

Interactive Visualization of High-Dimensional Voxel Features in Volumetric Path Tracing using Dimensionality Reduction and HDBSCAN

Tilen Smrdel,¹ Lara Smrdel,² Žiga Lesar,^{1,3,} and Ciril Bohak^{1,}

¹ University of Ljubljana, Faculty of Computer and Information Science,
Večna pot 113, 1000, Ljubljana, Slovenia

² University of Ljubljana, Faculty of Pharmacy,
Aškerčeva cesta 7, 1000 Ljubljana

³ University of Nova Gorica,
Vipavska 13, 5000 Nova Gorica, Slovenia

ts0277@student.uni-lj.si, lara.smrde@ffa.uni-lj.si, ziga.lesar@ung.si, ciril.bohak@fri.uni-lj.si

Abstract

This paper introduces an interactive visualization framework for exploring complex volumetric microscopy data, using multi-dimensional feature extraction and dimensionality reduction to improve structural analysis. Each voxel is assigned an 11-dimensional feature vector to capture both local and global context. After uniform sampling, we apply UMAP and t-SNE for dimensionality reduction, followed by HDBSCAN for density-based clustering. The resulting clusters are mapped back to the original volume to construct an interactive transfer function, with Gaussian blobs representing cluster centers. Expert evaluation of troponin-labeled cardiomyocytes shows that this analysis offers greater sensitivity in detecting treatment-induced structural remodeling than visual inspection of 2D slices. Non-treated samples display higher morphological heterogeneity, with multiple small regions of varying intensity, while treatment with 500 nM carfilzomib for 24 hours leads to fewer, more dominant phenotypic states characterized by localized protein clustering and nuclear enlargement. Our results demonstrate that this unbiased, quantifiable workflow successfully reveals treatment-associated structural changes and protein redistributions that remain undetected in conventional 2D immunofluorescence evaluation.

Keywords

Dimensionality Reduction, Density-Based Clustering, Interactive Transfer Functions, Volume Rendering, High-Dimensional Feature Extraction

1 INTRODUCTION

Advances in high-resolution volumetric microscopy have enabled the acquisition of increasingly detailed three-dimensional representations of biological structures [CB22]. Modern immunofluorescence (IF) imaging produces dense volumetric datasets containing rich spatial and intensity-based information about protein organization, cellular morphology, and treatment-induced structural changes. However, despite substantial progress in image acquisition, the exploration and interpretation of such data remain challenging. In practice, biological

analysis frequently relies on visual inspection of slices, maximum-intensity projections, or basic two-dimensional (2D) measurements, which provide only limited insight into the complex spatial organization present in volumetric datasets.

From a visualization perspective, the primary challenge lies in transforming high-dimensional volumetric data into representations that preserve meaningful structural relationships while remaining interpretable to domain experts. Subtle protein redistributions and localized morphological changes are often difficult to identify directly within raw voxel data due to visual clutter, overlapping structures, noise, and the inherent heterogeneity of biological samples. Traditional visualization approaches focus primarily on rendering intensity distributions, but they frequently lack mechanisms for revealing latent structural patterns, phenotypic similarities, or spatially localized variations within the volume. Consequently, important treatment-associated changes

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

may remain visually indistinguishable even when present within the data.

Recent advances in dimensionality reduction and unsupervised clustering offer new opportunities for visual analytics in volumetric microscopy. By embedding voxel-level features into lower-dimensional latent spaces, it becomes possible to organize complex structural relationships into interpretable visual representations that support exploratory analysis and pattern discovery. However, effectively integrating spatial context, local neighborhood information, and volumetric rendering into a unified analysis workflow remains an open challenge. In particular, there is a need for interactive visualization approaches that not only identify hidden phenotypic structures, but also allow these structures to be mapped back into their original spatial context within the biological sample.

To address these challenges, we propose an interactive visualization and analysis framework for volumetric microscopy data that combines high-dimensional feature extraction, dimensionality reduction, density-based clustering, and three-dimensional visual mapping. Our approach transforms voxel-level microscopy information into a structured feature space that captures both signal intensity and local spatial context. The resulting embeddings are clustered to identify phenotypic states, which are subsequently visualized within interactive transfer functions (TFs) using Gaussian blob representations. This enables users to explore spatially localized structural patterns and treatment-associated redistributions directly within the volumetric domain.

We demonstrate the applicability of the proposed framework by investigating the effects of carfilzomib, a proteasome inhibitor, on cardiomyocyte structure. Using our visualization pipeline, we identify treatment-associated redistributions of contractile proteins and subtle changes in nuclear morphology that are difficult to detect using conventional IF visualization techniques. These results highlight the potential of combining visual analytics and unsupervised learning for exploratory analysis of complex biological volumes.

The primary contributions of this work are as follows:

- **High-Dimensional Feature Integration:** We introduce an 11-dimensional feature vector for volumetric microscopy that combines voxel intensity, gradient magnitude, spatial coordinates, and the intensities of six immediate neighbors to capture local structural context.
- **Unbiased Structural Discovery:** We demonstrate a workflow using Uniform Manifold

Approximation and Projection (UMAP)/t-distributed Stochastic Neighbor Embedding (t-SNE) and Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) that identifies subtle protein redistributions and morphological patterns not apparent through standard IF analysis.

- **Comparative Dimensionality Reduction:** We evaluate the efficacy of UMAP versus t-SNE in a biological context, finding that UMAP provides greater intensity differentiation in identified clusters, particularly for highlighting treatment-associated nuclear phenotypes.

The rest of the paper is structured as follows. We first present related work in Section 2. Next, we present our approach in Section 3. We present results in Section 4, and discuss their quality in Section 5. Finally, we present the conclusions in Section 6.

2 RELATED WORK

Volumetric data visualization is essential across scientific disciplines, particularly in medical diagnostics and biology. Recent advancements have emphasized interactive, efficient, and accessible processing of three-dimensional datasets within web environments.

