CSRefiner: A lightweight framework for fine-tuning cell segmentation models with small datasets

Can Shi[1,2], Yumei Li[1], Jing Guo[1], Qiuling Chen[1], Tingting Cao[1], Sha Liao[1], Ao Chen[1], Mei Li[1,*], and Ying Zhang[1,*]

[1]BGI Research, Shenzhen 518083, China
[2]School of Artificial Intelligence, University of China Academy of Science, Beijing 100049, China

Abstract

Recent advances in spatial omics technologies have enabled transcriptome profiling at subcellular resolution. By performing cell segmentation on nuclear or membrane staining images, researchers can acquire single-cell spatial gene expression data, which in turn enables subsequent biological interpretation. Although deep learning-based segmentation models achieve high overall accuracy, their performance remains suboptimal for whole-tissue analysis, particularly in ensuring consistent segmentation accuracy across diverse cell populations. Moreover, manual annotation of complete or multiple tissue sections imposes prohibitive time and cost constraints that significantly impede research progress. To address these challenges, we present CSRefiner, a lightweight and efficient fine-tuning framework for precise whole-tissue single-cell spatial expression analysis. Our approach adapts pre-trained segmentation models using only a minimal set of annotated image patches. This study demonstrates CSRefiner's superior performance across various staining types and its compatibility with multiple mainstream models. Combining operational simplicity with robust accuracy, our framework offers a practical solution for real-world spatial transcriptomics applications.

Keywords: Cell segmentation; Fine-tuning; Spatial omics

1 Introduction

Spatially resolved transcriptomics (SRT) technologies have opened a new door for investigating the cellular microenvironment and cellular heterogeneity in complex tissues[1, 2, 3, 4, 5], these technologies were named the "Method of the Year 2020" by Nature Methods[6]. Recent advances in spatial omics now enable RNA capture at subcellular resolution, as demonstrated by platforms such as Stereo-seq[7] (0.5 μm), Visium HD (2 μm), and Seq-Scope[8] (0.5–0.8 μm). These high-resolution SRT methods enable researchers to investigate single-cell transcriptomes, unlocking insights into functional, developmental, and disease-related mechanisms of living organisms[1]. To obtain single-cell data, cell segmentation serves as the core step for extracting single nuclear or cell boundaries from cell staining images[9, 10, 11, 12, 13]. Those single-cell analysis pipelines in spatial transcriptomics (SRT) typically employ state-of-the-art deep learning models for cell segmentation, such as Cellpose[14, 15, 16, 17], StarDist[18], and the cell segmentation model within CellBin[19], a generalist framework for constructing single-cell expression matrices in spatial omics (hereafter referred to as CellBin).

Although current cell segmentation algorithms exhibit satisfactory performance under controlled conditions, their practical application encounters significant challenges due to cellular morphological heterogeneity and variations in image quality[13, 19, 20, 21]. Specifically in entire sections of complex biological tissues that contain multiple cell types, these algorithms often fail to maintain consistent segmentation accuracy across all cellular populations (see Results and Fig. 1A). Even if the model achieves a segmentation accuracy of more than 95% in the global evaluation, under-segmentation or over-segmentation of local areas may still lead to key segmentation errors, which will further affect the reliability of differential expression analysis, identification of rare cell subpopulations, and construction of cell spatial interaction networks, and even lead to erroneous biological conclusions. Manual annotation is often the

*Correspondence address. Mei Li and Ying Zhang, BGI Research, Shenzhen 518083, China. Email: limei1@genomics.cn; zhangying7@genomics.cn

ultimate solution for cell segmentation, but large or multiple samples will bring more manpower. Annotating a single 1 cm × 1 cm sample often takes an annotator half a month or even longer. This high-intensity manual effort not only greatly increases the research cost, but also seriously restricts the research progress.

In the face of the above challenges, we developed CSRefiner, a lightweight and efficient cell segmentation fine-tuning framework designed to obtain single-cell spatial expression data and thereby lay a critical foundation for downstream biological analyses. By introducing a small sample fine-tuning mechanism, the framework can quickly adapt the existing model to specific image features with only a small number of manually annotated samples, thereby significantly improving the segmentation accuracy. Currently, it is compatible with three mainstream cell segmentation models in the field of spatial transcriptomics, namely Cellpose, StarDist, and CellBin, and will be continuously expandable to support more models in the future. Additionally, it is applicable to a variety of staining methods, such as hematoxylin and eosin (H&E), single-stranded DNA (ssDNA), 4',6-diamidino-2-phenylindole (DAPI), and multiplex immunofluorescence (mIF). CSRefiner is particularly suitable for long-term research projects that require processing large quantities of samples. To facilitate evaluation and benchmarking, we also generated and publicly release two demo datasets consisting of formalin-fixed, paraffin-embedded (FFPE) DAPI- and fresh frozen (FF) H&E-stained mouse brain sections with paired expression matrices and annotation results. Collectively, these capabilities enable CSRefiner to significantly minimize manual intervention while preserving segmentation accuracy, particularly in critical regions. This synergistic enhancement not only accelerates the spatial transcriptomics analysis pipeline but also unlocks the potential for robust, large-scale investigations by providing researchers with a scalable and cost-effective annotation strategy.

2 Results

2.1 Limitations of pre-trained models in complex tissues

To investigate the performance of commonly used pre-trained segmentation models on spatial transcriptomics images, we first established two demo datasets that are newly generated and reported in this study: FFPE DAPI-stained and FF H&E-stained mouse brain sections. Each dataset includes paired images, expression matrices and downstream analyses, thereby providing valuable resources for benchmarking segmentation approaches.

For the DAPI-stained demo dataset, we evaluated four representative pre-trained models: two official versions of Cellpose (cyto, the basic residual U-Net model; and cpsam, a recently released variant that integrates a SAM-based Vision Transformer encoder to enhance generalization capability), StarDist (2D_versatile_fluo, optimized for fluorescence-stained images), and CellBin, a segmentation framework specifically designed for spatial transcriptomics data with an emphasis on high recall.

As shown in Fig. 1A, the mouse brain section exhibited region-specific morphological differences: the hippocampus was densely packed with cells and had blurred nuclear boundaries, rendering segmentation particularly challenging compared to non-hippocampal regions. Across four selected representative image patches, we observed that segmentation performance was generally higher in non-hippocampal areas, where both Cellpose-cyto and Cellpose-cpsam achieved satisfactory accuracy, with F1 scores greater than 0.85 and 0.90, respectively. In contrast, StarDist and CellBin performed less consistently. For CellBin in particular, high background noise often led to over-segmentation, where a single nucleus was split into multiple fragments, substantially lowering overall accuracy. In the hippocampus, segmentation performance declined markedly across all models. While Cellpose-cpsam achieved the best balance with an F1 score around 0.6, it still suffered from substantial under-segmentation. CellBin exhibited the opposite tendency: it achieved nearly 90% coverage by visual inspection, yet suffered from poor boundary precision and frequent over-segmentation, resulting in relatively low F1 scores (0.26 and 0.49, depending on patch). Comparable trends were observed in the H&E-stained demo dataset (Supplementary Fig.1), confirming the generality of these observations across staining modalities. In summary, these results highlight the limitations of pre-trained models in handling heterogeneous tissue morphologies. Despite their satisfactory performance in relatively simple regions, none of the tested algorithms maintained robust segmentation accuracy across the hippocampus, underscoring the need for adaptive approaches to achieve reliable single-cell resolution in complex biological tissues.

2.2 CSRefiner framework overview

As illustrated in Fig. 1B, CSRefiner is a lightweight and flexible pipeline designed to rapidly adapt pre-trained segmentation models to tissue-specific features in spatial transcriptomics images. The workflow consists of four streamlined steps. First, users construct a small training set by annotating representative patches from regions where baseline segmentation is suboptimal, such as densely packed or morphologically heterogeneous regions (e.g., hippocampus). For most models, patches of 256 × 256 pixels are used, except for StarDist H&E models which require 512 × 512 patches. A base model is then selected from supported algorithms (Cellpose, StarDist, or CellBin), with more

accurate base models requiring fewer training patches. Second, the chosen model is fine-tuned on the annotated patches through a unified, parameter-controlled script. Fine-tuning follows standard settings: Cellpose models are trained for 100 epochs as recommended by the official documentation, while StarDist and CellBin employ early stopping to optimize training efficiency. Third, the refined model is evaluated on the training set/testing set and iteratively refined by adding more annotated patches if necessary, ensuring that the model achieves satisfactory performance across challenging regions. Finally, the optimized model is applied to whole-slide images, and the resulting segmentation boundaries are integrated with the corresponding expression matrix to generate a CellBin gene expression format (cgef) file, a widely used standard in spatial transcriptomics, facilitating downstream single-cell analyses.

