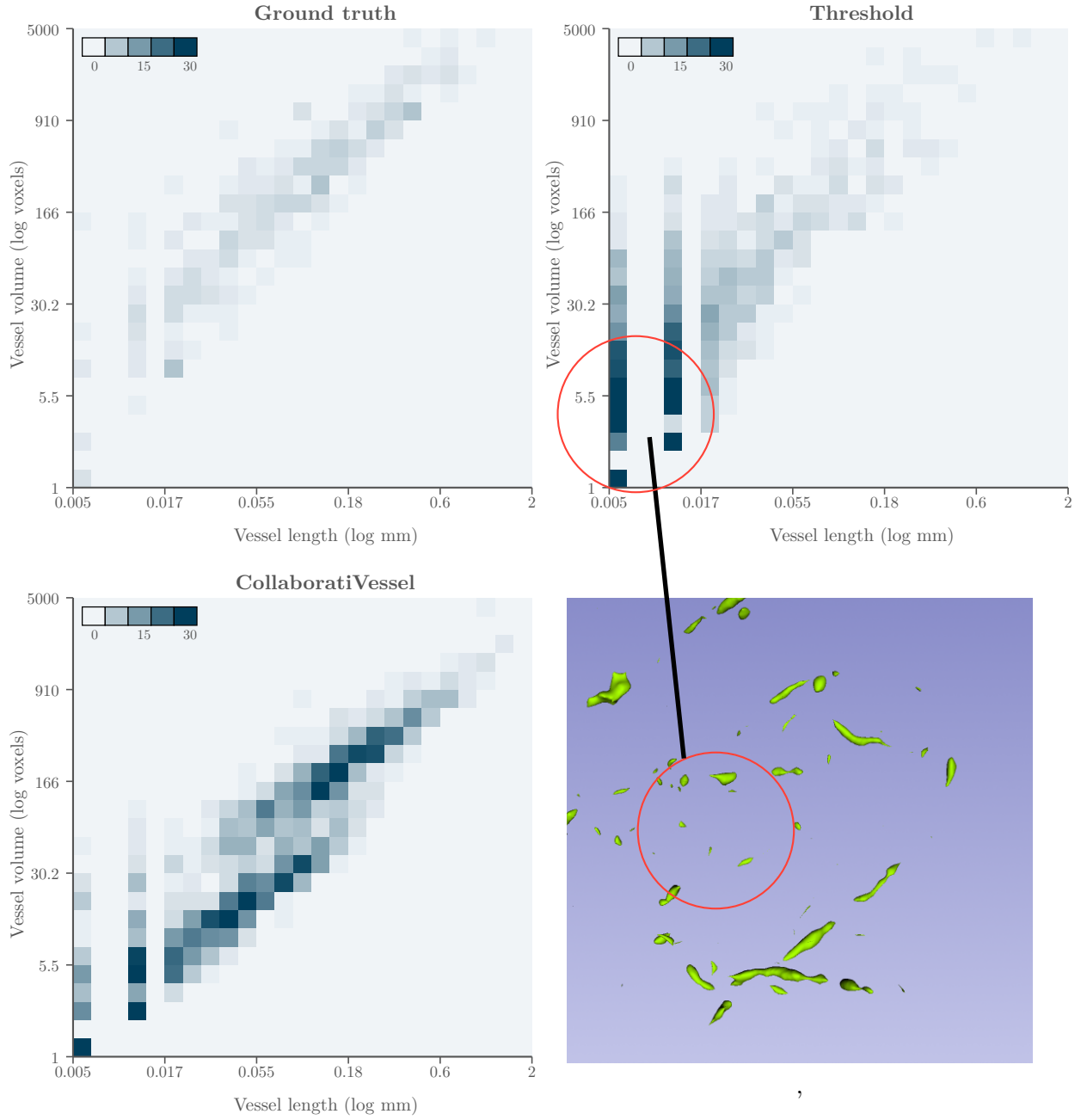


Figure 5.7: **Relationship between vessel volume - vessel length**: all predictions (incl false positives). A tubular vessel lays on the diagonal, as can be seen in the ground truth and CollaboratiVessel. Thresholding shows a high density of low volume predictions with short lengths, indicating blobs, and a generally smaller distribution of vessel sizes. CollaboratiVessel shows a tendency of predicting thinner vessels than the ground truth (a lower volume for a given vessel length). In **3D**: Sample 2 thresholding (Yellow) showcasing the small disconnected predictions

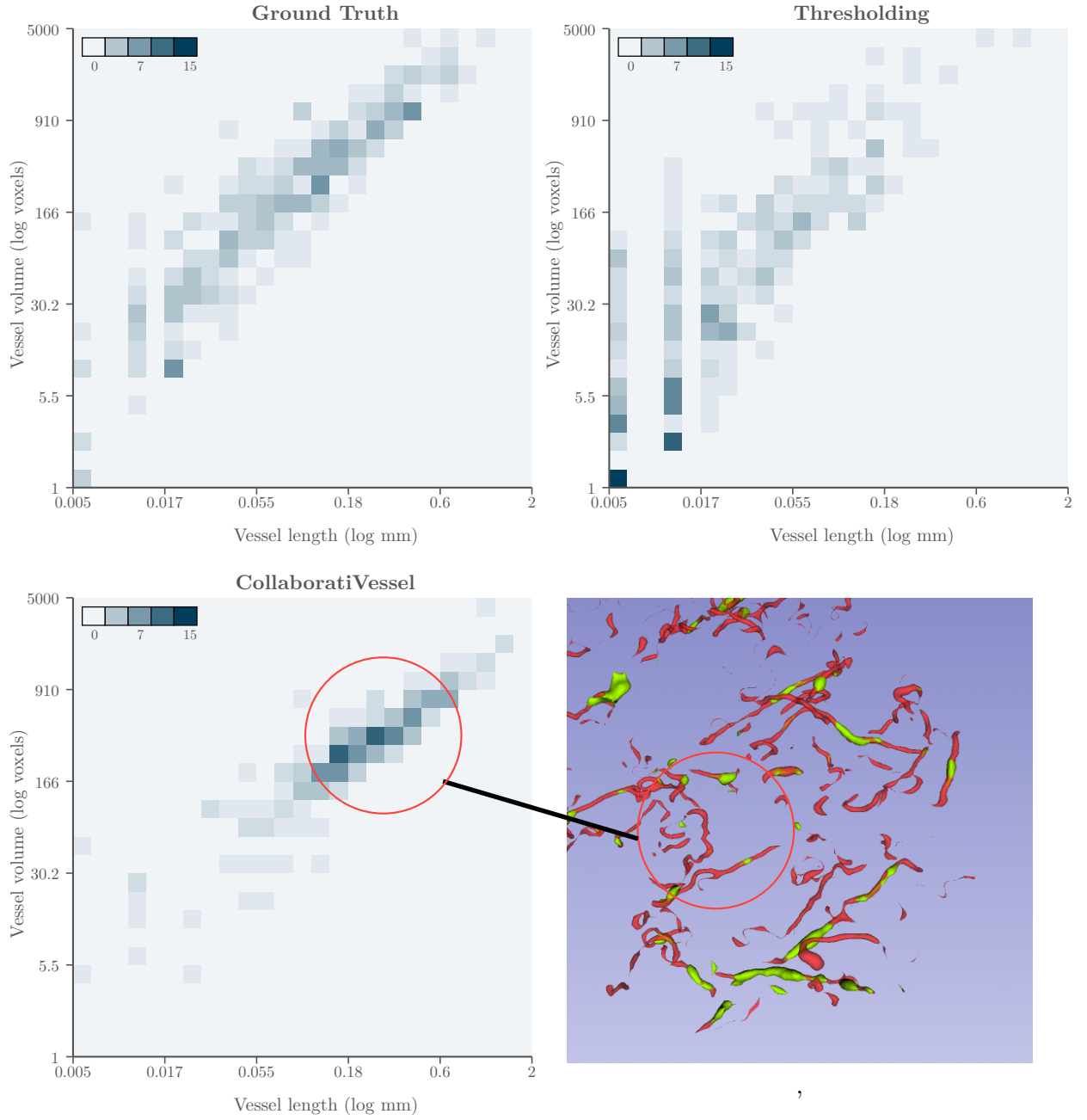


Figure 5.8: **Heatmaps of vessel volume/vessel length - only true predictions:** vessels for thresholding and CollaboratiVessel are only plotted if they correspond to at least one GT vessel, showing that many of the small predictions in thresholding and CollaboratiVessel are outside of the ground truth. In **3D: Sample 2** thresholding (Green) and CollaboratiVessel (Red) showcasing the connection of regions that are also captured by thresholding into longer vessels.

Figure 5.8 highlights that CollaboratiVessel is effectively connecting many small vessels into longer continuous ones. Sample 2 is shown, which had a higher thresholding cIDice than CollaboratiVessel, and where many false positives were present.

5.3.3 Quantifying connectivity

Beyond vessel size and length, it is interesting to investigate the known issue of voxel level metrics laid out in Error calculation Fig 2.4: vessels are not considered unitary, meaning that it is possible to miss vessels entirely without it being evident in the results, and reconstruction is not captured. To paliate this, a vessel level comparison is done using a bipartite analysis: for each predicted contiguous vessel, the corresponding contiguous ground truth vessel(s) are during metric calculation. This 0 to N matching allows us to identify, for each predicted vessel, how many ground truth vessels it touches. Figure 5.9

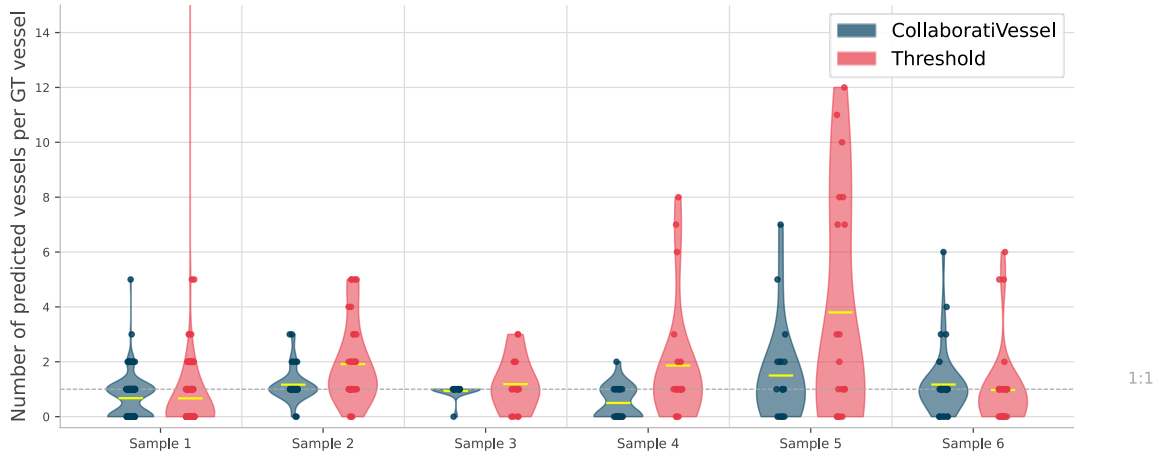


Figure 5.9: The distribution of number of predictions to each ground truth vessel: each violin shows how many predictions corresponded to each ground truth vessel. **0** = missed, the segmentation method did not identify the vessel, **1** = match, segmentation overlaps with one ground truth vessel **2+** = fragmented. Dashed line at 1.0 indicates 1:1 correspondence.

Dots show individual GT vessels. Yellow horizontal bar marks the mean.

Figure 5.9 shows thresholding generally results in a higher amount of predictions per GT vessel, with a wider spread, and more outlier values, indicating a worse matching. Individual runs as seen in as seen in Figure 5.10 and Figure 5.11 enable a more granular understanding by thanks to the ability to visualize the amount of predictions:

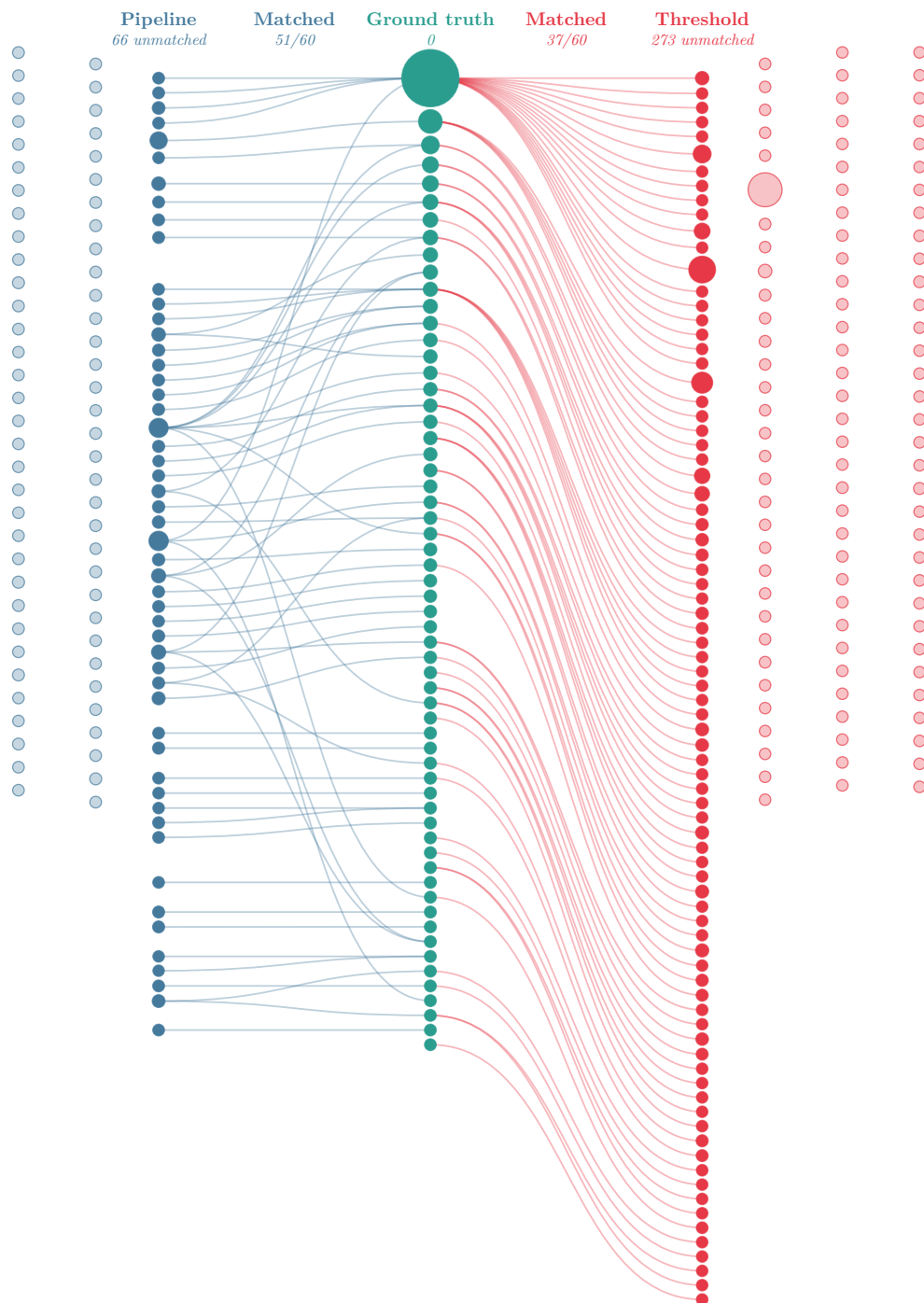


Figure 5.11: **Sample 1**, correspondence between unitary contiguous CollaboratiVessel predictions (blue), ground truth (green) and thresholding (red). Unmatched vessels are partially transparent. Thresholding here shows many small predictions matching to few ground truth vessels: thresholding has many disconnections

In Figure 5.11 to comparative advantage of CollaboratiVessel is particularly visible: vessel sensitivity is much higher, identifying 51 of 60 ground truth vessels, and vessels are much less fragmented, as can be seen when compared to thresholding: for each ground truth vessel there are few CollaboratiVessel predictions, against many in thresholding. This was previously visible qualitatively in Figure 5.2 and revisited in Figure 5.12 to highlight further the benefits of the implemented reconnection step.