2.1 Web-Based Volumetric Rendering

Interactive visualization of volumetric biological and medical datasets has been significantly advanced by platform-agnostic web technologies. Real-time platform-independent volumetric path tracing (VPT) in WebGL 2.0 [LBM18] enables physically-based light transport simulation directly in a browser. Frameworks such as Med3D demonstrate the feasibility of rendering high-resolution radiological and microscopy data while providing easier collaboration and platform independence compared to traditional desktop tools [Boh+20].

To support these computationally intensive tasks, efficient data management is crucial. Solutions like the Blocky Volume Package [LBM23] offer web-friendly storage and compression to address the memory demands of large volumetric datasets. In densely populated or “crowded” biological volumes, visibility management systems [Les+24] enable users to group object instances and control their opacity, selective sparsification, or colorization to reduce visual clutter and highlight relevant structures.

2.2 Dimensionality Reduction and Clustering

To analyze complex multidimensional voxel attributes, researchers use dimensionality reduction and clustering techniques. t-SNE [MH08] is widely used to visualize high-dimensional data by reducing it to a lower-dimensional representation while preserving local segmentation quality. More recently, UMAP [MHM18] has emerged as an efficient alternative that better preserves global data structure. These methods are often paired with density-based clustering algorithms, such as HDBSCAN [CMS13], which can identify meaningful regions and clusters of varying shapes through hierarchical density estimates.

2.3 Interactive Transfer Function Design

The design of TFs is a central challenge in making volumetric data interpretable. Automated approaches [SVG06] have used hierarchical clustering of material boundaries to map data features to visual properties. In multivariate settings, designing TFs based on user-selected samples [ZH13] has been shown to provide more intuitive exploration of complex volumes. Additionally, the use of “scented widgets” [WHA07]—user interface components embedded with visualizations of underlying data distributions—improves navigation by providing visual cues that guide users through high-dimensional attribute spaces. Beyond standard TFs, techniques such as indexed-points parallel coordinates [ZW17] are used to visualize multivariate correlations, assisting in the discovery of patterns within large scientific datasets.

Our framework builds on these advancements by integrating 11-dimensional feature vectors with UMAP, t-SNE and HDBSCAN to generate cluster-based TF primitives. By mapping these results to interactive Gaussian elements in a web-based path-tracing environment, we provide a data-driven approach to visibility control and structural analysis.

3 METHODS

To address the challenges of high-dimensional microscopy analysis, we extended the existing VPT framework [LBM18] to meet the specific requirements of confocal image data. To ensure broad compatibility with standard laboratory outputs, the system uses serialized image stacking as its primary input method. The workflow is organized into three distinct stages—preprocessing, clustering, and visualization—each serving a critical role in

transforming raw voxel data into an interpretable, clustered volume. The process begins with the acquisition of TIFF image stacks, which are transformed into a custom BVP volume format during the **Preprocessing** stage. In the **Clustering** stage, 11-dimensional feature vectors are sampled and processed using UMAP or t-SNE for dimensionality reduction, followed by HDBSCAN to identify phenotypic clusters. Finally, the **Visualization** stage maps these clusters back to voxel space, enabling interactive exploration through cluster-based transfer functions within the VPT environment. This integrated pipeline is illustrated in Figure 1.

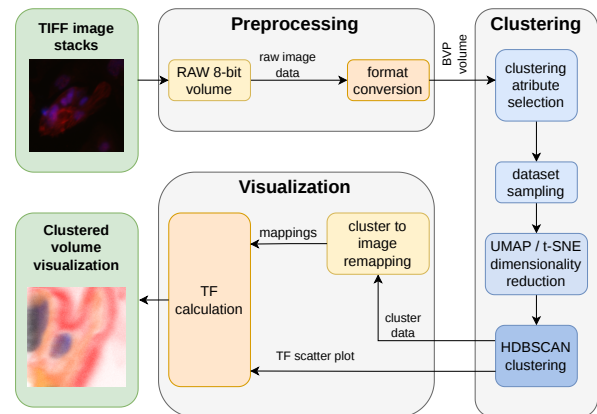


Figure 1: The proposed visualization and analysis pipeline.

3.1 Preprocessing

The preprocessing stage transforms raw microscopy data into a structured format suitable for high-dimensional analysis and real-time visualization within the VPT framework. This workflow includes image stack acquisition, data normalization, bit-depth conversion, and final transformation into the BVP format.

The process starts with acquiring image stacks using confocal microscopy, such as the Operetta CLS High-Content Analysis System. These datasets are typically exported in TIFF format to preserve full spatial resolution and essential metadata of the recorded slices. Each dataset consists of a sequence of 2D images, with each slice representing a consecutive layer of the biological sample.

To ensure compatibility with the visualization pipeline, the image stacks undergo a conversion process. This involves de-interweaving the multi-channel stacks and normalizing the images into raw 8-bit intensity data. We use FIJI¹, an industry-standard open-source image processing tool, for this task. FIJI was chosen for its ubiquity

¹<https://imagej.net/software/fiji/>

in pharmaceutical and microscopy research; since domain experts are already familiar with its environment, they can readily verify the integrity of the preprocessing results.

Once the raw 8-bit data is prepared, the image stack is converted into the BVP format [LBM23]. BVP is a specialized volume data format designed for web-based applications, optimized for efficient streaming and block-wise access to large volumetric datasets. Finally, the generated BVP file is uploaded to the VPT web application, serving as the primary input for the subsequent clustering and interactive visualization stages of the pipeline.

3.2 Clustering

After preprocessing, the pipeline advances to the clustering phase, where voxel-wise feature vectors are generated and grouped using nonlinear dimensionality reduction and density-based clustering. This stage is essential for producing the data needed for automated TF synthesis and interactive exploration within the VPT framework.

At the beginning of the clustering stage, the user selects the dimensionality reduction method and its associated hyperparameters. The system supports two approaches: UMAP and t-SNE. Regardless of the selected reduction method, the resulting low-dimensional data are grouped using HDBSCAN. These parameters are used for dataset construction and computational analysis.