In summary, CSRefiner provides an accessible and extensible framework that requires only limited manual annotations to achieve reliable segmentation while directly supporting downstream spatial transcriptomics analyses.

2.3 CSRefiner enhances segmentation accuracy and reduces manual effort

2.3.1 Visual improvement of segmentation performance

Compared with the baseline pre-trained models (Fig. 1A), CSRefiner substantially enhanced the segmentation quality across different regions of the mouse brain section (Fig. 1C). In non-hippocampal areas, segmentation became more refined: Cellpose-cyto and Cellpose-cpsam achieved F1 score of approximately 0.95 on representative patches, while StarDist improved to about 0.9 and CellBin to above 0.65. In the hippocampus, where dense cellular packing and blurred boundaries had previously led to severe under-segmentation, CSRefiner enabled the recovery of many additional cells, markedly improving local accuracy. Fine-tuned F1 scores in the hippocampus were approximately 0.7 for Cellpose-cyto, 0.8 for Cellpose-cpsam, 0.85 for StarDist, and 0.55 for CellBin. Although CellBin remained slightly less accurate than Cellpose and StarDist due to its emphasis on recall, it also exhibited clear improvements after fine-tuning. Collectively, these observations demonstrate that CSRefiner consistently enhances segmentation results in both easy and challenging tissue regions.

2.3.2 Quantitative improvements across evaluation metrics

We next assessed the impact of CSRefiner on segmentation accuracy using five standard evaluation metrics: precision, recall, F1 score, Jaccard index, and Dice coefficient (Fig. 2A–E). Boxplot comparisons showed improvements across all metrics for all tested models. Cellpose-cyto F1 increased from 0.54 to 0.78, Cellpose-cpsam from 0.79 to 0.91, StarDist from 0.42 to 0.87, and CellBin from 0.37 to 0.71. Similar gains were observed for precision, recall, Dice, and Jaccard indices. Notably, the improvement was particularly pronounced in models with weaker initial performance, such as StarDist and CellBin, while even the high-performing Cellpose-cpsam model achieved measurable gains. Moreover, the reduced variance in post-fine-tuning scores indicates enhanced consistency and robustness of the models across different image regions.

2.3.3 Reduction of annotation time and manual annotation burden

When applied to large tissue sections, CSRefiner greatly reduced annotation burden. We compared the overall time required for CSRefiner-assisted fine-tuning against full manual annotation (Fig. 2F). Based on an average annotation time of $\tilde{1}5$ minutes per 256×256 image patch, generating 20 training patches required $\tilde{3}00$ minutes. Model fine-tuning, evaluation, and inference added an additional $\tilde{1}00$ minutes, resulting in a total of $\tilde{4}00$ minutes ($\tilde{6}.7$ hours) for a complete CSRefiner workflow. In contrast, manual annotation of an entire whole-slide image would require approximately 10 days ($\tilde{3}6 \times$ longer) under comparable conditions. These results demonstrate that CSRefiner enables the generation of high-quality segmentation results within a fraction of the time, significantly lowering the cost and labor of manual annotation.

These findings demonstrate that CSRefiner not only enhances segmentation accuracy both visually and quantitatively but also offers a scalable, time-efficient solution for large-scale spatial transcriptomics projects.

2.4 Downstream biological impact of improved segmentation

To further evaluate the biological relevance of CSRefiner, we examined how improved segmentation accuracy influenced downstream spatial transcriptomics analyses.

2.4.1 Improved spatial mapping

Using StarDist, the model with the greatest performance gains, fine-tuned segmentation enabled the generation of high-quality cgef matrices. Subsequent cell type annotation with cell2location[22] produced spatial distributions that

closely matched established anatomical structures and biological priors (Fig. 2G). Notably, hippocampal subregions were delineated with high fidelity, confirming that CSRefiner-generated segmentations directly support biologically meaningful spatial mapping.

2.4.2 Increased detection of DEGs and enriched pathways

Differential expression analysis revealed that fine-tuning increased the number of significantly upregulated genes across multiple cell types. Ranking the differences between before and after fine-tuning results identified the six cell types with the largest increases in significant genes (Fig. 2H). Among these, hippocampal populations featured prominently, with Ext_Hpc_DG1 occupying the fourth position—consistent with visual inspection indicating improved segmentation in densely packed hippocampal regions.

As shown in Fig. 2I–J, we performed enrichment analysis of Ext_Hpc_DG1 using the GO Biological Process 2023 database[23, 24]. Before fine-tuning, this analysis highlighted a limited set of synaptic signaling processes, such as anterograde trans-synaptic signaling (GO:0098916) and chemical synaptic transmission (GO:0007268), supported by a small group of canonical synaptic genes (e.g., GRIA1, GRIN2A, CACNA1E, DLGAP1). By contrast, after fine-tuning, the same analysis revealed a broader and biologically coherent network of pathways, including axon guidance, neuron projection guidance, and axonogenesis (Fig. 2K). The expanded gene set (e.g., EPHA4/6/7, SLIT1, DSCAM/DSCAML1, KALRN, CAMK2A/B, GRIA1/2, GRIN2A) is shown in Fig. 2L and represents critical regulators of axonal connectivity and synaptic plasticity, in line with the known functions of hippocampal excitatory neurons[25, 26, 27, 28].

2.4.3 Enhanced cell annotation and quality metrics

At the tissue level, annotation visualizations showed that fine-tuning substantially increased the number of segmented and annotated cells across hippocampal subregions (Fig. 2M-N). Annotation confidence scores also shifted toward higher values, indicating more reliable cell type assignments (Fig. 2O). Quantitative assessment of cell morphology revealed that extreme outliers in cell size were eliminated, while overall cell area slightly increased, reflecting more regular and biologically plausible segmentation boundaries (Fig. 2P). Furthermore, cell type-specific counts demonstrated biologically consistent changes (Fig. 2Q): Ext_Hpc_CA1 and Ext_Hpc_DG1 cells increased markedly, Ext_Hpc_DG2 remained stable, and Ext_Hpc_CA2/CA3 exhibited modest decreases. These results align with expected patterns of hippocampal cytoarchitecture. Finally, fine-tuning raised the number of detected genes per cell (Fig. 2R), thereby enhancing the sensitivity of downstream transcriptomic analyses. By recovering expression signals from cells that would otherwise be under-segmented or omitted, CSRefiner improved both the resolution and the biological interpretability of spatial transcriptomics data.

These findings demonstrate that CSRefiner does not merely improve image-level segmentation accuracy, but also restores biologically meaningful signals at the transcriptomic level, thereby strengthening downstream single-cell resolution analyses in spatial transcriptomics.

2.5 Training set design for optimal performance

To evaluate how training set size influences model performance, we systematically varied the number of annotated cells used for fine-tuning and tested on independent held-out regions not included in the training set.

Fig. 3A shows that in the hippocampus, model performance improved gradually with increased training examples: F1 scores rose from 0.23 with no training cells, to 0.35 with 105 cells, 0.36 with 357 cells, and 0.49 with 1,430 cells. Accuracy and recall showed similar upward trends, and the performance curve did not plateau at 1,430 cells, suggesting further gains with additional annotations. In contrast, in non-hippocampal regions, performance improved rapidly with relatively few training examples: F1 increased from 0.33 with no training cells, to 0.67 with 105 cells, 0.84 with 357 cells, and 0.85 with 566 cells. Here, the performance curve approached saturation after $\tilde{3}00$ cells, and additional annotations provided diminishing returns (Fig. 3B). These observations suggest that regional complexity critically determines training requirements: dense and heterogeneous areas such as the hippocampus require larger training sets to approach optimal performance, whereas simpler regions converge with fewer annotations.

To quantitatively assess region-specific complexity, we compared six structural metrics between hippocampal and non-hippocampal images (Fig. 3C). Hippocampal cells exhibited smaller mean inter-cell distance (18.7 vs. 23.3) and lower compactness (1.38 vs. 1.46), indicating tighter packing. Cell density was higher (0.0039 vs. 0.0027), while distribution uniformity was lower (0.189 vs. 0.264), suggesting large-scale dense clustering rather than localized groups. Edge contrast was reduced (2.01 vs. 3.21), reflecting weaker boundaries that complicate segmentation. In contrast,

shape complexity was comparable between regions (13.99 vs. 13.63), implying that cell morphology was not a major differentiating factor.

