Check for updates

SOFTWARE TOOL ARTICLE

REVISED LMDmapper: an open-source desktop tool for spatial mapping of laser microdissection samples

[version 2; peer review: 2 approved]

Antton Alberdi[1], Jaime Ramirez[1], Nanna Gaun[1], Zoé Horisberger[2], Bryan Wang[2], Urvish Trivedi[2], Amalia Bogri[1]

[1]Globe Institute, University of Copenhagen, Copenhagen, Capital Region of Denmark, Denmark
[2]Department of Biology, University of Copenhagen, Copenhagen, Capital Region of Denmark, Denmark

v2 **First published:** 24 Jul 2026, **6**:257
https://doi.org/10.12688/openreseurope.24543.1
Latest published: 08 Sep 2026, **6**:257
https://doi.org/10.12688/openreseurope.24543.2

Abstract

Laser microdissection (LMD) enables researchers to isolate targeted microsamples from microscopy slide specimens for downstream molecular analyses. While traditionally employed for isolating eukaryotic cells from complex tissues, LMD is starting to be used for micro-scale spatial microbiome analyses, which require the precise location of the microsamples to be tracked for downstream spatial analyses. To address this need, we present LMDmapper, an open-source desktop application that allows designing, tracking and logging micron-scale spatial microsample data and metadata from LMD sessions. The software parses Leica Database LIF image files (containing stage coordinates), imports laser microdissection CSV exports (containing image pixel coordinates), transforms and maps image pixel coordinates into stage coordinates, and presents the resulting cut points together with user-defined plate layouts and collection metadata. LMDmapper supports a variety of microdissection designs, including multiple slides, specimens, collection plates and plate layouts. The application is implemented in TypeScript using Electron, React, Vite, and fast-xml-parser. LMDmapper outputs include a metadata CSV linking microsample identifiers to plate positions, collection information, image labels, pixel coordinates and stage coordinates, as well as the possibility to create overview images of the specimens, and automatically calculating distances between cutting points and regions of interest. With these capabilities, LMDmapper is intended as a practical bridge between microscope-side laser microdissection records and downstream spatial omics sample tracking.

Plain Language summary

Open Peer Review

Approval Status ✓ ✓

	1	2
version 2 (revision) 08 Sep 2026	✓ view	
version 1 24 Jul 2026	? view	✓ view

1. **Pierluigi Di Chiaro**, Azienda Unità Sanitaria Locale-IRCCS di Reggio Emilia, Reggio Emilia, Italy

2. **Yuliaxis Ramayo**, Institute for Research and Technology in Food and Agriculture (IRTA), Torre Marimon, Spain

Any reports and responses or comments on the article can be found at the end of the article.

Laser microdissection is a technique that lets researchers cut very small regions from a thin specimen section, such as selected cells or gut content, and collect them for DNA, RNA, protein, or metabolite measurements. Often, the scientific value of these measurements depends on knowing exactly where every collected sample came from. This can become difficult because the information is spread across microscope image files, exported tables, plate maps, collection notes, and later sample names. LMDmapper is a desktop application that helps researchers keep these pieces connected. A user can define a project layout, import microscopy imaging files, calculate the stage coordinates of each cut, inspect missing or ambiguous image links, manually correct coordinates when needed, and export a clean metadata table for downstream analysis. The tool also lets users align overview images and export cropped overview regions with the positions of collected samples. The software is open source and can be run as a packaged desktop application on macOS, Windows, and Linux. It is designed for laboratory workflows where careful tracking of microscopic sample position, plate well, image evidence, and microsample identity is needed after laser microdissection.

Keywords
Imaging, Laser capture microdissection, Micro-scale, Spatial metagenomics, Specimen

H2020 This article is included in the Horizon 2020 gateway.