To characterize the volumetric data, a voxel-wise feature representation is constructed for clustering and transfer function generation. Each voxel is represented by an 11-dimensional feature vector, as detailed in Table 1. The feature design is partially inspired by the work of Yoo et al. [YKJ25], who demonstrated that multi-attribute voxel descriptors improve the separability of structural patterns in dimensionality reduction-based visualization pipelines. In contrast to their iterative transfer-function editing workflow, our method focuses on automated structural grouping and direct reintegration of clustered features into interactive volumetric visualization.

The selected attributes were chosen to capture complementary structural properties relevant to volumetric microscopy analysis. The voxel intensity I represents the local signal response and serves as the primary descriptor of protein density. The gradient magnitude $|\nabla I|$ captures local intensity variation and highlights structural boundaries, enabling the clustering process to distinguish homogeneous regions from edges and transitional areas. Spatial coordinates (x, y, z) preserve positional context within the volume, which is particularly important

for identifying spatially localized biological structures and preventing structurally unrelated regions with similar intensity profiles from collapsing into the same cluster. Finally, the six directly adjacent neighboring voxel intensities $N_1, \dots, N_6$ encode the immediate local neighborhood around each voxel, providing contextual information about surrounding density distributions and improving robustness against noise and isolated intensity fluctuations.

The dimensionality of the feature vector was selected based on both computational and practical considerations. Yoo et al. [YKJ25] observed diminishing visualization benefits when substantially increasing the number of voxel attributes. In our experiments, the 11-dimensional representation provided a favorable balance between structural expressiveness, clustering stability, and computational efficiency, while enabling interactive processing of large volumetric microscopy datasets.

vector	type	description
I	F32	Voxel intensity at coordinates (x, y, z)
$ \nabla I $	F32	Gradient magnitude at the voxel position
(x, y, z)	$3 \times \text{F32}$	Voxel position within the volume
$N_1, \dots, N_6$	$6 \times \text{F32}$	Intensities of the six directly adjacent neighboring voxels

Table 1: Table of vectors used in the two-level TF.

Given the high computational cost of processing full datasets for large volumes, a uniform random sampling step is performed. A user-defined percentage of the generated vectors is selected to form a representative subset for the initial dimensionality reduction. The execution flow then diverges based on the selected algorithm:

- **UMAP:** The algorithm is first fitted to the sampled dataset to learn the manifold structure. Once the projection model is established, the transformation is applied slice-by-slice to the entire volume. This enables a global projection of the full dataset while keeping the fitting step computationally manageable.
- **t-SNE:** Both the fitting and transformation processes are restricted to the sampled dataset. Unlike UMAP, t-SNE does not produce a stable global transformation function that can be applied to out-of-sample data. Consequently, when using t-SNE, subsequent clustering and visualization are based exclusively on the sampled subset.

Both algorithms reduce the 11-dimensional vectors into 2-dimensional vectors, containing (x, y) coordinates. The resulting low-dimensional coordinates

are stored in an 8-bit typed array to minimize memory usage and enable efficient transfer. The system computes cluster centroids and radii from these coordinates, which define the parameters of the Gaussian bump primitives used for TF generation. The reduced data are then processed with HDBSCAN to assign cluster labels and identify noise points. The low-dimensional coordinates and cluster assignments are saved to a binary output file, while the Gaussian bump parameters are stored in a JSON file. Both files are returned to the client as inputs for the final visualization stage.

3.3 Visualization

The final stage of the pipeline is the visualization phase, where the multidimensional clustering results are remapped to the original spatial volume to support the construction of an interactive TF for volume rendering. This stage combines the original voxel intensity data with the low-dimensional clustering representation, enabling cluster-based exploration of the dataset within the VPT framework. At the beginning of this stage, the clustering results from the previous phase are loaded from the binary and JSON output files. The binary file contains the UMAP or t-SNE coordinates and their corresponding cluster assignments, while the JSON file provides the parameters describing the Gaussian bump primitives used for TF generation. These binary data are unpacked and stored in separate typed arrays; one array is sent to the TF canvas for interactive UI visualization, while the other is written into a 3D texture and uploaded to the GPU for direct access by the rendering shaders.

During rendering, the fragment shader simultaneously samples both the original volumetric intensity texture and the 3D texture containing the reduced dimensionality coordinates. The sampled UMAP/t-SNE values act as indices for the TF, which is visually displayed as a scatter plot within the TF canvas. This mechanism creates a bidirectional mapping between the abstract cluster space and the physical voxel space, allowing identified phenotypic cluster information to directly influence the visual properties of the rendered volume.

The output of this TF lookup is mathematically combined with the sampled intensity from the original volume, and the resulting values are sent to the fragment shader for final color and opacity computation. This process remaps clustered voxels to user-defined visual properties, enabling cluster-aware volume rendering that highlights specific biological structures.

The Gaussian bump primitives loaded from the JSON file are integrated into the TF editor (see

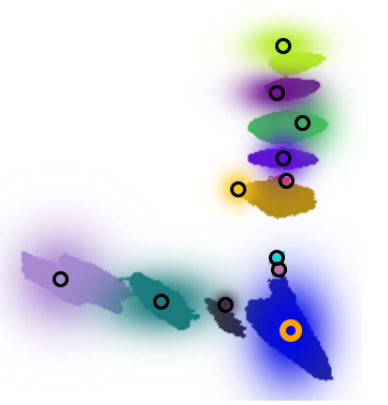


Figure 2: An example of generated TF for data clustered using UMAP and HDBSCAN.

Figure 2). Each bump represents a cluster centroid and radius computed during the clustering stage, defining a discrete colored region within the TF space. These primitives primarily determine how specific clusters are visualized in the final rendered volume. To ensure flexibility during exploratory analysis, the TF editor allows users to manually adjust, move, and resize these Gaussian bumps in real time. This interactive manipulation permits refinement of the visualization—such as isolating specific cellular subpopulations—without the need to repeat the computationally expensive clustering step.

