Eurographics Conference on Visualization (EuroVis) 2023
D. Archambault, R. Bujack, and T. Schreck
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 3

xOpat: eXplainable Open Pathology Analysis Tool Supplementary Material

J. Horák[1], K. Furmanová[1], B. Kozlíková[1], T. Brázdil[1], P. Holub[2], M. Kačenga[1], M. Gallo[1], R. Nenutil[3], J. Byška[1,4], and V. Rusňák[2]

[1]Masaryk University, Faculty of Informatics, Brno, Czech Republic
[2]Masaryk University, Institute of Computer Science, Brno, Czech Republic
[3]Department of Pathology, Masaryk Memorial Cancer Institute, Brno, Czech Republic
[4]University of Bergen, Department of Informatics, Bergen, Norway

Introduction

In this supplementary material, we present additional details about related work and performance evaluation of our tool.

1. Related Work

In Table 1, we present an overview of existing WSI viewers with respect to dependence on external services and supported features.

2. Performance Evaluation

During the evaluation, we measured the number of requests generated by the viewer and their duration. We compared asynchronous and synchronous access to the data. Asynchronous access was forced to wait for all tiles from separate files. While this synchronization step lowers the measured performance, it more precisely describes the real behavior of the viewer (since to render the final image, it is required that all the data are available). We also tested two versions of the synchronous transfer. First, an option where the data were simply concatenated into a single image. Second, the alternative where the data were sent as a single zip archive. Note that the second method is harder to integrate into the viewer (as the data must be extracted from the archive on the viewer side). Still, we hypothesized that the performance gain would be significant.

Furthermore, we compared the behavior of the image server with enabled and disabled server-side caching. The cache used was a small, simple object cache in RAM, although more advanced options were available (e.g., Memcached server for IIPImage). To simulate several real-world scenarios, we measured the performance when connecting to the server during a busy workday from:

- Eduroam university WiFi network where hundreds of students and machines are connected, but at the same time, the client was close to cloud services hosting the server and data, simulating typical set-up in a hospital.
- A private remote WiFi network simulating the scenario of pathologists working from home and connecting remotely to a server at a hospital (in our case, at the above-mentioned university).

The evaluation was performed on a notebook with a resolution of 1920×1080 pixels and integrated GPU (i.e., a less favorable but most-likely scenario). We attempted to render a single tissue file together with 3, 9, and 18 distinct pyramidal grayscale image data sources at once. While the tissue file used JPEG compression, the data sources used PNG lossless compression. Finally, the data were rendered using *edges*, *heatmap*, and *bi-polar heatmap*.

Using our *Storytelling* plugin, we recorded a query-intensive series of zoom-ins, zoom-outs, and movements at different resolutions (see Figure 1). The simulation lasted 13.5 seconds and was repeatedly re-played to measure the number of generated requests and their duration. To test the caching performance the same scenario without profiling was run first to initialize the server cache. The *Profiling* plugin, which we used to measure the performance, waited for all requests to finish, offered data export, and rendered a box plot of the result directly in the viewer using *Vega* graph visualization library module.

Figure 1: *Tracking of the viewport position during profiling. The position is encoded in opacity: the more the viewport zooms in, the more opaque the color; rendered with WebGL module*—heatmap.

Our evaluation confirms that asynchronous access to data is a communication-intensive process. As shown in Figure 2, a single user can generate more than 1000 requests even for a small dataset. This could easily cause problems when multiple clients use the same server.

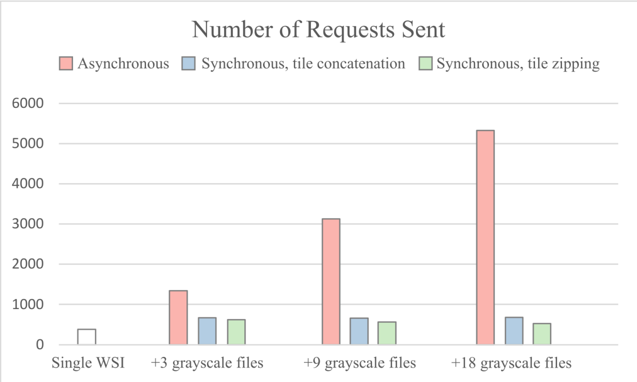

Figure 2: *The graph shows the average number of requests sent from the viewer to the server in different data scenarios within 13.5 seconds-long simulation loop. Note that the number of asynchronous requests was computed from the average amount of tiles requested during the simulation loop.*

Regarding the time required to fulfill the request, the difference in server-side caching (Figure 3, Figure 4) was negligible compared to the difference in the immediate network load, which had a much more significant impact on the measured values.