Corresponding author: Antton Alberdi (antton.alberdi@sund.ku.dk)

Author roles: Alberdi A: Conceptualization, Data Curation, Formal Analysis, Funding Acquisition, Investigation, Methodology, Project Administration, Resources, Software, Supervision, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Ramirez J**: Data Curation, Formal Analysis, Investigation, Writing – Review & Editing; **Gaun N**: Formal Analysis, Investigation, Writing – Review & Editing; **Horisberger Z**: Formal Analysis, Investigation, Writing – Review & Editing; **Wang B**: Data Curation, Formal Analysis, Investigation, Writing – Review & Editing; **Trivedi U**: Project Administration, Resources, Supervision, Writing – Review & Editing; **Bogri A**: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Software, Supervision, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This work was supported by the European Union's Horizon Europe research and innovation programme under grant agreement No. 101000309, 3D'omics,and the Danish National Research Foundation's grant DNRF143.
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2026 Alberdi A *et al*. This is an open access article distributed under the terms of the Creative Commons Attribution License , which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Alberdi A, Ramirez J, Gaun N *et al.* **LMDmapper: an open-source desktop tool for spatial mapping of laser microdissection samples [version 2; peer review: 2 approved]** Open Research Europe 2026, **6**:257
https://doi.org/10.12688/openreseurope.24543.2

First published: 24 Jul 2026, **6**:257 https://doi.org/10.12688/openreseurope.24543.1

REVISED **Amendments from Version 1**

Following the suggestions of the reviewers, we have improved the explanations about data export and included a new GeoJSON format to enhance compatibility with other software. We have also clarified the compatibility range of the software and added a performance and scalability analysis to the manuscript. The case studies are now accompanied with a new figure that showcases the mentioned details, and the discussion contains a new paragraph highlighting the limitations of LMDmapper. Version 1.5.0 of LMDmapper has also been released, incorporating all requested features as well as others.

Any further responses from the reviewers can be found at the end of the article

Introduction

Laser capture microdissection, also called laser microdissection (LMD), is used to isolate defined small regions of interest from heterogeneous samples under direct microscopic visualisation.[1] This physical isolation preserves spatial context while enabling downstream molecular assays such as genomic, transcriptomic, proteomic, and metabolomic analyses.[1,2] The method is particularly useful in cases when bulk homogenisation would mix molecular signals from adjacent compartments.[2]

Modern laser microdissection workflows are increasingly connected to spatial biology and image-analysis pipelines. Commercial laser microdissection software provides tools for selecting, dissecting, navigating, and documenting cuts, including sample overviews, collector tracking, and shape-location records. Other open-source tools address complementary parts of the workflow, such as transferring regions of interest between serial sections,[3] or integrating image-analysis algorithms with laser dissection platforms.[4]

Despite these advances, post-dissection curation remains difficult in workflows where microscope image files, cut-coordinate exports, collector positions, plate configurations, and downstream sample identifiers are stored as separate records. In Leica LMD workflows, LIF (Leica Image File Format; *.lif) files contain stage coordinates (µm), image data, and acquisition metadata, while CSV exports contain complementary information such as cut shapes, image labels, and pixel coordinates (px). The lack of an accessible tool for reconciling these records makes it difficult to build spatially aware LMD sample libraries with traceable Cartesian coordinates and supporting image evidence.

LMDmapper addresses this post-dissection curation need. The tool is not intended to drive the microscope or generate cutting instructions. Instead, it focuses on mapping and validating the relationship between Leica LIF images, laser microdissection CSV exports, specimen and plate layouts, collector positions, cut-point coordinates, collection metadata, and overview images. The aim is to provide an auditable bridge between laser microdissection acquisition records and the structured metadata required for downstream molecular analysis.

Methods

Implementation

LMDmapper is implemented as a cross-platform desktop application using Electron,[5] React,[6] Vite,[7] and fast-xml-parser.[8] The core parsing procedure of the software, where output files of the Leica instrument are reconciled with the metadata, is written in TypeScript[9] and operates locally on the user's files. The software addresses a major limitation of the LIF files with regard to the absolute spatial coordinates of the cutting position through a two-step process.

