SEGMENTATION TOOLS
March 15, 2024

Cell SEGMENTATION Tools: Pros and Cons

March 15, 2024

Contents

1 Summary

The research initially focused on the Cellpose library, spurred by prior correspondences and dataset acquisitions. However, as the project progressed, the team expanded its scope to include exploration and development of other tools within the field. Throughout this process, tools such as Cellpose (Versions 2 and 3), SSAM, TIA Toolbox, SVETLANA, among others, were investigated. Following this phase, serving as a proof of concept (POC) research and development stage, there is now a necessity to provide precise answers to certain questions, eliminating ambiguity and clarifying subsequent stages and phases of the project. Notably, a team of six individuals, led by Dr. Lashgari and comprising experts in computer science and neuroscience, has been formed. They have initiated work and are striving to progress towards a cohesive research and development phase with a structured approach. Furthermore, efforts have been made to ensure that all implementation stages are visible in a Git repository, accessible at: `https://github.com/msmsadegh/Cell-Counting-Cell-Segmentation/tree/main`. However, it is important to mention that, due to data protection measures, the datasets will not be hosted on this platform.

2 Current Questions

1. What type of image format does the network require as input?
2. Are there only three methods of image acquisition, or will you be introducing additional types of cells during our research?
3. Could you provide images with cells manually annotated for validating our model's predictions against these reference images?
4. Is there a requirement to classify cells in addition to segmenting them?
5. Why is cell segmentation important, and what steps follow this process?

3 Introduction

Cell segmentation, a critical step in image analysis, plays a pivotal role, particularly in the field of microscopy. In this comprehensive report, we delve into the methodologies, algorithms, and technologies that drive cell segmentation tools, aiming to uncover advancements, challenges, and applications across diverse scenarios. We navigate through the intricacies of data augmentation, preprocessing strategies, and deep learning architectures, seeking insights into how these tools compare to traditional image-processing techniques. Additionally, we explore the training strategies employed, the evaluation metrics used for performance assessment, and the challenges and limitations encountered in the realm of cell segmentation. Our journey extends to the integration of cell segmentation tools with other technologies and examines the landscape of open-source tools and datasets. By the end of this exploration, we aim to provide a nuanced understanding of the state-of-the-art in cell segmentation and the accessibility and reproducibility of proposed methodologies.

As an example of the tools we have researched, we have explored TIA-Toolbox, Cellpose (Versions 2 and 3), SSAM, SVETLANA, among others. Ultimately, we intend to draw conclusions on whether it is better to utilize existing tools or even implement a neural network tailored to specific needs, contingent upon the responses provided to us. [2]

4 Exploring Cell Segmentation Tools

4.1 Cellpose

The initial Cellpose model was developed by Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu as mentioned in [3]. Cellpose is an open-source, generalist algorithm for cellular

segmentation that can be used on various types of cells from different culture lines. The authors have made the code available on GitHub[1].

The latest version of Cellpose, known as Cellpose 3, has endeavored to address deficiencies such as noise, blurred images, and undersampling. This effort aims to enhance its analysis and modeling capabilities.

- **Pros:**
  - Pre-trained models available, facilitating easier implementation.
  - User-friendly interface, making it accessible to a broad user base.
- **Cons:**
  - Limited ability to segment multiple objects in one run may require additional processing steps.

4.2 CellProfiler

CellProfiler is a widely used open-source image analysis software designed specifically for biological imaging applications, including cell segmentation and pose estimation. It includes various built-in modules for common tasks such as background subtraction, particle analysis, and feature extraction.

Pros:

- User-friendly interface
- Extensive documentation
- Large community of users

Cons:

- May not be as flexible or customizable as other libraries for advanced applications.

4.3 Svetlana

- **Pros:** Open-source plugin for Napari, supporting manual and automatic segmentation.
- **Cons:** Lack of detailed limitations in the provided source.

4.3.1 Project Description and Installation:

The `napari_svetlana` plugin aims to classify segmentation algorithm outputs, processing 2D, 3D, and multichannel images. Installation involves setting up a Python 3.9 Conda environment, activating it, and installing via pip or Napari plugin manager. Note that manual installation steps may be required for Cupy and Cudatoolkit if using a Cuda compatible GPU, especially on Windows.

4.3.2 Tutorial and Similar Plugins:

Svetlana offers advanced features like data augmentation and contextual information reduction. Refer to the YouTube tutorial and documentation for guidance. Similar plugins include Joel Luethi's `napari feature classifier` and Robert Haase's `apoc` for pixels and objects classification.

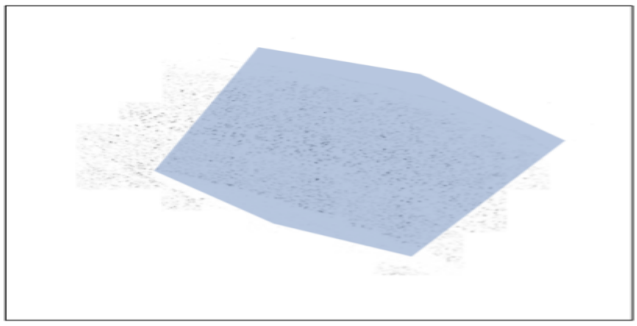
[1]

[1] `https://github.com/carsenstringer/cellpose`

4.4 SSAM

Launching the SSAM project requires the utilization of the Conda environment on the Linux operating system, with activation of Python version 3 and R within the Conda environment. However, specific version details and download procedures have not been provided.