Overall, these results confirm that hippocampal tissue poses greater segmentation challenges due to denser cellular organization and lower boundary clarity, consistent with qualitative visual inspection. Optimal training set design should therefore consider both region type and structural complexity, balancing annotation effort with expected performance gains.

3 Discussion

Accurate cell segmentation is fundamental to spatially resolved transcriptomic analysis, as errors at this stage propagate to downstream tasks such as cell type annotation, differential expression analysis, and pathway enrichment analysis. In this study, we introduce CSRefiner, a flexible framework designed to fine-tune compatible cell segmentation models for histological images. Through comparative evaluation, we demonstrate that CSRefiner substantially improves segmentation accuracy across multiple architectures, reduces annotation time by more than an order of magnitude, and enhances biological interpretability of downstream analyses. Additionally, we have released two high-quality datasets containing paired images, segmentation annotations, gene expression matrices, and downstream analyses, providing a valuable resource for benchmarking and future method development.

A key finding is that model fine-tuning tailored to tissue-specific features markedly outperforms direct application of pre-trained models. In challenging regions such as the hippocampus, fine-tuned models achieved notable gains in F1 score and other segmentation metrics, recovering cells that were otherwise missed. These improvements translated directly into more coherent spatial maps, more robust differential gene expression signals, and broader, biologically consistent pathway enrichment results. Thus, CSRefiner bridges a critical gap between generic segmentation tools and the unique requirements of spatial transcriptomics.

Our results further highlight the importance of training set design. Performance scaling analyses revealed that simpler regions reach saturation with relatively few annotated cells, whereas complex structures such as the hippocampus require substantially larger training sets to achieve comparable accuracy. Quantitative metrics of tissue organization—such as cell density, compactness, and boundary clarity—correlated well with segmentation difficulty, providing a principled way to estimate annotation requirements for different tissue contexts. These insights may inform annotation strategies in future large-scale studies, enabling an optimized balance between manual effort and segmentation performance.

While CSRefiner demonstrates robust improvements, several limitations warrant consideration. Although fine-tuning reduces annotation requirements compared with full retraining, some manual effort is still necessary, especially in highly complex regions. Moreover, integration with emerging multimodal data, such as joint spatial proteomics, may require further adaptation of the framework.

In conclusion, CSRefiner establishes a practical and extensible approach for enhancing cell segmentation in spatial transcriptomics. By combining the strengths of existing deep learning models with targeted fine-tuning, it enables more reliable extraction of single-cell features, thereby improving the fidelity of downstream biological analyses. Combined with the release of new data, this work also brings community resources. In addition to expanding to more models, future developments can also consider interactive annotation to prepare training sets and multimodal data input, further lowering the barriers to achieving accurate and scalable spatial analysis.

4 Methods

4.1 Tissue samples and data acquisition

4.1.1 Experimental animals and ethical approval

Two mouse brain tissue sections were used in this study. Brain tissues were collected from 6-week-old C57BL/6 male mice, provided by the Guangzhou Institutes of Biomedicine and Health, Chinese Academy of Sciences, in collaboration with the BGI laboratory. All animal experiments followed institutional ethical guidelines and were approved by the relevant committees.

4.1.2 Tissue preparation and staining

FFPE DAPI-stained mouse brain section: Fresh mouse brains were dissected and fixed in 4% paraformaldehyde (Servicebio, G1121) at room temperature for 24-48 hours, followed by dehydration, paraffin embedding, and sectioning into 5 μm-thick slices using standard histopathological protocols. Sections were mounted on Stereo-seq Chip N after

flattening in a 45°C water bath, then baked at 42°C for 3 hours and dried overnight. Deparaffinization, decrosslinking (95°C with FFPE Decrosslinking Reagent; STOmics, 211KN114-EA), and post-decrosslinking fixation (-20°C methanol) were performed according to the STOmics Stereo-seq Transcriptomics Set protocol[29]. DAPI staining (1 μm/mL, room temperature, 10 minutes in the dark) was conducted after washing with 0.1×SSC buffer (Thermo Fisher Scientific, AM9770) supplemented with RNase inhibitor (GCATBio, LS-EZ-E-00006Q).

FF H&E-stained mouse brain section: Dissected mouse brains were snap-frozen in dry ice-cooled Tissue-Tek OCT (Sakura, 4583) and sectioned into 10 μm-thick coronal slices using a Leica CM1950 cryostat. Sections were mounted on Stereo-seq Chip N, incubated on a 37°C slide dryer for 5 minutes, and fixed in -20°C methanol for 30 minutes. Adjacent sections on glass slides were stained with H&E using a Solabio G1121 kit (hematoxylin for 8 minutes, eosin for 3 minutes) following the manufacturer's instructions, then dehydrated, cleared, and mounted with neutral resin.

4.1.3 Imaging data acquisition

FFPE DAPI-stained mouse brain section: Fluorescence images were acquired using a STOmics Microscope Go Optical system (Scanner Version 1.2.2, 20×/0.5 NA objective, Ximea Mc124 camera) in epi-fluorescence mode (excitation: 358 nm, emission: 461 nm). Image quality was verified via the StereoMap software (imageQC module).

FF H&E-stained mouse brain section: Brightfield images were captured using a Motic PA53 FS6 microscope (PA53Scanner 1.0.0.14, 10×/0.75 NA objective, PA53 FS6 SCAN S5LITE RGB module) in epi-brightfield mode, with images saved in TIFF format.

4.1.4 Gene expression matrix generation

FFPE and FF sections on Stereo-seq chips were processed for RNA capture, reverse transcription, and cDNA amplification following the STOmics Stereo-seq Transcriptomics Set protocol[29]: FFPE samples used random primers for total RNA capture and FFPE Decrosslinking Reagent for decrosslinking, while FF samples underwent methanol fixation and short-time permeabilization. Libraries were constructed via cDNA fragmentation (Tn5 transposase) and PCR, then sequenced on MGI DNBSEQ-T1 (FFPE) or DNBSEQ-Tx (FF) sequencers.

Raw sequencing data were processed with the SAW pipeline[30]: low-quality reads were filtered, CID sequences were mapped to chip coordinates (1 base mismatch allowed), and reads were aligned to the mm10 genome (STAR software[31], MAPQ > 10). UMIs for the same CID and gene were collapsed to eliminate PCR duplicates, and spatial gene expression matrices were generated by integrating gene counts with coordinates.

4.2 Model selection, patch generation, and training set construction

To enable efficient fine-tuning with limited annotated data, CSRefiner encourages users to construct a compact yet informative training set from the target images. The process begins with the selection of a suitable pre-trained model. Specifically, users are advised to apply the available pre-trained segmentation models supported by CSRefiner (e.g., Cellpose, StarDist, or CellBin) to the target tissue image and visually inspect the initial segmentation. This assessment, typically carried out using widely adopted visualization tools such as ImageJ, involves evaluating both global coverage and local boundary accuracy. The model that demonstrates the most satisfactory overall performance is then chosen as the starting point for fine-tuning.

Following model selection, the full tissue image is divided into smaller patches for training. CSRefiner provides a lightweight cropping utility that supports batch processing of whole-slide images and allows users to customize patch sizes to facilitate this step. With the exception of StarDist's H&E-optimized pre-trained model ('2D_versatile_he'), which requires training patches of 512 × 512 pixels, all other models operate with patches of 256 × 256 pixels. For convenience, CSRefiner also includes a padding utility that can extend 256 × 256 images to the 512 × 512 size without altering cell morphology. This is achieved by symmetrically reflecting the image content at the borders (border reflection padding), which preserves the original cellular structures while avoiding artificial distortions introduced by resizing or interpolation. This ensures compatibility with different base models while avoiding the need to separately generate patches of multiple resolutions.

When selecting patches for annotation, priority should be given to regions where segmentation quality is noticeably suboptimal—for example, areas with blurred cell boundaries, overlapping or densely packed cells, or uneven staining that leads to under-segmentation or over-segmentation. To maintain model generalizability, a small proportion of morphologically simple and sparsely populated regions should also be included, although overrepresentation of these regions is discouraged to prevent model forgetting effects.