Interestingly enough, the university network was able to handle many asynchronous requests well, probably as it is designed to scale well with the number of users (Figure 3, Figure 4). On the other hand, when connecting to the server remotely from a private network, the number of requests started to be a problem. In this scenario, even a naive server-side concatenation performed better (Figure 5, Figure 6).

Nevertheless, as expected, in both scenarios using the synchronous request in connection with compressing the data into a zip archive was either on par or was significantly overperforming the remaining two options, especially when it comes to large datasets.

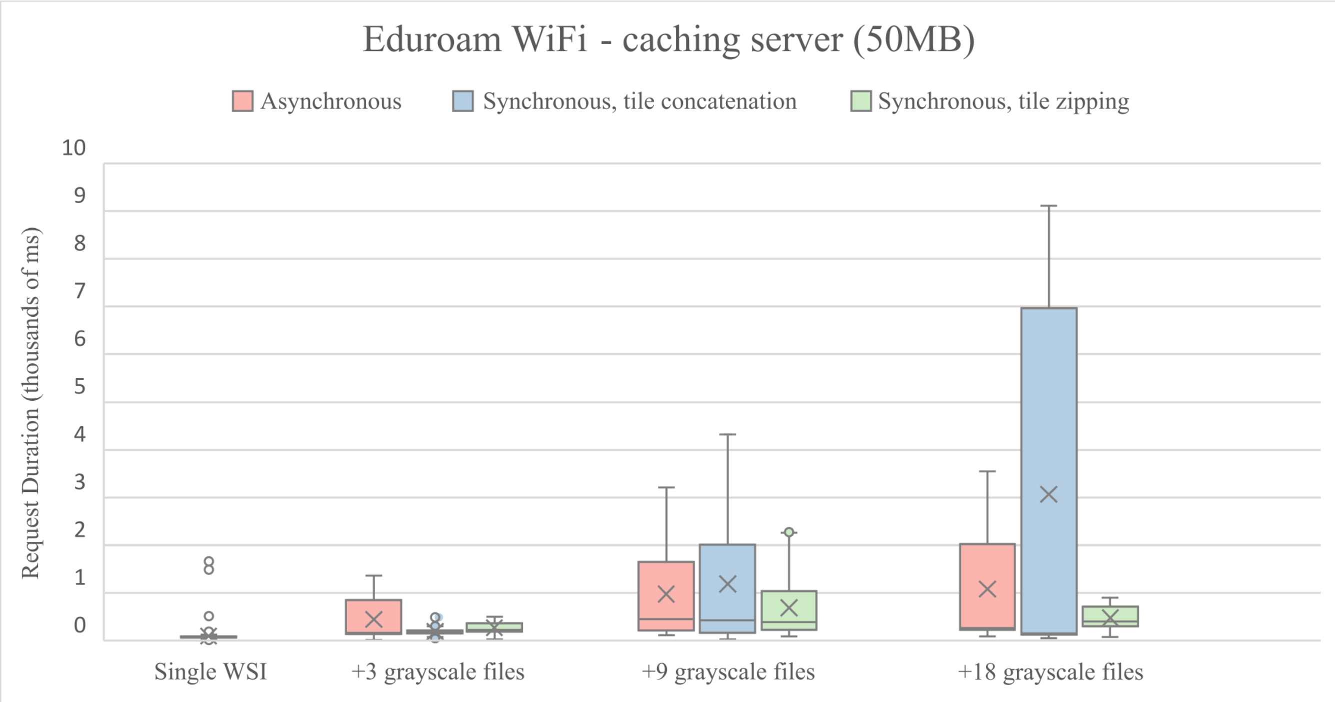

Figure 3: *Distribution of duration of individual requests (in seconds) for the case where up to 19 files (one tissue image with additional grayscale data layers) were rendered at once on the university WiFi network with server-side caching turned on (limit of 50MB).*