First, LMDmapper reads the 'LMSDataContainerHeader' XML section of the LIF files using UTF-8 or UTF-16LE detection, and parses metadata from 'Element' nodes. For each microscopy image capture, LMDmapper extracts the element name, memory block identifier, memory size, dimensions, channel count, bit resolution, stage position, collector holder position, acquisition timestamp, and laser settings. The parser currently marks an element as display-supported when it is 2D RGB, 8-bit, three-channel, interleaved image data and the recorded memory size matches width times height times three bytes. The parser caches file-level metadata and resolves image data by scanning the binary section for memory block tokens in ASCII or UTF-16LE. It can load full images or generate reduced-size thumbnails. Stage positions (µm) are read from Leica metadata fields, normalised internally, and used as the centre point of an image element in the coordinate map.

Second, the complementary CSV parser reads laser microdissection export fields. This file includes the pixel coordinates (px) indicating the exact location of the laser cutting points within the image element recorded in the previous step, which is essential to calculate the absolute position of the microsample cutting point. Finally, LMDmapper combines the stage coordinates with the pixel coordinates to yield the absolute Cartesian coordinates of all cutting points.

All outputs are written in open, documented formats so that curated records can leave the application without a converter. Tabular outputs — the metadata table, the plate configuration, and the overview crop coordinates — are exported as CSV, and session state is retained as plain JSON. Spatial annotations are exported as GeoJSON:[10] closed contours become Polygon features, open contours LineString features, and cutting points Point features named by microsample identifier. Geometry is written in the pixel frame of the aligned overview image and carries the annotation properties expected by QuPath,[11] so contours and cutting positions can be imported directly onto the corresponding image, and are equally readable by Fiji/ImageJ,[12] napari,[13] or any GeoJSON-aware library. Because laser microdissection yields one bulk molecular profile per dissected region, these regions correspond to observations in the AnnData model[14] used by Scanpy,[15] and to columns of the expression matrix in Seurat.[16]

Operation

LMDmapper can be used as a packaged desktop application on Windows (10+), macOS (Monterey 12+), and Linux (Ubuntu 18.04+, Fedora 32+, Debian 10+) operating systems, which correspond to the nominal compatibility range of the embedded Electron runtime rather than an exhaustively tested matrix. Installers for each platform and architecture are built automatically for every release by the repository's continuous-integration workflow.

The software contains six workflow tabs that are used before (Project and Design), during (Collection) and after (Metadata, Coordinates, and Overview) the laser microdissection session (Fig. 1). The expected user workflow is:

Before the LMD session:

1. **Start a session:** Define the session users to log changes.

2. **Configure the project**: Declare session name, cryosection or specimen identifiers, stage positions, number of plates, and plate-to-cryosection assignments. LMDmapper supports multiple sampling designs.

3. **Define the plate design**: For each of the microsample collecting plates, define the location of positive samples, controls, disabled wells, and microsample numbering (Fig. 1A).



Figure 1. Screenshots of four workflow tabs. A) Design page showing the randomly assigned locations of different types of controls and regular samples across two plates. **B)** Metadata page containing sample details in tabular format. **C)** Coordinates page after loading the LIF and CSV files. Cutting points are indicated by blue markers, with the cursor position and details of the hovered image displayed in tooltips. A warning sign indicating unexpected sample information is shown above the canvas. **D)** Overview page showing cutting locations overlaid on a full-slide image. Blue dots represent regular samples, while orange dots represent membrane controls. A manually generated blue contour delineates the host tissue and is used to automatically calculate distances to the closest reference.

During the LMD session:

4. **Log microsample collection**: For each microsample cutting, declare the number of attempts for generating the sample and the number of microsamples visually checked in the collector. These data are logged in the collection tab, and displayed on the metadata table (Fig. 1B).

After the LMD session:

5. **Import files**: Load Leica LIF files and laser microdissection CSV files for each cryosection to calculate laser microdissection locations and visualise them in a two-dimensional space (Fig. 1C).

6. **Inspect the coordinate map**: Check image footprints, fix cut points, and detect missing or suspicious assignments.

7. **Enter collection metadata**: Enrich the metadata of each session with plate notes, collection method, date, time, temperature, humidity, and free-text notes.

8. **Align overview images**: Align pre-cut and post-cut overview images to the stage coordinate system, draw contours, compute crop cut positions.