Annotation of the selected patches can be performed using any standard software familiar to the user. In practice, QuPath is recommended for its accessibility and compatibility. CSRefiner provides a step-by-step tutorial for generating cell annotations with QuPath. However, QuPath typically produces semantic masks (all cells assigned a common label), whereas instance segmentation models such as Cellpose and StarDist require instance masks, in which each cell is assigned a unique integer identifier. To address this mismatch, CSRefiner offers a conversion script that automatically transforms semantic masks into instance masks, ensuring training compatibility.

Finally, CSRefiner organizes the annotated images and corresponding masks into a standardized training file list. This file management system simplifies subsequent fine-tuning, enables reproducible execution of the workflow, and facilitates version tracking of training datasets. Based on empirical results, we recommend preparing at least $\tilde{3}00$ annotated cells to achieve satisfactory fine-tuning performance. If the outcome of the first fine-tuning round does not meet expectations, additional annotated samples can be incrementally added. In such cases, to avoid catastrophic forgetting, all available annotated data should be combined and the model re-fine-tuned starting from the original pre-trained weights, rather than continuing from the previously fine-tuned checkpoint.

4.3 Base models and fine-tuning procedures

4.3.1 Cellpose (cyto, cyto3, nuclei, etc.)

Cellpose is a general cell segmentation model based on a U-Net architecture with residual connections. It achieves instance segmentation by predicting vector flows originating from cell centers. It has officially released multiple pre-trained models (such as cyto, cyto3, and nuclei) for different tissue types and staining methods.

CSRefiner integrates an automated fine-tuning process based on Cellpose's official training interface 'cellpose.train'. Users can quickly load training data by providing the path file of the image and the corresponding mask pairs. CSRefiner supports grouping based on the sample identifier (SN) in the image file name and uses an optional ratio to split the data into training and validation sets to ensure balanced coverage of each sample during training.

Training utilizes GPU acceleration by default. The optimizer is stochastic gradient descent (SGD) with an initial learning rate of 0.1 and weight decay of 1e-4. The number of training epochs can be configured based on task requirements. After each epoch, the training and validation losses are recorded and saved as JSON log files. Additionally, the loss curve is automatically generated to help users evaluate training progress and model convergence.

The model fine-tuning script is integrated into the unified command-line interface (CLI), supporting one-click switching of different Cellpose pre-trained models. Upon completion, the model weights and log files are automatically saved to a timestamped directory, facilitating version control and reproducibility of experimental results.

4.3.2 Cellpose-SAM (cpsam)

Cellpose-SAM is a new architecture introduced in Cellpose 4.0, which combines the image encoder capabilities of the Segment Anything Model (SAM) released by Meta AI and the flow field prediction mechanism of Cellpose for segmenting cell images. Unlike the original SAM, which relies on a prompt-based decoder, Cellpose-SAM discards the decoder entirely and retains only the pure visual encoder. This encoder directly outputs the vector flows required by Cellpose, making it more suitable for instance segmentation tasks in dense biological images.

In CSRefiner, we provide an independent training script adapted to Cellpose-SAM. The training process is consistent with the previous version of Cellpose, including key steps such as data loading, SN grouping, and loss recording. Users can complete model customization without in-depth understanding of the underlying architecture. Notably, following official recommendations, the Cellpose-SAM training script uses a smaller batch size (1) and a lower learning rate (1e-5) to ensure training stability and robustness.

This architecture has significant advantages in areas with blurred edges and densely populated cells, and is particularly suitable for segmentation fine-tuning scenarios of highly heterogeneous tissue images such as spatial transcriptomes.

4.3.3 StarDist

StarDist is a nucleus segmentation framework tailored to microscopy images, modeling objects as star-convex polygons rather than binary masks. Its backbone is a U-Net encoder–decoder that extracts multi-scale features, followed by two parallel output heads: one predicts object probabilities, and the other predicts radial distances along predefined angles to reconstruct polygons. Final instances are selected through non-maximum suppression, ensuring one unique polygon per nucleus.

CSRefiner incorporates a customized fine-tuning script for StarDist. Notably, the official StarDist framework lacks explicit recommendations for selecting training parameters and does not support the integration of an early stopping function into its training pipeline—two critical limitations that hinder the determination of optimal training epochs. To address this challenge, a single-epoch iterative training strategy was developed. In detail, the training process is executed in iterations of one epoch each, with the loss value recorded at the conclusion of every individual epoch. This design enables real-time dynamic monitoring of the validation set loss; once the model reaches a converged state (characterized by stable or non-decreasing validation loss, patience=30), the training process can be terminated prematurely, thereby simulating the functionality of a standard early stopping mechanism. Users can initialize models from scratch, from official pre-trained weights, or from previously saved checkpoints.

For fluorescent nuclear stains (e.g., DAPI, ssDNA), images are preprocessed as single-channel grayscale with percentile-based intensity normalization; for H&E images, three-channel RGB input is supported. As with other models, logs and weight files are version-controlled, and training curves are automatically generated to assist performance evaluation.

4.3.4 CellBin

CellBin is a segmentation network designed for biomedical images. It is based on the BCDU-Net D3 architecture and introduces ConvLSTM units in the decoder path to enhance spatial context modeling. With a three-level densely connected decoding structure, CellBin significantly improves segmentation performance on irregular or blurry cell boundaries, making it particularly suitable for tissue images with complex staining patterns or high background noise.

The CellBin fine-tuning process in CSRefiner is implemented based on TensorFlow 2.x. The training process uses a custom DataGenerator class to load and augment data, and the standard fit() interface provided by Keras is used to complete the training process. Preprocessing steps differ based on the staining type. For H&E images, color normalization and intensity normalization are applied using Contrast Limited Adaptive Histogram Equalization (CLAHE). For ssDNA&DAPI images, grayscale conversion, percentile-based intensity normalization, and adaptive histogram equalization are performed.

The model training uses the Adam optimizer and binary cross-entropy loss function. The ReduceLROnPlateau strategy is used to dynamically adjust the learning rate, and early stopping (patience=30 epochs) is implemented to prevent overfitting. The training and validation losses are recorded after each round of training, and a learning curve is plotted to assist analysis. Users can either train the model from scratch or fine-tune it using a pre-trained .hdf5 weight file. All training outputs (model weights, log files) will be saved uniformly to support reproducibility and downstream inference.

After obtaining a fine-tuned segmentation model, users can apply it to perform inference on complete tissue images or a batch of unannotated image patches to generate high-quality cell segmentation results. CSRefiner provides a unified and efficient inference interface that supports large-scale image processing while ensuring compatibility with downstream spatial transcriptomics analysis pipelines.

4.4 cgef file generation

cgef is a widely adopted file format in spatial transcriptomics, designed to integrate gene expression matrices with cell segmentation results. CSRefiner incorporates a dedicated utility for generating '.cgef' files by combining raw expression matrices ('.gem.gz' or '.gef' formats) with segmentation masks, leveraging the 'cgef_writer_cy.generate_cgef' function from the gefpy library for the core file generation process. The principle is to map each expression point to its corresponding cell segmentation region and associate the segmented cell mask with the spatially resolved expression matrix. For each cell instance, the gene counts from all expression points within its mask are aggregated to generate a cell gene expression matrix annotated with the cell centroid coordinates.

The workflow comprises three key steps. First, input expression matrices undergo standardization: '.gem.gz' files are converted to '.gef' format using an integrated wrapper module. Second, segmentation masks are validated against matrix dimensions, with automatic resizing applied when necessary to ensure alignment with the expression grid. Third, the system classifies masks as either semantic or instance-based, performing conversion of instance masks to semantic format when required before invoking the final '.cgef' generation process.

Output files are generated in blocks (default size: 256×256 pixels) to accommodate large-scale datasets while minimizing memory consumption. The utility automatically cleans up intermediate files and maintains comprehensive logs for full process traceability. This streamlined pipeline enables reproducible integration of cell segmentation data with transcriptomic quantification, facilitating downstream analytical tasks including cell annotation, clustering, and differential expression analysis.

4.5 Cell segmentation performance evaluation metrics

In order to comprehensively evaluate the performance of different models and fine-tuning schemes in the cell instance segmentation task, we use five classic evaluation indicators: Precision, Recall, F1 Score, Jaccard Index and Dice Coefficient. These indicators focus on measuring the overlap and matching quality between the predicted mask and the true mask, and are suitable for quantitative comparison of the generalization ability of the model in small sample training scenarios.