Through this integrated process, the clustering results are remapped to image space, incorporated into the TF, and used to compute the final volumetric rendering, thereby completing the comprehensive visualization pipeline.

3.4 Implementation Details

The system architecture uses a robust client-server model designed for high-performance data processing and real-time visualization. The client-side application is developed with standard HTML5, CSS, and vanilla JavaScript, intentionally avoiding additional frameworks to maintain a lightweight footprint and ensure direct control over the rendering pipeline. The server is implemented in JavaScript and runs on the Node.js runtime environment, enabling efficient asynchronous communication. Data exchange between the client and server occurs through structured HTTP GET and POST requests. The client UI (shown in Figure 3) provides an interface for uploading volumetric datasets and defining clustering parameters, which the server then delegates for intensive processing.

To manage the substantial computational load required for dimensionality reduction and clustering,

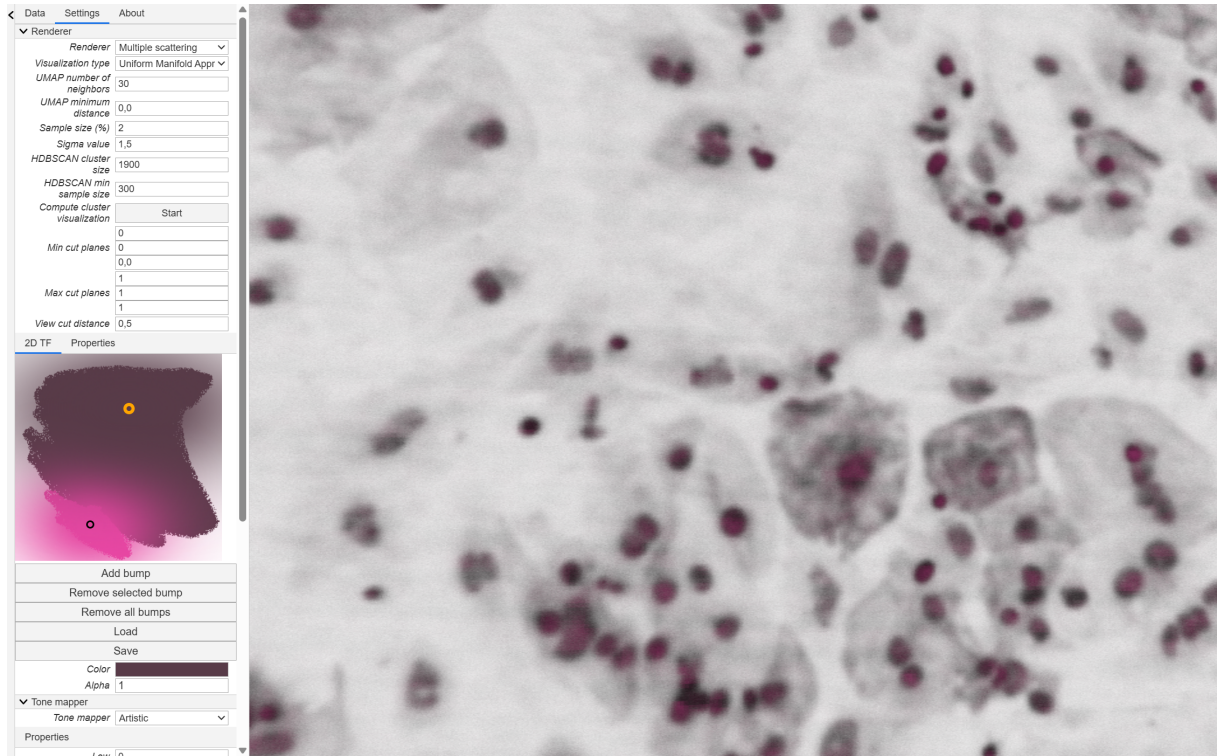


Figure 3: The client-side UI built upon the VPT framework. The interactive render view (right) provides real-time feedback of the volumetric output. The control panel (left) facilitates data management, selection of volume rendering algorithms, and fine-tuning of TF parameters and the post-processing pipeline.

the server uses the Node.js `spawn`² functionality to start a separate Python child process. This hybrid approach enables the system to take advantage of Python’s extensive scientific ecosystem while maintaining a responsive web interface. Within this child process, several specialized libraries are used to process the 11-dimensional voxel dataset. NumPy³ is used for dataset construction, selected primarily for its support of typed arrays and its ability to perform accelerated operations on large data structures. For the core analytical tasks, dimensionality reduction is performed by the `openTSNE`⁴ and `umap-learn`⁵ packages, while density-based grouping is handled using the `hdbscan`⁶ package.

The implementation of these algorithms differs significantly: t-SNE is fitted only on the uniform random samples and performs dimensionality reduction on that subset. In contrast, UMAP is first fitted on the uniform samples to learn the manifold, after which the transformation is applied to each vol-

ume slice individually to achieve superior results. Consequently, the UMAP pipeline is more resource-intensive, as HDBSCAN must process a much larger volume of data. Although HDBSCAN maintains a complexity of $O(N \log N)$, the increased value of N in the UMAP workflow leads to higher memory consumption. This multi-layered implementation ensures that high-volume microscopy data is processed with high precision and then efficiently returned to the VPT framework for interactive visualization.