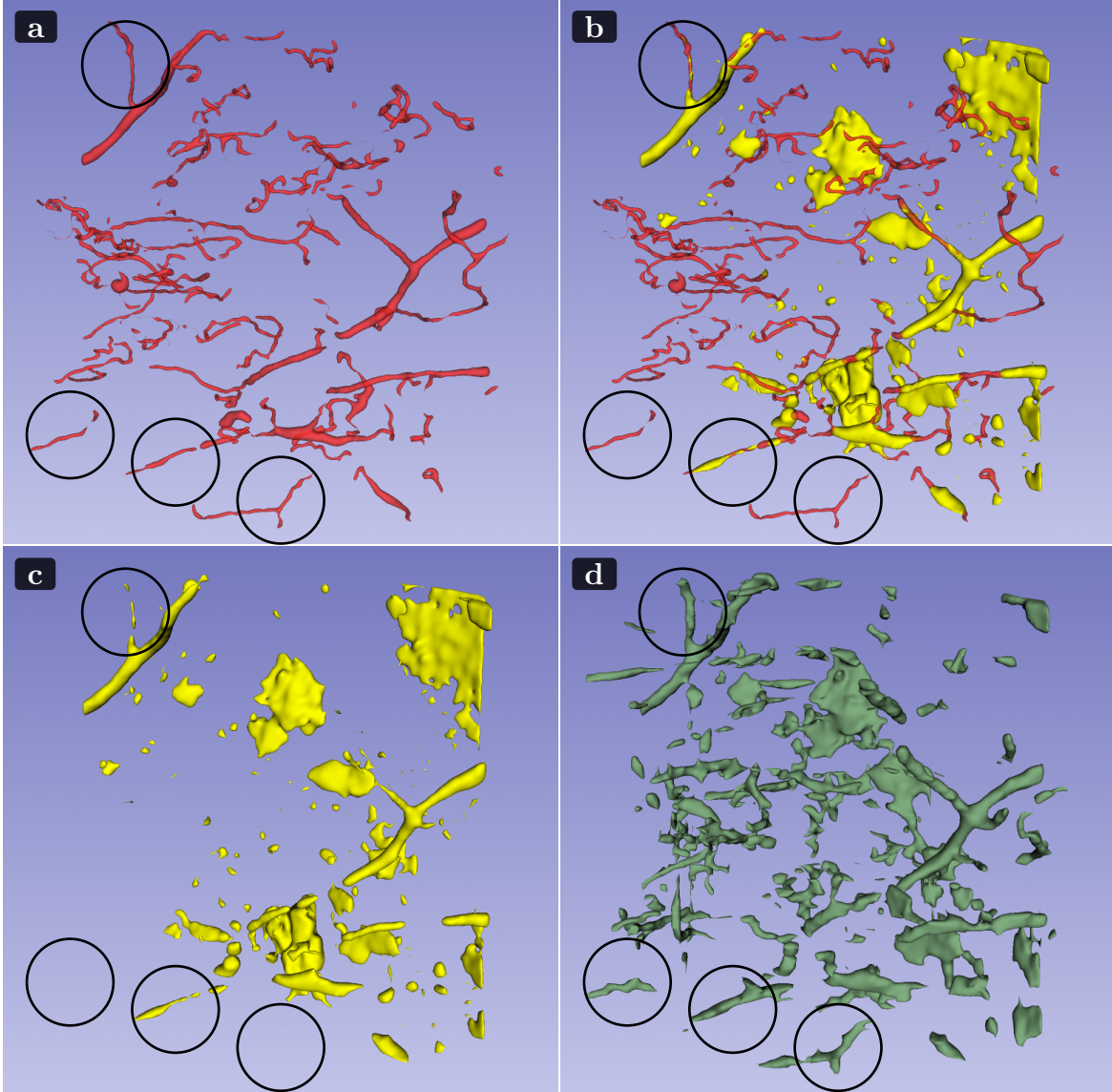


Figure 5.12: **Sample 1**, 3D visuals highlighting the many small disconnected predictions of thresholding against CollaboratiVessel, in black: vessels either missed by thresholding, or containing disconnections.

5.3.4 Evaluation takeaways

Following the evaluation of CollaboratiVessel against thresholding using voxel level cIDice and higher order heatmaps and bipartite matching, as well as examining the outputs of CollaboratiVessel in 3D views, it is clear that cIDice fails to capture the complexity of predicting vessels with sufficient accuracy for it to be used as a reliable measure of vessel prediction performance. The biases towards vessel size drowns out the real signal of vessel presence. Additionally, cIDice completely fails to capture the metrics highlighted in the heatmaps: many small disconnected predictions are not relevant for downstream tasks, but still artificially boost the performance of cIDice for thresholding. The bipartite matching also highlights the need for a metric that can evaluate something akin to a reconnection rate: how good a method is at not splitting a continuous prediction into sub predictions.

A worthwhile extension to evaluation would be further development of these vessel level metrics and a procedure for abstracting away the noise induced by manual annotations.

Chapter 6

Conclusion and future perspectives

The work laid out in this thesis set out to answer the question of *How can an open-source microvasculature extraction pipeline be developed, able to be used by non-computer scientists, leveraging classical segmentation methods and sparse user-driven input to produce improved segmentations when compared to thresholding on CECT data across a diverse dataset?*, question to which CollaboratiVessel was created to answer: a 3D Slicer plugin that combines vesselness and intensity priors with vessel reconstruction steps leveraging user placed points with minimal hyperparameter tuning.

The principal contributions of this work can be summarized as follows: **A user friendly 3D Slicer plugin**, CollaboratiVessel, was created. It requires no coding to operate, and can export binary segmentations compatible with downstream analysis tools. The UI was designed with prospective users in mind, with pre-tuned hyperparameters and offering only the necessary options. The associated code was written in a modular fashion to enable simple changes and future expansion to other datasets. Collaborativessel leverages **a framework for extrapolative prediction** with a different segmentation paradigm to thresholding or machine learning. As opposed to purely hard evidence based methods like thresholding that do not extrapolate or interpolate vessels, or purely data driven methods for extrapolation and interpolation that require learning from data, a multistep approach is successfully used. This approach enforces priors and connects vessels based on evidence to successfully extract connected vessel segmentations.

A user-in-the-loop segmentation approach was followed, that differentiates this work from classical deep learning approaches to solving challenging segmentation tasks. The algorithm, user and data are all treated as working together towards a same goal, with user inputs leveraged in the loop and outputs directly shown the user in a 3DSlicer segmentation window, enabling rapid iteration and adjustment by modifying hyperparameters or revising segmentations with built-in 3D Slicer tools if required.

Finally, **A manually annotated evaluation dataset** was created, with six 3D subvolumes spanning four tumour samples annotated, a valuable resource for future work on CECT microvasculature segmentation. These annotations were used in **a framework for evaluating prediction quality** beyond voxel-level Dice and cDice. The evaluation work in this thesis explores the use of ways of evaluating vessel segmentation relevant for future

work on the data, using heatmaps to visualize tubularity and bipartite vessel matching to capture fragmentation and missed-vessel rates.

6.1 Limitations

The initial goal, being very wide in scope and problem definition, is naturally satisfied within certain constraints:

Memory and inference scale Due to the size of scans and limitations with how 3D Slicer currently handles them, CollaboratiVessel operated on manually cut subvolumes. Tiling of memory intensive steps was implemented, but without an improvement in base memory use in 3D Slicer, it will not suffice to infer on datasets more than approximately a fifth memory size. Inference time also varies widely due to the iterative nature of certain logic steps, taking 10 to 30 minutes per subvolume, with large variations independent of volume size due to the amount of predicted vessels and their size. These two limitations (memory and inference time) limit immediate adoption and constitute the largest barriers to adoption.

Annotation quality and bias is also a concern: annotations were produced by a single non-domain-expert annotator. Vessels not clearly visible in 2D slice views were not annotated, and as a result little extrapolation was done. This results in a bias towards larger and more visible vessels, and a systematic undersegmentation of small and low-contrast vessels or vessels in a volume with a gradient. This benefits thresholding, whose prediction mechanism does not infer vessel connections, and who does not enforce vessel size variation or continuity and penalises the connectivity-aware extrapolation.

Sample size and diversity The six subvolumes only explored four datasets, a small subset of all provided datasets. As a result performance may not generalise well to out of distribution datasets.

User variability and shape bias The user-placed seed points and vessel size prior (and its enforcement) introduce some annotator-dependent variability in the placement of seeds. The user selected vessel size and standard deviation enforce a tubular shape prior through Frangi and ridge extraction. This risks undersegmenting vessels, and introducing a bias in downstream tasks that may not be aware of this prior, as well as hinder the comparison of data between different segmentation techniques.

6.2 Future work

Future work can be plotted into three main directions: improvements to inference speed, vessel extraction performance and generalization and evaluation metrics:

Inference speed is currently a limiting factor to the evaluation and use of CollaboratiVessel: improvements would allow inference on larger volumes, faster iteration and better exploration of hyperparameters. This could be achieved by either improving 3D Slicer itself, or moving the principal data handling tasks outside of 3D Slicer to be able to use libraries that would enable techniques such as *streaming* or *chunking*. It may also be possible to get

around the limitations of 3D Slicer by automating the process of cutting chunks, importing, analyzing and exporting their results.

Despite using libraries constructed with speed in mind for this work, a full C++ implementation of the plugin could reduce memory and compute time substantially, as well as enabling parallelism currently blocked by python and its single thread nature. It may be relevant to ask the question of the position of 3D Slicer as a tool for inference, or if it should be purely used as a tool for visualizing and working on the input points/hyperparameters and final results.

Extraction performance could be improved by exploring and implementing an automated configuration of the advanced hyperparameters based on the inference volume size and dimensions. It would also be beneficial to expose more hyperparameters that are currently hard coded, and integrate some method of configuring them automatically, such as with a grid search of parameters based on user placed points. Extra data outside of the current distribution could also be tested, relevant to understand the shift in performance as the task at hand changes.

Improvements in evaluation such as automated methods for measuring and comparing vessels across methods is a relevant direction to explore, as better metrics enables a better understanding of what contributions matter and how they impact outputs. Developing an algorithm without these metrics forces slow and deliberate step by step exploration of the search space of possible algorithms, as opposed to enabling automatic searching. These improved metrics could also better reflect the downstream tasks which vessel extraction is intended to enable.

Additionally, a method of comparison between volumes would be needed to enable comparative studies, and methods for standardization such as vessel extraction or export based on characteristics (such as distance from the outer shell) would be relevant. Current binary label export binarizes outputs, losing much of the information about possible vessels that CollaboratiVessel has generated, discarding elements that could be extracted from the skeletonized outputs such as the centerlines, branching ratios or vessel diameter distributions.

Finally, supplemental annotations from one or more annotators would enable inter-annotator agreement measurements, as well as improve tumor coverage and provide a more reliable ground truth. Alternatively, or as a supplement, simulated data (synthetic vasculature with known geometry) would be relevant to allow the creation of data with a known ground truth, and through ablation, style transfer or other methods could be made to emulate the current data, offering sufficient data volumes for more detailed and precise evaluation and potentially utilizing deep learning.

6.2.1 Data based extensions

As illustrated in Figure 2.2 (*Vascular segmentation methods positioned by data need and topological awareness*), we exist in the upper left quadrant: one that avoids the need for large amounts of data or machine learning, but that trades it for higher complexity in terms of vessel understanding, and extrapolation/interpolation based on hard coded algorithms. The delivered work was built as a *bootstrapping* tool: the first step in an iterative process to help extract vessels, which themselves could be used as a baseline for future learning

based methods. Each subvolume produced by a user running CollaboratiVessel is a potential training sample for a learned model. Given an expert with both some knowledge of vessels as well as knowledge of learning based techniques, a segmentation network potentially including topology awareness could augment CollaboratiVessel while still keeping elements such as the user-in-the-loop seed mechanism for guiding inference on new samples.

This would close the loop described in the problem statement: classical methods provide an accessible entry point, the tool accumulates annotations, and the annotations eventually enable the data-driven performance that classical methods cannot reach without requiring the upfront annotation investment that made data-driven approaches inaccessible at the outset.

Note of the author:

When the first pairs of scissors were developed for mass production, right handed-ness was not just the majority of potential users but the implicit default due to societal and educational factors. As development effort shifted from solving the functional problem to improving the user experience and outcomes, left handed scissors began to appear and they became more widespread, used by a wider range of skill levels, from children to professional tailors.

In a sense, this thesis is the initial scissors of microvasculature, while also existing in a highly specific niche of 3D structure segmentation: a solution to the broad problem of vasculature extraction that does not necessarily satisfy all users, or produce an ideal result, but still functionally brings a solution to the problem of challenging vascular extraction better than the thresholding used before. The door is now open to iterations that can either make the solution higher performance by continuing the bootstrapping towards data intensive methods, or more fit for a specific purpose or user by simplifying or automating extraction, or even adapt it to other problem areas that may have noisy data from which tubular structures must be extracted.