Precision: measures how many of the positive samples predicted by the model are real cell instances. A high precision indicates that the model prediction is more conservative and has fewer false positives (FP).

$$precision = \frac{TP}{TP + FP} \tag{1}$$

Recall: indicates how many of the real cell instances are correctly identified by the model. A high recall indicates that the model is more capable of finding all targets and has fewer missed detections (FN).

$$recall = \frac{TP}{TP + FN} \tag{2}$$

F1 Score: the harmonic mean of precision and recall, which comprehensively reflects the model's precision and recall capabilities and is an important measure of the model's overall performance.

$$F1score = \frac{2 \times (precision \times recall)}{precision + recall} \tag{3}$$

Jaccard Index: measures the ratio of the intersection and union between the predicted mask and the true mask. It is a commonly used overlap indicator in instance segmentation tasks. The higher the value, the more accurate the prediction.

$$jaccard = \frac{TP}{TP + FP + FN} \tag{4}$$

Dice Coefficient: similar to Jaccard, it also measures the degree of overlap between masks, but is more sensitive to small targets and is often used to evaluate fine structure segmentation in the field of biological images.

$$dice = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{5}$$

Among them, TP (True Positive) indicates the number of cell instances correctly identified by the model, FP (False Positive) indicates false positives, and FN (False Negative) indicates missed positives. We use the IoU matching strategy to match the predicted mask with the true mask one by one, and set a threshold (default is IoU > 0.5) to determine whether the match is established.

All metric results are computed based on the model's predicted masks and the corresponding groundtruth masks, and are recorded in .xlsx files for tracking and reproducibility. We use Python's matplotlib[32] library to visualize these metrics, including bar charts and box plots, which clearly illustrate the differences in average performance and inter-sample stability across models.

4.6 Quantification of cellular morphological and spatial metrics

All cellular images were acquired in TIFF format, accompanied by corresponding binary segmentation masks that delineate the boundaries of individual cells. Masks were converted to boolean arrays (where True values represented cellular regions and False values represented the background) to enable precise region-of-interest (ROI) analysis. Image preprocessing and metric calculation were performed using Python[33] with core libraries including scikit-image[34] for image processing, NumPy[35] for numerical computations, pandas[36] for data organization, and SciPy[37] for spatial distance calculations.

A custom CellMetricsCalculator class was developed to automate the extraction of six key cellular metrics, encompassing spatial arrangement, density, morphological complexity, and boundary integrity. For each image-mask pair, cellular regions were first labeled using connected-component analysis, and region properties were extracted to quantify area, perimeter, and centroid coordinates of individual cells. Detailed calculations for each metric are described below.

4.6.1 Mean Distance

Specifically, the average nearest-neighbor distance quantifies the mean spatial separation between each cell and its closest neighboring cell, reflecting the overall dispersion of cells within the field of view.

For all labeled cells, centroid coordinates (x_i, y_i) (where $i = 1, 2, \ldots, N$ and N is the total number of cells) were computed. A pairwise Euclidean distance matrix between all centroids was generated using the formula:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{6}$$

Where d_{ij} is the Euclidean distance between the centroids of cell i and cell j. Diagonal elements of the distance matrix (representing self-distance) were set to infinity, and the minimum distance for each cell ($d_{min,i} = \min(d_{i1}, d_{i2}, \ldots, d_{iN}, i \neq j)$) was identified. The average nearest-neighbor distance was calculated as the arithmetic mean of all $d_{\text{min},i}$:

$$MeanDistance = \frac{1}{N} \sum_{i=1}^{N} d_{min,i} \tag{7}$$

4.6.2 Compactness

Compactness characterizes the degree of cell packing by normalizing nearest-neighbor distance to cellular size, with higher values indicating looser packing and lower values indicating tighter packing.

Average cellular area $\bar{A}$ was computed as the mean area of all labeled cells:

$$\bar{A} = \frac{1}{N} \sum_{i=1}^{N} A_i \tag{8}$$

Where A_i is the area (in pixels^2) of cell i. Compactness was derived by normalizing the squared average nearest-neighbor distance (to account for area units) by the average cellular area:

$$\text{Compactness} = \frac{(\text{Mean Distance})^2}{\bar{A}} \tag{9}$$

4.6.3 Density

Cell density quantifies the number of cells per unit area, reflecting the overall confluency and population density of the cellular monolayer.

Total cellular area (A_{total}) was calculated as the sum of pixels in the binary mask (i.e., the total area occupied by all cells). Cell density was computed as the ratio of the total number of cells (N) to A_{total}:

$$Density = \frac{N}{A_{total}} \tag{10}$$

4.6.4 Distribution

Distribution assesses the heterogeneity of local cell density, with higher values indicating uneven spatial distribution (e.g., formation of cell clusters) and lower values indicating uniform dispersion.

For each cell i, Euclidean distances to all other cells were calculated, sorted, and the distances to the $k = 3$ nearest neighbors (excluding the cell itself) were summed ($D_{i,k} = \sum_{j=1}^{k} d_{min,i,j}$). Local density for cell i was defined as $\frac{k}{D_{i,k}}$ (inverse of the average distance to k neighbors). Population-level mean (μ_{local}) and standard deviation (σ_{local}) of local densities were computed. Clustering was derived as the coefficient of variation (CV) of local densities:

$$Distribution = \frac{\sigma_{local}}{\mu_{local}} \tag{11}$$

4.6.5 Shape Complexity

Shape Complexity describes the irregularity of cellular shapes, with higher values indicating more complex (e.g., branched, irregular) cell boundaries and lower values indicating simpler (e.g., circular, oval) shapes. This metric is derived from the relationship between cellular perimeter and area, adapted for population-level analysis.

For each cell i, perimeter (P_i) and area (A_i) were extracted from region properties. Population-level averages of squared perimeter ($\overline{P^2}$) and area ($\overline{A}$) were computed:

$$\overline{P^2} = \frac{1}{N}\sum_{i=1}^{N} P_i^2, \quad \overline{A} = \frac{1}{N}\sum_{i=1}^{N} A_i \tag{12}$$

Morphological complexity was defined as the ratio of $\overline{P^2}$ to $\overline{A}$:

$$\text{Morphological Complexity} = \frac{\overline{P^2}}{\overline{A}} \tag{13}$$

4.6.6 Edge Contrast

Edge Contrast quantifies the contrast between cellular edges and the background, reflecting the sharpness of cell boundaries in the original image.

RGB images (if applicable) were converted to grayscale via intensity averaging across color channels. A gradient magnitude image was generated using the Sobel operator (filters.sobel), which highlights edge regions by measuring intensity changes across the image. Gradient values were partitioned into two regions: Cell-edge gradient: Gradient values within the binary mask (i.e., edges of cellular regions), with mean intensity μ_{cell}. Background gradient: Gradient values outside the binary mask (i.e., background edges), with mean intensity μ_{bg}. Boundary clarity was computed as the ratio of μ_{cell} to μ_{bg}:

$$Edge\ Contrast = \frac{\mu_{cell}}{\mu_{bg}} \tag{14}$$

4.7 Downstream transcriptomic analysis

To assess the biological impact of improved segmentation, downstream transcriptomic analyses were performed using the Stereo Data Analysis Solution (SDAS, available at `https://github.com/STOmics/SDAS`). SDAS is a dedicated platform for advanced spatial transcriptomics analysis, providing modules for cell annotation, differential expression testing, and functional enrichment. The resulting cgef file output by the CSRefiner pipeline was converted to an AnnData object using Stereopy[38] before being input into SDAS.

4.7.1 Cell type annotation

Cell type annotation was performed with the cell2location[22] module within SDAS. For each segmented cell, annotation scores across all reference types were computed, and the cell was assigned to the type with the highest posterior probability. Annotation confidence was quantified by the distribution of assignment scores, and population-level shifts in score distributions between pre-trained and fine-tuned results were compared. Annotation outputs were visualized in spatial maps and used as the basis for downstream analyses.

4.7.2 Differential expression gene analysis

Differential expression gene (DEG) testing was conducted for each annotated cell type using the Wilcoxon rank-sum test implemented in SDAS. Genes with false discovery rate (FDR) < 0.05 and absolute log-fold-change > 1 were defined as significant DEGs. To capture global trends, we quantified the number of significant DEGs per cell type before and after fine-tuning, with particular focus on the six most enriched cell populations identified in the hippocampal region (e.g., Ext_Hpc_DG1, CA1–CA3 subfields). Comprehensive DEG tables, including both raw statistical results and filtered sets, were generated for reproducibility.