In SSAM, primary visual cortex (VISp) data from mice, multiplexed using smFISH, is employed. Initially, data is downloaded and unpacked. Subsequently, the mRNA point table is loaded, where each row delineates an mRNA point, with columns containing coordinates and target genes. Necessary columns are loaded into a dataframe.



If the dataset is in a different format, it necessitates reshaping and conversion to a similar format before proceeding.

Since SSAM analysis operates at the cellular level, coordinates are converted from the laboratory system to micrometers.

To instantiate an SSAMDataset object, four arguments are required:

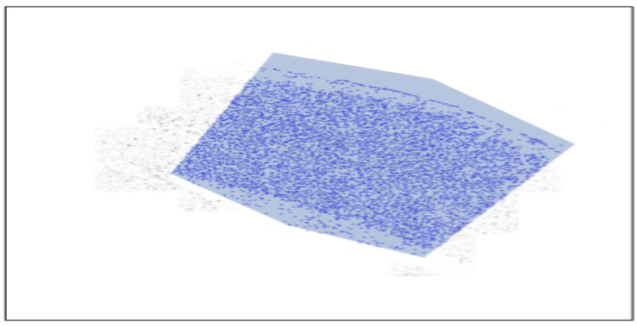
- A list of gene names present in the experiment
- Lists containing coordinates of each gene: coordlist
- Image width
- Image height

Width and height are computable from the image dimensions. Data is grouped based on genes, creating a gene names list, and obtaining coordinate lists for each gene type.

Subsequently, the SSAMDataset object is imported. Following data loading, SSAM converts discrete mRNA locations to mRNA density using kernel density estimation.

With the SSAMDataset object `ds`, SSAMAanalysis analysis is initialized.

For selective analysis of a sample portion, MASK can be utilized, restricting image parts for local maximum sampling (input_mask) and confining SSAM cell type map generation to specific regions (output_mask).



Local maxima aid in determining stabilization parameters, image variance, and cluster identification. In this section, local maxima contribute to variance stabilization. The `find_localmax` function identifies local density maxima, with gene expression thresholds of 0.027 for individual genes and 0.2 for total gene expression.

Following local maxima identification, they can be plotted. In cases where many local maxima lie outside tissue regions, the k-NN method filters out "wandering" local maxima. However, as mask segmentation is employed in this instance, this issue is mitigated. Identified local maxima are then utilized to compute stabilization and variance parameters using `sctransform`.

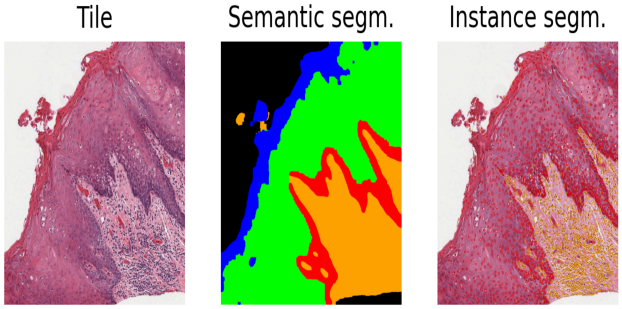
Output generation for this stage was successfully achieved.

An error occurred during the normalization stage, currently under troubleshooting. The error may originate from either the R language or the Ubuntu 22.04 Linux operating system being used.

Experienced users of the R package are invited to offer guidance, as despite confirmation of program installation, the requisite output for this stage may not be accurately produced.

4.5 TIA-Toolbox

The TIA Toolbox utilizes two methods, semantic segmentation and instance segmentation. Semantic segmentation is valuable for identifying different types of nuclei and labeling them accordingly, providing insightful information for our project. On the other hand, instance segmentation can be useful in different layers of the project, offering versatility and adaptability. Additionally, the TIA Toolbox has demonstrated good accuracy in its performance, further highlighting its potential usefulness for our project.



5 Recent Developments and Key Results

- One notable finding in our research journey was the discovery of the "Segment Anything" model by Meta, which presents a promising approach to segmentation tasks. While we have encountered several projects related to cell segmentation using this model, further experimentation is needed to assess its alignment with our specific requirements. This model is available at segment-anything.com.

- Another tool that caught our attention is the TIA Toolbox, which primarily focuses on the segmentation of cancer cells. However, it is recommended to delve into its capabilities by inspecting its code and documentation for better understanding. The documentation for the TIA Toolbox is accessible at tia-toolbox.readthedocs.io.

- Additionally, PyTorch offers several useful libraries for segmentation tasks, which will be explored in the next stages, as familiarity and utilization of these libraries can significantly aid our work. Tutorials and documentation for these libraries can be found at pytorch.org.

- Models for semantic segmentation, suitable for labeling projects, are available and could be beneficial for our purposes. These models can be explored at paperswithcode.com.

- Our team's GitHub repository serves as a central hub for collaboration, where over time and across different branches, the project will be completed. You can access the repository at github.com.

References

[1] Clément Cazorla, Renaud Morin, and Pierre Weiss. "Svetlana: a Supervised Segmentation Classifier for Napari". In: (2023). FFHAL-03927879F. URL: https://hal.inria.fr/hal-03927879.

[2] "Segmentation Software Usability and Performance". In: *WayScienceLab Journal* (2022). URL: https://www.waysciencelab.com/2022/11/01/segmentation.html.

[3] Carsen Stringer et al. "Cellpose: a generalist algorithm for cellular segmentation". In: *Nature Methods* 18 (2021), pp. 100–106. DOI: 10.1038/s41592-020-01018-x.