Appendices

Appendix A

Appendices

1.1 List of software for working with 3D data

Software	License model	Scriptable
Commercial / closed source		
Amira	Proprietary, paid	Yes
Analyze	Proprietary, paid	Limited
Aphelion	Proprietary, paid	Yes
Avizo	Proprietary, paid	Yes
CTAn / CTVol	Proprietary, paid (Bruker)	Limited
Dragonfly3D	Proprietary; FreED (free, non-commercial)	Yes (Python)
Image-Pro	Proprietary, paid	Yes
Imaris	Proprietary, paid	Yes
MeVisLab	Proprietary; free academic	Yes
Mimics	Proprietary, paid	Yes
Octopus	Proprietary, paid	Limited
ScanIP	Proprietary, paid	Yes
VG Studio	Proprietary, paid	Yes
Zeiss Inspect	Proprietary, paid	Limited

Software	License model	Scriptable
Open source / freeware		
3D Slicer	BSD (modified)	Yes (Python, C++)
SimVascular	BSD	Yes
VesselKnife	Open source (research)	Yes
VisIt	BSD	Yes (Python)
FIJI / ImageJ	GPLv2 / Public domain	Yes (Java, macros)
Blender	GPLv3	Yes (Python)
Chimera / ChimeraX	Free academic	Yes (Python)
Dream3D	BSD	Yes
IMOD	GPLv2	Yes
MeshLab	GPLv3	Yes
OsiriX (Lite)	LGPL (Lite); proprietary (MD)	Limited
ParaView	BSD	Yes (Python)

Table 1.2: Landscape of tools for working with 3D data, which range from intended for medical or biological use to broader 3D analysis of microCT scans, grouped by licensing model. *Scriptable* indicates if an extension interface exists.

1.2 Nanotom scan parameters

Parameter	Value
Voxel size	6 μm (isotropic)
Number of images	2400
Exposure time	500 ms
Total scan time	20 minutes
Voltage / current	60 kV / 420 μA
Source-detector distance	224.999 mm
Source-sample distance	3.499 mm

Table 1.3: Acquisition parameters as detailed by Wlodarski, pp35 [\[50\]](#)

1.3 Compute characteristics

CollaboratiVessel pipeline steps each have their own CPU and memory cost. We divide the processing into three main steps:

1. Initial Frangi inference: multicore thanks to TubeTK, and limited memory use thanks to tiling
2. Local intensity probability: single core CPU and light on memory as it is iterative on a small subarea.
3. Reconnection process: single core but has the potential for extremely high memory use as inference size increases due to⁵



Figure 1.1: Performance regimes, top to bottom: **(1)** frangi inference, **(2)** single core ops, **(3)** sawtooth RAM use

In order to combat memory overflows, SWAP size was increased.

⁵As mentioned in Scikit docs, `route_through_array` has the option `partition_size` that could be relevant

1.4 Environmental and CO2 impact of the thesis

The main sources of CO2 impact of this thesis are the use of computational resources and the human factor.

1. During the thesis, the laptop of the writer was used: A legion 7i slim with 48GB DDR5, i7-13700H, 8GB RTX4070.
 1. **Production CO2:** The compliance document provided by Lenovo details an estimated carbon footprint of 429g +/- 86g CO2e. For a lifetime of 8 years, and an estimated thesis length of 3 months equivalent full time, this equates to **13.41Kg** CO2e.
 2. **Usage CO2:** The total estimated time dedicated to the coding and writing of the thesis is estimated based on the credits and approximate time investment per credit: 25 credits at 30 hours per credit equates to 750 hours. The laptop was measured over one 2h coding + writing session as using 40.89Wh, equating to a continuous use of 20.45w. This works out to 15.34 kWh, with an estimated Belgian CO2/kWh by the AIB (Association of Issuing Bodies) in 2024 of 131.73 g/kWh, resulting in total emissions of **2.02Kg** CO2.
 3. **AI Utilization:** The impact of LLMs from large providers such as OpenAI is an ongoing research topic and difficult to measure, on top of which are layered issues like the use of Piccolo, UCLouvains aggregation service. A total of 1280 credits were used on Piccolo, with an unknown impact. Claude Code was also used, with a total of 3 separate conversations containing 41 cumulative prompts. The impact of AI use is thus left out of the estimation, with a best guess below.
2. The thesis required meetings, for which the main source of CO2 emissions is the travel to/from Louvain-La-Neuve. This travel was done primarily by train, and the amount of travel directly attributable to the thesis was of 21 round trips, with a distance per trip of 93.2Km at an estimated 16.6g CO2e/Km, equating to **32.49Kg** CO2e.

There was minimal extra data storage required to complete this thesis, however it is important to note that the large files handled do incur emissions if duplicated and stored across devices outside the user's own computer.

The prior calculations equate to a total approximate impact of **47.92 Kg CO2e**.

1.5 Use of LLMs and AIs

The main two AI assistants used in this thesis were Piccolo, offered by UCLouvain and used for writing feedback, and ClaudeCode, by Anthropic, for the expansion and creation of the plugin.

When using Claude Code, multiple events of note occurred:

1. The use of Claude Code for code audits in performance increases generally resulted in many more code changes and solution adjacent changes. For example, when auditing the code to improve performance on subsampled sections, ClaudeCode carried out over hundreds of minor modifications on top of the proscribed main changes
2. The use of coding assistants (Piccolo and Claude) for scripts to help in data management (subsampling and transformations) often resulted in minor discrepancies in outputs that required manual revision
3. The process of requesting Claude Code to add comments also resulted in a re-organization of the code. Readability improved from the perspective of the writer, but the question of originality remains.
4. AI assistants were exceptionally relevant and helpful for the creation of diverse data visualizations and graphing, as well as the creation of validation loops and tools.

1.6 Development computer specifications

Development computer:

Lenovo legion 7i slim, 48GB DDR5, i7-13700H, 8GB RTX4070

Laboratory computer provided to students:

i7-13700, 32GB, T1000 8GB

Appendix B

Supplemental figures

2.1 Visualizations of results

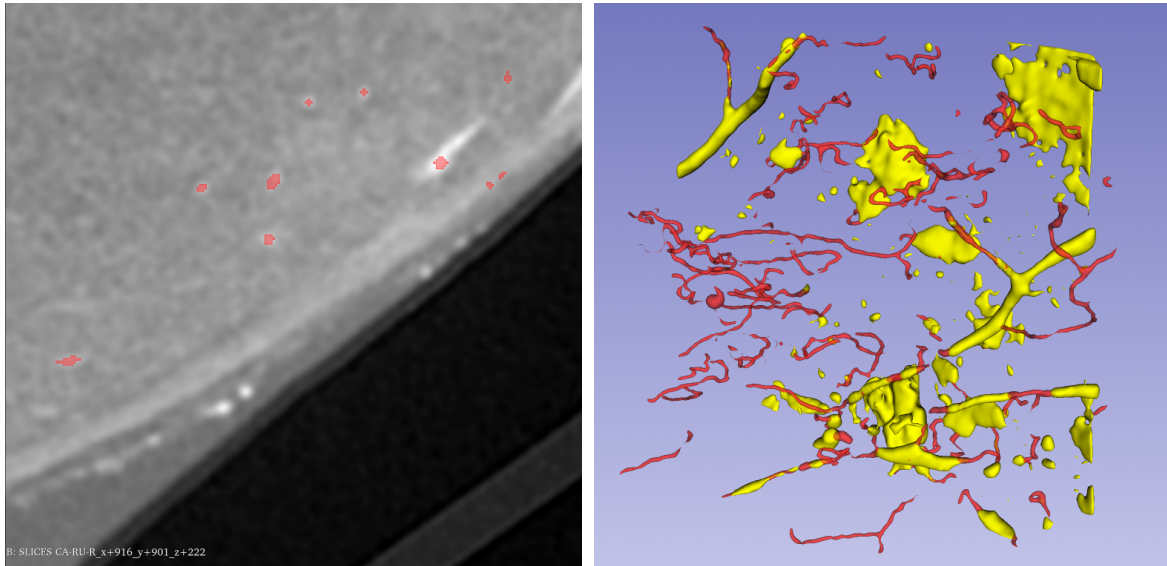


Figure 2.1: CA-RU-R (1) Outer section: **Yellow**: thresholding, **Red**: vessels. Good contrast, more continuous and better defined vessels with some large areas of non vessel-like high valued points that are successfully rejected by CollaboratiVessel

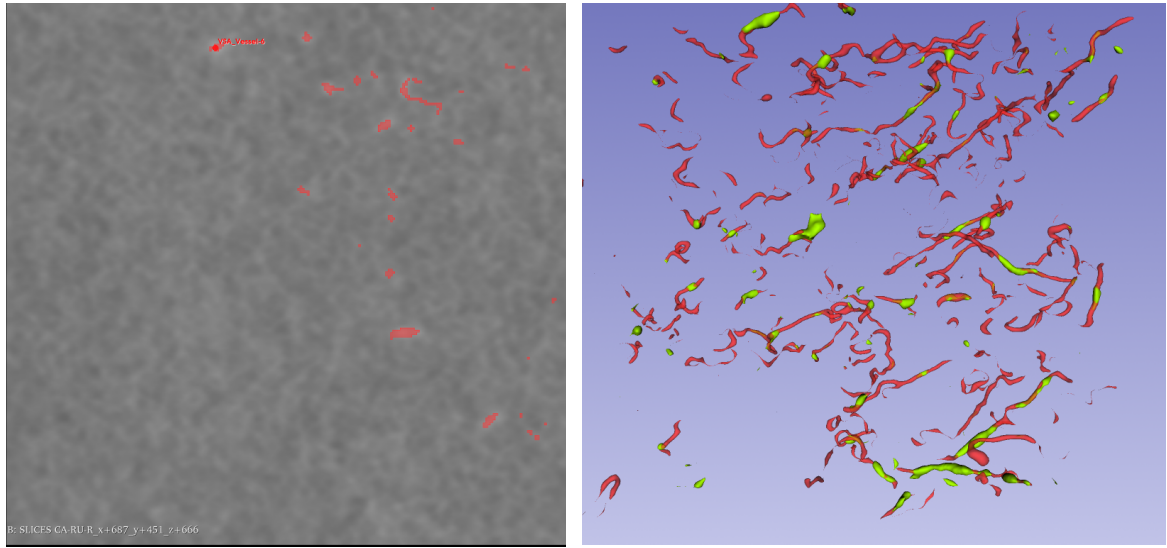


Figure 2.2: CA-RU-R (2) inner **Yellow**: thresholding, **Red**: vessels. Central section: challenging, with low contrast, highly disconnected vessels.

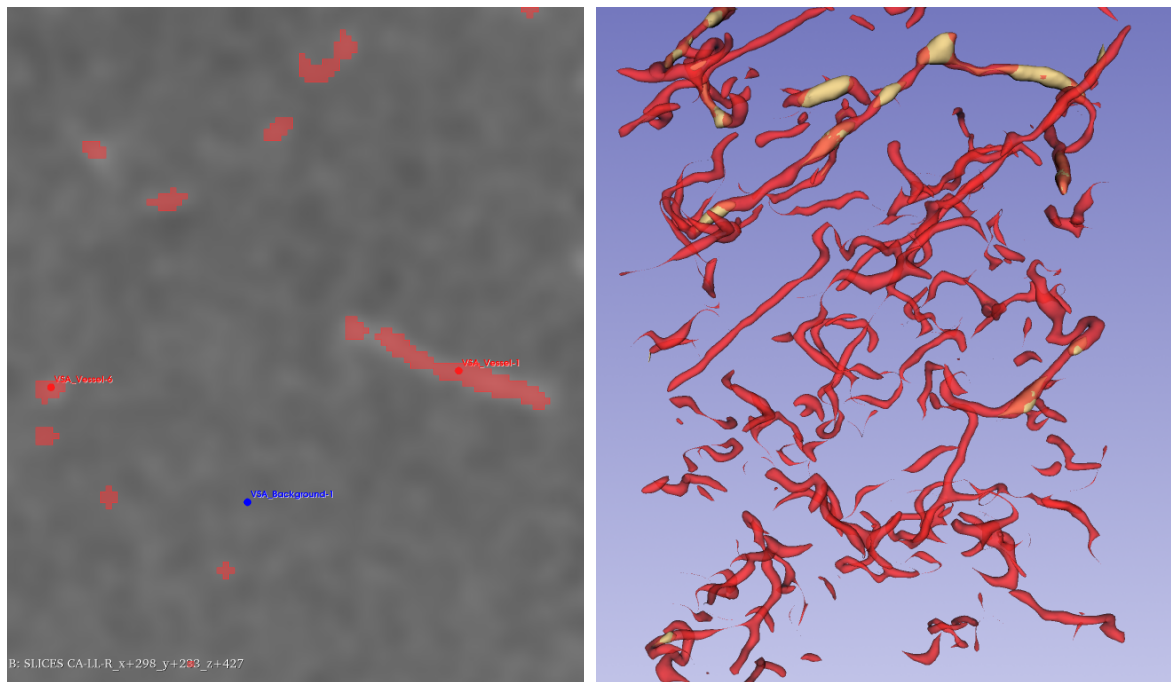


Figure 2.3: CA-LL-R **Yellow**: thresholding, **Red**: vessels. Central section with low contrast, highly disconnected vessels. Vessel prediction shows extensive extrapolation towards bottom slices, which have a gradient and are more noisy, resisting thresholding.