9. **Export crops:** export the crop as a PNG with a JSON spatial reference, a CSV of crop and stage coordinates, and a GeoJSON of contours and cut points for third-party image analysis such as QuPath.

10. **Export metadata**: Choose visible and exported columns, and export a metadata CSV, including one contour-distance column per drawn contour (Fig. 1D).

11. **Save the session**: As a '*.lmd' file for a single session or combine multiple saved sessions into a '.mlmd' workspace.

Performance and scalability

Because laser microdissection sessions can combine several cryosections, multiple plates, and full-slide overview images, we characterised the file-handling performance of the tool on synthetic sessions of increasing size. The measurements exercise the same main-process functions the application calls over its inter-process interface: header parsing and element listing, on-demand thumbnail generation, full-resolution image loading, and session save and load (Table 1).

Listing images took 0.8-25 ms across all cases, and was fastest for the 618 MB file because it contains only two image elements and therefore the smallest XML header. Displaying the first image of a file cost 43-179 ms, dominated by the one-time offset index; every further image was displayed in 3-7 ms, including a 12000 × 9000 pixel slide overview. Peak additional memory stayed below 160 MB while listing and browsing thumbnails. Loading a full-resolution element is the exception, costing time and memory of the order of the element itself: 75 ms and roughly 245 MB of transient memory for a 309 MB overview image. Session files remained small and fast: a session with 960 cut points occupied 0.5 MB and saved or loaded in under 2 ms, and a workspace merging six sessions with 2304 cut points in total occupied 1.7 MB and saved or loaded in about 4 ms.

Three implementation choices determine the positive scaling behaviour. First, listing the images in a LIF file reads only the 'LMSDataContainerHeader' XML section, streamed in 1 MB chunks, so listing cost follows the number of image elements rather than the size of the file. Second, image data is read on demand per element, and thumbnails are generated by strided row reads, so thumbnail memory depends on the requested thumbnail size and one source row rather than on the source image. Third, the first image displayed from a multi-element file triggers a single pass over the binary section that indexes every memory-block offset; every subsequent image is served from that cached index. One hard limit follows from the parser design: the XML header must be found within the first 128 MB of the file. Files whose header exceeds that limit are rejected with an explicit message rather than partially read. However, the header of a 500-element LIF file is 0.37 MB, so the limit corresponds to header sizes far beyond those produced by the sessions we have encountered.

Table 1. LMDmapper's LIF file performance benchmarks. Details include session shape, image dimensions, access latency, and peak added memory.

Session shape	Images	Image size (px)	.lif size (MB)	XML header (KB)	List images (ms)	First image (ms)	Further images (ms)	Full-resolution image (ms)	Peak added memory (MB)
Single cryosection	24	1024 × 1024	72	18	2.1	44	6.9	0.8	100
Two plates, one cryosection	96	1024 × 1024	288	73	6.5	126	7.1	1.3	146
Large field of view	24	2048 × 2048	288	18	2.3	99	7.2	2.6	156
Element-dense header	500	256 × 256	94	381	24.8	43	3.0	0.2	98
Full-slide overview	2	12000 × 9000	618	2	0.8	179	6.1	75.0	245

Open Research Europe 2026, 6:257 Last updated: 01 OCT 2026

Use cases

Curating a single laser microdissection session

A common LMDmapper use case is a session with one or two 96-well plates and one or more cryosections. The user first records the session name, cryosection identifiers, stage positions, and plate layout. Wells are classified as containing samples, controls, or not used. LMDmapper then generates microsample codes that remain tied to the plate and cryosection design.

The user imports the LIF files and CSV exports for each cryosection. LMDmapper parses the image metadata, reconstructs source columns from the CSV rows, routes the source columns onto the configured plate segments, and computes stage coordinates for wells with linked image names and pixel coordinates. The Metadata tab then provides one row per plate well, including plate, well, cryosection, sample type, microsample code, collection label, image labels, pixel coordinates, stage coordinates, size, status, notes, and optional distances to user-defined contours. Each row links the downstream microsample identifier to its tissue-derived coordinate evidence and collection context. For wells without valid coordinate evidence, LMDmapper marks the row as pending, warning, or bad depending on sample type and available information (Fig. 2A).