4 RESULTS AND EVALUATION

4.1 Visual Results

To demonstrate the practical application of our framework, we present representative visualizations of cardiomyocyte samples processed through our integrated pipeline (see Figure 4). These images illustrate the system’s ability to map abstract high-dimensional clusters back to the original volumetric space, providing a clear visual distinction between different biological states. The images also demonstrate the use of different volume rendering techniques: (a) Emission-Absorption Model and (b) Volumetric Path Tracing, each highlighting different aspects of the data. The data show non-treated cardiomyocytes clustered using t-SNE. By utilizing cluster-aware VPT, we can effectively isolate and

² https://nodejs.org/api/child_process.html

³ <https://numpy.org/doc/2.4/reference/arrays.ndarray.html>

⁴ <https://github.com/pavlin-polcar/openTSNE>

⁵ <https://umap-learn.readthedocs.io/en/latest/>

⁶ https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html

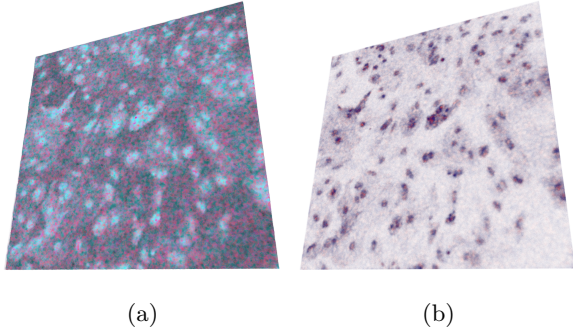


Figure 4: Showcase of visualizing the confocal microscopy data of non-treated cardiomyocytes in using two rendering techniques: (a) Emission-Absorption Model – pronouncing tropamin-rich regions and (b) Volumetric Path Tracing – pronouncing cell nuclei.

colorize specific structural features that are often conflated in standard intensity-based renderings.

4.2 Parameter Sensitivity and Practical Tuning

Both t-SNE and UMAP are sensitive to parameter selection, as their hyperparameters directly influence neighborhood preservation, cluster compactness, and the visual separation of embedded structures. Since exhaustive exploration of the parameter space is computationally expensive for large volumetric datasets, we focused on identifying stable parameter ranges that produced visually consistent clustering results across multiple microscopy samples.

For t-SNE, the perplexity parameter had the strongest influence on the resulting embeddings. Lower perplexity values produced fragmented local clusters with increased sensitivity to noise, while excessively large values reduced local structural separation and merged biologically distinct regions. In practice, values between 30 and 50 provided the most stable balance between local detail preservation and global cluster organization for the tested microscopy datasets (see Figure 5). The learning rate primarily affected convergence stability; values that were too low resulted in compressed embeddings, whereas excessively high values produced unstable cluster layouts. The selected configuration of perplexity 45 and learning rate 410 yielded visually stable embeddings while maintaining reasonable computation times.

For UMAP, the number of neighbors controlled the balance between local and global structure preservation. Smaller neighborhood sizes emphasized fine-grained local variations but introduced fragmented clusters, while larger values produced

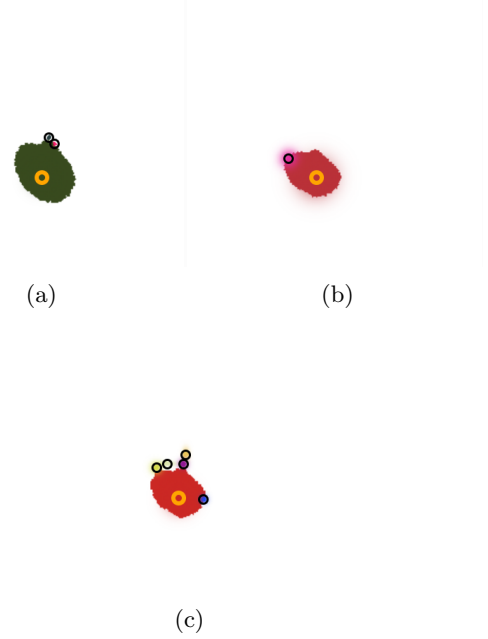


Figure 5: TFs with different t-SNE perplexity values run on $360 \times 360 \times 40$ sized volume with 5% of the samples used for clustering computation. The remaining parameters were equal for all run variants, with learning rate set to a value of 410, iterations set to 290 and exaggeration set to 1.0. The HDBSCAN cluster size parameter was set to a value of 1000 and minimum sample size was set to 5. (a) t-SNE TF with perplexity value 10 (b) t-SNE TF with perplexity value 30 (c) t-SNE TF with perplexity value 50

smoother embeddings with reduced structural detail. The minimum distance parameter strongly affected cluster compactness in the latent space. Setting the minimum distance close to zero produced tighter and more visually separable clusters, which improved subsequent HDBSCAN processing and transfer function generation. Empirically, we found that neighborhood sizes between 20 and 40 provided stable visual separation across datasets (see Figure 6).

Although parameter tuning remains necessary, our experiments indicate that the framework is relatively robust within moderate parameter ranges. Small parameter adjustments typically altered cluster compactness and separation rather than completely changing the detected structural patterns. This reduces the need for exhaustive blind parameter searches and enables practical iterative refinement guided by visual inspection of the resulting embeddings and volumetric renderings.

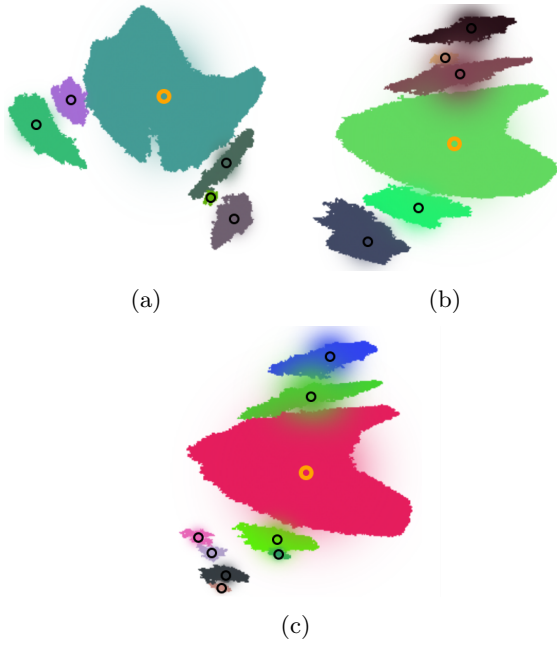


Figure 6: TFs with different UMAP number of neighbors values run on $270 \times 270 \times 40$ sized volume with 5% of the samples used for clustering computation. The minimum distance parameter was set to a value of 0, with HDBSCAN cluster size parameter set to a value of 1000 and minimum sample size set to a value of 5. (a) UMAP TF with number of neighbors value 10. (b) UMAP TF with number of neighbors value 30. (c) UMAP TF with number of neighbors value 50.