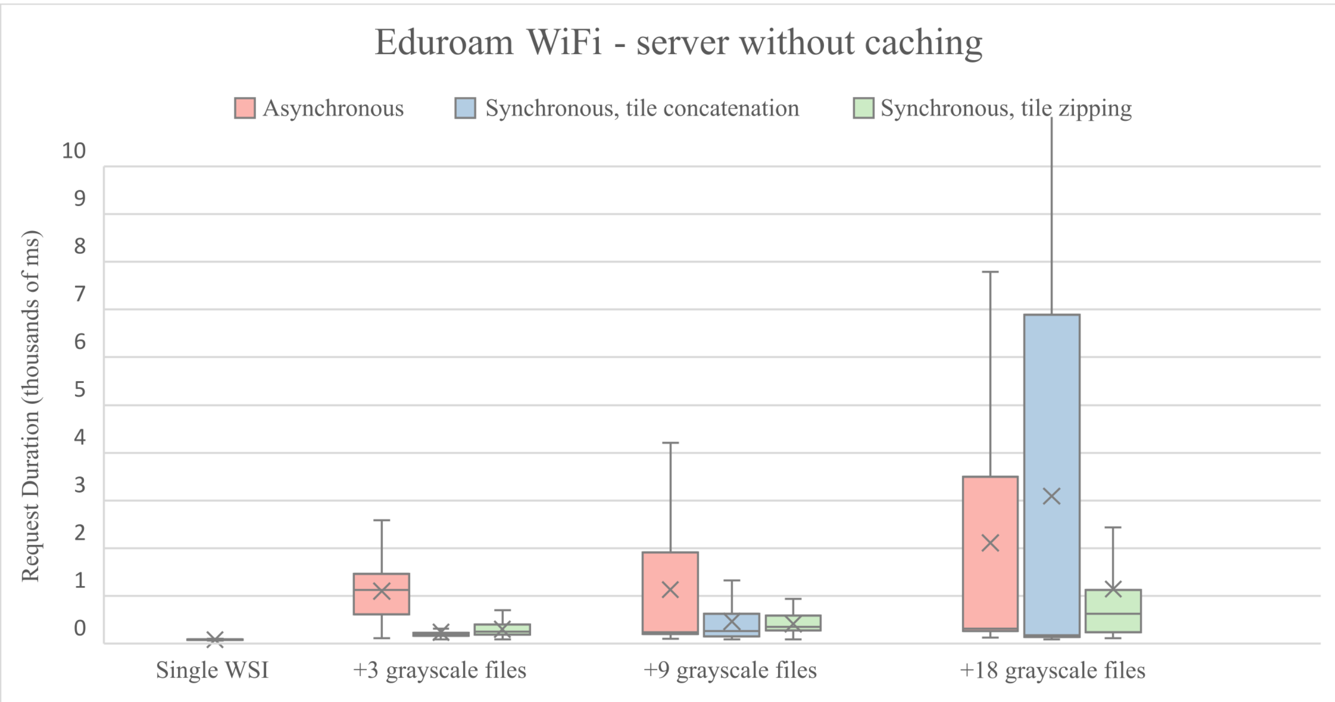

Figure 4: *Distribution of duration of individual requests (in seconds) for the case where up to 19 files (one tissue image with additional grayscale data layers) were rendered at once on the university WiFi network with server-side caching turned off.*