Resolving missing image links and orphan images

Laser microdissection exports may contain incomplete image labels or may be missing images. LMDmapper handles these cases by distinguishing linked image records from orphan images. Orphan images are image elements present in the '*.lif' file but not assigned to a resolved metadata row.

In the Coordinates view, the user can filter orphan pre-cut and post-cut images, group them by collector row, inspect hover metadata, and assign an orphan image to a missing plate position. LMDmapper can also preserve original CSV pixel coordinates for a well even when its image link is removed, allowing the predicted cut marker to be displayed during reassignment. If no image evidence is available, the user can enter manual stage coordinates and link them to a metadata row. Manual coordinates and manual image assignments remain in warning status until explicitly confirmed, preserving an audit trail for downstream users.

The expected output is a corrected metadata table in which formerly missing or ambiguous wells have explicit manual-assignment status, image labels where available, resolved coordinates, and user notes describing the correction (Fig. 2B).

Exporting overview crops and spatial annotations

For reporting, quality control, or downstream spatial analysis, users may need an overview image crop that contains a subset of collected cut points. LMDmapper allows the user to import pre-cut and post-cut overview images, align them to the stage coordinate system, and show cut points, image footprints, orphan images, and control samples. The user can draw a selection rectangle, set an export size, compute which cut points fall within the crop, and export both the rendered crop image PNG and a CSV table of crop-local pixel coordinates. Users can also draw contours on the overview image, such as tissue boundaries or regions of interest. LMDmapper calculates the shortest distance from each cut point to each visible contour and exposes those distances as optional metadata columns.

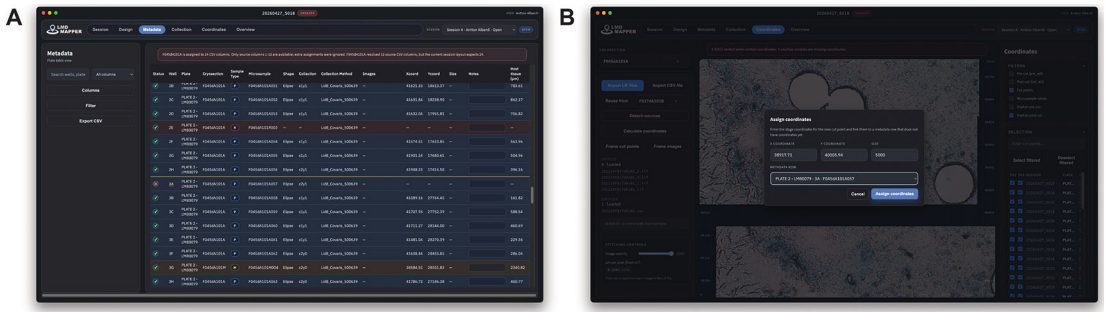

Figure 2. Screenshots representing the capabilities of LMDmapper. A) A metadata table that containing not only the cutting point coordinates (Xcoord, Ycoord) but also the measurements of each cutting point to a user-defined contour (Host tissue µm). Note that the differently coloured well 2E represents a reaction control with no coordinate information, while 3G indicates a membrane control. The well 3A is the only one with an error status, because LMDmapper was unable to find its corresponding image in the LIF file. **B)** A cutting points coordinates view over the canvas of spatially referenced microscopy images, where a window to manually assign a well membership to an orphan image is showcased.

Discussion

LMDmapper addresses a practical reproducibility problem in laser microdissection workflows: Maintaining a transparent link between image evidence, spatial coordinates, plate positions, collection records, and downstream sample identifiers. By keeping the session state in structured local files and exporting plain CSV outputs, the tool is designed to fit into downstream omics and laboratory information-management workflows without requiring a server or proprietary database export.