4.3 Performance Evaluation

To evaluate the efficiency of the proposed clustering pipeline, we conducted a series of experiments on multiple volumetric datasets with increasing resolution. The evaluation focuses on runtime performance and memory consumption for the two supported dimensionality reduction approaches: t-SNE combined with HDBSCAN and UMAP combined with HDBSCAN.

All experiments were performed on a high-end consumer system with an Intel Core i9-13900HX CPU, 32 GB of DDR5 system memory and an Nvidia RTX 4070 Laptop GPU with 8 GB of video memory. All computations for preprocessing and clustering steps were performed strictly on the CPU. This choice was motivated by the memory constraints typical of consumer-grade GPUs, ensuring a fair comparison of algorithmic efficiency without limitations imposed by specialized hardware memory boundaries. The GPU was utilized only for the visualization step, mapping the calculated coordinates and rendering the volume on screen.

For experimental consistency, fixed parameter configurations were applied across all datasets, includ-

ing a uniform sampling rate of 2% of the total voxel population. The t-SNE configuration used a perplexity of 45, a learning rate of 410, and 290 iterations. The UMAP setup used 30 neighbors and a minimum distance of 0. HDBSCAN parameters were adjusted slightly between methods to account for inherent differences in data distribution, with larger cluster and minimum sample sizes assigned for UMAP.

Performance metrics were recorded for three volume resolutions: $180 \times 180 \times 40$, $270 \times 270 \times 40$, and $360 \times 360 \times 40$. For the smallest resolution, t-SNE significantly outperformed UMAP in both runtime and memory efficiency. As detailed in Table 2, the t-SNE pipeline completed in 78.23 seconds—less than half the time required by UMAP. Memory usage showed an even larger disparity; while t-SNE remained light on resources, the UMAP pipeline reached a peak consumption of approximately 13 GB (Figure 7).

As volume size increased, this performance gap remained consistent. At the $270 \times 270 \times 40$ resolution, the runtime for t-SNE rose to 530.67 seconds, compared to 728.39 seconds for UMAP. Peak memory usage for UMAP scaled sharply, exceeding 27 GB, whereas t-SNE required approximately 10 GB. This indicates that the UMAP + HDBSCAN pipeline exhibits a significantly steeper increase in memory requirements relative to dataset size.

Critically, the UMAP-based pipeline failed to process the largest dataset ($360 \times 360 \times 40$) due to an out-of-memory error. In contrast, the t-SNE pipeline demonstrated greater robustness under these constrained conditions, successfully completing the task in 676.74 seconds with a peak memory usage of approximately 18 GB. These results suggest that while both methods are computationally intensive, the t-SNE configuration provides superior stability for high-resolution volumetric analysis on standard consumer hardware.

4.4 Expert Evaluation

Domain experts evaluated the proposed framework using volumetric IF microscopy data of cardiomyocytes under baseline and treatment conditions. The goal of the evaluation was not biological diagnosis, but assessment of whether the visualization pipeline could reveal structural differences that are difficult to observe using conventional intensity-based rendering.

The proposed clustering-based visualization consistently highlighted differences in spatial organization between the two sample groups. Non-treated samples exhibited more heterogeneous and spatially

Volume Resolution	t-SNE + HDBSCAN (s)	UMAP + HDBSCAN (s)
$180 \times 180 \times 40$	78.23	164.76
$270 \times 270 \times 40$	530.67	728.39
$360 \times 360 \times 40$	676.74	OOM*

Table 2: Runtime comparison of clustering methods. * denotes an Out-of-Memory event.

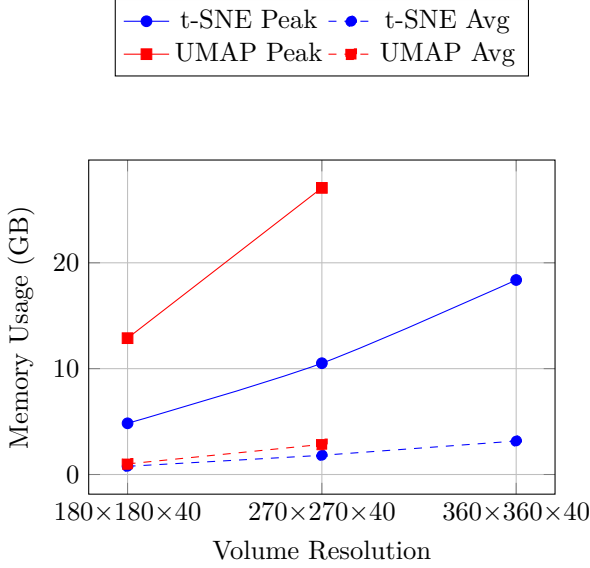


Figure 7: Comparison of peak and average memory usage for t-SNE and UMAP across different volume sizes. UMAP fails for the largest dataset due to memory constraints.

dispersed cluster distributions, whereas treated samples showed more concentrated and structurally localized regions (Figures 8 to 10). These patterns were difficult to distinguish using standard IF inspection alone, but became visually apparent after dimensionality reduction and cluster-aware rendering.