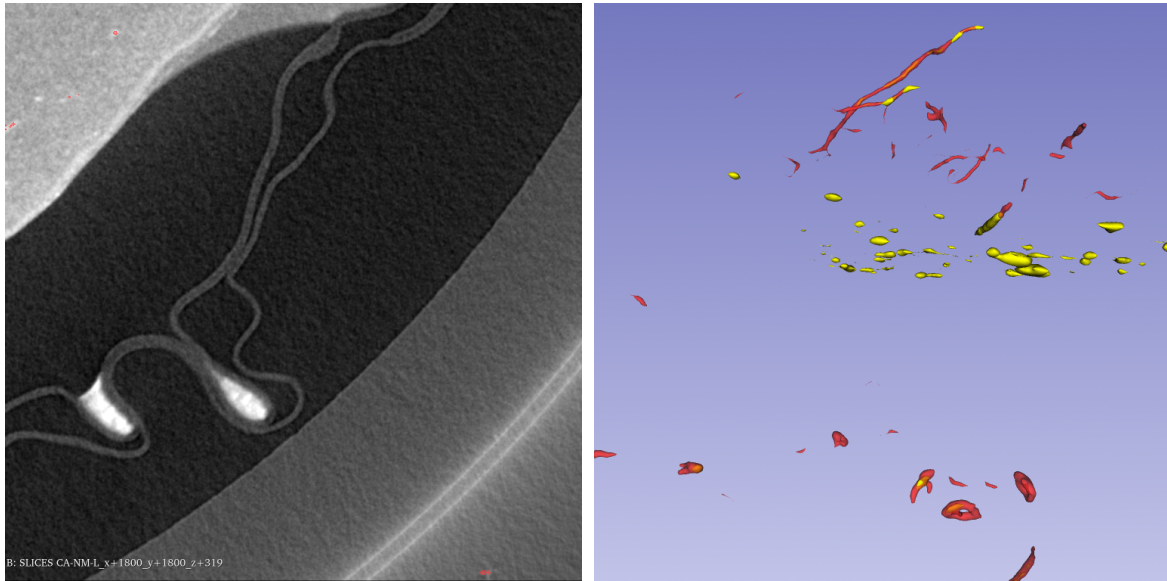


Figure 2.4: CA-NM-L (1) **Yellow**: thresholding, **Red**: vessels. Thresholding fails to reject noisy out of volume elements. Pipeline incorrectly picks up on some vessel-like structures outside of volume.

2.2 Vessel heatmaps

2.2.1 Collated heatmaps

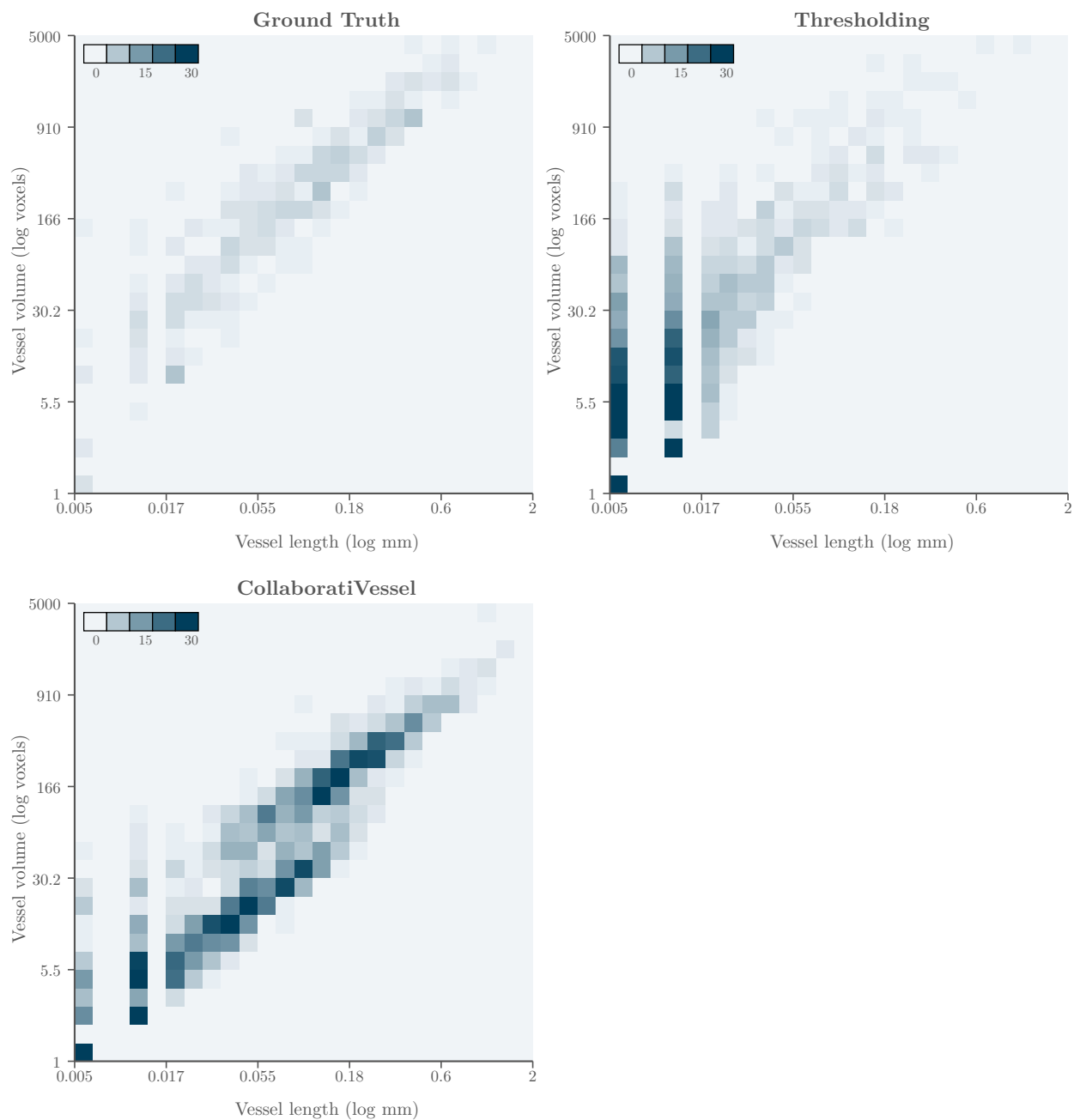


Figure 2.5: Heatmaps of vessel volume/vessel length - all predictions.

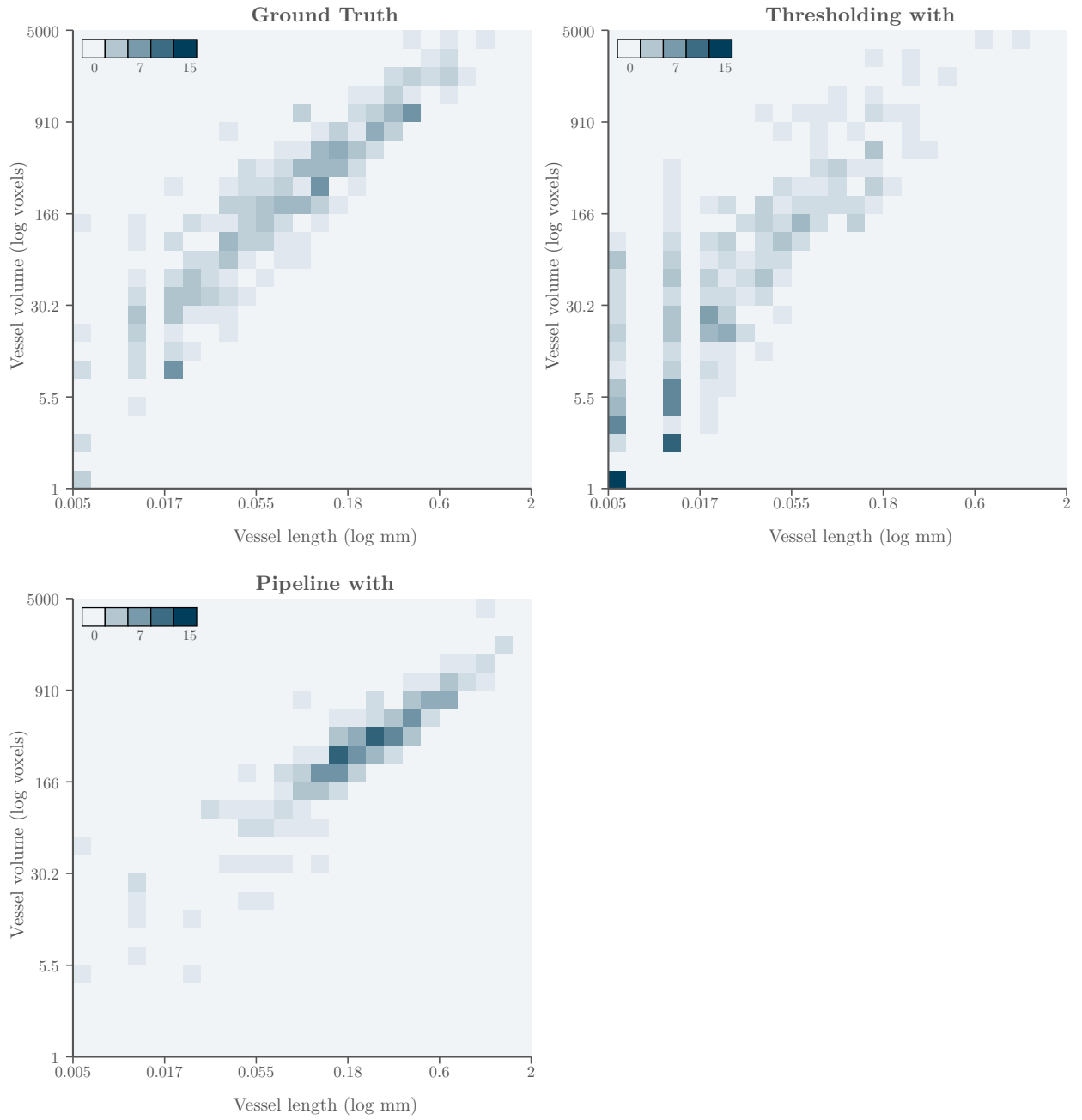


Figure 2.6: **Heatmaps of vessel volume/vessel length - only true predictions:** vessels for thresholding and pipeline are only plotted if they correspond to at least one GT vessel.

2.2.2 Individual heatmaps:

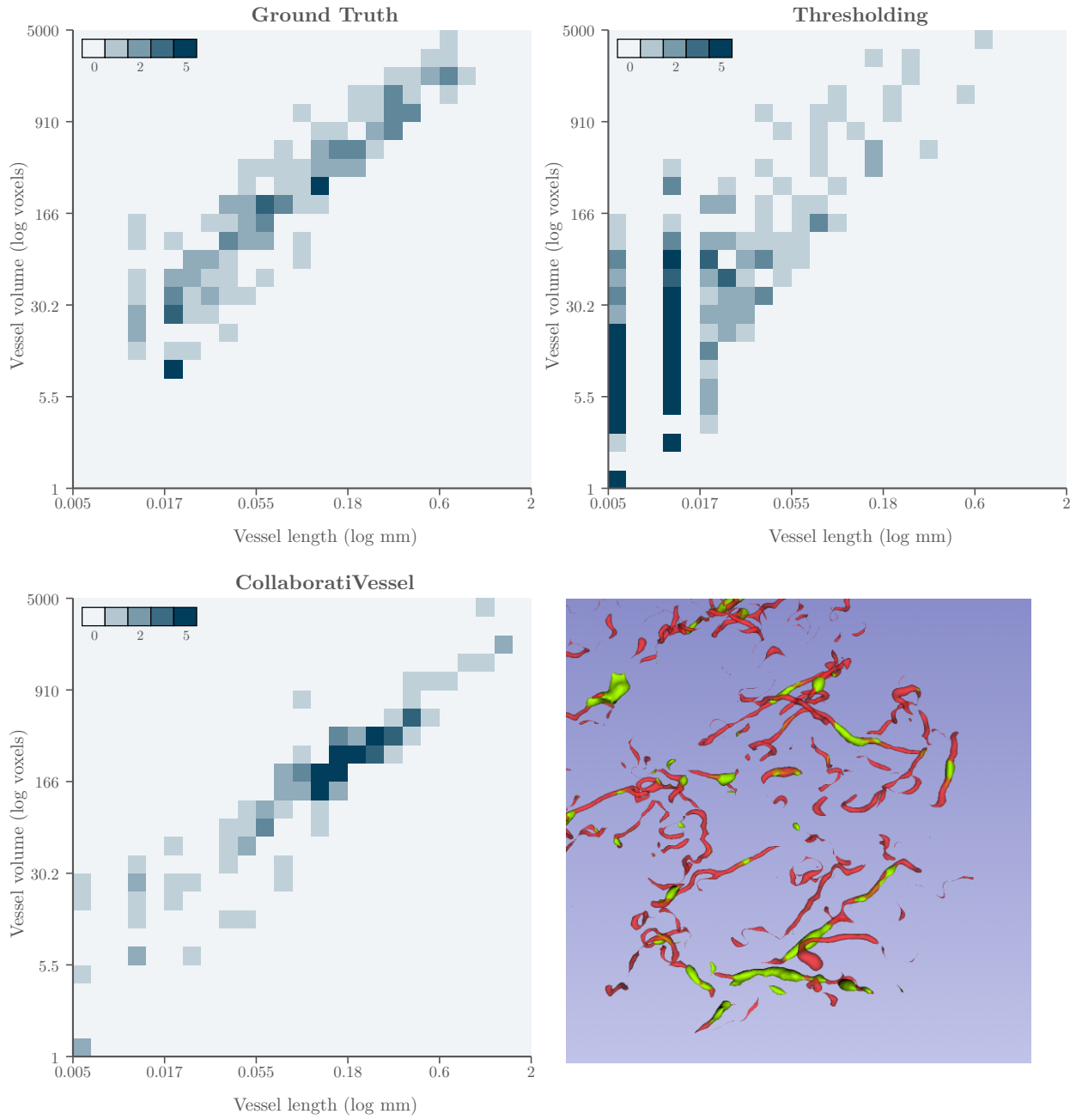


Figure 2.7: CA-RU-R (1): Yellow: thresholding, Red: vessels. predicted vessels tend towards longer sizes, and have a tighter size distribution.