LMDmapper is agnostic to the staining and to any other treatment the specimen underwent before laser microdissection, because it operates on the exported images and coordinates rather than on the tissue signal itself. Haematoxylin and eosin (H&E), immunohistochemistry (IHC), and immunofluorescence (IF) images can therefore all be used as reference images, whether they are read from LIF files or imported as overview layers. Alignment of images acquired from consecutive tissue sections, however, is not supported: overview alignment applies only scale and offset within a single section, and our own benchmarking showed that, at the spatial scale LMDmapper resolves, the tissue deformation between consecutive cryosections is too large to be corrected by such a transformation. Transferring regions of interest across serial sections therefore remains outside the current scope.

The application is most useful when a dissection workflow has enough complexity that manual spreadsheet curation becomes error-prone. Examples include split plates, multiple cryosections, positive and negative controls, shared CSV sources, re-cuts, missing image labels, manually corrected coordinates, and multiple saved sessions that need to be compared. The explicit warning states for inferred or manual assignments are intended to reduce hidden curation bias by making uncertain mappings visible at export time.

Ethics and consent

Ethical approval and consent were not required.

Data availability code and software availability

Source code available from:

- github.com/anttonalberdi/lmdmapper
- Archived software available from: 10.5281/zenodo.20827562
- License: MIT License.

Acknowledgements

LMDmapper was developed within the project 3D'omics, which received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. No 101000309.

References

1. Emmert-Buck MR, Bonner RF, Smith PD, *et al.*: **Laser capture microdissection.** *Science.* 1996; **274**(5289): 998–1001.
2. Espina V, Wulfkuhle JD, Calvert VS, *et al.*: **Laser-capture microdissection.** *Nat. Protoc.* 2006; **1**(2): 586–603.
3. Schaaf JP, Mitchell D, Ogata J, *et al.*: **ShapeUpLMD: An automated pipeline for spatially optimized laser microdissection in multi-omic tissue profiling.** *bioRxiv. bioRxiv.* **Publisher Full Text**
4. Hipp JD, Johann DJ, Chen Y, *et al.*: **Computer-aided laser dissection: A microdissection workflow leveraging image analysis tools.** *J. Pathol. Inform.* 2018; **9**(1): 45.
5. Electron Contributors: Electron: 2026. **Reference Source**
6. Meta Open Source: React: 2026. **Reference Source**
7. Vite Contributors: Vite: 2026. **Reference Source**
8. Natural Intelligence: fast-xml-parser: 2026. **Reference Source**
9. Microsoft: TypeScript: 2026. **Reference Source**
10. Butler H, Daly M, Doyle A, *et al.*: *The geojson format.* 2016. **Reference Source**
11. Bankhead P, Loughrey MB, Fernández JA, *et al.*: **QuPath: Open source software for digital pathology image analysis.** *Sci. Rep.* 2017; **7**(1): 1–7. **Publisher Full Text**
12. Schindelin J, Arganda-Carreras I, Frise E, *et al.*: **Fiji: an open-source platform for biological-image analysis.** *Nat. Methods.* 2012; **9**(7): 676–682. **PubMed Abstract** | **Publisher Full Text** | **Free Full Text**
13. Sofroniew N, Lambert T, Bokota G. *napari: a multi-dimensional image viewer for Python.* **Reference Source** | **Reference Source**

14. Virshup I, Rybakov S, Theis FJ, *et al.*: **anndata: Access and store annotated data matrices.** *J. Open Source Softw.* 2024; **9**(101): 4371.
Publisher Full Text

15. Wolf FA, Angerer P, Theis FJ: **SCANPY: large-scale single-cell gene expression data analysis.** *Genome Biol.* 2018; **19**(1): 15.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

16. Hao Y, Stuart T, Kowalski MH, *et al.*: **Dictionary learning for integrative, multimodal and scalable single-cell analysis.** *Nat. Biotechnol.* 2024; **42**(2): 293–304.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

Open Peer Review

Current Peer Review Status: ✔ ✔

Version 2

Reviewer Report 17 September 2026

https://doi.org/10.21956/openreseurope.27160.r80661

© **2026 Di Chiaro P.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

✔ **Pierluigi Di Chiaro**
Azienda Unità Sanitaria Locale-IRCCS di Reggio Emilia, Reggio Emilia, Italy

no further comments

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: expert in spatial transcriptomics and data analysis including LMD applications