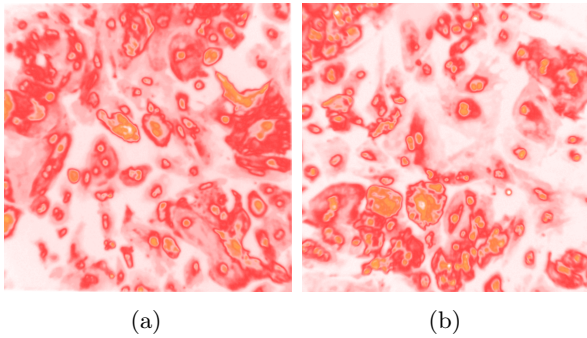


Figure 8: Density-based analysis of the troponin channel revealing treatment-induced structural remodeling. (a) Non-treated cardiomyocytes showing dispersed troponin distribution. (b) Cardiomyocytes treated with 500 nM carfilzomib for 24h.

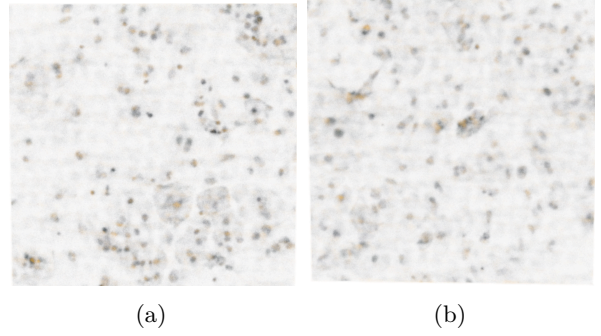


Figure 9: UMAP visualization of clustered troponin and DAPI channels. (a) Non-treated cardiomyocytes exhibit traces of dispersed troponin distribution with multiple small regions of varying intensity. (b) Cells treated with 500 nM carfilzomib for 24h show locally concentrated and clustered high-density regions, suggesting a redistribution of contractile proteins and structural remodeling.

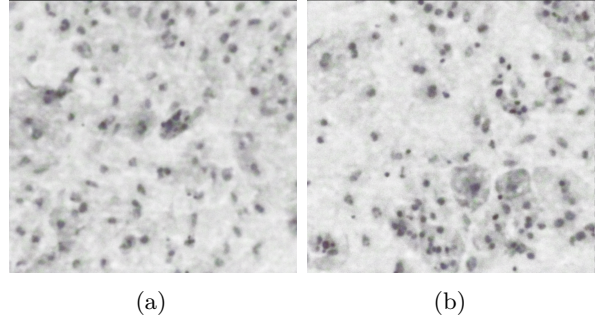


Figure 10: t-SNE visualization of DAPI and troponin channels. (a) Non-treated cardiomyocytes exhibit a heterogeneous and dispersed troponin distribution with multiple small regions of varying intensity. (b) Cells treated with 500 nM carfilzomib for 24h show locally concentrated and clustered high-density regions, suggesting treatment-induced structural remodeling.

Both t-SNE and UMAP successfully separated distinct structural regions; however, the methods produced different visual characteristics. t-SNE generated more fragmented local groupings, while UMAP produced more compact and visually separable clusters, which experts found easier to interpret during exploratory analysis. In particular, UMAP improved the visibility of localized high-density regions and enhanced differentiation between dominant phenotypic structures.

The expert feedback further indicated that the framework supports identification of subtle morphological variations, including differences in protein organization and nuclear structure. Although these findings require additional biological validation, the results demonstrate that the proposed visualization pipeline can expose structural patterns that are not readily observable through traditional volume rendering approaches.

5 DISCUSSION

The evaluation of the proposed framework demonstrates a high level of efficacy in handling complex microscopy data, while also highlighting specific trade-offs between dimensionality reduction algorithms and hardware constraints.

5.1 Comparison of Dimensionality Reduction Methods

A primary focus of our system evaluation was the comparative performance and visual utility of UMAP and t-SNE. From a visualization perspective, UMAP provided superior intensity differentiation in the identified clusters, making subtle structural features more apparent in the VPT environment. This advantage is likely due to UMAP’s ability to better preserve the global manifold structure of the 11-dimensional feature space, compared to the more locally focused t-SNE.

However, this visual advantage is offset by significant differences in computational robustness. Our performance benchmarks show that UMAP exhibits a much steeper increase in memory requirements relative to dataset size, resulting in out-of-memory failures on high-resolution volumes (e.g., $360 \times 360 \times 40$) when processed on high-end consumer hardware. In contrast, t-SNE demonstrated greater stability under constrained memory conditions, successfully completing all tested resolutions. Furthermore, UMAP’s ability to learn a stable global transformation allows it to project full volumes slice-by-slice, whereas t-SNE remains limited to the sampled subset, a critical distinction for researchers requiring full-volume segmentation.

5.2 Interactive Exploration and Transfer Function Design

The integration of cluster-based Gaussian primitives into the TF editor greatly enhances the intuitiveness of volume exploration. Traditional direct volume rendering (DVR) workflows require manual, often trial-and-error manipulation of 1D or 2D histograms. In contrast, our system enables users to interact directly with cluster centroids and radii in the low-dimensional scatter plot, effectively “scents” the

navigation widgets with structural context. This bidirectional mapping ensures that visualization refinement – such as isolating specific cellular sub-populations – can be performed in real time without re-executing the expensive clustering pipeline.

5.3 Expert-Led Structural and Phenotypic Discovery

The expert evaluation provided critical qualitative evidence that the framework’s data-driven approach aligns with actual biological phenomena. A major takeaway from the domain experts was the system’s enhanced sensitivity in detecting subtle structural changes. For example, while traditional visual inspections might reveal fewer differences between samples, our density-based and clustering models successfully identified treatment-associated redistributions of contractile proteins.

The experts noted that the framework’s ability to highlight high-density troponin clusters in treated cells—interpreted as protein accumulation or structural collapse—provided a more nuanced understanding of cellular stress than simple intensity analysis. Furthermore, the reduction in phenotypic heterogeneity observed in carfilzomib-treated samples suggests that the drug drives cardiomyocytes toward fewer morphological states. This quantifiable shift in cluster distribution, along with the identification of specific nuclear phenotypes characterized by slightly larger and more prominent nuclei in treated cells, demonstrates that the system can capture complex morphometric trends that remain hidden in conventional 2D evaluations.