4.7.3 Functional enrichment analysis

Significant DEGs were further analyzed for functional relevance using GO Biological Process (2023) annotations via Enrichr[39]. Upregulated and downregulated gene sets were tested separately, and enrichment results were exported as tabular files and visualized through multiple modalities, including bubble plots (summarizing gene counts, rich factors, and adjusted significance), waterfall plots (depicting representative genes within top enriched pathways), and, where appropriate, heatmaps of gene expression patterns across GO terms. These analyses highlighted biological pathways more comprehensively recovered after fine-tuning, thereby linking improved segmentation to enhanced interpretability of spatial transcriptomic data.

5 Data availability

The two datasets that support the findings of this study have been deposited into CNGB Sequence Archive (CNSA) of the China National GeneBank DataBase (CNGBdb) with accession number CNP0007731: `https://db.cngb.org/search/project/CNP0007731/`. Additionally, we have uploaded these datasets to Zenodo[40].

6 Code availability

Project home page: `https://github.com/STOmics/cs-benchmark`

7 Authors' Contributions

Project administration and supervision: Mei Li, Ying Zhang

Algorithm development and implementation: Can Shi

Data provision: Yumei Li, Jing Guo, Qiuling Chen, Tingting Cao

Data selection and processing: Can Shi, Yumei Li, Ying Zhang

Data annotation: Can Shi

Code Testing: Can Shi

Manuscript writing and figure generation: Can Shi, Ying Zhang

Manuscript review: Can Shi, Mei Li, Ying Zhang, Sha Liao, Ao Chen

8 Competing interests

The authors declare they have no competing interests.

9 Acknowledgments

We thank the China National GeneBank for providing data storage support for this study and the National Key R&D Program of China (2022YFC3400400) for funding. We also thank Zhonghan Deng, Manqi Liang, Jiaxue Chen and Zhihan Ren for their assistance.

References

[1] Haoyang Li, Qinan Hu, Zhaowen Qiu, Hui Xiong, Yuhui Hu, and Xin Gao. Stp: single-cell partition for subcellular spatially-resolved transcriptomics. *Nature Communications*, 16(1):4665, 2025.

[2] Zhiyuan Yuan, Fangyuan Zhao, Senlin Lin, Yu Zhao, Jianhua Yao, Yan Cui, Xiao-Yong Zhang, and Yi Zhao. Benchmarking spatial clustering methods with spatially resolved transcriptomics data. *Nature Methods*, 21(4):712–722, 2024.

[3] Katy Vandereyken, Alejandro Sifrim, Bernard Thienpont, and Thierry Voet. Methods and applications for single-cell and spatial multi-omics. *Nature Reviews Genetics*, 24(8):494–515, 2023.

[4] Zaira Seferbekova, Artem Lomakin, Lucy R Yates, and Moritz Gerstung. Spatial biology of cancer evolution. *Nature Reviews Genetics*, 24(5):295–313, 2023.

[5] Jeffrey R Moffitt, Emma Lundberg, and Holger Heyn. The emerging landscape of spatial profiling technologies. *Nature Reviews Genetics*, 23(12):741–759, 2022.

[6] A Xiaowei. Method of the year 2020: Spatially resolved transcriptomics. *Nat. Methods*, 18(1):10–1038, 2021.

[7] Ao Chen, Sha Liao, Mengnan Cheng, Kailong Ma, Liang Wu, Yiwei Lai, Xiaojie Qiu, Jin Yang, Jiangshan Xu, Shijie Hao, et al. Spatiotemporal transcriptomic atlas of mouse organogenesis using dna nanoball-patterned arrays. *Cell*, 185(10):1777–1792, 2022.

[8] Chun-Seok Cho, Jingyue Xi, Yichen Si, Sung-Rye Park, Jer-En Hsu, Myungjin Kim, Goo Jun, Hyun Min Kang, and Jun Hee Lee. Microscopic examination of spatial transcriptome using seq-scope. *Cell*, 184(13):3559–3572, 2021.

[9] Raimo Hartmann, Hannah Jeckel, Eric Jelli, Praveen K Singh, Sanika Vaidya, Miriam Bayer, Daniel KH Rode, Lucia Vidakovic, Francisco Díaz-Pascual, Jiunn CN Fong, et al. Quantitative image analysis of microbial communities with biofilmq. *Nature microbiology*, 6(2):151–156, 2021.

[10] Yaron M Sigal, Ruobo Zhou, and Xiaowei Zhuang. Visualizing and discovering cellular structures with super-resolution microscopy. *Science*, 361(6405):880–887, 2018.

[11] Van K Lam, Jeff M Byers, Michael C Robitaille, Logan Kaler, Joseph A Christodoulides, and Marc P Raphael. A self-supervised learning approach for high throughput and high content cell segmentation. *Communications Biology*, 8(1):780, 2025.

[12] Tatsuya Nobori, Alexander Monell, Travis A Lee, Yuka Sakata, Shoma Shirahama, Jingtian Zhou, Joseph R Nery, Akira Mine, and Joseph R Ecker. A rare primer cell state in plant immunity. *Nature*, 638(8049):197–205, 2025.

[13] Jun Ma, Ronald Xie, Shamini Ayyadhury, Cheng Ge, Anubha Gupta, Ritu Gupta, Song Gu, Yao Zhang, Gihun Lee, Joonkee Kim, et al. The multimodality cell segmentation challenge: toward universal solutions. *Nature methods*, 21(6):1103–1113, 2024.

[14] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature methods*, 18(1):100–106, 2021.

[15] Marius Pachitariu and Carsen Stringer. Cellpose 2.0: how to train your own model. *Nature methods*, 19(12):1634–1641, 2022.

[16] Carsen Stringer and Marius Pachitariu. Cellpose3: one-click image restoration for improved cellular segmentation. *Nature methods*, 22(3):592–599, 2025.

[17] Marius Pachitariu, Michael Rariden, and Carsen Stringer. Cellpose-sam: superhuman generalization for cellular segmentation. *bioRxiv*, pages 2025–04, 2025.

[18] Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers. Cell detection with star-convex polygons. In *International conference on medical image computing and computer-assisted intervention*, pages 265–273. Springer, 2018.

[19] Mei Li, Huanlin Liu, Qiang Kang, Shuangsang Fang, Min Li, Jiajun Zhang, Fei Teng, Dan Wang, Weixuan Cen, Zepeng Li, et al. Cellbin: a highly accurate single-cell gene expression processing pipeline for high-resolution spatial transcriptomics. *Biorxiv*, pages 2023–02, 2023.

[20] Can Shi, Jinghong Fan, Zhonghan Deng, Huanlin Liu, Qiang Kang, Yumei Li, Jing Guo, Jingwen Wang, Jinjiang Gong, Sha Liao, et al. Cellbindb: a large-scale multimodal annotated dataset for cell segmentation with benchmarking of universal models. *GigaScience*, 14:giaf069, 2025.

[21] Gihun Lee, SangMook Kim, Joonkee Kim, and Se-Young. Yun. Mediar: Harmony of data-centric and model-centric for multi-modality microscopy. In Jun Ma, Ronald Xie, Anubha Gupta, José Guilherme de Almeida, Gary D. Bader, and Bo Wang, editors, *Proceedings of The Cell Segmentation Challenge in Multi-modality High-Resolution Microscopy Images*, volume 212 of *Proceedings of Machine Learning Research*, pages 1–16. PMLR, 28 Nov–09 Dec 2023.

[22] Vitalii Kleshchevnikov, Artem Shmatko, Emma Dann, Alexander Aivazidis, Hamish W King, Tong Li, Rasa Elmentaite, Artem Lomakin, Veronika Kedlian, Adam Gayoso, et al. Cell2location maps fine-grained cell types in spatial transcriptomics. *Nature biotechnology*, 40(5):661–671, 2022.

[23] Gene Ontology Consortium. Expansion of the gene ontology knowledgebase and resources. *Nucleic acids research*, 45(D1):D331–D338, 2017.

[24] Gene Ontology Consortium. The gene ontology resource: 20 years and still going strong. *Nucleic acids research*, 47(D1):D330–D338, 2019.

[25] Vasiliy V Reshetnikov, Polina E Kisaretova, Nikita I Ershov, Anastasia S Shulyupova, Dmitry Yu Oshchepkov, Natalia V Klimova, Anna V Ivanchihina, Tatiana I Merkulova, and Natalia P Bondar. Genes associated with cognitive performance in the morris water maze: an rna-seq study. *Scientific Reports*, 10(1):22078, 2020.

[26] John Lisman, Ryohei Yasuda, and Sridhar Raghavachari. Mechanisms of camkii action in long-term potentiation. *Nature reviews neuroscience*, 13(3):169–182, 2012.