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Reviewer Report 14 August 2026

https://doi.org/10.21956/openreseurope.26589.r78198

© **2026 Ramayo Y.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

✔ **Yuliaxis Ramayo**
Institute for Research and Technology in Food and Agriculture (IRTA), Torre Marimon, Caldes de Montbui, Spain

Here, the authors present *LMDmapper*, an open-source Electron/React desktop application designed to reconcile Leica LIF stage coordinates with laser microdissection CSV pixel-coordinate exports, producing a unified metadata table that links microsample identifiers to plate positions,

image evidence, and Cartesian coordinates. The tool addresses a genuine and, to the best of my knowledge, previously unmet need. I just have few minor suggestions for the authors

Minor Comments:

1. *No performance or scalability data.* Sessions with multiple plates, multiple cryosections, and full-slide overview images could plausibly involve large LIF files and many hundreds of cutting points. A brief note on tested dataset sizes, memory footprint, or loading times (even approximate) would help prospective users anticipate whether the tool will hold up on their own session sizes.
2. The "Operation" section lists supported OS versions (Windows 10+, macOS Monterey 12+, Ubuntu 18.04+, Fedora 32+, Debian 10+) but does not state whether these were all actually tested, or whether this reflects Electron's nominal compatibility range; worth clarifying which.
3. Figure 1's caption is informative, but the main text never explicitly points the reader to panel (b) when describing metadata entry (step 7) , a small cross-referencing gap.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Computational biology, hologenomics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 28 Aug 2026

Antton Alberdi

We appreciate the positive comment of the reviewer. Minor Comments: No performance or scalability data. Sessions with multiple plates, multiple cryosections, and full-slide overview images could plausibly involve large LIF files and many hundreds of cutting points. A brief note on tested dataset sizes, memory footprint, or loading times (even approximate) would help prospective users anticipate whether the tool will hold up on their own session sizes.
>> Following this request, we have added a performance characterisation in the manuscript

(Methods, "Performance and scalability", Table 2). Sessions were benchmarked at 24–500 images per .lif file, file sizes of 72–618 MB, individual images up to 12000 × 9000 pixels, and sessions of up to 960 cut points, with merged workspaces of 2304 cut points. Listing the images in a file takes 0.8–25 ms and depends on the number of image elements rather than file size; the first image displayed from a file costs 43–179 ms because of a one-time memory-block index pass, and every further image 3–7 ms irrespective of its dimensions. Peak additional memory stays below 160 MB while browsing, rising to roughly the size of the image only when a full-resolution element is displayed. Session files remain under 2 MB and save or load in a few milliseconds. The benchmark is now included in the repository as npm run benchmark, and the full output, method, and known limits (including the 128 MB .lif header ceiling) are documented in docs/benchmark.md, so the measurements can be reproduced on other hardware.
The "Operation" section lists supported OS versions (Windows 10+, macOS Monterey 12+, Ubuntu 18.04+, Fedora 32+, Debian 10+) but does not state whether these were all actually tested, or whether this reflects Electron's nominal compatibility range; worth clarifying which. >> The listed versions are the nominal compatibility range of the Electron runtime that LMDmapper embeds (Electron 40), not a matrix of operating systems on which we have run the application. Specifically, the macOS binary produced by this Electron version declares a minimum system version of 12.0, and Windows 10 or later and glibc 2.28-based distributions (Ubuntu 18.04, Debian 10, Fedora 32 and later) are the platforms declared as supported by the Electron project. Installers for all distributed platform and architecture combinations are built automatically for every release by the continuous-integration workflow included in the repository, using the macOS (Apple silicon and Intel), Windows, and Ubuntu 24.04 (x64 and arm64) runner images of GitHub Actions; the same workflow lints and builds the source tree on Linux for every change. Figure 1's caption is informative, but the main text never explicitly points the reader to panel (b) when describing metadata entry (step 7) , a small cross-referencing gap. >> We have now added a reference to 1b into the manuscript text.

Competing Interests: No competing interests were disclosed.