5.4 Limitations

Despite the advantages of the proposed framework, several technical and operational limitations were identified during evaluation. A primary constraint is the reliance on CPU-based computation, a design choice necessitated by the limited memory capacity typical of consumer-grade GPUs. This leads to significantly longer processing times for high-resolution datasets, with runtimes exceeding 10 minutes for the largest tested volumes. The performance and accuracy of the analysis depend heavily on fluorescence intensity, which is inherently sensitive to both sample preparation and the image acquisition process. Variability in staining quality or image capture settings on microscope can cause technical fluctuations in the raw signal, potentially affecting the resulting density calculations and clustering outcomes.

Furthermore, while the workflow is designed to be unbiased and quantifiable, it is highly sensitive to initial parameter configurations. The final

clustering and visualization results depend heavily on user-defined settings, such as sampling rates and HDBSCAN thresholds, which can vary across datasets. Currently, the methodology is not entirely intuitive for domain experts without computational backgrounds, and there is a risk of over-interpretation if results are not rigorously validated against raw image data. Future iterations could address these issues by implementing automated parameter optimization or GPU-accelerated clustering to enhance the overall interactivity and robustness of the pipeline.

Experts concluded that this workflow enables detection of subtle structural patterns beyond standard IF evaluation and provides quantitative insight into cellular heterogeneity. However, the results depend strongly on fluorescence intensity, so observed differences may reflect variations in staining or imaging rather than true structural changes. Therefore, validation against raw image data and complementary biological approaches remains essential.

6 CONCLUSION

This paper presents an interactive framework for exploring complex volumetric microscopy data using 11-dimensional feature extraction and density-based clustering. By integrating UMAP/t-SNE for dimensionality reduction with HDBSCAN, the system effectively maps phenotypic states back to a volumetric path-tracing environment.

Expert evaluation on troponin-labeled cardiomyocytes demonstrated that this workflow offers greater sensitivity in detecting treatment-induced remodeling compared to standard visual inspection. Results showed that 500 nM carfilzomib treatment induces quantifiable protein clustering and reduces phenotypic heterogeneity. Performance benchmarks highlighted critical trade-offs: while UMAP provided superior visual differentiation, t-SNE offered greater computational robustness and successfully processed higher-resolution volumes where UMAP encountered memory failures.

Future work will focus on GPU-accelerated clustering and automated parameter optimization to further enhance interactivity and accessibility for domain experts. This unbiased, quantifiable approach offers a powerful tool for spatially resolved mapping of structural reorganization in biological samples.

ACKNOWLEDGEMENTS

Images were obtained by asist. dr. Živa Zajec (research group of assoc. prof. dr. Martina Gobec), University of Ljubljana, Faculty of Pharmacy.

This research was conducted as part of the basic research project Cell visualization of unified microscopic data and procedurally generated sub-cellular structures [project number J2-50221], funded by the Slovenian Research and Innovation Agency (Javna agencija za znanstvenoraziskovalno in inovacijsko dejavnost RS) from the state budget.

REFERENCES

- [Boh+20] C. Bohak et al. “Web-Based 3D Visualisation of Biological and Medical Data”. In: *Biomedical Visualisation: Volume 6*. Springer, 2020, pp. 1–18. DOI: 10.1007/978-3-030-37639-0_1.
- [CB22] J. Czajkowska and M. Borak. “Computer-Aided Diagnosis Methods for High-Frequency Ultrasound Data Analysis: A Review”. In: *Sensors* 22.21 (2022), p. 8326. DOI: 10.3390/s22218326.
- [CMS13] R. J. Campello et al. “Density-Based Clustering Based on Hierarchical Density Estimates”. In: *Advances in Knowledge Discovery and Data Mining*. Springer, 2013, pp. 160–172.
- [LBM18] Ž. Lesar et al. “Real-time interactive platform-agnostic volumetric path tracing in WebGL 2.0”. In: *Web3D 2018 : proceedings*. 2018, pp. 1–7.
- [LBM23] Ž. Lesar et al. “Blocky Volume Package: a Web-friendly Volume Storage and Compression Solution”. In: *WSCG 2023 : 31. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision : Plzen, Czech Republic, May 15 – 19, 2023 : proceedings*. University of West Bohemia, 2023.
- [Les+24] Ž. Lesar et al. “Volume conductor: interactive visibility management for crowded volumes”. In: *The Visual Computer* 40.2 (2024), pp. 1005–1020. DOI: 10.1007/s00371-023-02828-8.
- [MH08] L. van der Maaten and G. Hinton. “Visualizing Data using t-SNE”. In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605.
- [MHM18] L. McInnes et al. “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction”. In: *arXiv preprint arXiv:1802.03426* (2018).
- [SVG06] P. Sereda et al. “Automating transfer function design for volume rendering using hierarchical clustering of material

- boundaries”. In: *Euro Vis.* 2006, pp. 243–250.
- [WHA07] W. Willet et al. “Scented Widgets: Improving Navigation Cues with Embedded Visualizations”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1129–1136. DOI: 10.1109/TVCG.2007.70589.
- [YKJ25] S. Yoo et al. “Two-Level Transfer Functions Using t-SNE for Data Segmentation in Direct Volume Rendering”. In: *IEEE Transactions on Visualization and Computer Graphics* 31.9 (2025), pp. 5948–5961. DOI: 10.1109/TVCG.2024.3484471.
- [ZH13] L. Zhou and C. Hansen. “Transfer function design based on user selected samples for intuitive multivariate volume exploration”. In: *2013 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 2013, pp. 73–80.
- [ZW17] L. Zhou and D. Weiskopf. “Indexed-points parallel coordinates visualization of multivariate correlations”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.6 (2017), pp. 1997–2010.