[27] Keith K Murai, Louis N Nguyen, Fumitoshi Irie, Yu Yamaguchi, and Elena B Pasquale. Control of hippocampal dendritic spine morphology through ephrin-a3/epha4 signaling. *Nature neuroscience*, 6(2):153–160, 2003.

[28] Kelly A Jones, Deepak P Srivastava, John A Allen, Ryan T Strachan, Bryan L Roth, and Peter Penzes. Rapid modulation of spine morphology by the 5-ht2a serotonin receptor through kalirin-7 signaling. *Proceedings of the National Academy of Sciences*, 106(46):19575–19580, 2009.

[29] Yu Zhao, Young Li, Ying He, Junqi Wu, Yi Liu, Xinxing Li, Zhaoxun Li, Qiaomei Yuan, Jialuo Li, Xinya Zhang, et al. Stereo-seq v2: Spatial mapping of total rna on ffpe sections with high resolution. *Cell*, 2025.

[30] Chun Gong, Shengkang Li, Leying Wang, Fuxiang Zhao, Shuangsang Fang, Dong Yuan, Zijian Zhao, Qiqi He, Mei Li, Weiqing Liu, et al. Saw: an efficient and accurate data analysis workflow for stereo-seq spatial transcriptomics. *GigaByte*, 2024:1–12, 2024.

[31] Alexander Dobin, Carrie A Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R Gingeras. Star: ultrafast universal rna-seq aligner. *Bioinformatics*, 29(1):15–21, 2013.

[32] John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(03):90–95, 2007.

[33] Guido VanRossum and Fred L Drake. *The python language reference*, volume 561. Python Software Foundation Amsterdam, The Netherlands, 2010.

[34] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014.

[35] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *nature*, 585(7825):357–362, 2020.

[36] Wes McKinney et al. Data structures for statistical computing in python. *scipy*, 445(1):51–56, 2010.

[37] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

[38] Shuangsang Fang, Mengyang Xu, Lei Cao, Xiaobin Liu, Marija Bezulj, Liwei Tan, Zhiyuan Yuan, Yao Li, Tianyi Xia, Longyu Guo, et al. Stereopy: modeling comparative and spatiotemporal cellular heterogeneity via multi-sample spatial transcriptomics. *Nature Communications*, 16(1):3741, 2025.

[39] Maxim V Kuleshov, Matthew R Jones, Andrew D Rouillard, Nicolas F Fernandez, Qiaonan Duan, Zichen Wang, Simon Koplev, Sherry L Jenkins, Kathleen M Jagodnik, Alexander Lachmann, et al. Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. *Nucleic acids research*, 44(W1):W90–W97, 2016.

[40] Can Shi, Yumei Li, Jing Guo, Qiuling Chen, Tingting Cao, Sha Liao, Ao Chen, Mei Li, and Ying Zhang. Csrefiner: A lightweight framework for fine-tuning cell segmentation models with small datasets, 2025.

10 Figures

Description for Figure 1:
(A) Representative segmentation results from four pre-trained models (Cellpose-cyto, Cellpose-cpsam, StarDist-2D_versatile_fluo, and CellBin) for FFPE mouse brain tissue with DAPI-stained. Four regions are shown, including hippocampal and non-hippocampal areas. Red contours indicate ground-truth manual annotations, and yellow contours indicate model-predicted segmentation boundaries, the corresponding F1 scores are labeled for quantification.
(B) Schematic overview of the CSRefiner workflow, including training set preparation, model selection and fine-tuning, performance evaluation, and downstream generation of CellBin GEF files.
(C) Segmentation results after fine-tuning with CSRefiner, corresponding to the regions shown in panel A.

Description for Figure 2:
(A–E) Quantitative evaluation of segmentation performance before and after fine-tuning across four representative models (Cellpose-cyto, Cellpose-cpsam, StarDist, and CellBin). Boxplots show improvements in (A) precision, (B) recall, (C) F1 score, (D) Jaccard index, and (E) Dice coefficient. The significance mark line and P value are added to the box plot. The number of "*" from 1 to 3 represents a P value less than 0.05, 0.01, and 0.001.
(F) Time required for manual whole-slide annotation ($\tilde{1}0$ days) versus CSRefiner-assisted workflow ($\tilde{4}00$ minutes).
(G) Spatial maps of cell type annotations generated by cell2location using cgef matrices from the fine-tuned StarDist model.
(H) Numbers of Top 6 significantly upregulated genes across major cell types before and after fine-tuning.
(I–J) Gene ontology enrichment analysis results for Ext_Hpc_DG1 before fine-tuning.
(K–L) Gene ontology enrichment analysis results for Ext_Hpc_DG1 after fine-tuning.
(M–N) Visualization of segmented and annotated cells in hippocampal subregions before and after fine-tuning.
(O) Distribution of annotation scores.
(P) Distribution of cell areas.
(Q) Cell counts of hippocampal subtypes.
(R) Numbers of detected genes per cell.

Description for Figure 3:
(A) Model performance on the hippocampus (not included in training) after fine-tuning the model using different numbers of annotated cells. The line graph on the right visualizes precision, recall, and F1 score for different numbers of training cells.
(B) Same as panel A, model performance on non-hippocampal regions (not included in training).
(C) Quantitative comparison of regional structural complexity between hippocampal and non-hippocampal images. Metrics include mean distance between cells, compactness, density, distribution, shape complexity, and edge contrast.

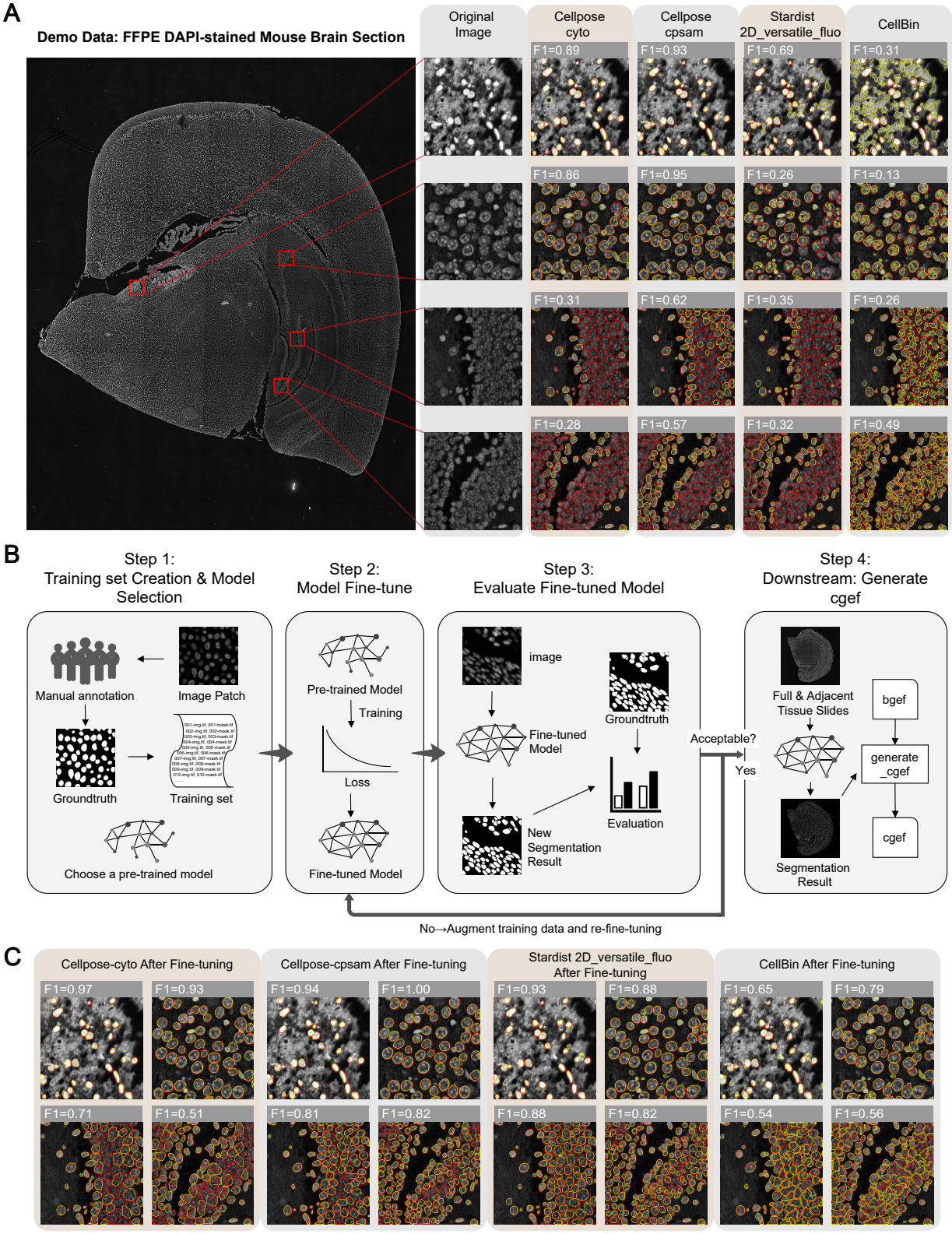

Figure 1: **Performance of pre-trained and fine-tuned segmentation models on FFPE DAPI-stained mouse brain section, and schematic workflow of CSRefiner.**