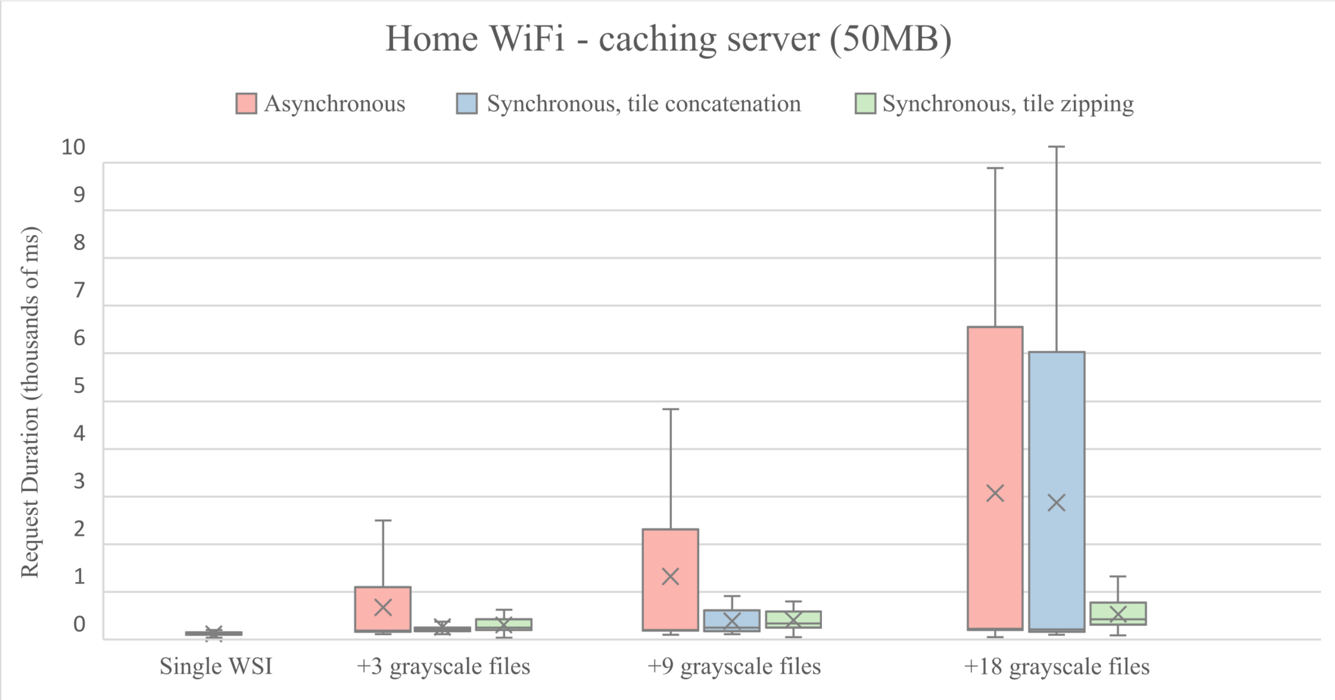

Figure 5: *Distribution of duration of individual requests (in seconds) for the case where up to 19 files (one tissue image with additional grayscale data layers) were rendered at once on the private WiFi network with server-side caching turned on (limit of 50MB).*

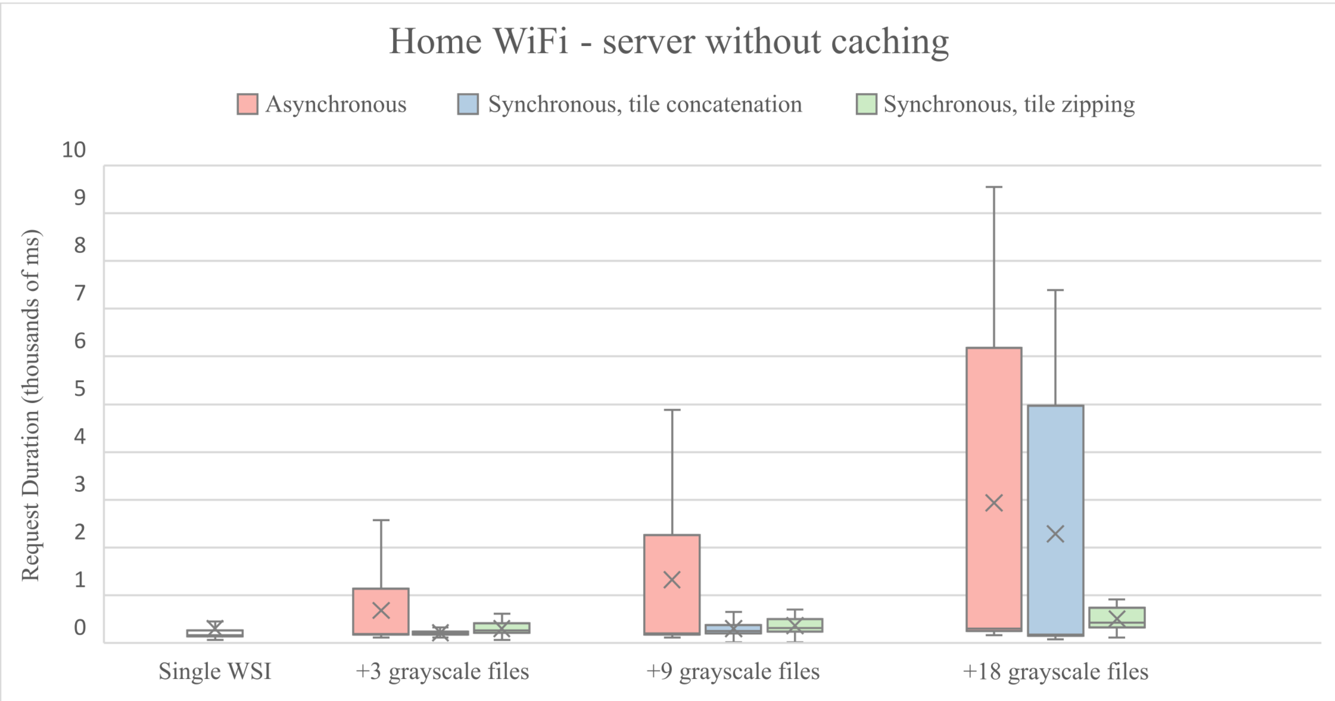

Figure 6: *Distribution of duration of individual requests (in seconds) for the case where up to 19 files (one tissue image with additional grayscale data layers) were rendered at once on the private WiFi network with server-side caching turned off.*