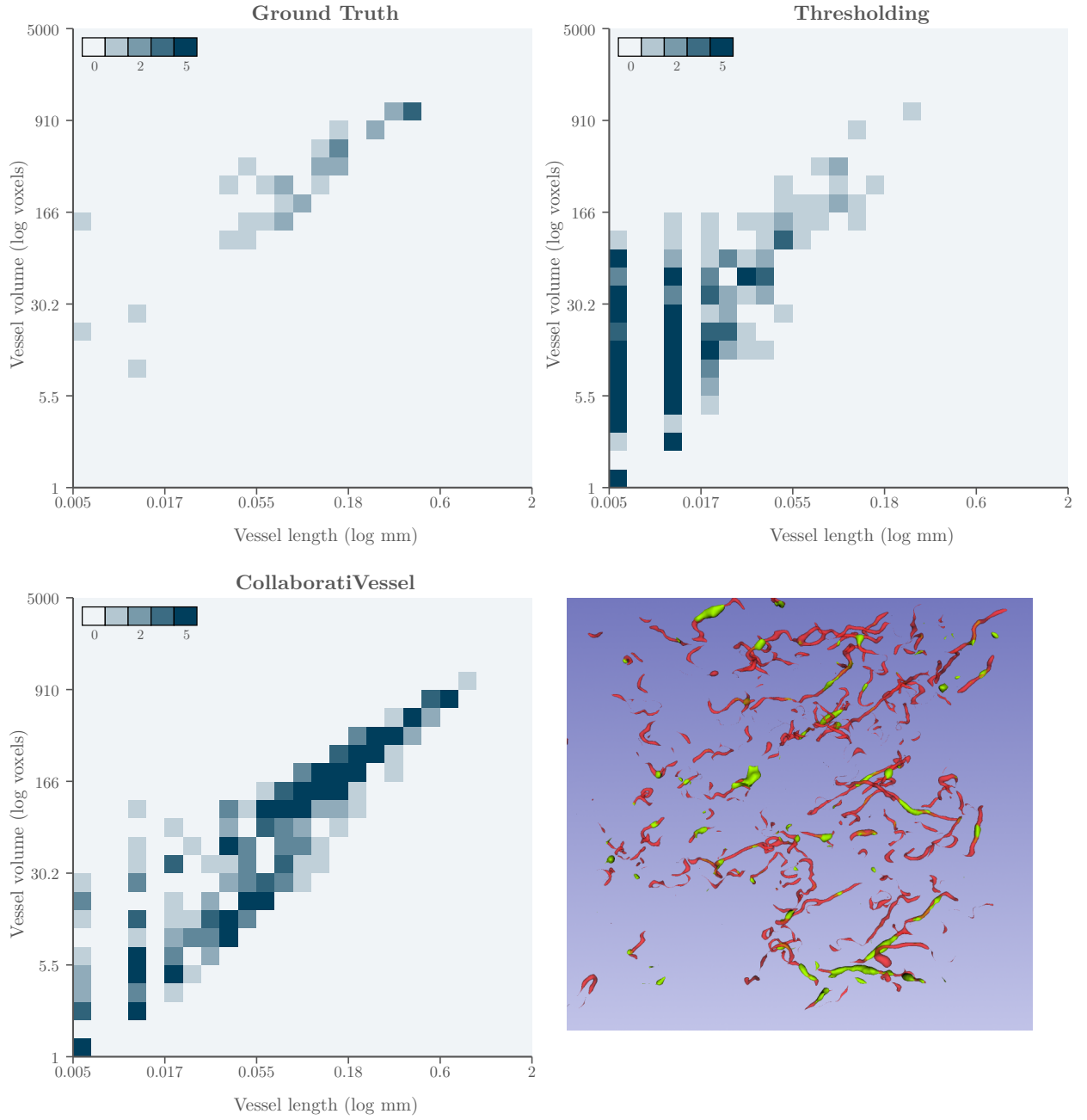


Figure 2.8: CA-RU-R (2): Yellow: thresholding, Red: vessels. extrapolation, as well as smaller vessels being detected and a broader distribution of small vessel sizes than seen on other samples such as CA-RU-R (1).

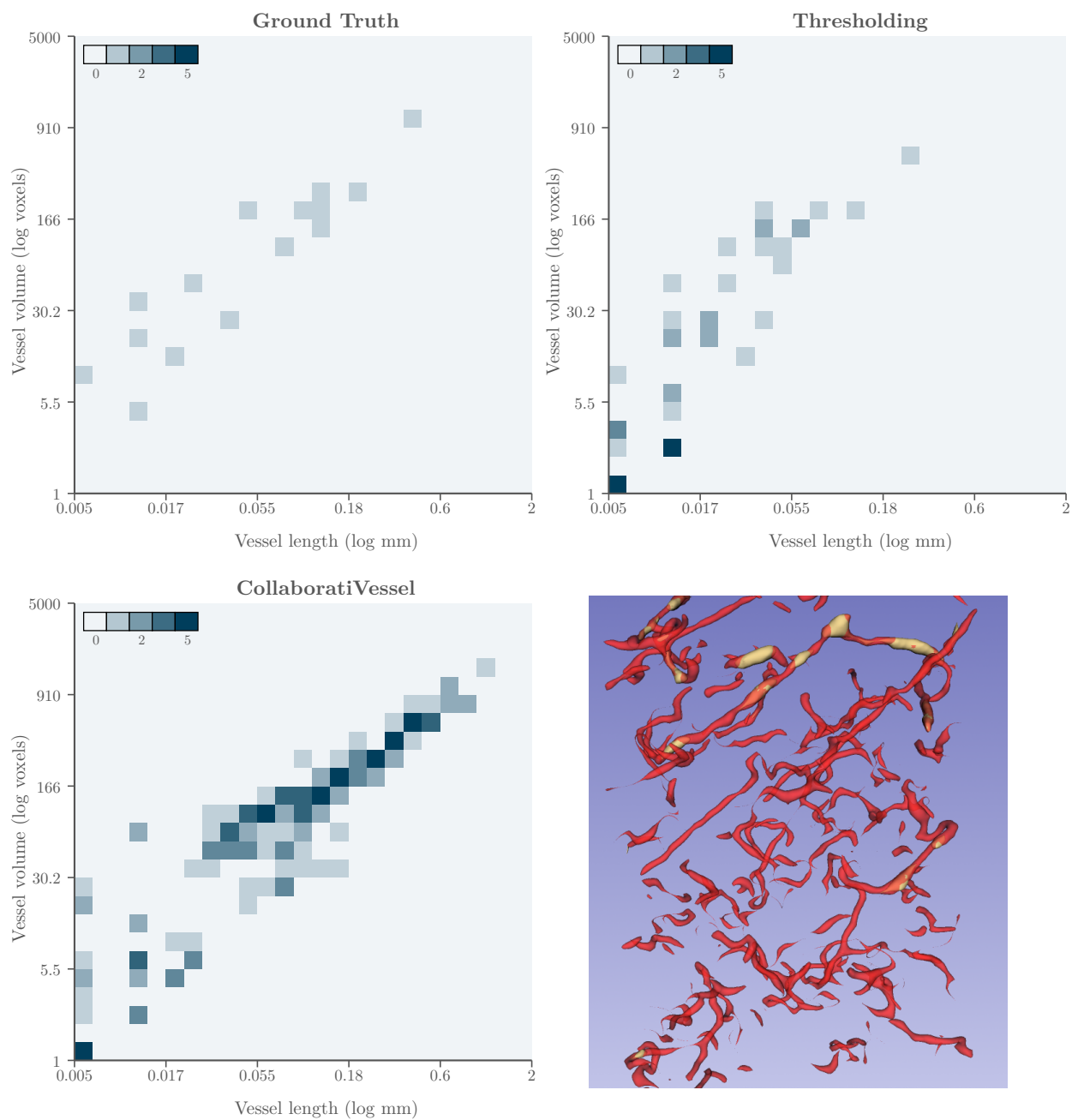


Figure 2.9: CA-LL-R: Yellow: thresholding, Red: vessels. A challenging example with extrapolation by CollaboratiVessel.

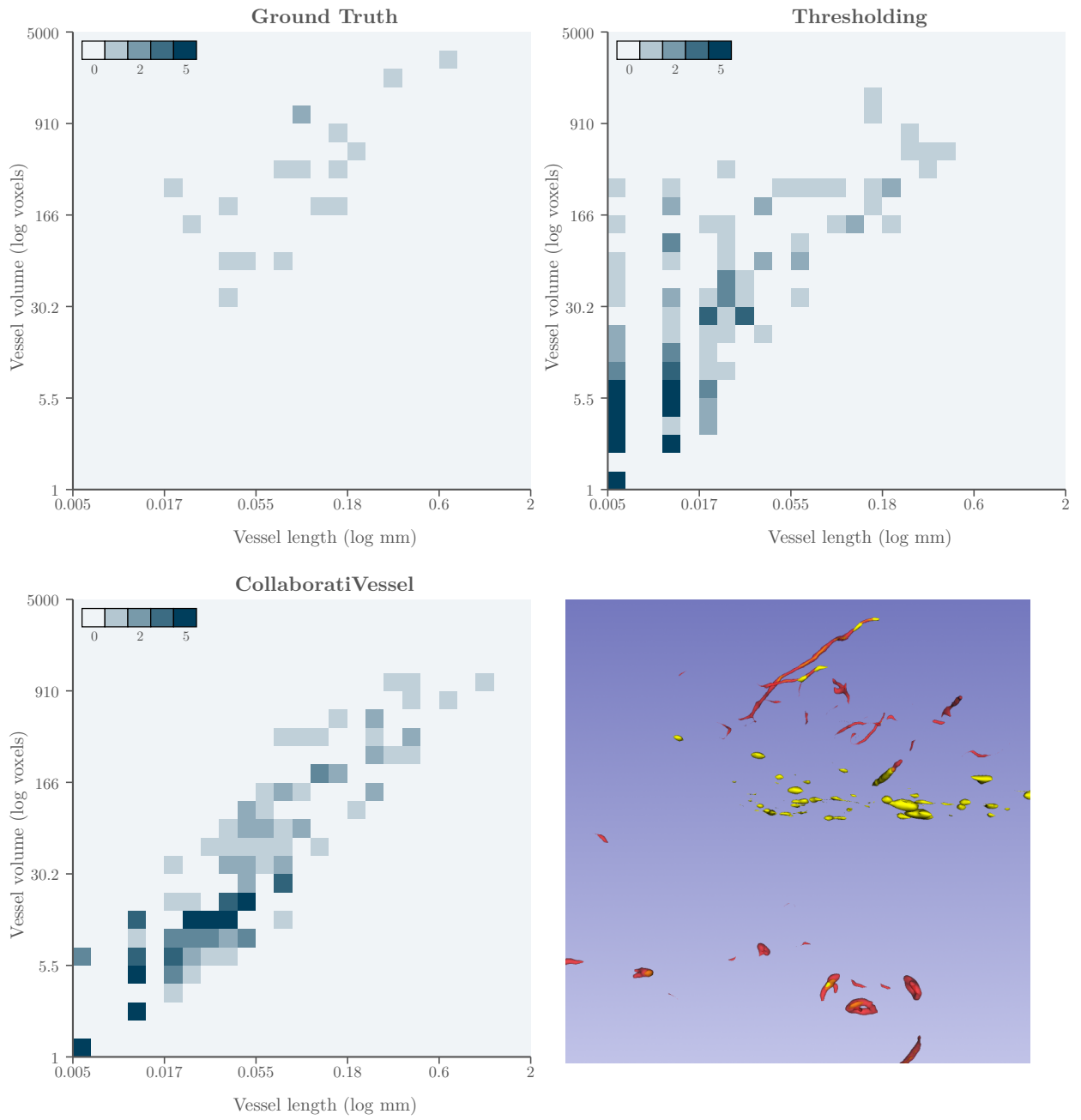


Figure 2.10: CA-NM-L (1) Yellow: thresholding, Red: vessels - the small vessels can be seen on the distribution, as well as the wider spread in vessel sizes.

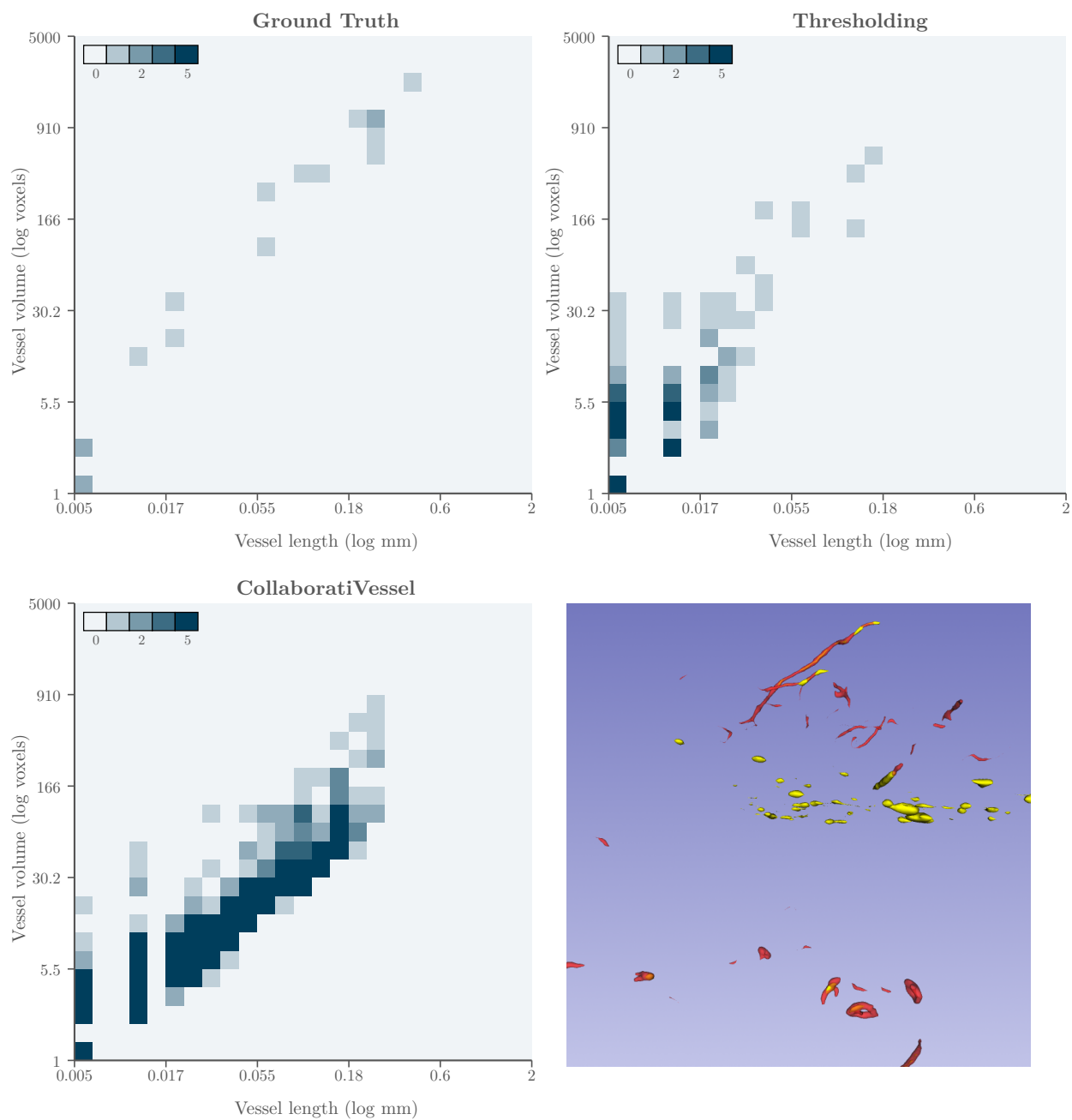


Figure 2.11: CA-NM-L (2) Yellow: thresholding, Red: vessels - the only sample that required modification of vessel parameters: vessel size was adjusted +2 and std deviation +2, to 6 and 5 voxels respectively.