Reviewer Report 12 August 2026

https://doi.org/10.21956/openreseurope.26589.r78196

© 2026 Di Chiaro P. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Pierluigi Di Chiaro
Azienda Unità Sanitaria Locale-IRCCS di Reggio Emilia, Reggio Emilia, Italy

The authors developed LMDmapper, an open-source desktop application for designing, tracking, and documenting micron-scale spatial samples and associated metadata generated during laser microdissection sessions. This software addresses an important gap in the LMD field by providing

a structured system for organizing images, sample information, and experimental metadata after microdissection. However, several aspects of the manuscript are not yet sufficiently clear for potential users. In particular, some software functions and workflows require a more detailed visual explanation, and the advantages, limitations, and possible downstream applications of the exported outputs should be discussed more clearly.

Improve the quality and readability of Figure 1.
The resolution of the screenshots should be increased, as several elements are currently difficult to read. Rectangles, arrows, numbered labels, or short annotations should be added to better indicate the specific function illustrated in each panel. In particular, panel D should more clearly show how to align overview images, draw contours, compute crop-cut positions, and export overview crops as PNG files together with the corresponding CSV files. Additional screenshots or intermediate panels should be included, if necessary, to make these steps easier to understand.

Add a figure illustrating the main use cases of LMDmapper.
Similarly, a dedicated figure should be added to the Use Cases section. This would help clarify how the software can be used to resolve missing image links and orphan images, as well as to export overview crops and spatial annotations.

Briefly expand the Discussion on the advantages and limitations of LMDmapper in the era of spatial biology.
The authors should more clearly discuss the advantages and limitations of their tool. For example, they should clarify whether additional H&E, IHC, or IF images acquired from consecutive tissue sections can be imported and aligned with the image used for LMD. Another important point is whether the exported images, contours, and spatial coordinates of the microdissected regions can be used in third-party software, such as QuPath for image analysis, or integrated with molecular outputs, such as transcriptomic data, using tools such as Seurat or Scanpy.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: expert in spatial transcriptomics and data analysis including LMD applications

I confirm that I have read this submission and believe that I have an appropriate level of

expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 28 Aug 2026

Antton Alberdi

We appreciate the timely review of our manuscript. We address the reviewer's comments in the following.
Improve the quality and readability of Figure 1. >> We have uploaded a new version of the figure in a different format. The resolution of the screenshots should be increased, as several elements are currently difficult to read. Rectangles, arrows, numbered labels, or short annotations should be added to better indicate the specific function illustrated in each panel. In particular, panel D should more clearly show how to align overview images, draw contours, compute crop-cut positions, and export overview crops as PNG files together with the corresponding CSV files. Additional screenshots or intermediate panels should be included, if necessary, to make these steps easier to understand. >> Following this request, we have increased the resolution of the images. However, instead of adding new images in this initial figure, we have now included a figure containing two more screenshots to better explain the functionality of LMDmapper. These figures incorporate the details requested by the reviewer in this comment. Add a figure illustrating the main use cases of LMDmapper. Similarly, a dedicated figure should be added to the Use Cases section. This would help clarify how the software can be used to resolve missing image links and orphan images, as well as to export overview crops and spatial annotations. >> See response above. Briefly expand the Discussion on the advantages and limitations of LMDmapper in the era of spatial biology. The authors should more clearly discuss the advantages and limitations of their tool. For example, they should clarify whether additional H&E, IHC, or IF images acquired from consecutive tissue sections can be imported and aligned with the image used for LMD. >> We have expanded the Discussion to address this. Another important point is whether the exported images, contours, and spatial coordinates of the microdissected regions can be used in third-party software, such as QuPath for image analysis, or integrated with molecular outputs, such as transcriptomic data, using tools such as Seurat or Scanpy. >> Following this request, we have released a new version, 1.5.0, which includes the possibility to export contours as an RFC 7946 GeoJSON file. Geometry is written in the pixel frame of the aligned overview image and carries QuPath's annotation properties, and is equally readable by ImageJ/Fiji, napari, or any GeoJSON-aware library. This is now mentioned in the manuscript as well.

Competing Interests: No competing interests were disclosed.