Name	Image Server Independence	Flexible Data Handling		Collaboration Support			Analysis Support	Customizable UI	Annotations	License	Platform
		User-Defined Mapping of Data to Visualization Layers	Arbitrary Layering from Multiple Sources	Storytelling Capabilities	Session Export	Live Collaboration					
ASAP [Com]	N/A	Limited	No	No	No	No	both	No	Yes	OSS	Desktop
CytoBrowser [RL21]	Custom server	Yes, but requires coding	Yes	No	Yes	Yes	No	No	Yes	OSS	WebApp
Cytomine [MRS*16]	Custom server	No	No	No	Yes	Yes	custom	No	Yes	OSS	WebApp
Minerva [HRM*20]	Custom server	No	No	Yes	Yes	No	No	Yes	Yes	OSS	WebApp
Image-Pro [Med]	Custom server	Limited	Yes, only as multichannel	Yes	No	No	built-in	Yes	Yes	Commercial	Desktop
ImageScope [Ape]	Closed ecosystem	No	No	No	No	No	both	No	Yes	Freeware	Desktop
Jeong et al. [JSH*13]	Custom server	No	No	No	No	Yes	No	No	Yes	N/A	Desktop/Mobile
HALO [Ind]	Closed ecosystem	No	No	No	No	No	built-in	No	Yes	Commercial	WebApp
napari [SLN*22]	Custom server	Yes, but requires coding	Yes, but requires coding	No	No	No	both	Yes	Yes	OSS	Desktop
OpenHi2 [PZD*19]	Custom server	Yes, but requires coding	No	No	No	Yes	custom	No	Yes	OSS	WebApp
Orbit [SSV20]	OMERO	No	No	No	Yes	No	both	No	Yes	OSS	Desktop
PathViewer [Gle]	OMERO	Limited	No	No	Yes	Yes	No	Yes	Yes	Commercial	WebApp
Pathomation [Pat]	Custom server	No	No	No	No	No	both	No	Yes	Freeware	WebApp
QuPath [BLF*17]	OMERO	Limited	Yes, only as multichannel	No	No	No	both	No	Yes	OSS	Desktop
Scope2Screen [JKW*22]	Custom server	Limited	Yes, only as multichannel	Yes	Yes	No	built-in (no AI)	No	Yes	OSS	WebApp
Sectra [Sec]	Closed ecosystem	No	No	No	No	No	both	Yes	Yes	Commercial	WebApp
TisUUmaps [PAA*23]	Custom server	Yes, but requires coding	Yes	No	Yes	No	custom	No	Yes	OSS	WebApp
Visiopharm Viewer [Vis]	Closed ecosystem	No	No	No	No	No	both	Yes	Yes	Commercial	Desktop
xOpat	Any server	Yes via UI	Yes	Yes	Yes	No	both in progress	Yes	Yes	OSS	WebApp

Table 1: *Comparison of image viewers' characteristics. Custom server = an implementation provided by the authors, typically with limited compatibility. Flexible Data Handling: Limited = typically fixed set of available configuration options for given data. Analysis support: built-in = integrated algorithms; custom algorithms = connectors to external services or ability to integrate own algorithms; both = built-in and custom. License: OSS = Open-source software.*

References

[Ape] APERIO. *Aperio ImageScope – Pathology Slide Viewing Software*. URL: https://www.leicabiosystems.com/digital-pathology/manage/aperio-imagescope/ (visited on 03/24/2022) 5.

[BLF*17] BANKHEAD, PETER, LOUGHREY, MAURICE B., FERNÁNDEZ, JOSÉ A., et al. "QuPath: Open Source Software for Digital Pathology Image Analysis". *Scientific Reports* 7.1 (2017), 16878. ISSN: 2045-2322. DOI: 10.1038/s41598-017-17204-5 5.

[Com] COMPUTATIONAL PATHOLOGY GROUP. *Automated Slide Analysis Platform*. URL: https://computationalpathologygroup.github.io/ASAP/ (visited on 10/12/2022) 5.