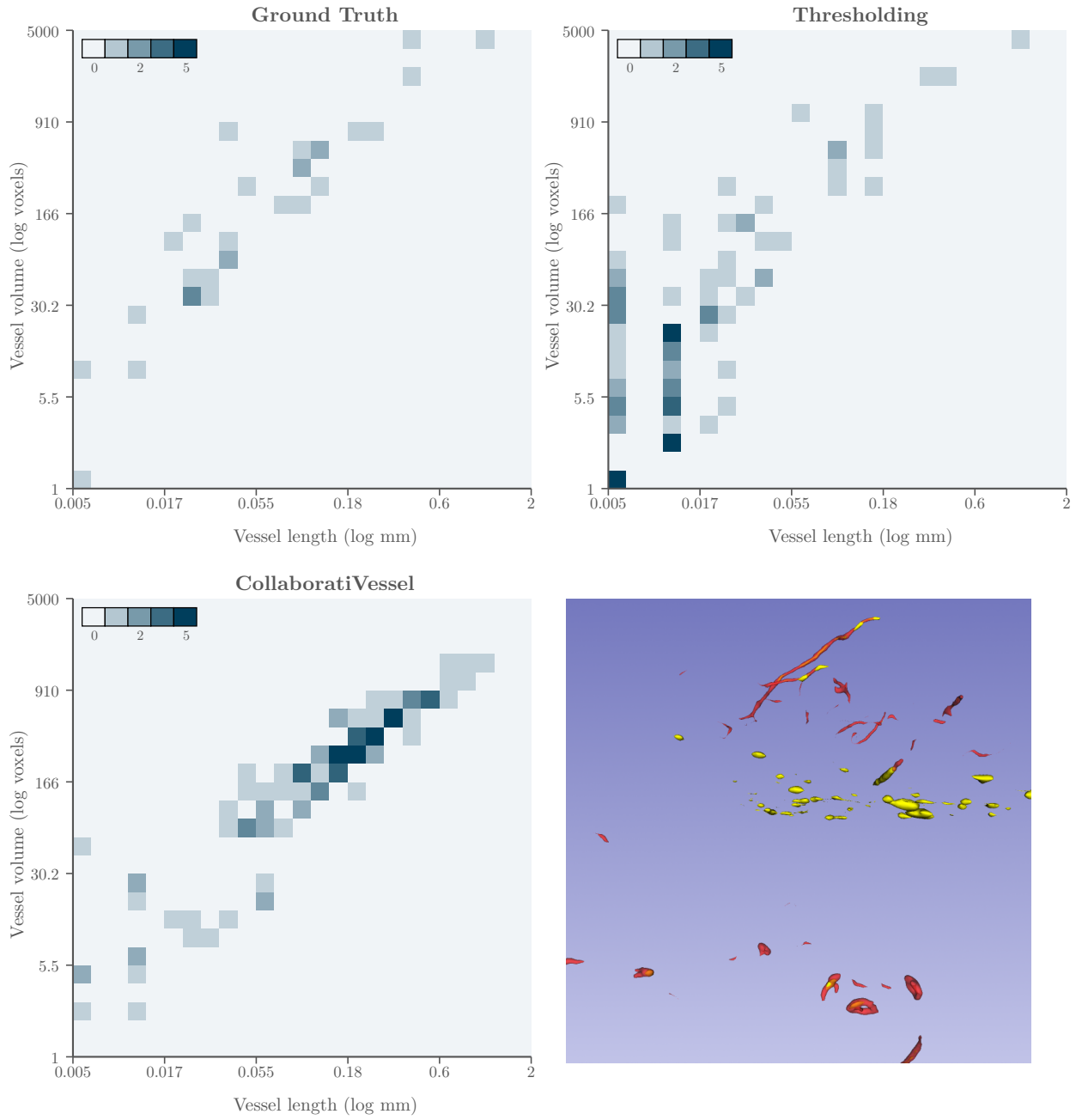


Figure 2.12: CA-LL-L1 Yellow: thresholding, Red: vessels

2.3 Matching graphs

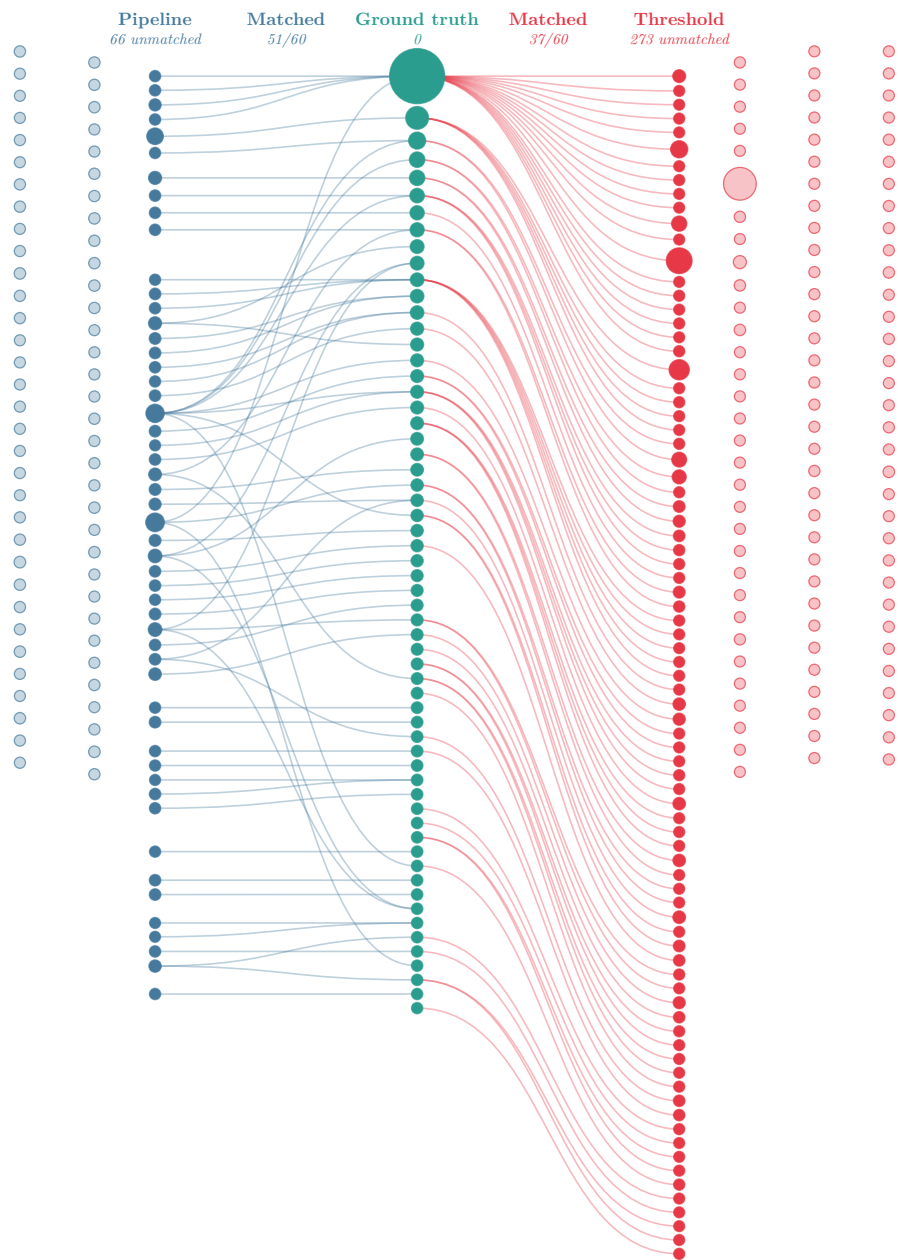


Figure 2.13: CA-RU-R 222

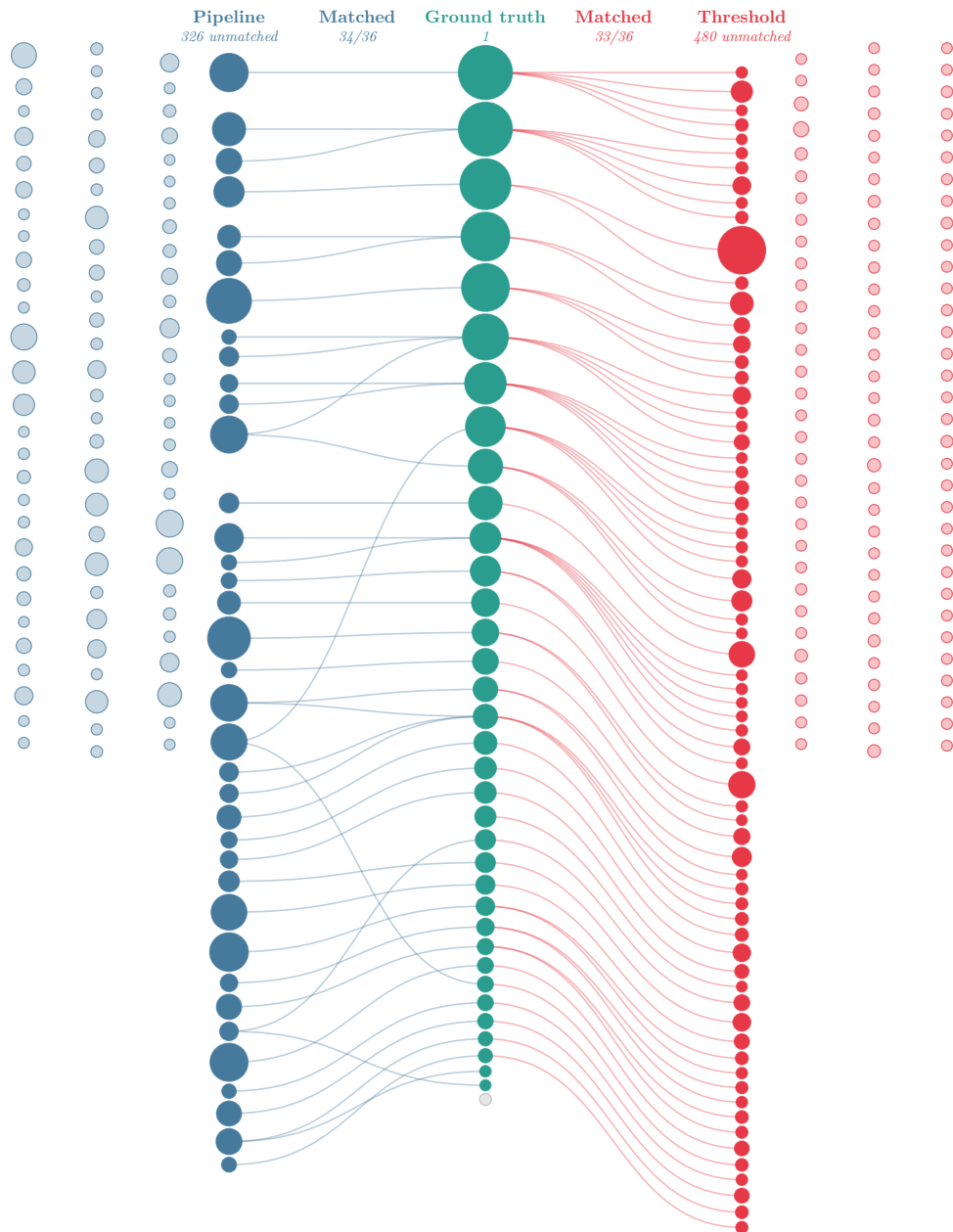


Figure 2.14: CA-RU-R 666

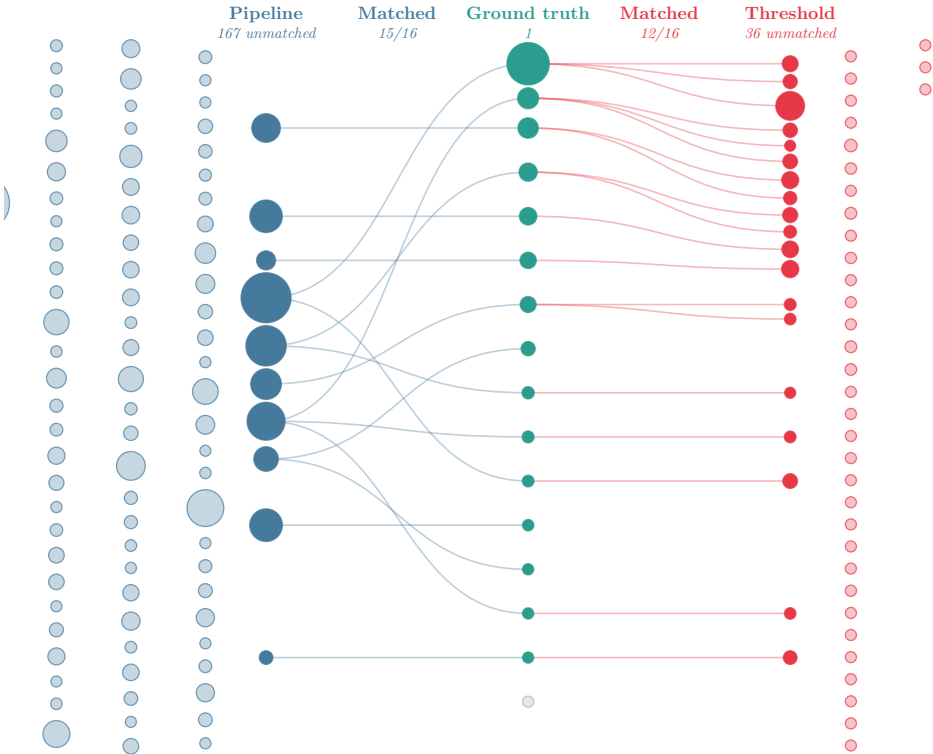


Figure 2.15: CA-LL-R

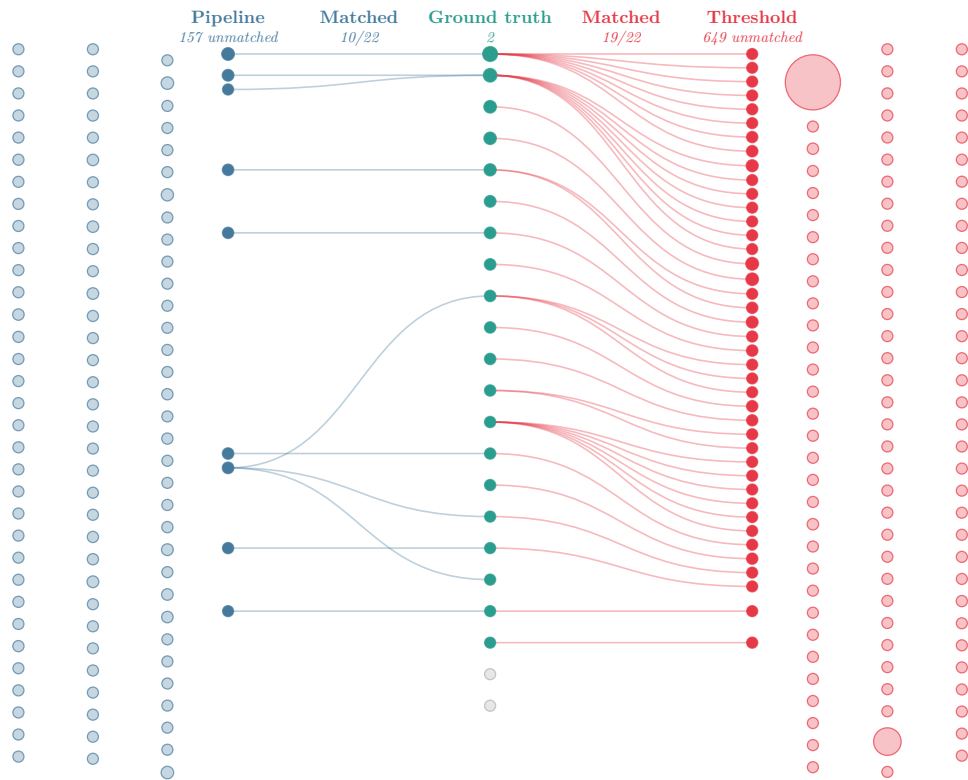


Figure 2.16: CA-NM-L (1)