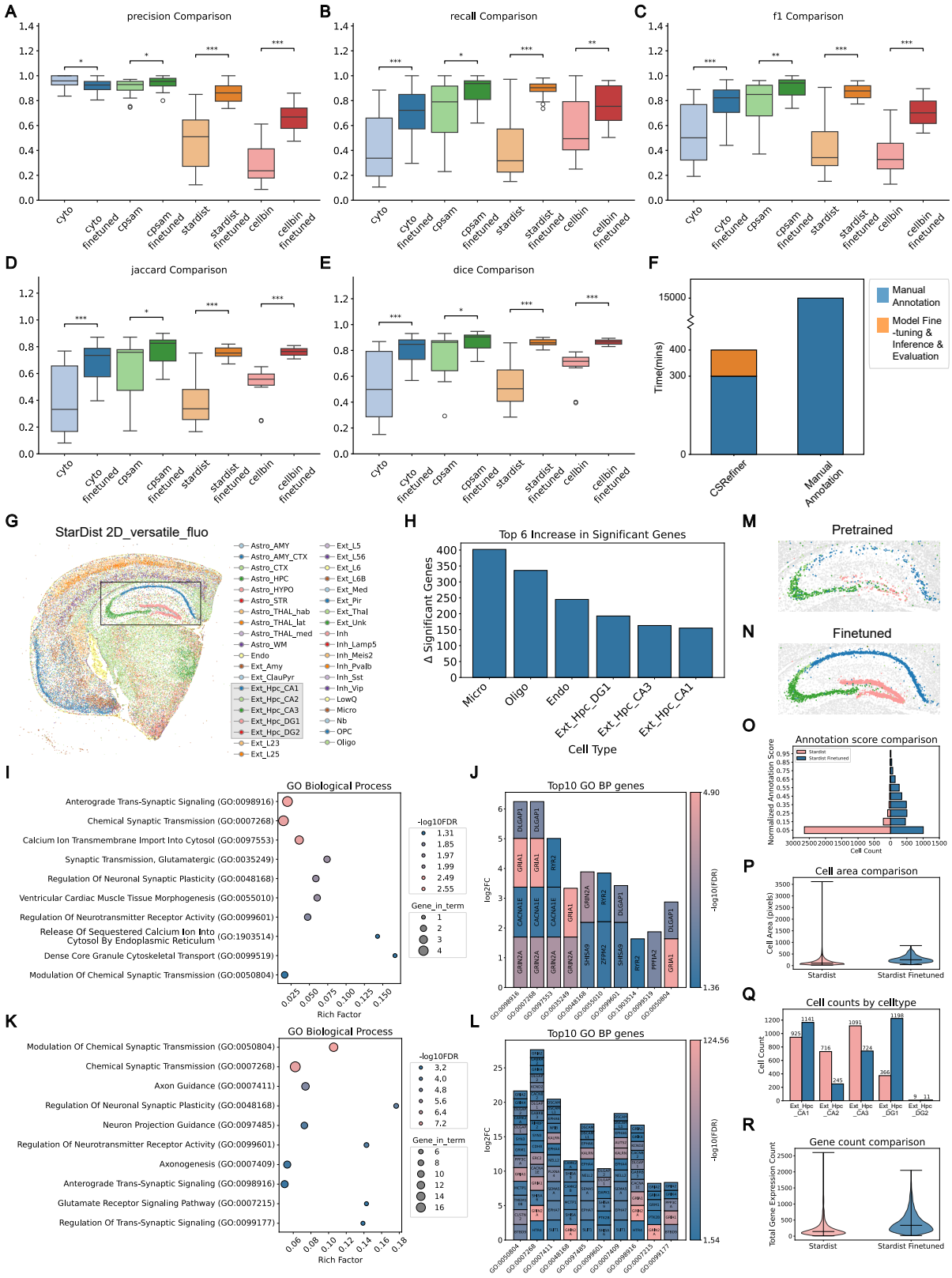

Figure 2: **Segmentation performance and downstream biological analyses improved by CSRefiner.**

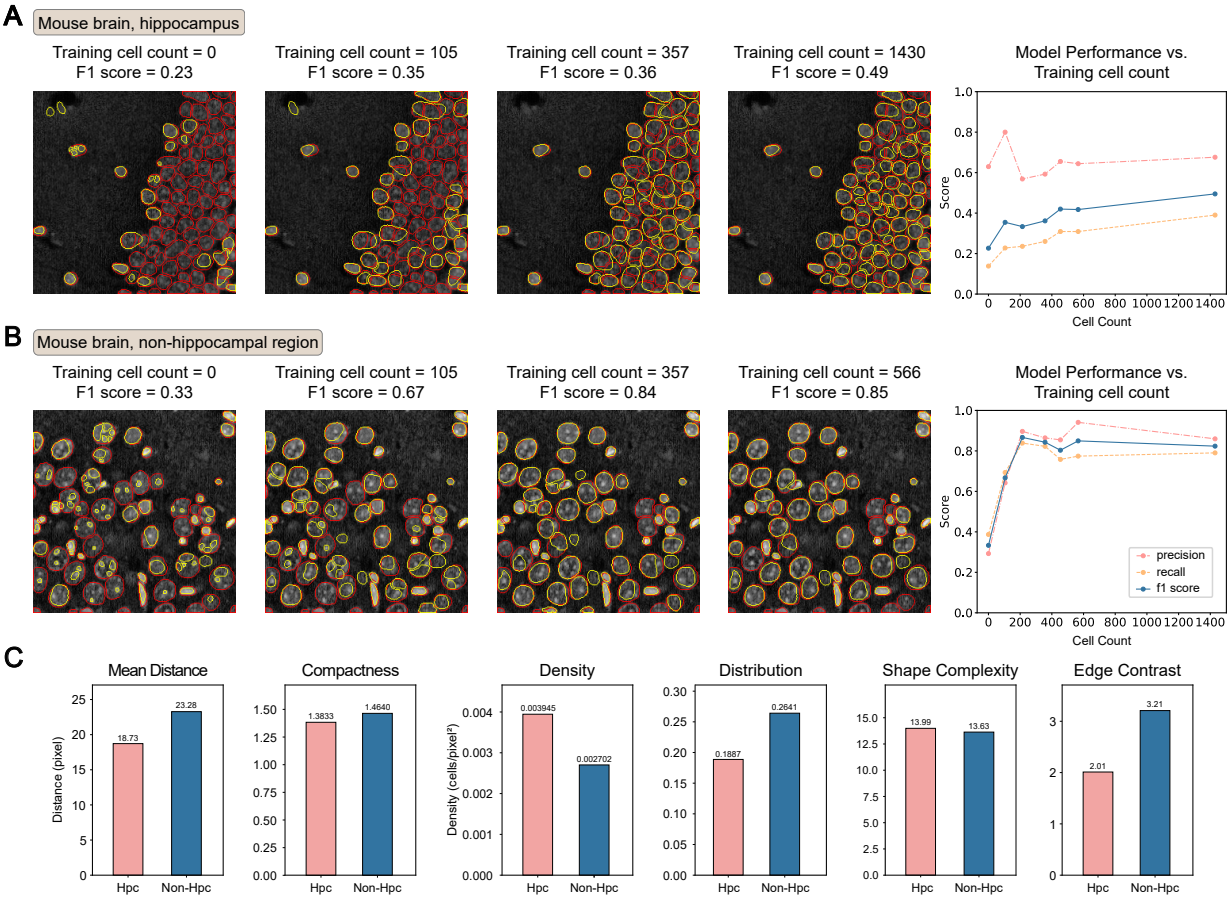

Figure 3: **Training set size and tissue complexity influence segmentation performance.**