[Gle] GLENCOE SOFTWARE. *PathViewer: An interactive visualization, analysis, and annotation tool specifically tailored for the digital pathology workflow*. URL: https://www.glencoesoftware.com/products/pathviewer/features/ (visited on 10/26/2022) 5.

[HRM*20] HOFFER, JOHN, RASHID, RUMANA, MUHLICH, JEREMY L., et al. "Minerva: a light-weight, narrative image browser for multiplexed tissue images". *Journal of Open Source Software* 5.54 (2020), 2579. DOI: 10.21105/joss.02579. URL: https://doi.org/10.21105/joss.02579 5.

[Ind] INDICA LABS INC. *HALO – Image Analysis Platform for Quantitative Tissue Analysis in Digital Pathology*. URL: https://indicalab.com/halo/ (visited on 10/11/2022) 5.

[JKW*22] JESSUP, JARED, KRUEGER, ROBERT, WARCHOL, SIMON, et al. "Scope2Screen: Focus+Context Techniques for Pathology Tumor Assessment in Multivariate Image Data". *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2022), 259–269. DOI: 10.1109/TVCG.2021.3114786 5.

[JSH*13] JEONG, W., SCHNEIDER, J., HANSEN, A., et al. "A Collaborative Digital Pathology System for Multi-Touch Mobile and Desktop Computing Platforms". *Computer Graphics Forum* 32.6 (2013), 227–242. DOI: 10.1111/cgf.12137. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12137 5.

[Med] MEDIA CYBERNETICS. *Image-Pro*. URL: https://www.mediacy.com/imagepro (visited on 03/24/2022) 5.

[MRS*16] MARÉE, RAPHAËL, ROLLUS, LOÏC, STEVENS, BENJAMIN, et al. "Cytomine: An Open-Source Software For Collaborative Analysis Of Whole-Slide Images". *Diagnostic Pathology* 1.8 (2016). DOI: 10.17629/www.diagnosticpathology.eu-2016-8:151 5.

[PAA*23] PIELAWSKI, NICOLAS, ANDERSSON, AXEL, AVENEL, CHRISTOPHE, et al. "TissUUmaps 3: Improvements in Interactive Visualization, Exploration, and Quality Assessment of Large-Scale Spatial Omics Data". *bioRxiv : the preprint server for biology* (2023). DOI: 10.1101/2022.01.28.478131. URL: https://www.biorxiv.org/content/early/2023/01/16/2022.01.28.478131 5.

[Pat] PATHOMATION. *PMA.Start – Universal Whole Slide Image Viewer for Digital Pathology*. URL: https://free.pathomation.com (visited on 10/12/2022) 5.

[PZD*19] PUTTAPIRAT, PARGORN, ZHANG, HAICHUAN, DENG, JINGYI, et al. "OpenHI2 — Open source histopathological image platform". *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2019, 2696–2701. DOI: 10.1109/BIBM47256.2019.8983322 5.

[RL21] RYDELL, CHRISTOPHER and LINDBLAD, JOAKIM. "CytoBrowser: A Browser-Based Collaborative Annotation Platform for Whole Slide Images". *F1000Research* 10 (Mar. 22, 2021), 226. ISSN: 2046-1402. DOI: 10.12688/f1000research.51916.1. URL: https://f1000research.com/articles/10-226/v1 (visited on 03/14/2023) 5.

[Sec] SECTRA AB. *Sectra Digital Pathology Solution*. URL: https://medical.sectra.com/product/sectra-digital-pathology-solution/ (visited on 10/12/2022) 5.

[SLN*22] SOFRONIEW, NICHOLAS, LAMBERT, TALLEY, NUNEZ-IGLESIAS, JUAN, et al. *Napari/Napari: 0.4.15*. Version v0.4.15. Zenodo, Mar. 10, 2022. DOI: 10.5281/ZENODO.3555620. URL: httpcope://zenodo.org/record/3555620 (visited on 10/03/2022) 5.

[SSV20] STRITT, MANUEL, STALDER, ANNA K., and VEZZALI, ENRICO. "Orbit Image Analysis: An open-source whole slide image analysis tool". *PLOS Computational Biology* 16 (2020), 1–19. DOI: 10.1371/journal.pcbi.1007313 5.

[Vis] VISIOPHARM. *Visiopharm Viewer – AI-driven Pathology Software*. URL: https://visiopharm.com/visiopharm-digital-image-analysis-softwarefeatures/viewer (visited on 10/12/2022) 5.