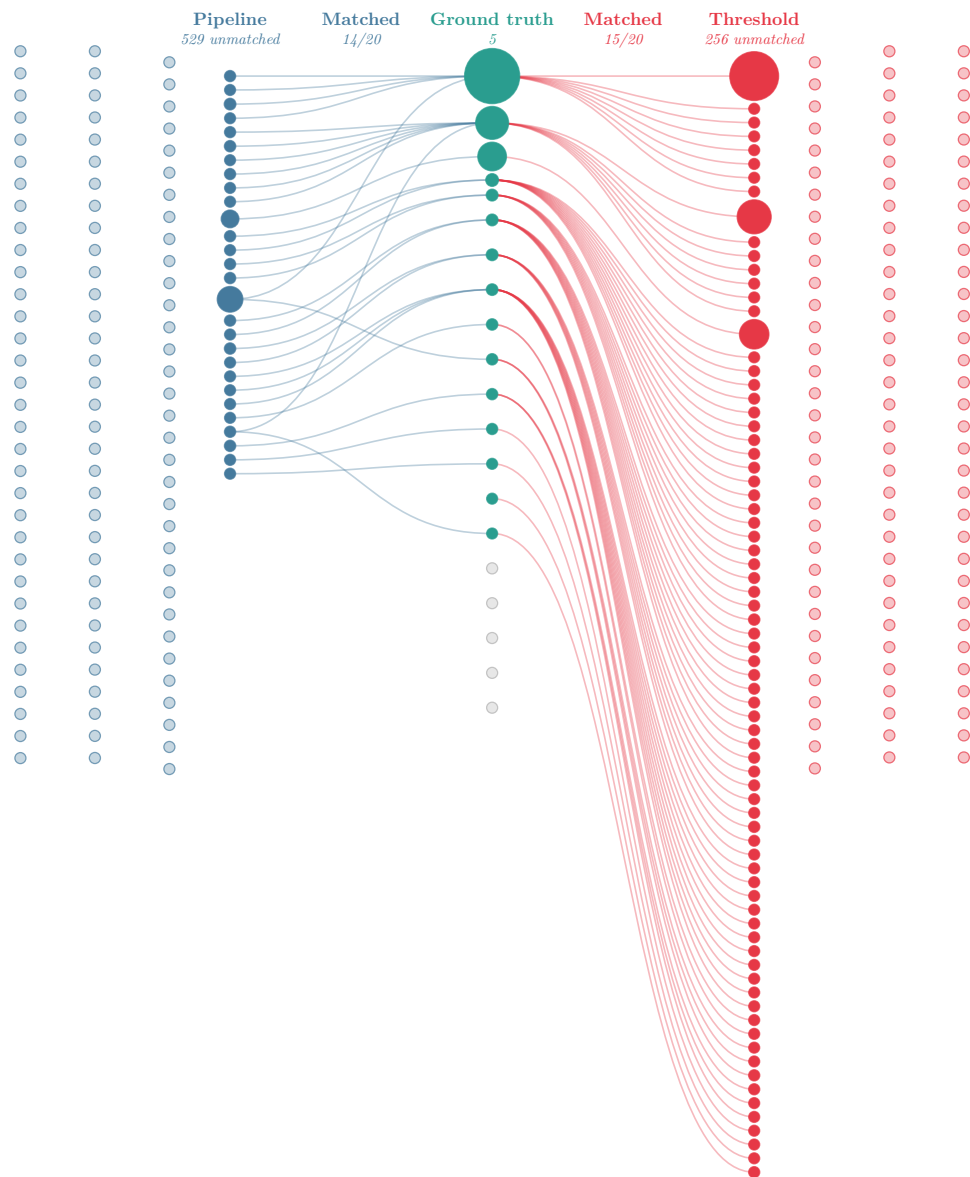


Figure 2.17: CA-NM-L (2)

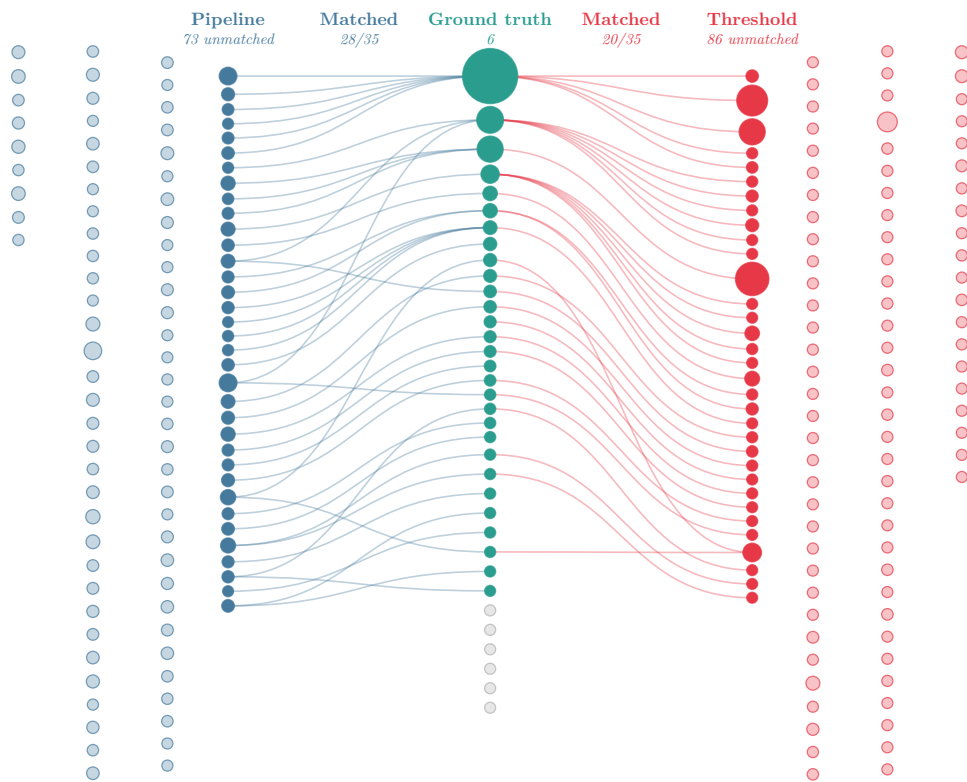


Figure 2.18: CA-LL-L1

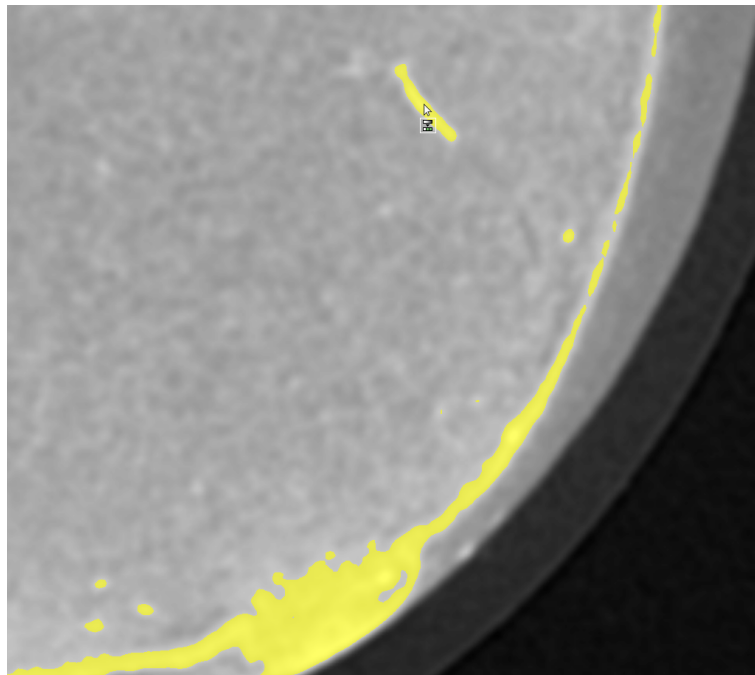


Figure 2.19: The shortcoming of threshold based segmentation visualized, with a “shell” of high valued outside being included whe the threshold accepts the vessel segment.

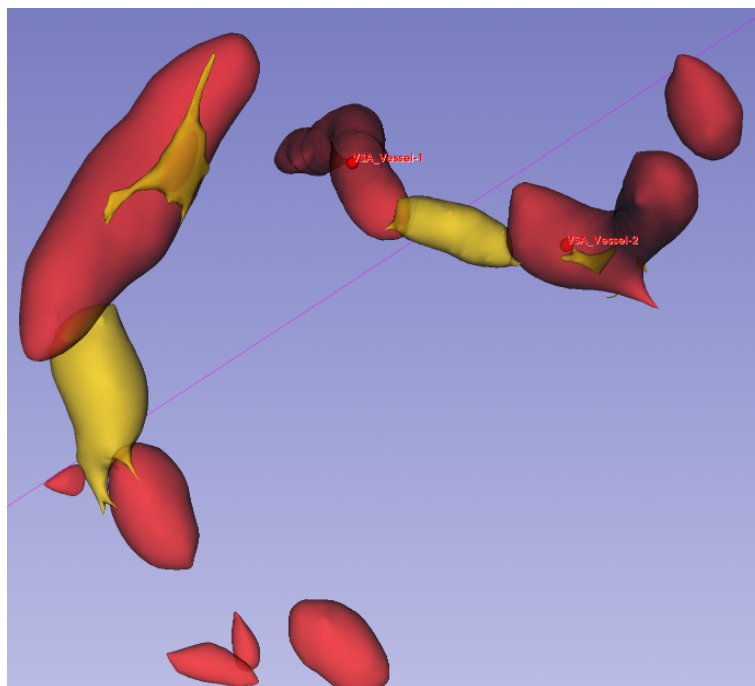


Figure 2.20: Structurally aware gap bridging to reconnect predicted vessels: **Red:** vessels as identified by other steps. **Yellow:** Bridges between tubular endpoints.⁶

⁶3DSlicer smooths visualizations in 3D without combining different classes. The final segmentation here is uniform and continuous.

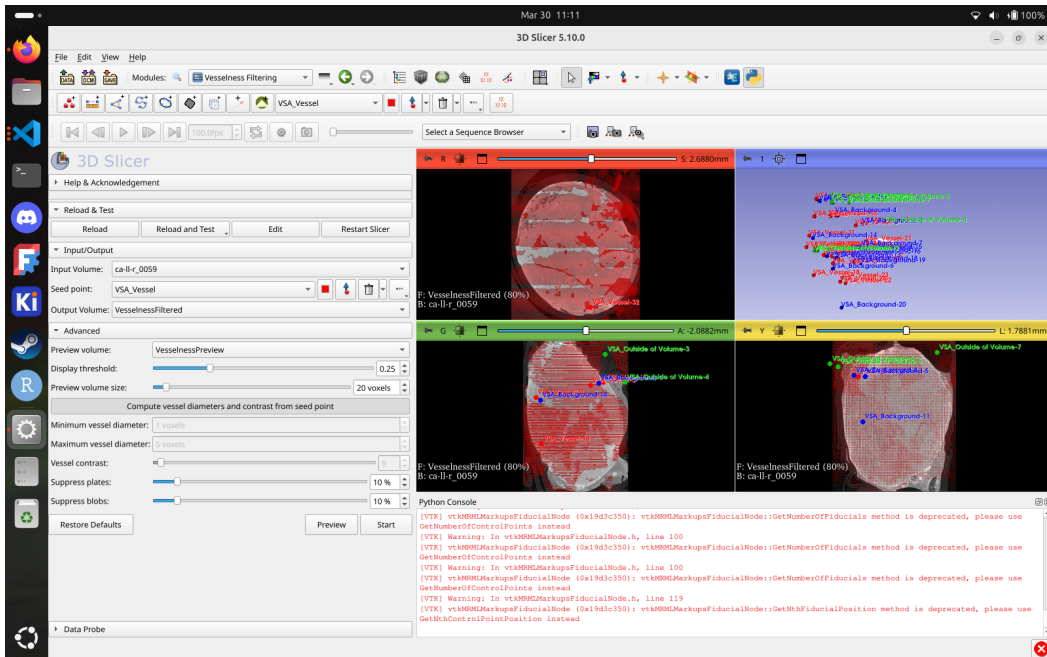


Figure 2.21: Demonstration of one of the failing plugins and the lack of user feedback on what went wrong

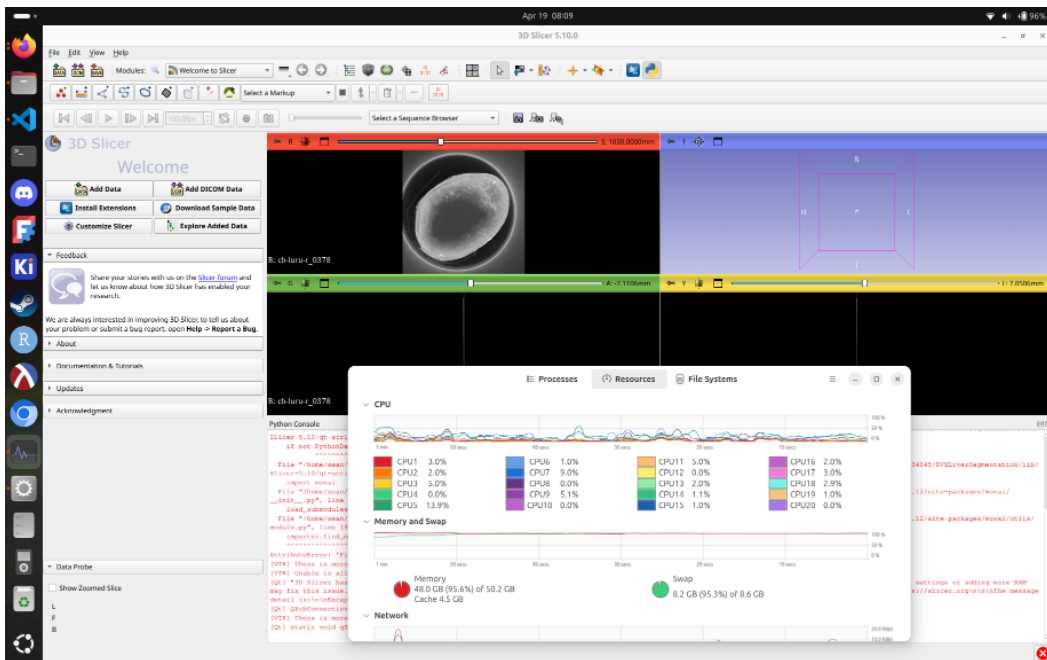


Figure 2.22: RAM use when loading a large dataset, showcasing the issue with 3D Slicers method of loading data into memory: the entire volume must be kept alive, meaning that other operations are not possible.

Bibliography

- [1] T. Balcaen *et al.*, “Revealing the Three-Dimensional Murine Brain Microstructure by Contrast-Enhanced Computed Tomography,” *Frontiers in Neuroscience*, vol. 17, p. 1141615, 2023, doi: [10.3389/fnins.2023.1141615](https://doi.org/10.3389/fnins.2023.1141615).
- [2] L. He, L. R. Long, S. Antani, and G. R. Thoma, “Histology Image Analysis for Carcinoma Detection and Grading,” *Computer Methods and Programs in Biomedicine*, vol. 107, no. 3, pp. 538–556, Sept. 2012, doi: [10.1016/j.cmpb.2011.12.007](https://doi.org/10.1016/j.cmpb.2011.12.007).
- [3] J. M. d. S. e. Silva, I. Zanette, P. B. Noël, M. B. Cardoso, M. A. Kimm, and F. Pfeiffer, “Three-dimensional non-destructive soft-tissue visualization with X-ray staining micro-tomography,” *Scientific Reports*, vol. 5, no. 1, p. 14088, Sept. 2015, doi: [10.1038/srep14088](https://doi.org/10.1038/srep14088).
- [4] S. de Bournonville, S. Vangrunderbeeck, and G. Kerckhofs, “Contrast-Enhanced MicroCT for Virtual 3D Anatomical Pathology of Biological Tissues: A Literature Review,” *Contrast Media and Molecular Imaging*, vol. 2019, p. 8617406, 2019, doi: [10.1155/2019/8617406](https://doi.org/10.1155/2019/8617406).
- [5] A. Khimchenko *et al.*, “Extending two-dimensional histology into the third dimension through conventional micro computed tomography,” *NeuroImage*, vol. 139, pp. 26–36, 2016, doi: <https://doi.org/10.1016/j.neuroimage.2016.06.005>.
- [6] A. Maes *et al.*, “Cryogenic contrast-enhanced microCT enables nondestructive 3D quantitative histopathology of soft biological tissues,” *Nature Communications*, vol. 13, no. 1, p. 6207, Oct. 2022, doi: [10.1038/s41467-022-34048-4](https://doi.org/10.1038/s41467-022-34048-4).
- [7] Z.-M. S. D. C. Ko-Chin Chen Alon Arad, “High-definition neural visualization of rodent brain using micro-CT scanning and non-local-means processing,” *BMC Med. Imaging*, vol. 18, p. 38, Oct. 2018, doi: [10.1186/s12880-018-0280-6](https://doi.org/10.1186/s12880-018-0280-6).
- [8] T. Balcaen, S. Vangrunderbeeck, W. D. Borggraeve, and G. Kerckhofs, “Contrast-Enhancing Staining Agents for Ex Vivo Contrast-Enhanced Computed Tomography: A Review,” *Tomography of Materials and Structures*, p. 100052, Jan. 2025, doi: [10.1016/j.tmater.2025.100052](https://doi.org/10.1016/j.tmater.2025.100052).
- [9] L. Leyssens, C. Pestiaux, and G. Kerckhofs, “A Review of Ex Vivo X-ray Microfocus Computed Tomography-Based Characterization of the Cardiovascular System,” *International Journal of Molecular Sciences*, vol. 22, no. 6, p. 3263, Mar. 2021, doi: [10.3390/ijms22063263](https://doi.org/10.3390/ijms22063263).
- [10] D. Lesage, E. D. Angelini, I. Bloch, and G. Funka-Lea, “A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes,” *Medical Image Analysis*, vol. 13, no. 6, pp. 819–845, 2009, doi: <https://doi.org/10.1016/j.media.2009.07.011>.
- [11] F. G. S. B. D. M. J. M. S.-A. R. H. G. M. N. R. K. W. L. F. K. T. L. Josef Ehling Benjamin Theek, “Micro-CT Imaging of Tumor Angiogenesis Quantitative Measures Describing

- Micromorphology and Vascularization,” *Am. J. Pathol.*, vol. 184, pp. 431–441, Feb. 2014, doi: [10.1016/j.ajpath.2013.10.014](https://doi.org/10.1016/j.ajpath.2013.10.014).
- [12] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979, doi: [10.1109/TSMC.1979.4310076](https://doi.org/10.1109/TSMC.1979.4310076).
- [13] M. Mancas, B. Gosselin, and B. Macq, “Segmentation using a region-growing thresholding,” in *Image Processing: Algorithms and Systems IV*, E. R. Dougherty, J. T. Astola, and K. O. Egiazarian, Eds., SPIE, 2005, pp. 388–398. doi: [10.1117/12.587995](https://doi.org/10.1117/12.587995).
- [14] P. Soille and L. Vincent, “Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 13, no. 6, pp. 583–598, June 1991, doi: [10.1109/34.87344](https://doi.org/10.1109/34.87344).
- [15] Y. Sato *et al.*, “Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images,” *Medical Image Analysis*, vol. 2, no. 2, pp. 143–168, 1998, doi: [https://doi.org/10.1016/S1361-8415\(98\)80009-1](https://doi.org/10.1016/S1361-8415(98)80009-1).
- [16] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, “Multiscale vessel enhancement filtering,” in *Medical Image Computing and Computer-Assisted Intervention — MICCAI’98*, W. M. Wells, A. Colchester, and S. Delp, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 130–137.
- [17] T. Jerman, F. Pernuš, B. Likar, and Ž. Špiclin, “Beyond Frangi: an improved multiscale vesselness filter,” in *Medical imaging 2015: Image processing*, 2015, pp. 623–633.
- [18] J. Lamy, O. Merveille, B. Kerautret, and N. Passat, “A Benchmark Framework for Multiregion Analysis of Vesselness Filters,” *IEEE Transactions on Medical Imaging*, vol. 41, no. 12, pp. 3649–3662, 2022, doi: [10.1109/TMI.2022.3192679](https://doi.org/10.1109/TMI.2022.3192679).
- [19] I. O. team, “Frangi Vesselness – ImageJ.”
- [20] F. Benmansour and L. D. Cohen, “Tubular structure segmentation based on minimal path method and anisotropic enhancement,” *International Journal of Computer Vision*, vol. 92, no. 2, pp. 192–210, 2011.
- [21] R. Izzo, D. Steinman, S. Manini, and L. Antiga, “The Vascular Modeling Toolkit: A Python Library for the Analysis of Tubular Structures in Medical Images,” vol. 3, no. 25. p. 745, 2018. doi: [10.21105/joss.00745](https://doi.org/10.21105/joss.00745).
- [22] D. Lin *et al.*, “\textit{E pluribus unum}: prospective acceptability benchmarking from the Contouring Collaborative for Consensus in Radiation Oncology crowdsourced initiative for multiobserver segmentation,” *Journal of Medical Imaging*, vol. 10, no. S1, p. S11903, Feb. 2023, doi: [10.1117/1.JMI.10.S1.S11903](https://doi.org/10.1117/1.JMI.10.S1.S11903).
- [23] M. Zamir, “Arterial branching within the confines of fractal L-system formalism,” *The Journal of General Physiology*, vol. 118, no. 3, pp. 267–276, Sept. 2001, doi: [10.1085/jgp.118.3.267](https://doi.org/10.1085/jgp.118.3.267).
- [24] X. Yi, E. Walia, and P. Babyn, “Generative adversarial network in medical imaging: A review,” *Medical Image Analysis*, vol. 58, p. 101552, 2019, doi: <https://doi.org/10.1016/j.media.2019.101552>.
- [25] G. Tetteh *et al.*, “DeepVesselNet: Vessel Segmentation, Centerline Prediction, and Bifurcation Detection in 3-D Angiographic Volumes,” *Frontiers in Neuroscience*, vol. 14, p. 592352, 2020, doi: [10.3389/fnins.2020.592352](https://doi.org/10.3389/fnins.2020.592352).
- [26] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” *CoRR*, 2016, [Online]. Available: <http://arxiv.org/abs/1603.02754>

-
- [27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: [10.1145/3065386](https://doi.org/10.1145/3065386).
 - [28] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation.” [Online]. Available: <https://arxiv.org/abs/1505.04597>
 - [29] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation.” [Online]. Available: <https://arxiv.org/abs/1606.06650>
 - [30] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-End Object Detection with Transformers,” *CoRR*, 2020, [Online]. Available: <https://arxiv.org/abs/2005.12872>
 - [31] A. Kirillov *et al.*, “Segment Anything,” 2023, [Online]. Available: <https://arxiv.org/abs/2304.02643>
 - [32] S. A. A. K. J. P. & K. H. M.-H. Fabian Isensee Paul F. Jaeger, “nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation,” *Nature Methods*, vol. 18, pp. 203–211, 2021, doi: [10.1038/s41592-020-01008-z](https://doi.org/10.1038/s41592-020-01008-z).
 - [33] F. Semeraro, A. M. Quintart, S. F. Izquierdo, and J. C. Ferguson, “TomoSAM: A 3D Slicer extension using SAM for tomography segmentation,” *SoftwareX*, vol. 31, p. 102218, 2025, doi: <https://doi.org/10.1016/j.softx.2025.102218>.
 - [34] Y.-C. Chen *et al.*, “Coronary Artery Segmentation in Cardiac CT Angiography Using 3D Multi-Channel U-net.” [Online]. Available: <https://arxiv.org/abs/1907.12246>
 - [35] F. Milletari, N. Navab, and S.-A. Ahmadi, “V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation.” [Online]. Available: <https://arxiv.org/abs/1606.04797>
 - [36] A. A. Taha and A. Hanbury, “Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool,” *BMC medical imaging*, vol. 15, no. 1, p. 29, 2015.
 - [37] Y. Zhou *et al.*, “CF-Loss: Clinically-relevant feature optimised loss function for retinal multi-class vessel segmentation and vascular feature measurement,” *Medical Image Analysis*, vol. 93, p. 103098, 2024, doi: <https://doi.org/10.1016/j.media.2024.103098>.
 - [38] “topolosses - Python package providing topology-aware losses for segmentation tasks.” Accessed: Apr. 29, 2026. [Online]. Available: <https://pypi.org/project/topolosses/>
 - [39] S. Shit *et al.*, “clDice - a Topology-Preserving Loss Function for Tubular Structure Segmentation,” *CoRR*, 2020, [Online]. Available: <https://arxiv.org/abs/2003.07311>
 - [40] Suprosanna and Jocpae, “jocpae/VesselGraph: First public release.” [Online]. Available: <https://zenodo.org/record/5301621>
 - [41] A. Kirillov, K. He, R. B. Girshick, C. Rother, and P. Dollár, “Panoptic Segmentation,” *CoRR*, 2018, [Online]. Available: <http://arxiv.org/abs/1801.00868>
 - [42] C. T. C. Inc., “Dragonfly FreeD License Model.” [Online]. Available: <https://dragonfly.comet.tech/en/products/license-models/freed-license>
 - [43] Y. He *et al.*, “SPROUT: A User-friendly, Scalable Toolkit for Multi-class Segmentation of Volumetric Images,” *bioRxiv*, 2024, doi: [10.1101/2024.11.22.624847](https://doi.org/10.1101/2024.11.22.624847).
 - [44] K.-C. J. F. J. F.-R. J.-C. P. S. B. C. J. D. F. F. S. M. B. J. A. S. M. J. P. S. K. R. Fedorov A. Beichel R., “3D Slicer as an Image Computing Platform for the Quantitative Imaging Network,” *Magnetic Resonance Imaging*, vol. 30, no. (9), pp. 1323–1341, Nov. 2012, doi: [10.1016/j.mri.2012.05.001](https://doi.org/10.1016/j.mri.2012.05.001).

- [45] A. Affane *et al.*, “The R-Vessel-X Project,” *arXiv preprint*, 2025.
- [46] E. B. S. R. Aylward, “Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction,” *IEEE Transactions on Medical Imaging*, vol. 21, pp. 61–75, 2002, doi: [10.1109/42.993126](https://doi.org/10.1109/42.993126).
- [47] P. M. Szczypiński, A. Klepaczko, and R. Olbrycht, “VesselKnife—software for the analysis of tubular structures in biomedical images,” *Computer Methods and Programs in Biomedicine*, vol. 268, p. 108817, 2025.
- [48] A. Updegrove, N. M. Wilson, J. Merkow, H. Lan, A. L. Marsden, and S. C. Shadden, “SimVascular: an open source pipeline for cardiovascular simulation,” *Annals of biomedical engineering*, vol. 45, no. 3, pp. 525–541, 2017.
- [49] L. for Percutaneous Surgery, “Tutorial on 3D Slicer python scripting and programming.” [Online]. Available: <https://github.com/Slicer/SlicerProgrammingTutorial>
- [50] L. Wlodarski, “Contrast-enhanced micro-CT for 3D X-ray based histology to assess the influence of antiangiogenic drugs on tumor vascularization and necrosis.” 2024.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl