



臺大醫學院研發分處 第一共同研究室顯微影像核心

IMAGEJ顯微影像分析 與程式設計

零基礎的學生也能掌握基本顯微影像分析能力



海報網址



報名網址

2025 3.3-4.28 周一 13:30-14:30 共7堂
影像前處理、AI應用、自動化分析

課程資訊及授課教師

2025/3/3(一) 【生物影像分析概論】
溫榮峯 中央研究院 生化所 生物影像核心設施
研究助師

2025/3/10(一) 【生物影像流程與小組討論編組】
許紹君 臺灣大學分子影像重點技術平台
助研究專家

2025/3/17(一) 【影像分析自動化】
張仁乾 日本理化學研究所
專門技術員

2025/3/24(一) 【互動式影像分析流程建立】
朱韋臣 中央研究院 細生所 公共儀器室影像組
專案研發學者

2025/3/31(一) 【物件追蹤分析】
黃紀穎 中央研究院 植微所 細胞核心實驗室光學顯微鏡組
專案研究人員

2025/4/7(一) 【AI: 機器學習與深度學習工具介紹】
羅安琦 臺灣大學分子影像重點技術平台
副技師

2025/4/28(一) 小組發表
許紹君 臺灣大學分子影像重點技術平台 助研究專家
朱韋臣 中央研究院 細生所共儀像組 專案研發學者

主辦單位：臺大醫學院研發分處 第一共同研究室顯微影像核心
協辦單位：中央研究院 生物化學研究所
地點：基醫大樓講堂區 5 樓 未來教室 (原508教室)

課程簡介

本課程將介紹生物影像的基本元素、如何利用FIJI進行影像前處理、影像切割、特徵萃取、程式設計與編程、互動式影像分析流程與GPU加速、AI(機器學習與深度學習工具)、物件追蹤、常用的資料庫以及如何分享自己的作品。將視報名人數進行小組發表與討論，利用工作中學習的方式提升課程效果。

課程目標

希望零基礎的學生參與課程後，都能具備基本分析顯微影像的能力。

上課須知

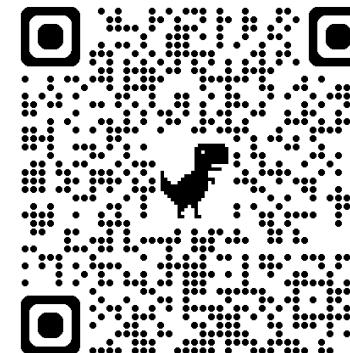
- 即日起開放報名，報名方式如下：
○ 提供姓名、EMAIL、任職/就學單位、實驗室主持人姓名。
○ 以一張A4篇幅文字說明實驗目的與欲解決的問題，並以一張投影片頁面作為輔助材料。
- 優先錄取自備影像分析問題並想透過課程學會如何自己解決問題者。
- 課程會同步紀錄影音並於課後上傳至教學影音平台。
- 每堂課皆會點名，上課出勤不得缺課超過一堂。
- 需自備筆電。

招生人數：實體招收24人，線上30人。
報名截止日：額滿為止，恕不開放現場候補。
聯絡人：第一共同研究室顯微影像核心 林思廷 szuting@ntu.edu.tw

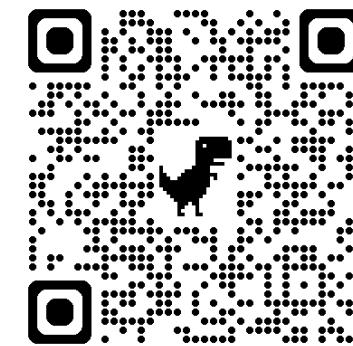


上課注意事項：

- 教室內禁止攜帶食物飲料入內，僅允許“白開水”，請大家將食物飲料放置於教室外的桌上。
- 請實體與線上學員掃描以下QR code進行線上簽到。
- 請線上學員於課程開始前關閉自己的麥克風。
- 線上學員若有問題，請先按下“舉手”，或於聊天室寫下問題，將於課程結束後在場地時間允許下，安排QA時間。
- 現場學員發問時請使用麥克風才可進行收音。



線上簽到



課程材料與資訊連結

ImageJ 顯微影像分析與程式設計-互動式影像分析流程建立

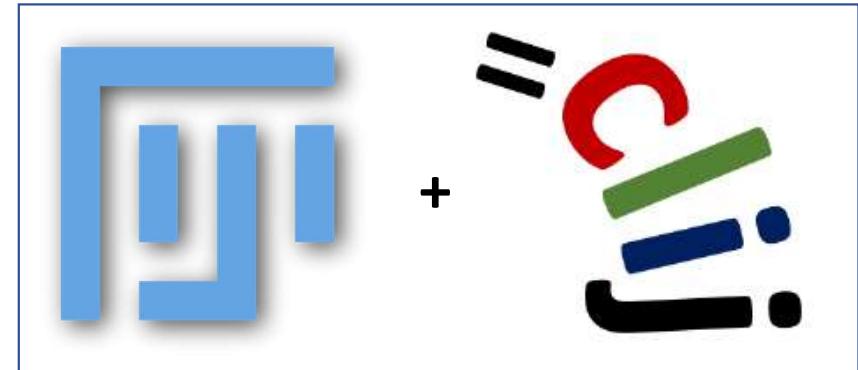
Interactive Bioimage Analysis and GPU-Accelerated Code Generation with CLIJ

Wei-Chen CHU (朱韋臣)

R&D Scientist

ICOB Imaging Core, Academia Sinica, Taiwan

EABIAS (East Asia Bioimage Analysts' Society)
<https://eabias.github.io/>



This document is licensed by Wei-Chen CHU under [CC-BY 4.0 license](#) unless mentioned otherwise



@weichen01.bsky.social

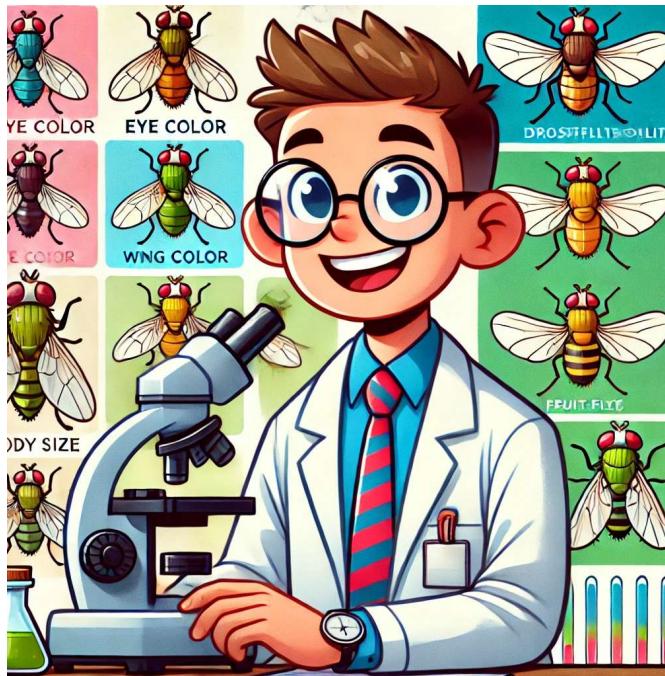


Wei-Chen Chu
ICOB Imaging Core

My background



Undergraduate ~ Graduate student
@Dept. Life sciences, NCHU



Ph.D. @IMB, Academia Sinica
PostDoc @RIKEN BDR, Japan



R&D Scientist
@ICOB imaging Core, Academia Sinica



@weichen01.bsky.social

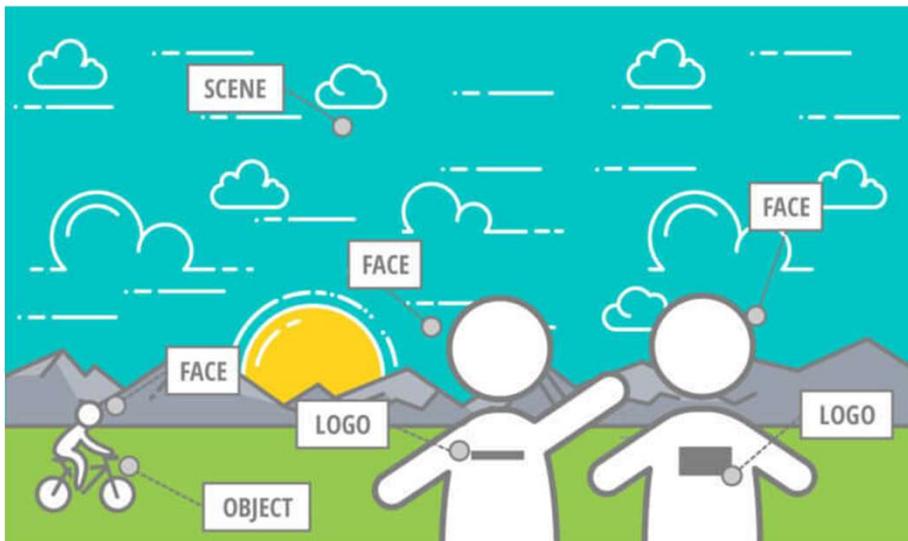


Wei-Chen Chu
ICOB Imaging Core

BioImage Analysis?

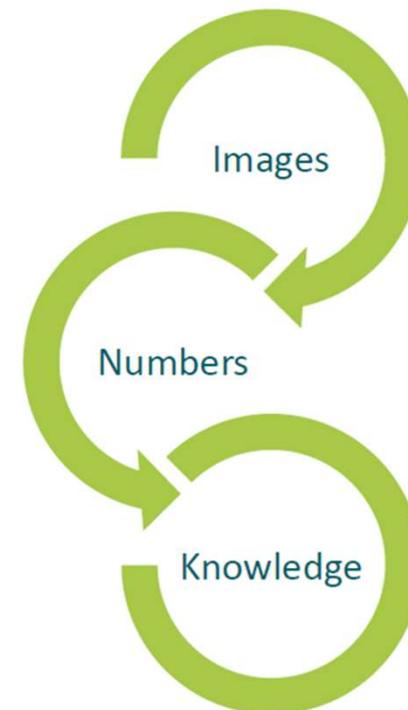
Image Analysis?

Image analysis (also known as “computer vision” or image recognition) is the ability of computers to **recognize attributes** within an image.



BioImage Analysis?

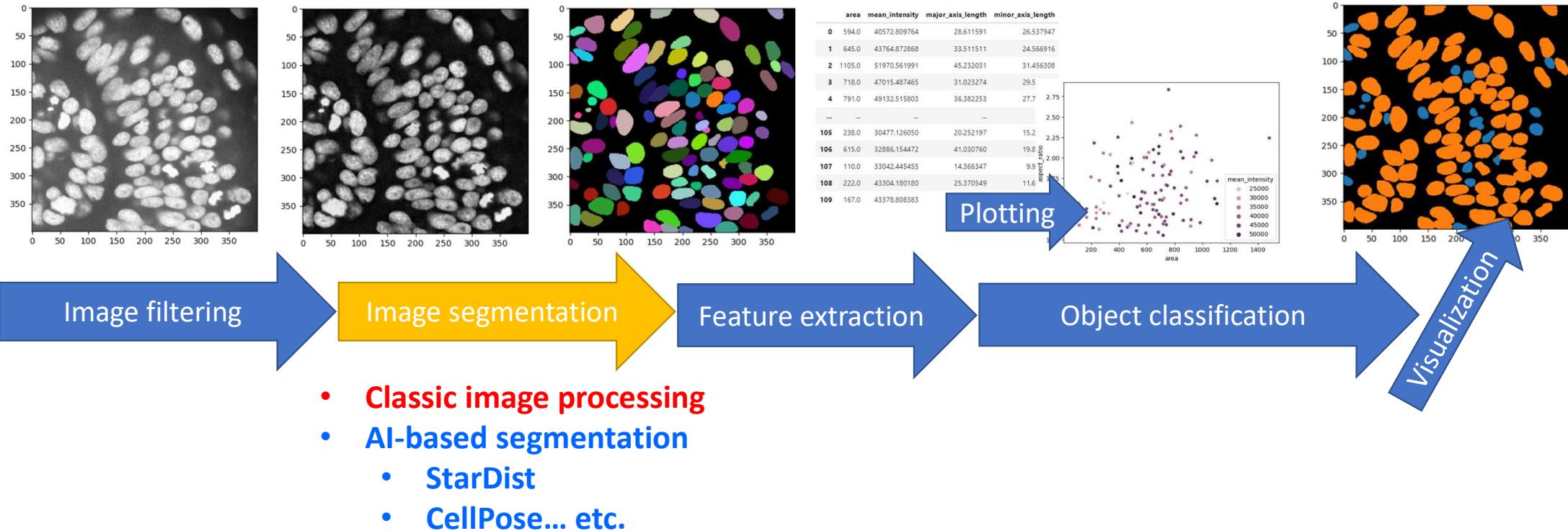
Understanding and quantifying microscopy, medical or any other calibrated image data.



- Objective
- Reliable
- Repeatable
- Reproducible
- Replicable

Bioimage Analysis Workflow

- Goal: Quantify observations, substantiate conclusions with numbers



Modified from [Open-Access Training Materials of Dr. Robert Haase](#), licensed CC-BY 4.0

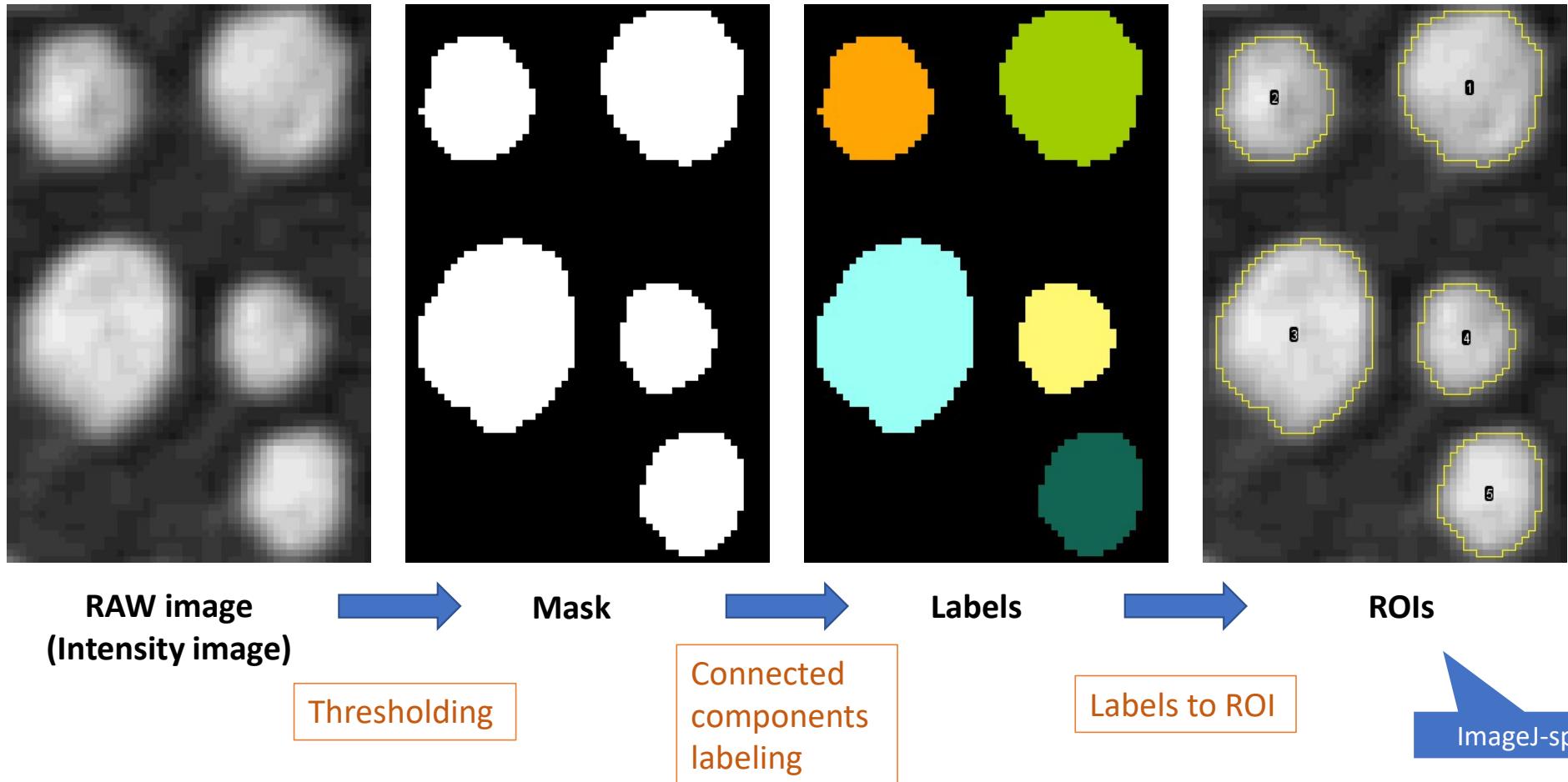


@weichen01.bsky.social

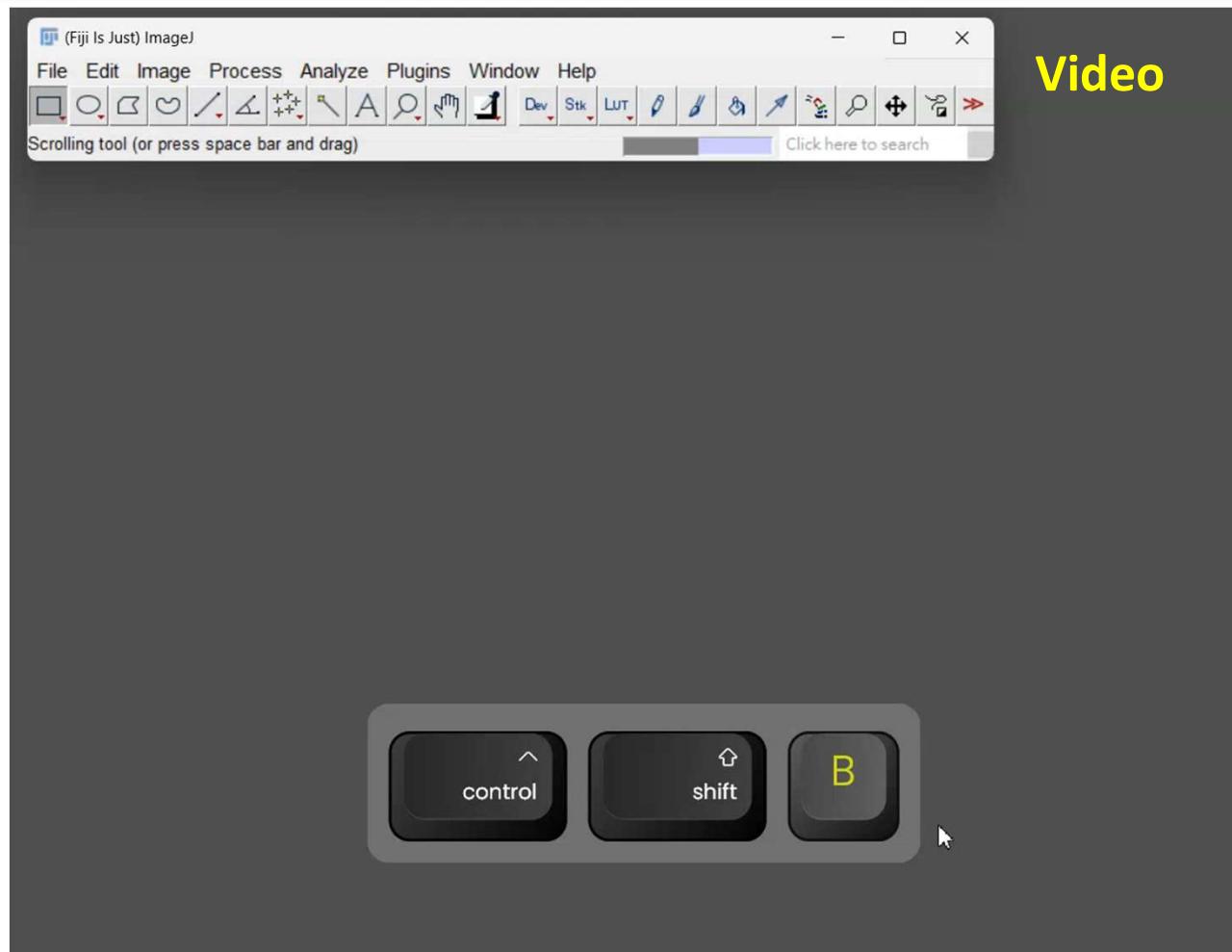
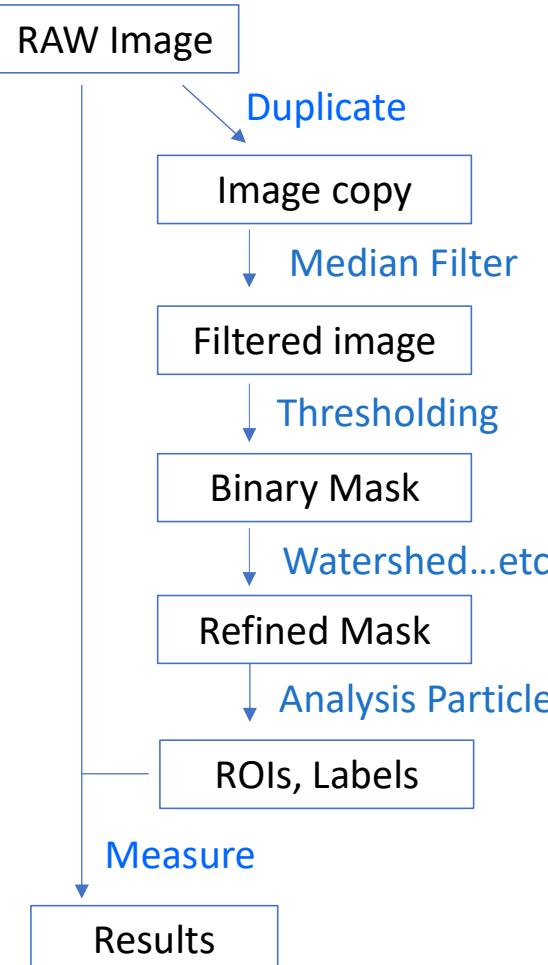


Wei-Chen Chu
ICOB Imaging Core

Segmentation: Mask, Label, ROI



Recap: Classic workflow in FIJI/ImageJ

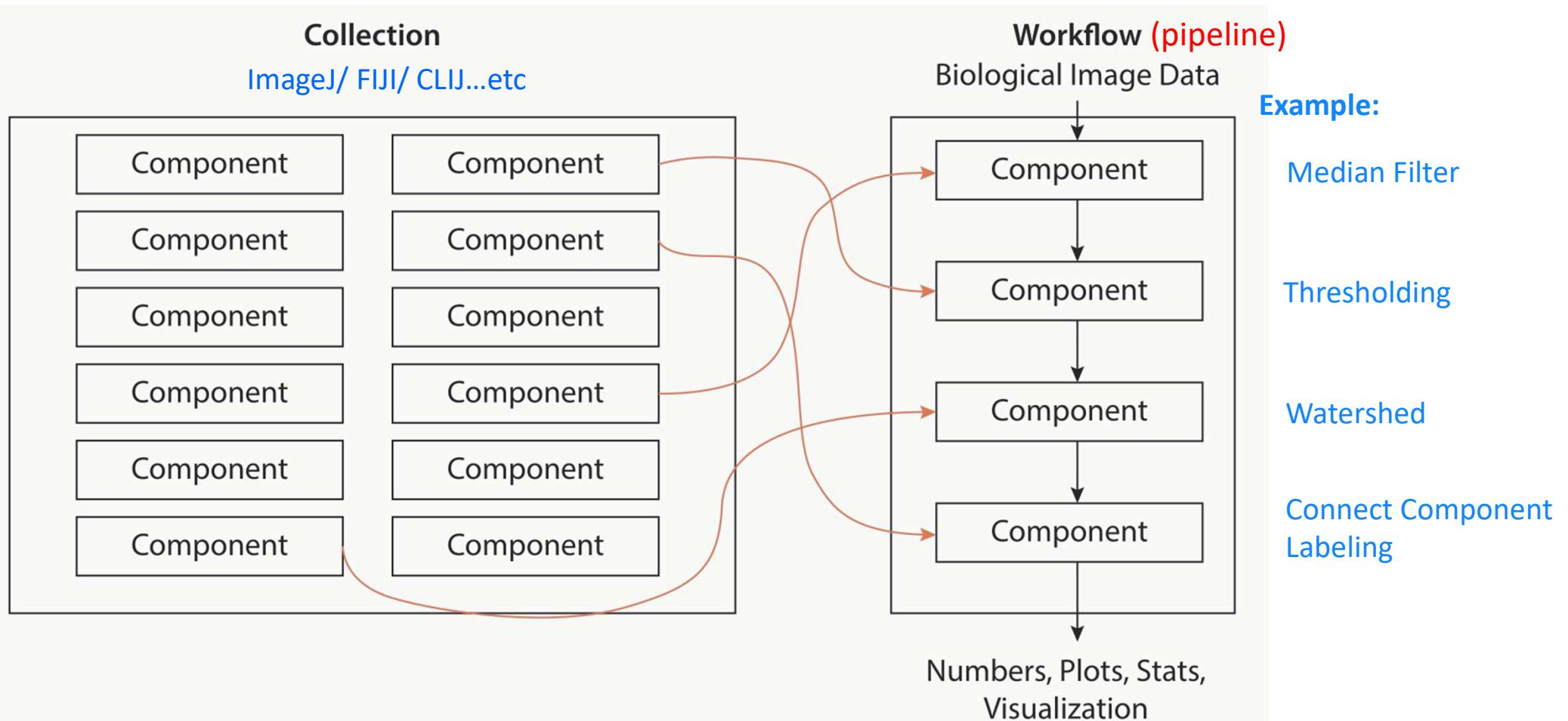


@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

Software -> components, collection and workflow



Modified from Kota Miura and Nataša Sladoje (2019) Bioimage Data Analysis Workflows,
licensed [CC-BY 4.0](#)

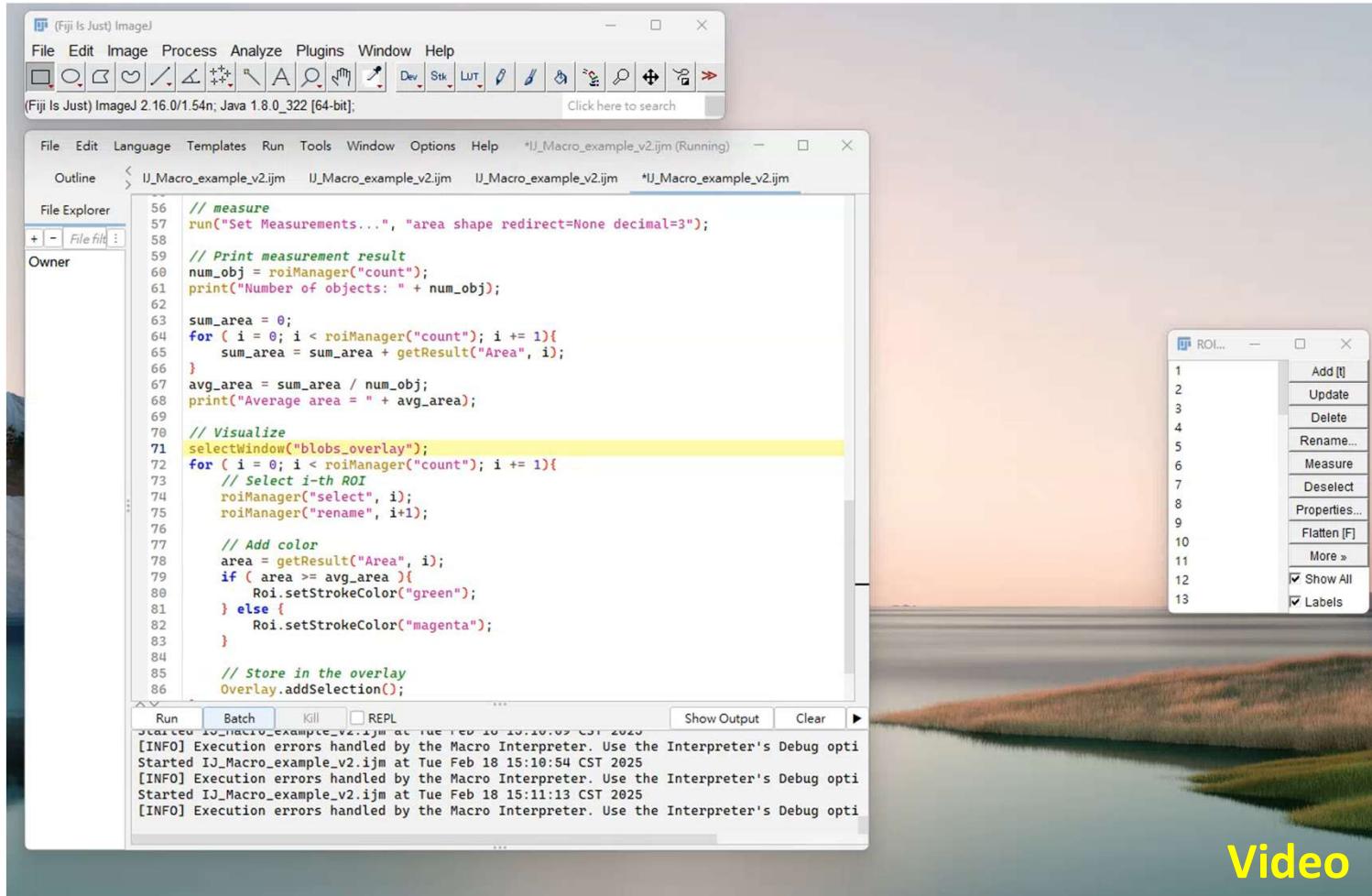


@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

ImageJ macro allows other people reproduce your workflow and results



Video

ImageJ macro:
Great documentation for
your bioimage analysis
workflow!

ImageJ Macro modified from Dr. Jen-Chien Chang

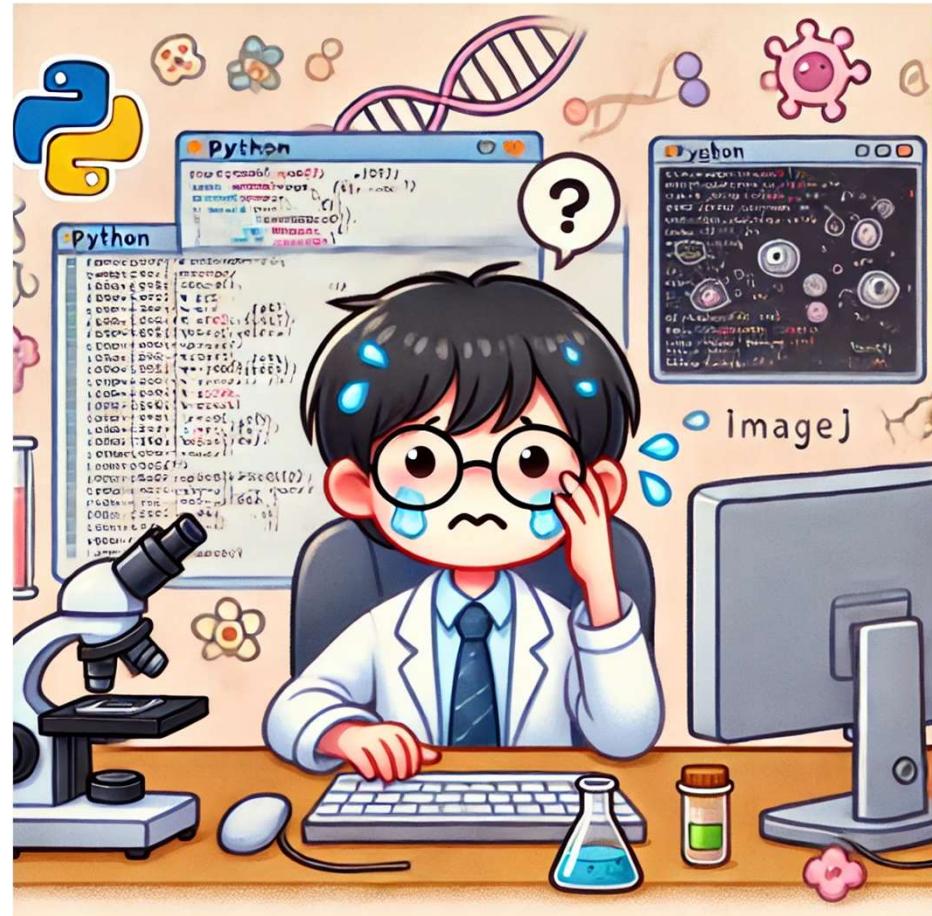
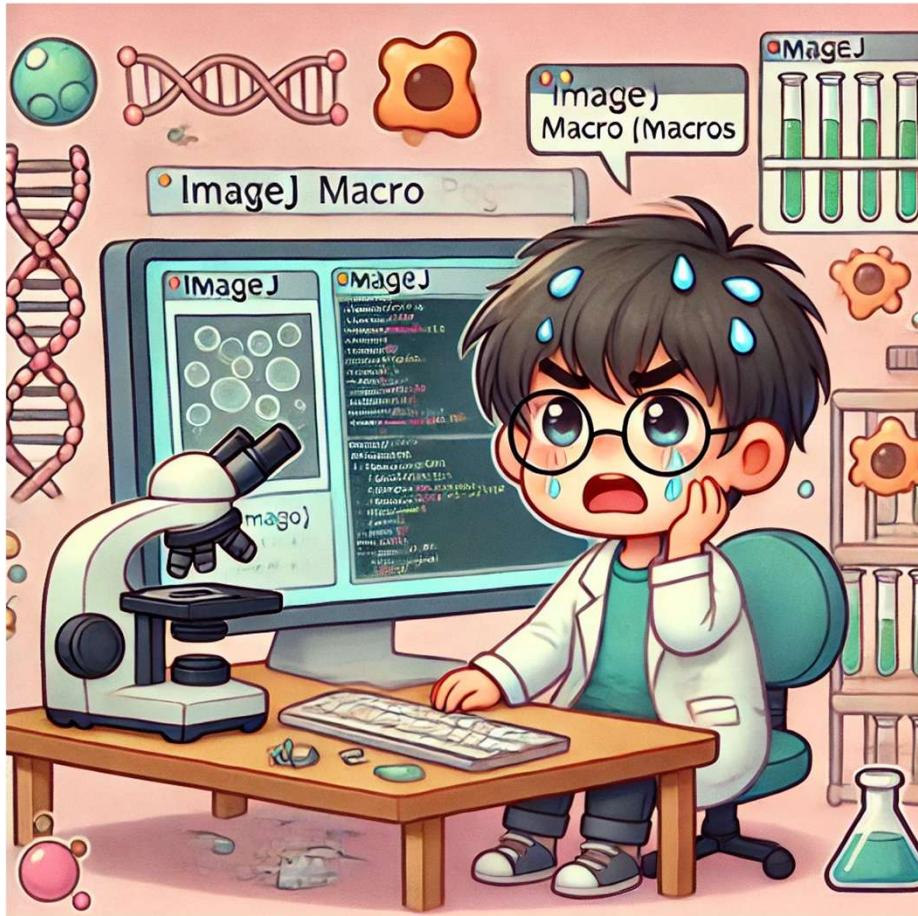


@weichen01.bsky.social



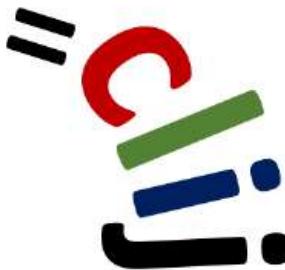
Wei-Chen Chu
ICOB Imaging Core

However, many biologists are not good for programming.

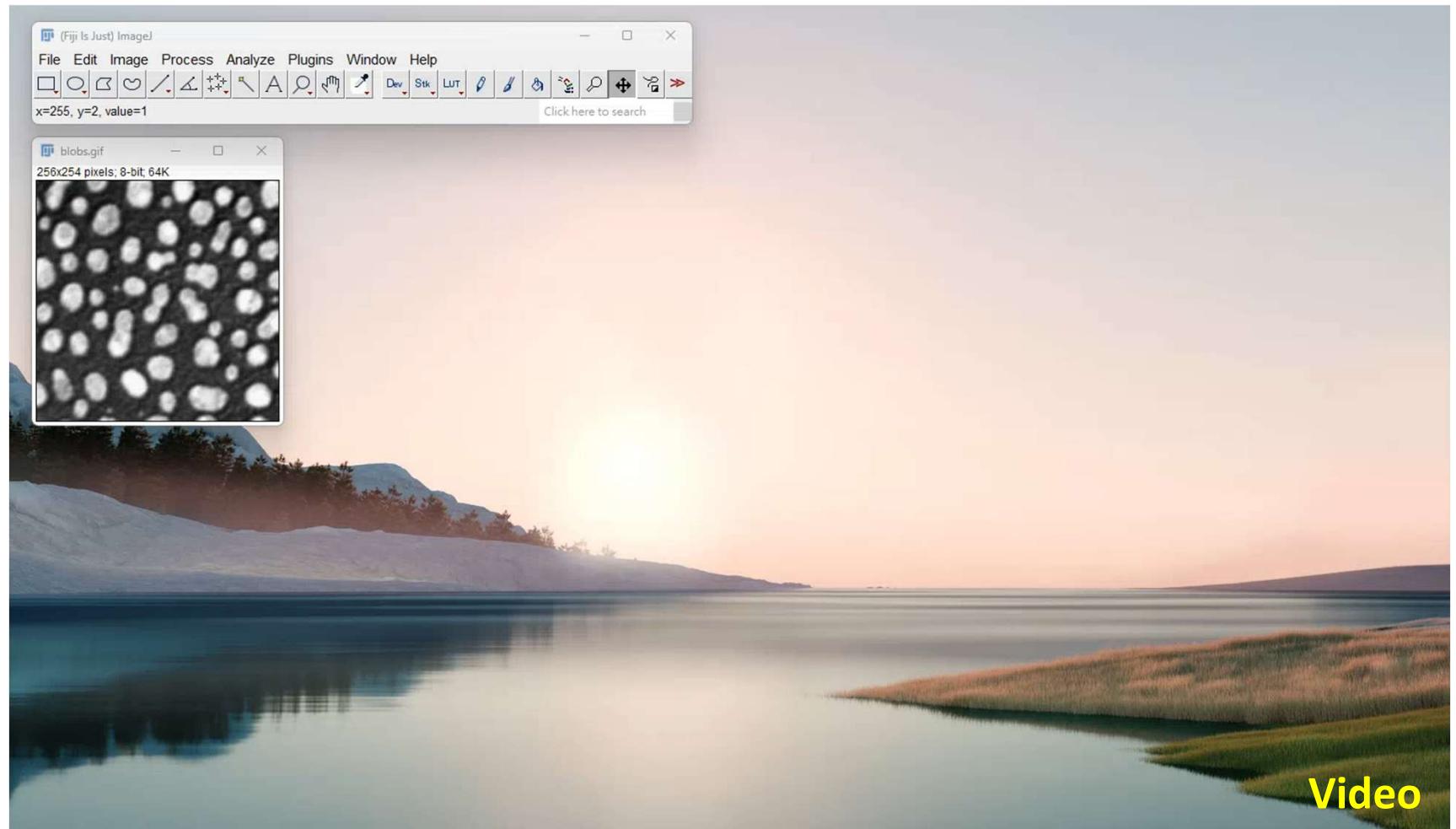


How can an image analysis workflow be built more efficiently?

- Interactive workflow
- Code generation
- Data visualization
- Documentation
- GPU acceleration



<https://clij.github.io/>



@weichen01.bsky.social

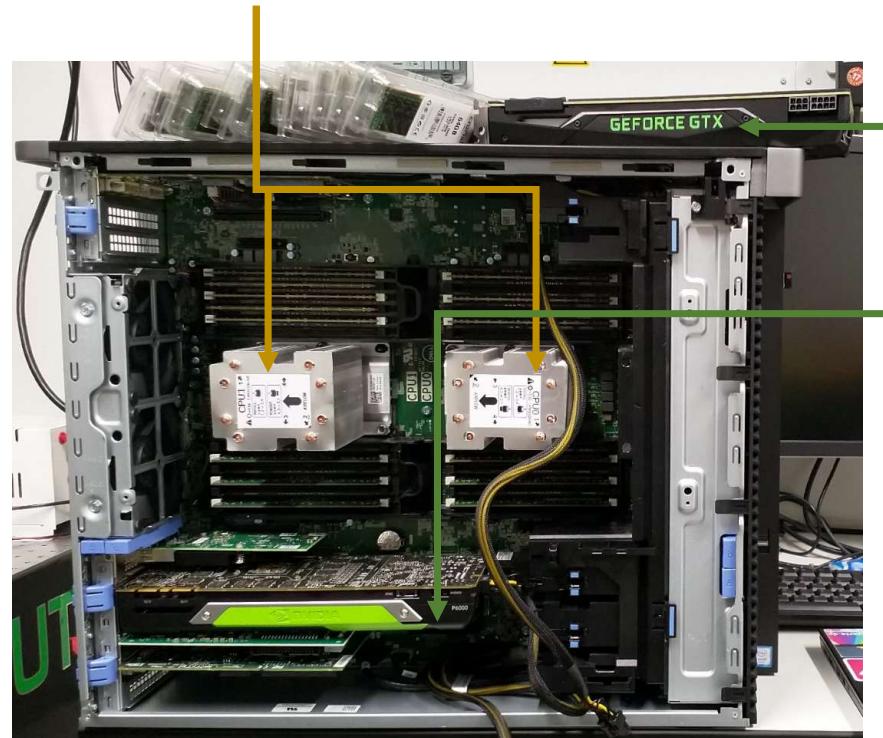


Wei-Chen Chu
ICOB Imaging Core

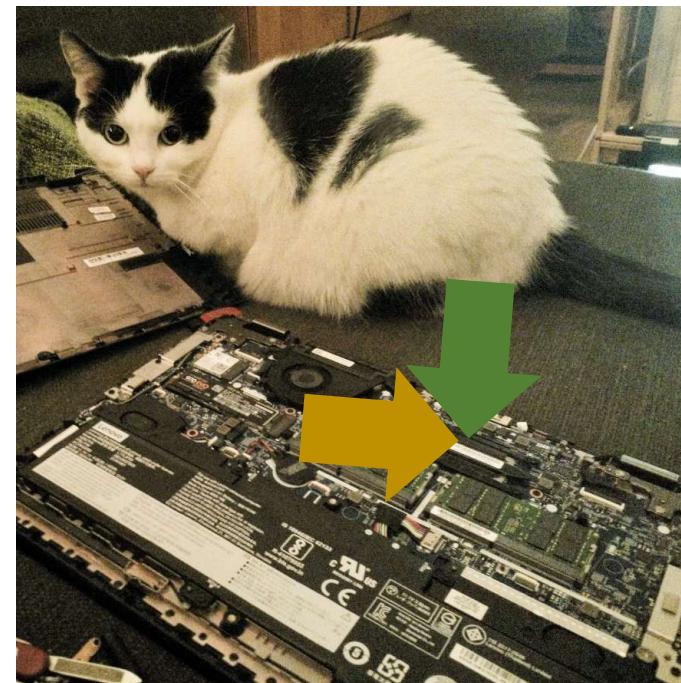
Graphics Processing Units (GPU)

- Typical computers contain Graphics Processing Units

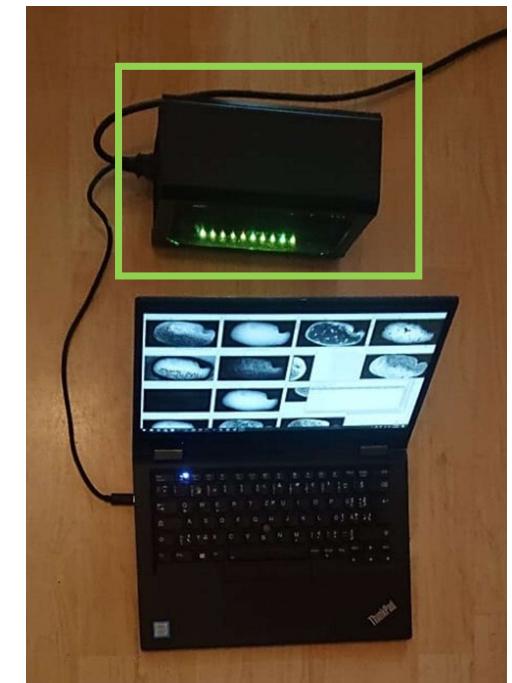
Central Processing Unit (CPU)



Graphics Processing Unit (GPU)

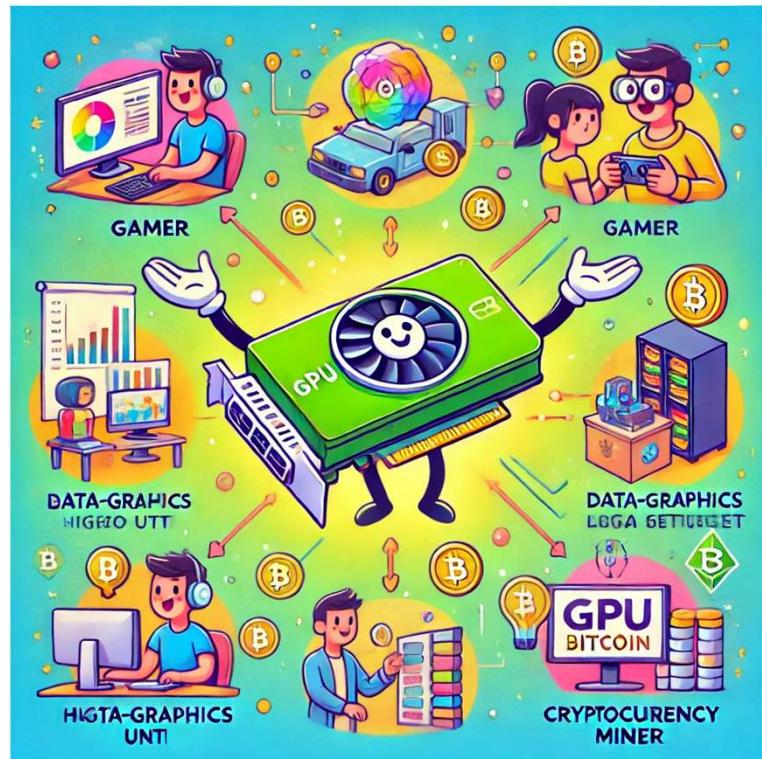


Most laptops contain *integrated* GPUs



Alternative: *external* GPUs

GPU: Graphics processing unit



CUDA

(Compute Unified
Device Architecture)



OpenCl

(Open Computing
Language)

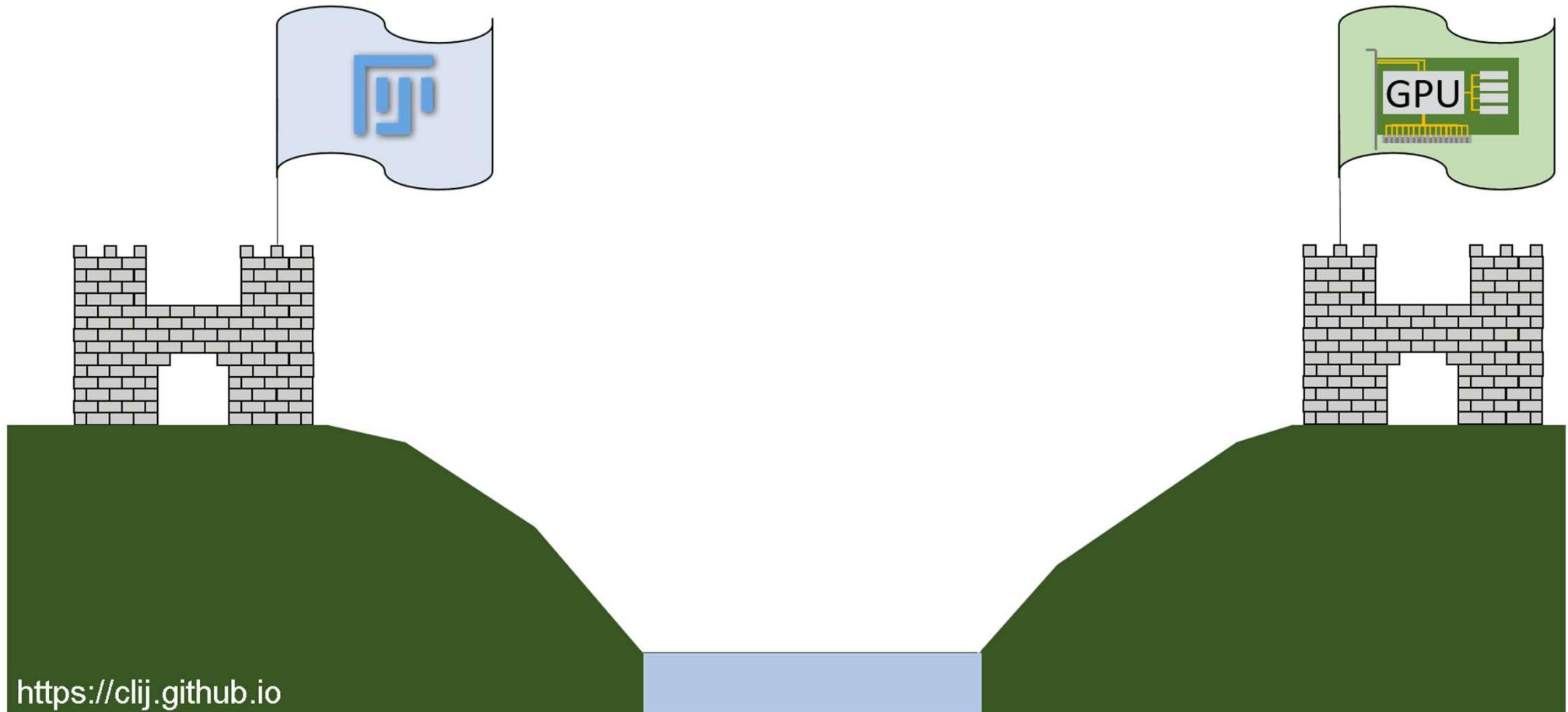


@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

What is CLIJ



Adapted from [Open-Access Training Materials of Dr. Robert Haase](#), licensed [CC-BY 4.0](#)



@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

Integrated GPU-acceleration into ImageJ / Fiji using ImageJ Macro

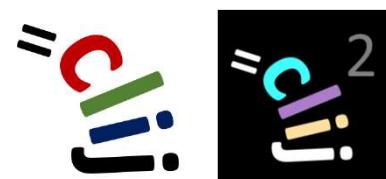


Dr. Robert Haase

Center for Scalable Data
Analytics and Artificial
Intelligence (ScaDS.AI)
Dresden/Leipzig, Leipzig
University, Germany

CLIJ

OpenCL ImageJ



<https://clij.github.io/>

clEsperanto

OpenCL 世界語
(International
auxiliary language)

cle . _

<https://clesperanto.github.io/>



Dr. Stéphane Rigaud

Image Analysis Hub,
Institut Pasteur, Paris



@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

Image processing using FIJI/ ImageJ



Video

1.4 GB

Image data source: Daniela Vorkel, Myers lab, MPI-CBG/CSBD

Adapted from [Open-Access Training Materials of Dr. Robert Haase](#), licensed [CC-BY 4.0](#)

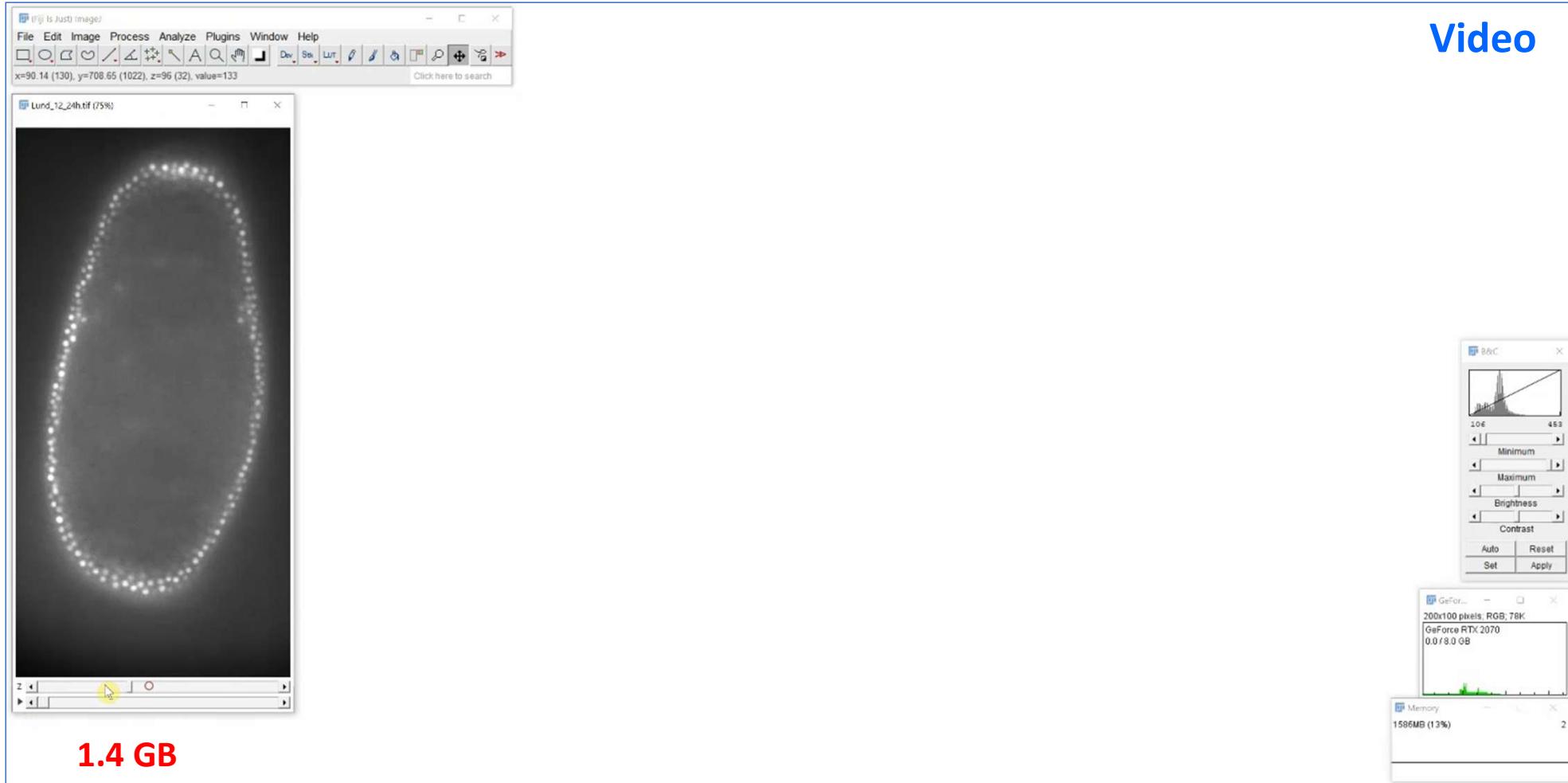


@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

Image processing using CLIJ



Adapted from [Open-Access Training Materials of Dr. Robert Haase](#), licensed [CC-BY 4.0](#)



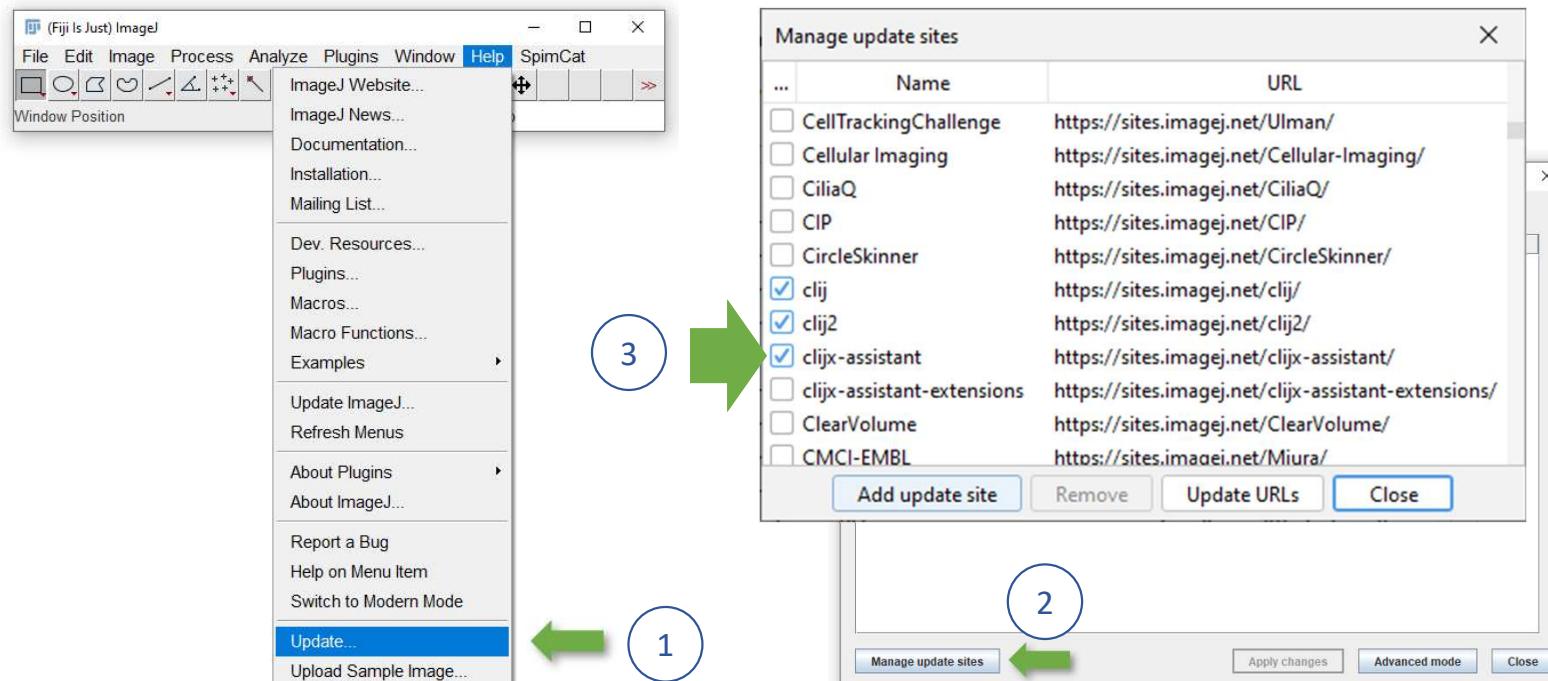
@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

CLIJ: installation

- Just activate the CLIJ update sites, in menu Help > Update...



<https://clij.github.io/clij2-docs/installationInFiji>

Basic installation:

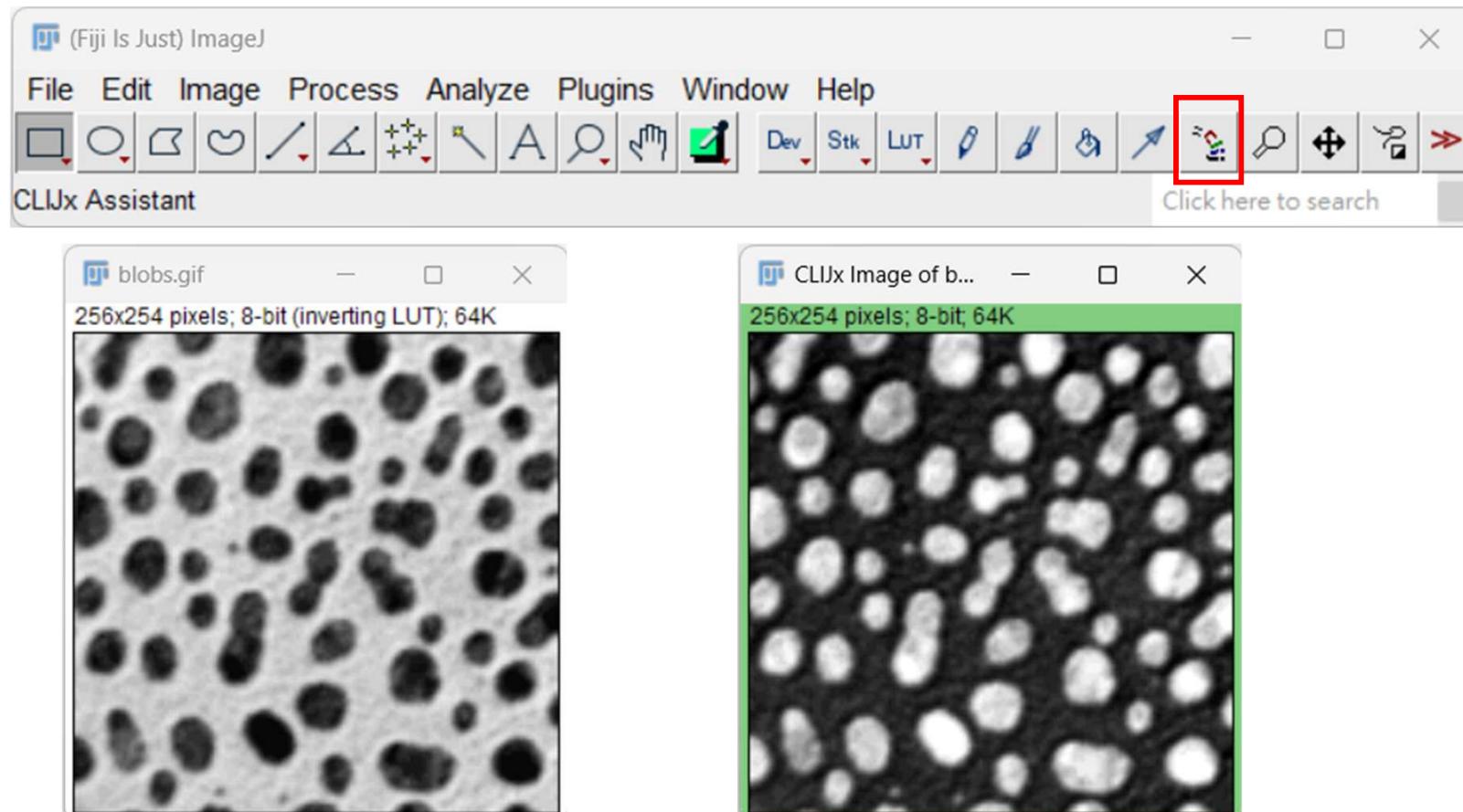
- clij
- clij2
- IJMND

Advanced installation:

- 3D ImageJ Suite
- BoneJ
- IJPB-Plugins experimental
- clijx-assistant
- clijx-assistant-extensions
- BioVoxel 3D box



CLIJ-Assistant



(Default :Auto Brightness and Contrast ON)

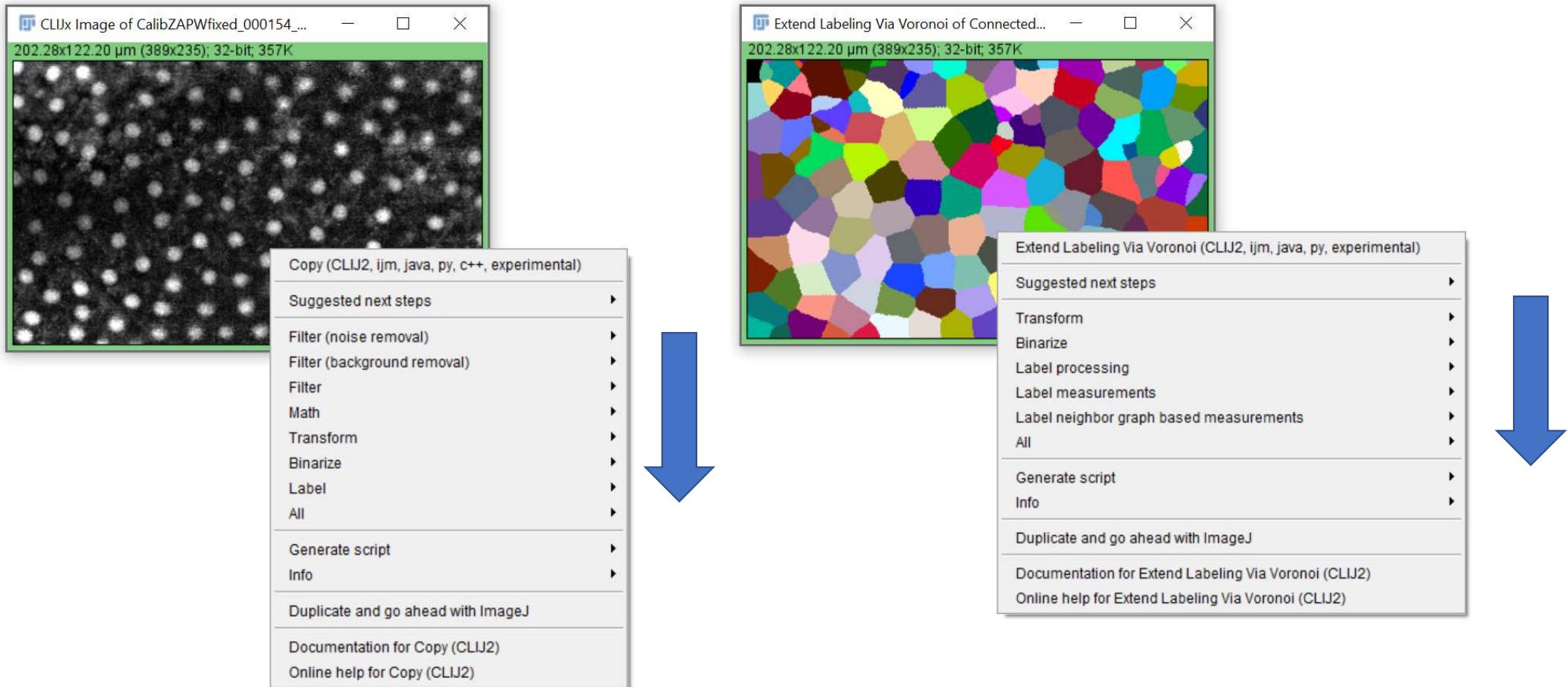


@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

The menu order is intentional: From preprocessing to analysis



Adapted from [Open-Access Training Materials of Dr. Robert Haase](#), licensed [CC-BY 4.0](#)

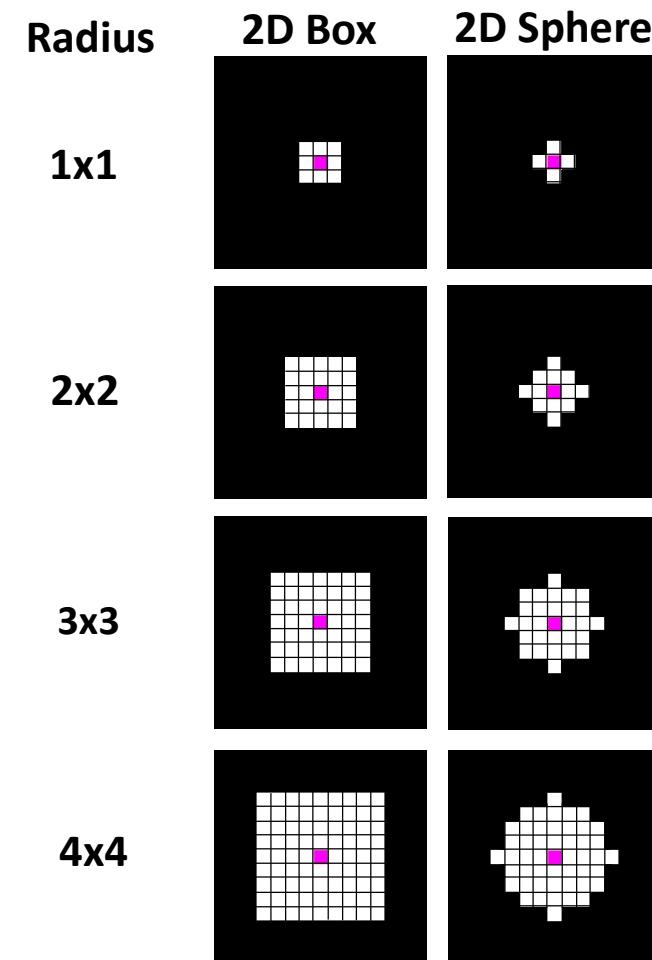
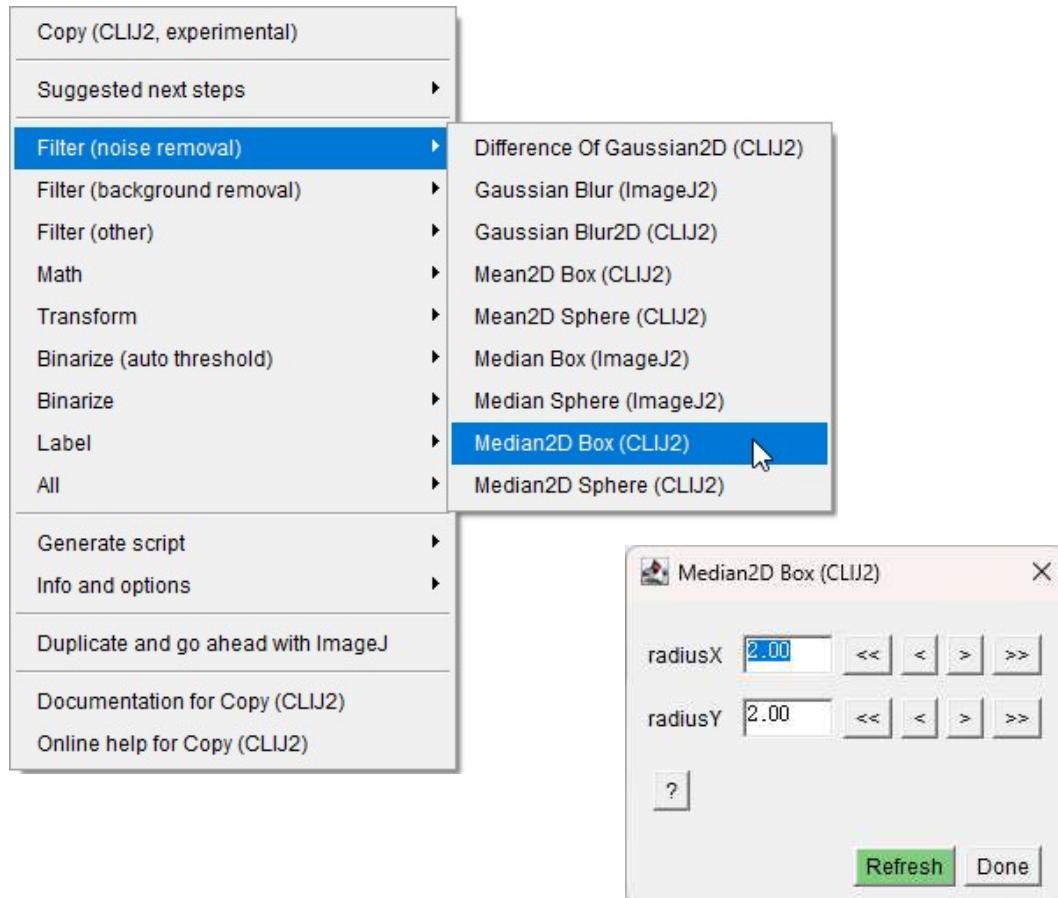


@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

Filters kernel: Box vs Sphere



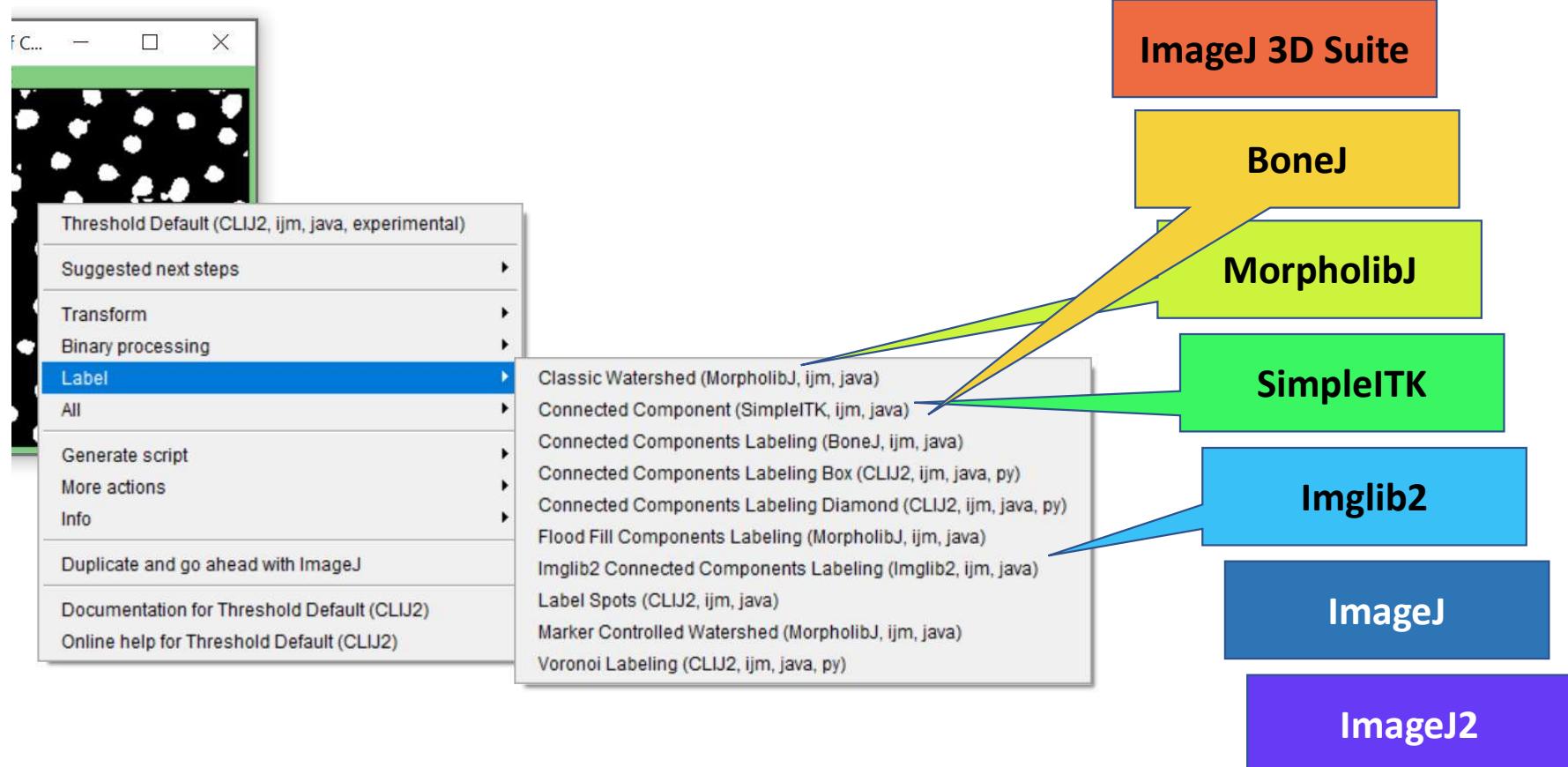
@weichen01.bsky.social



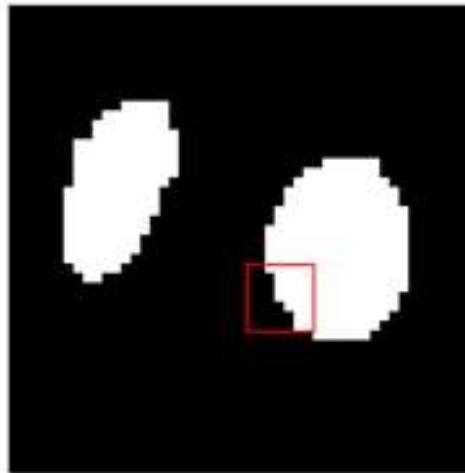
Wei-Chen Chu
ICOB Imaging Core

Extensibility

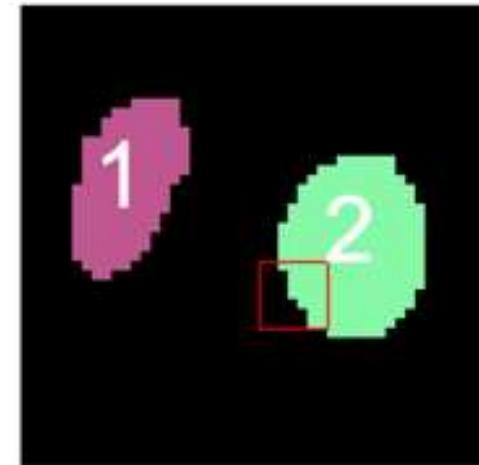
- Install: <https://clij.github.io/assistant/installation#extensions>



Connected component labeling/analysis (CCL/CCA)



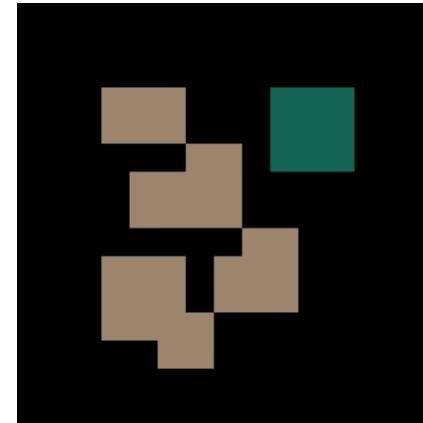
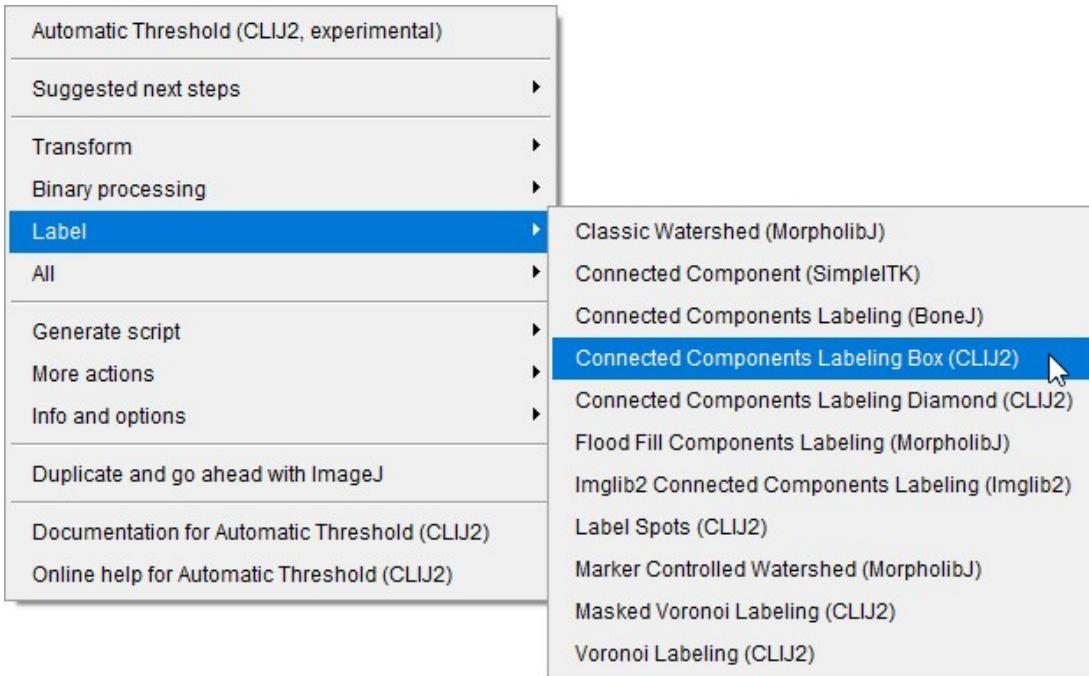
connected
component
analysis



| | | | | |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 |

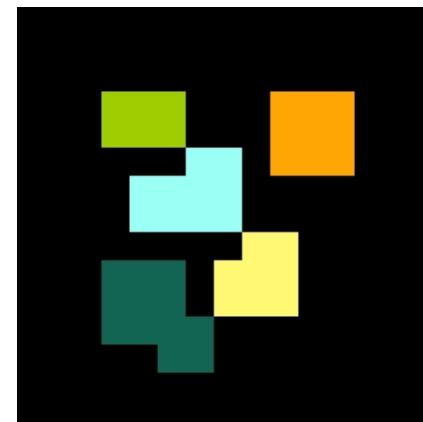
| | | | | |
|---|---|---|---|---|
| 0 | 0 | 2 | 2 | 2 |
| 0 | 0 | 2 | 2 | 2 |
| 0 | 0 | 2 | 2 | 2 |
| 0 | 0 | 0 | 2 | 2 |
| 0 | 0 | 0 | 0 | 2 |

CCL: Box vs Diamond



Box (8-connectivity)

ImageJ Analyze Particles:
8-connectivity



Diamond (4-connectivity)

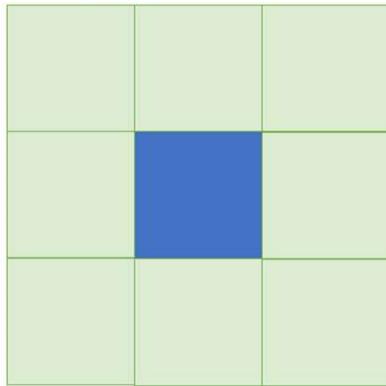


@weichen01.bsky.social

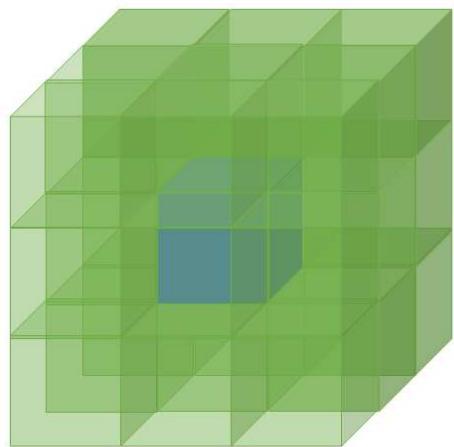


Wei-Chen Chu
ICOB Imaging Core

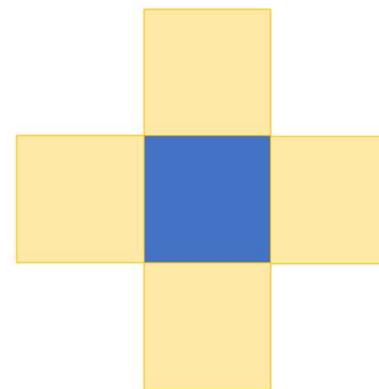
CCL: Box vs Diamond in 2D and 3D



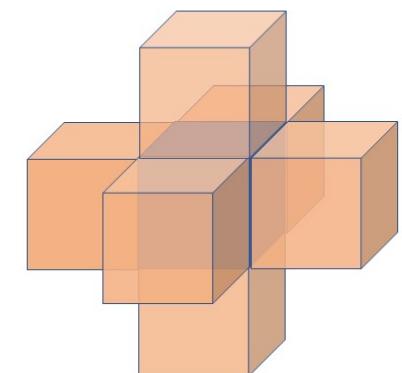
8-connectivity



26-connectivity



4-connectivity



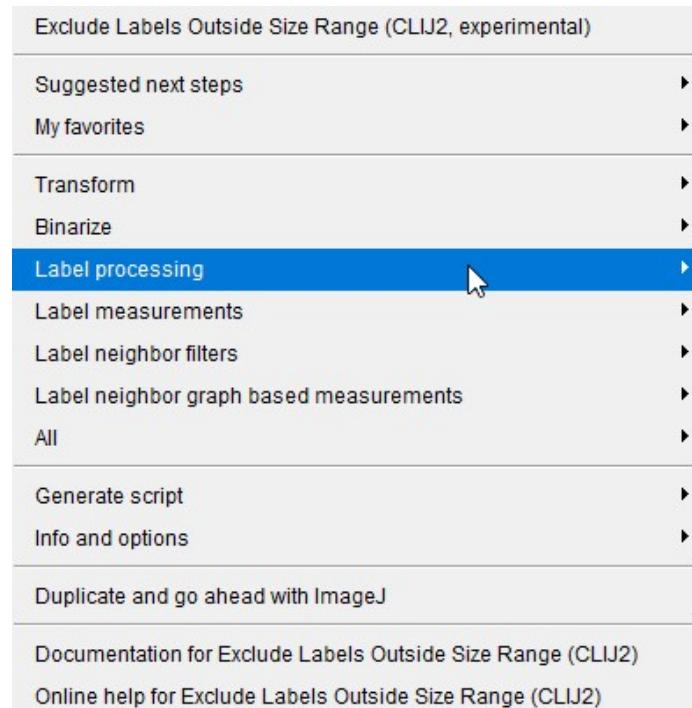
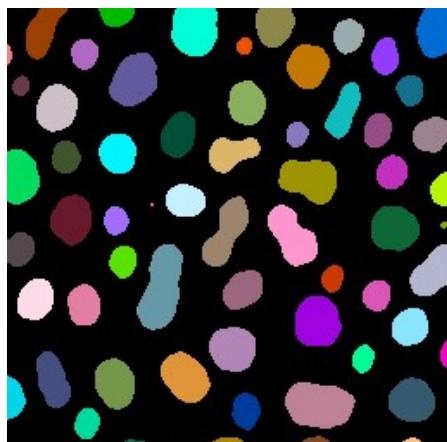
6-connectivity

Box

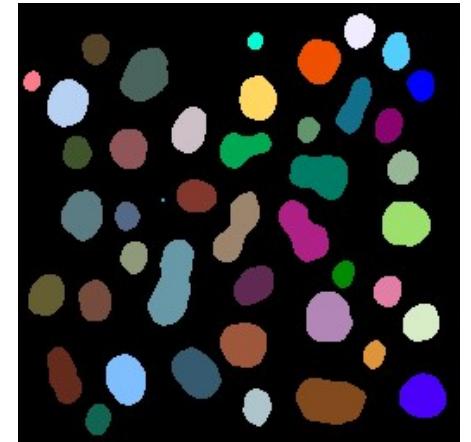
Diamond



Label processing



Exclude Label On
Edges



Exclude Label
Outside Size Range



@weichen01.bsky.social



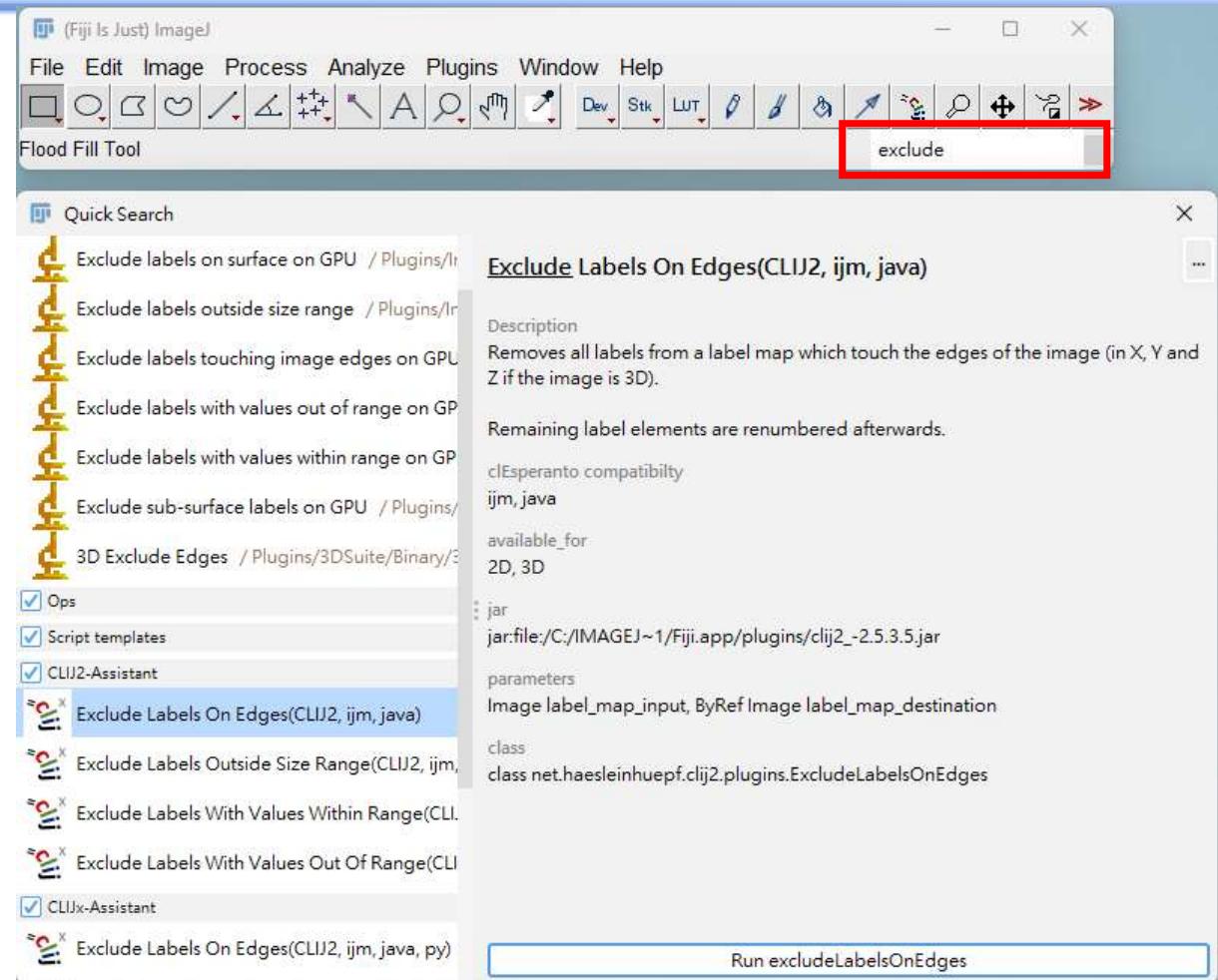
Wei-Chen Chu
ICOB Imaging Core

If you start from a mask or label image -> Use search bar

- **AI-based segmentation**



CLIJ2-Assistant
CLIJx-Assistant

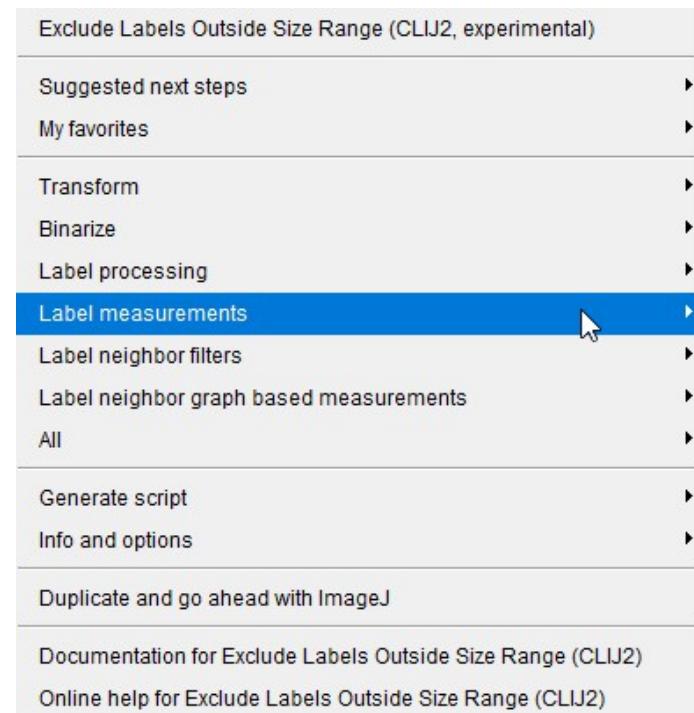
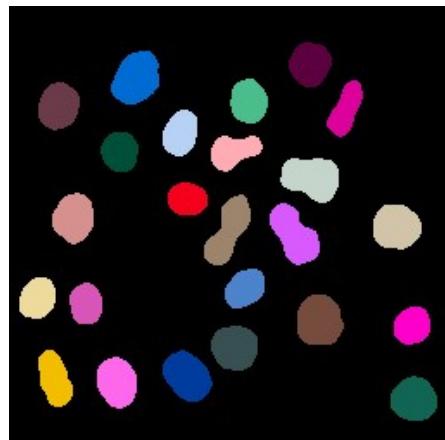


@weichen01.bsky.social

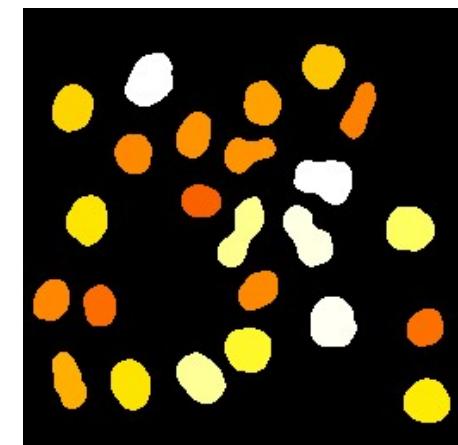


Wei-Chen Chu
ICOB Imaging Core

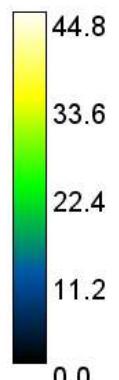
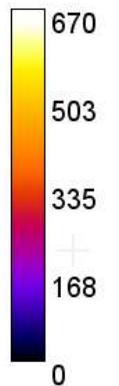
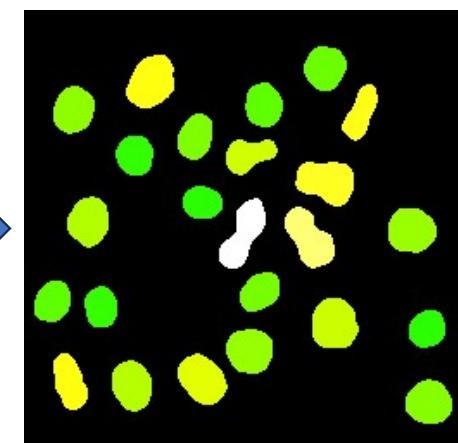
Label measurements (-> parametric image)



Pixel Count Map



Max. Feret Map



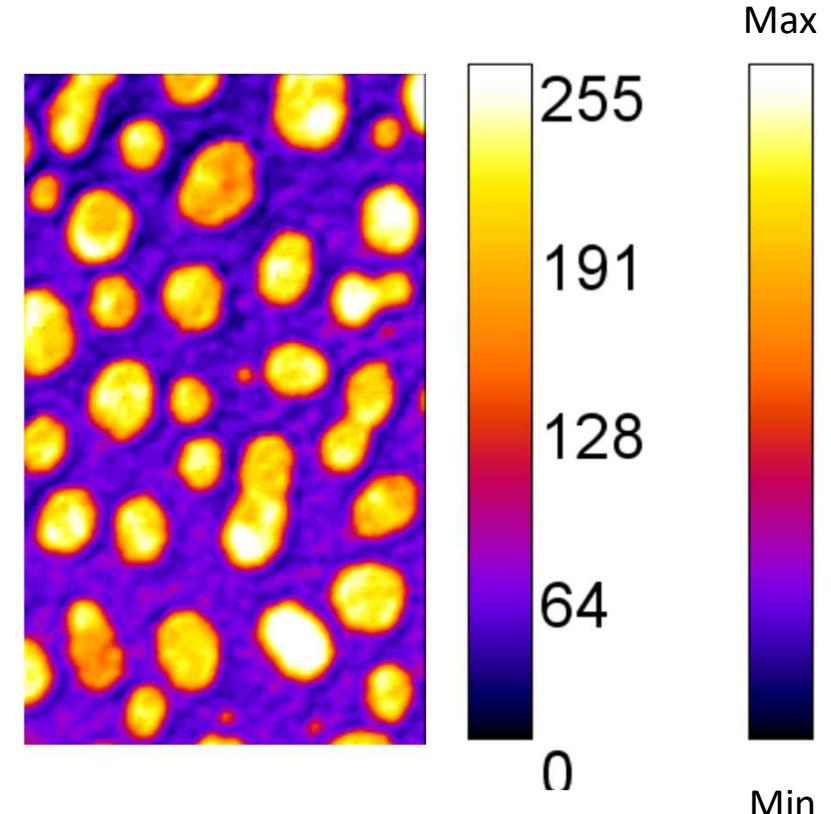
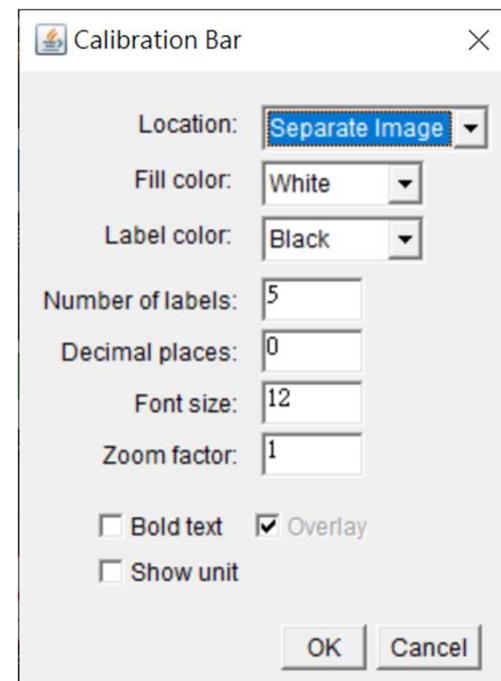
@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

Calibration Bar (Color scale)

- Analysis -> Tools -> Calibration Bar
- Only work for single channel image
- Brightness and Contrast setting affect the Calibration Bar!

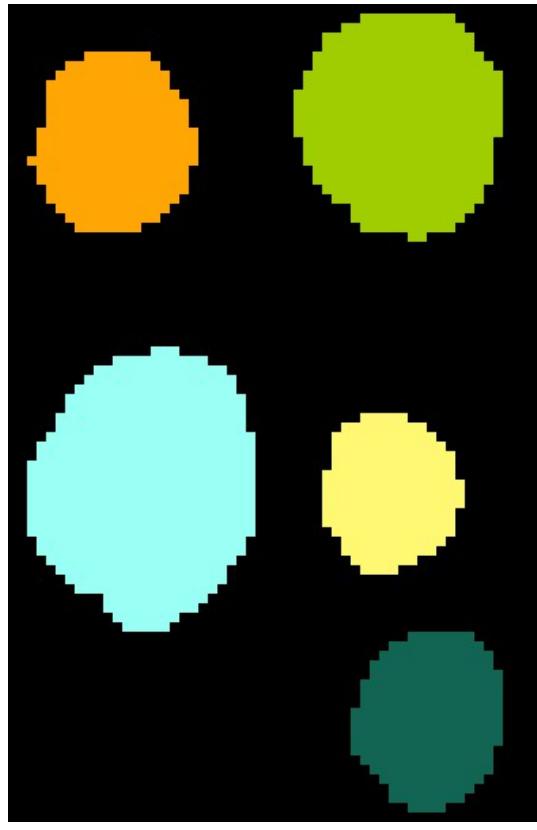


@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

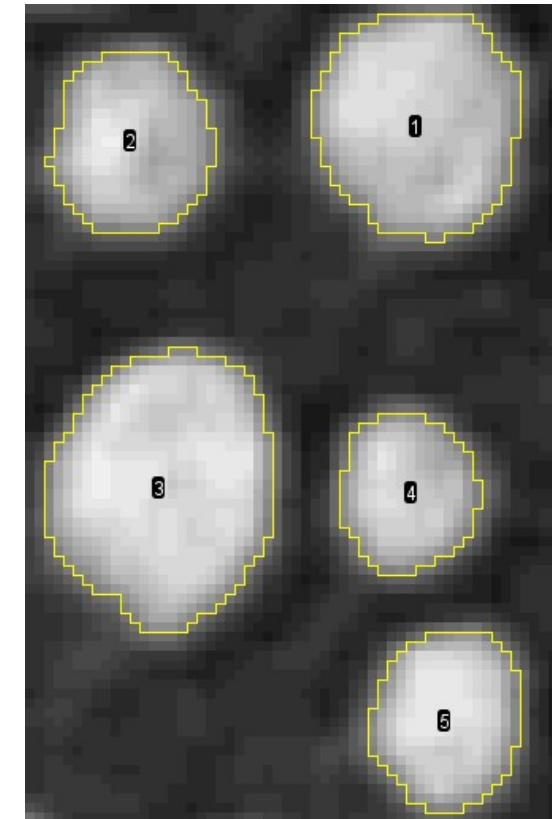
Measurements with labels



Labels



- `pullLabelsToROIManager (CLIJ2)`
- Label to ROI ([BIOP/ijp-LaRoMe](#))
- **BioVoxel 3D box**
 - **Labels to 2D Roi Manager**
 - Labels to 3D Roi Manager



ROIs

ImageJ-specific

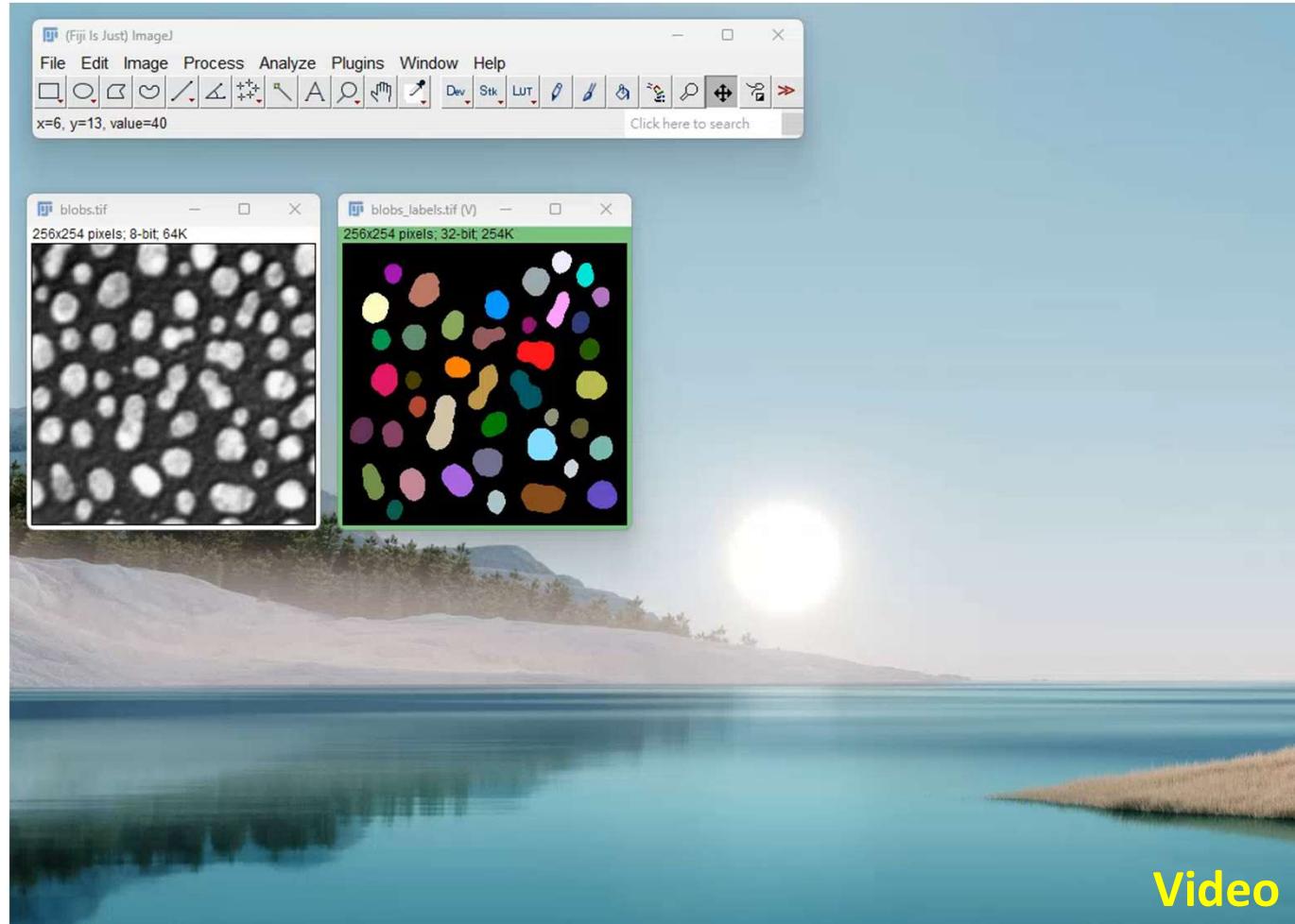


@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

Label to ROI and measurement



Video

Plugins -> BioVoxcel 3D box
-> Labels to 2D Roi Manager



<https://biovoxxel.github.io/bv3dbox/>

[BioVoxcel 3D Box - GPU-accelerated 2D and 3D Image Processing and Analysis - YouTube](#)



@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

Parametric image and label processing

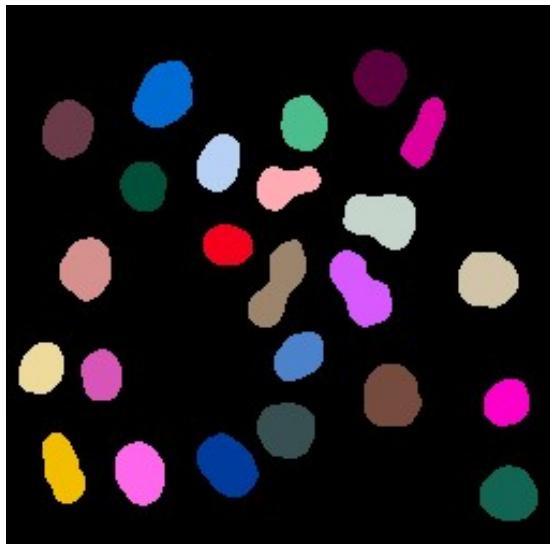


Image Label Map Input

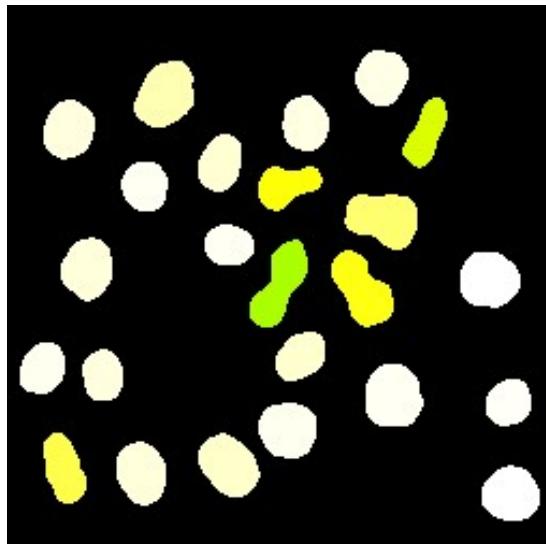
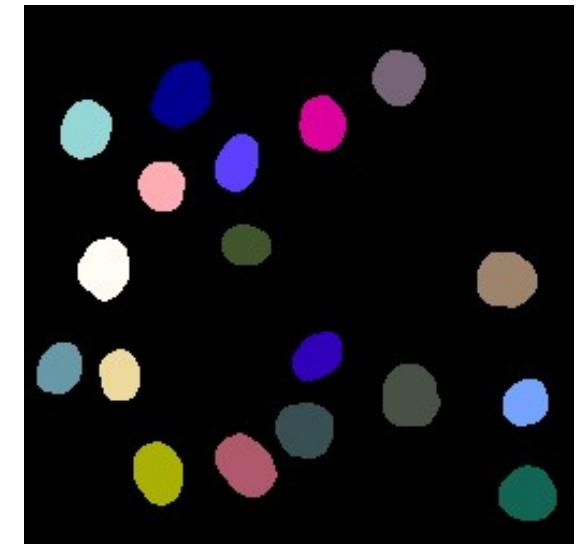
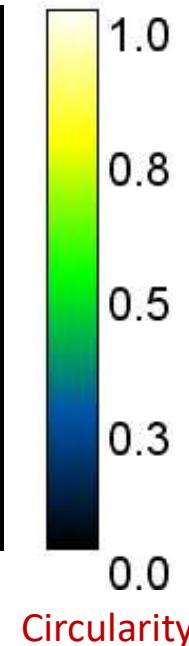


Image Value Vector



Exclude the label with circularity
Map with value within 0 ~ 0.88

Exclude Labels with value within/out of range

Min/ Mean/ Max Intensity, Circularity, Feret
Diameter, Perimeter, Extension ratio....etc.

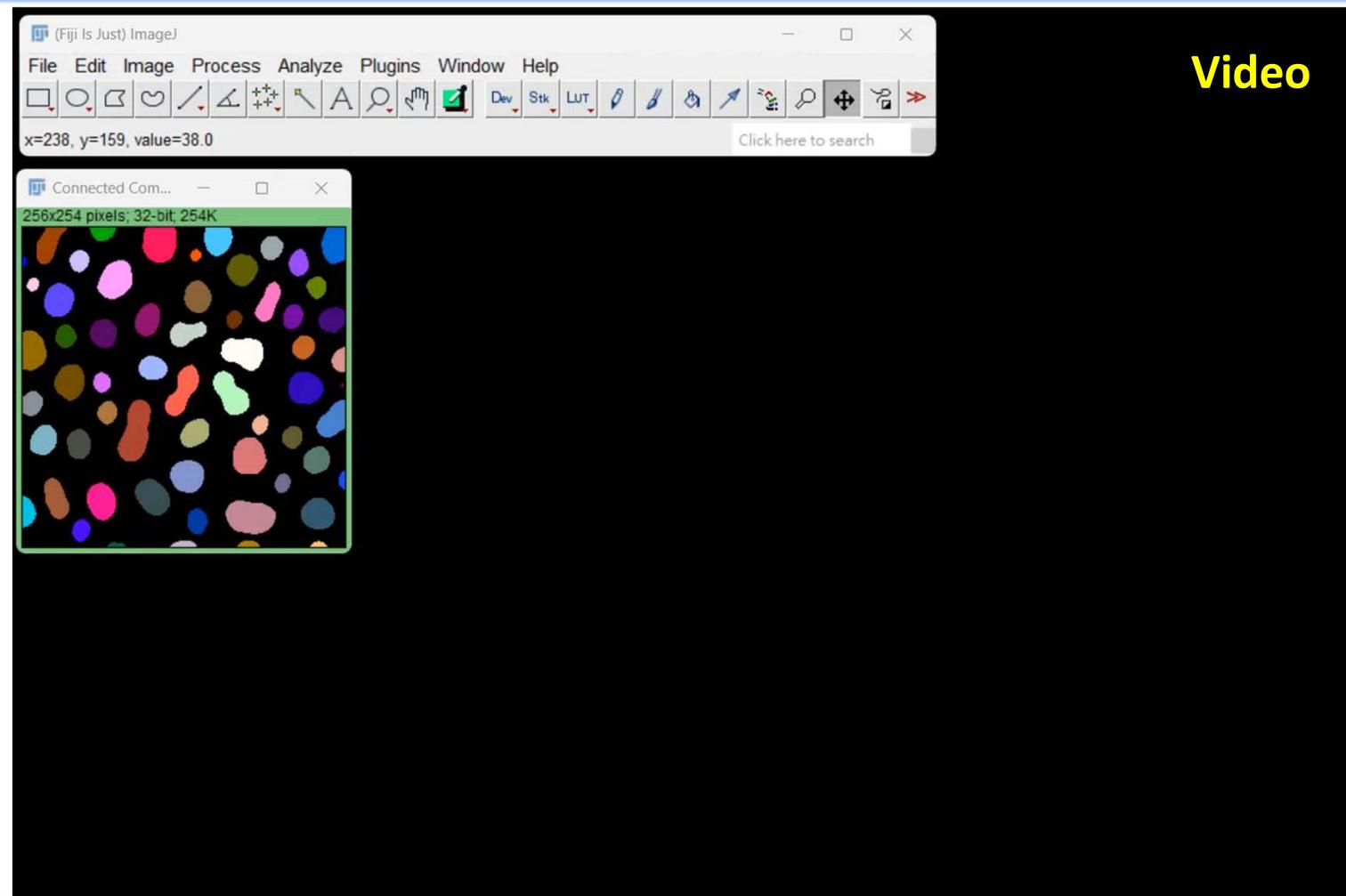
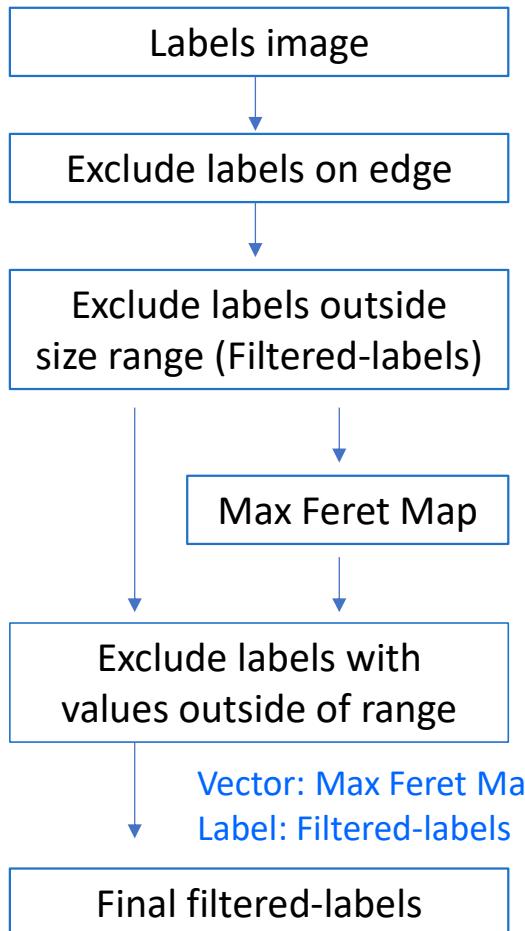


@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

Label processing demo

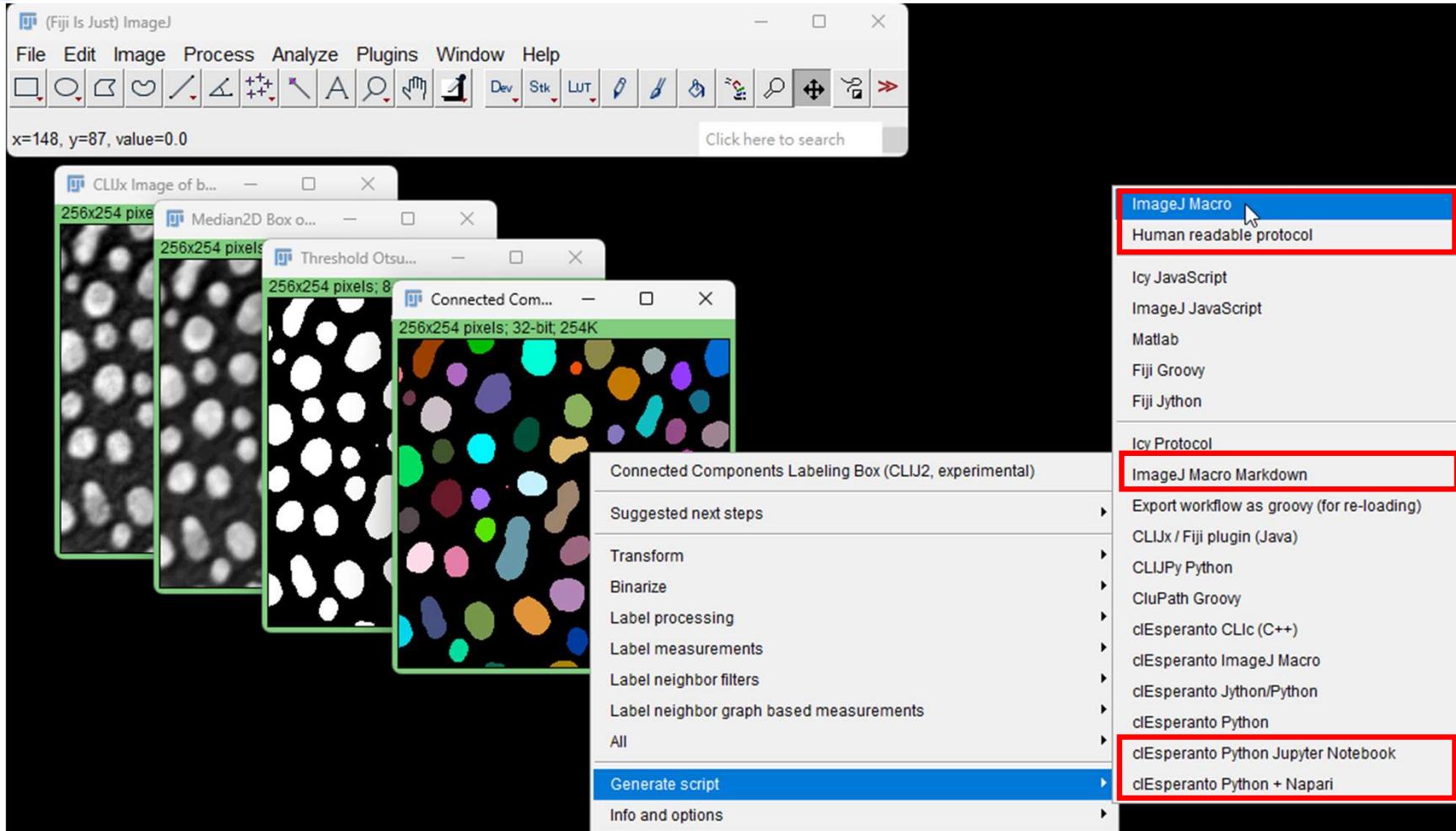


@weichen01.bsky.social

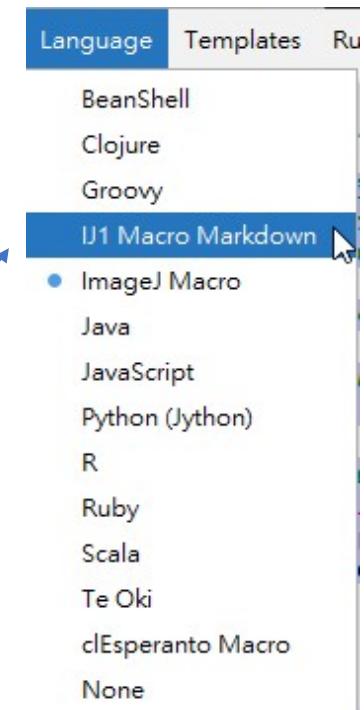


Wei-Chen Chu
ICOB Imaging Core

Interactive workflow design and code generation



Fiji Script editor:



@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

CLIJ2: What every ImageJ Macro script must have

Load data

```
1 // Load data
2 run("Cell Colony (31K)");
3
4 // initialize GPU
5 run("CLIJ2 Macro Extensions", "cl_device=");
6 Ext.CLIJ2_clear();
7
8 // push image to GPU
9 input_image = getTitle();
10 Ext.CLIJ2_push(input_image);
11
12 // process image
13 sigma = 5;
14 Ext.CLIJ2_gaussianBlur2D(input_image, result_image, sigma, sigma);
15
16 // optional: release input data
17 Ext.CLIJ2_release(input_image);
18
19 // pull result back from GPU
20 Ext.CLIJ2_pull(result_image);
21
22 // clean up by the end
23 Ext.CLIJ2_clear();
```

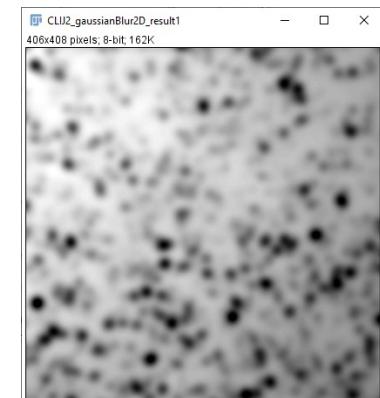
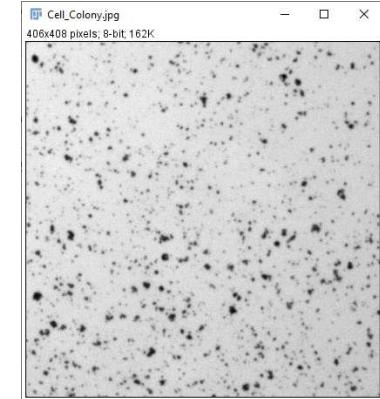
Initialize GPU

Push

Process images

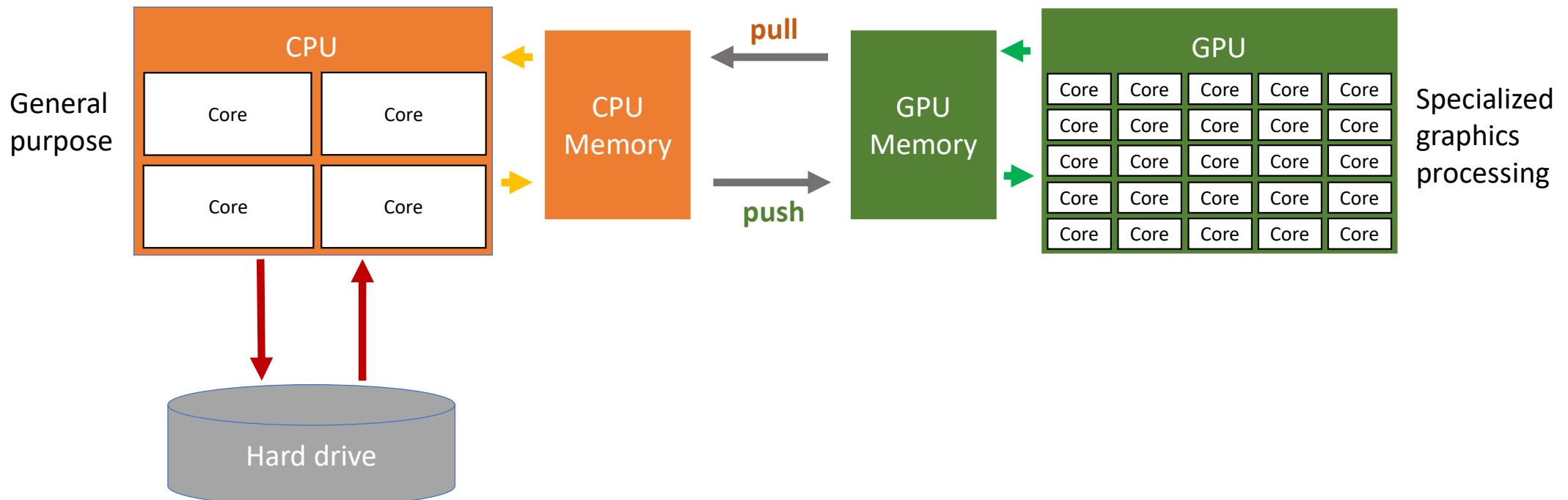
Pull

Cleanup

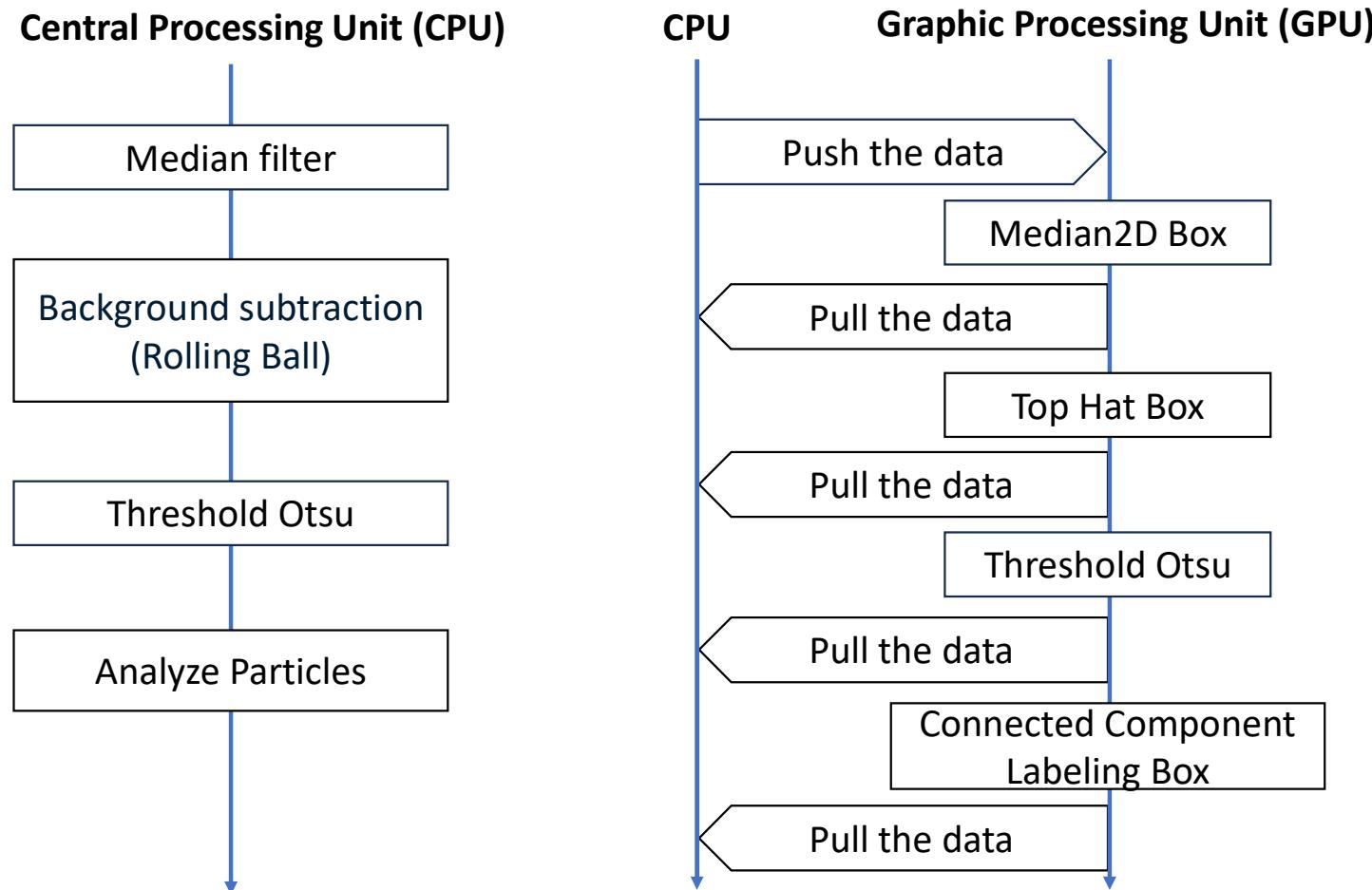


GPUs allow real-time image processing

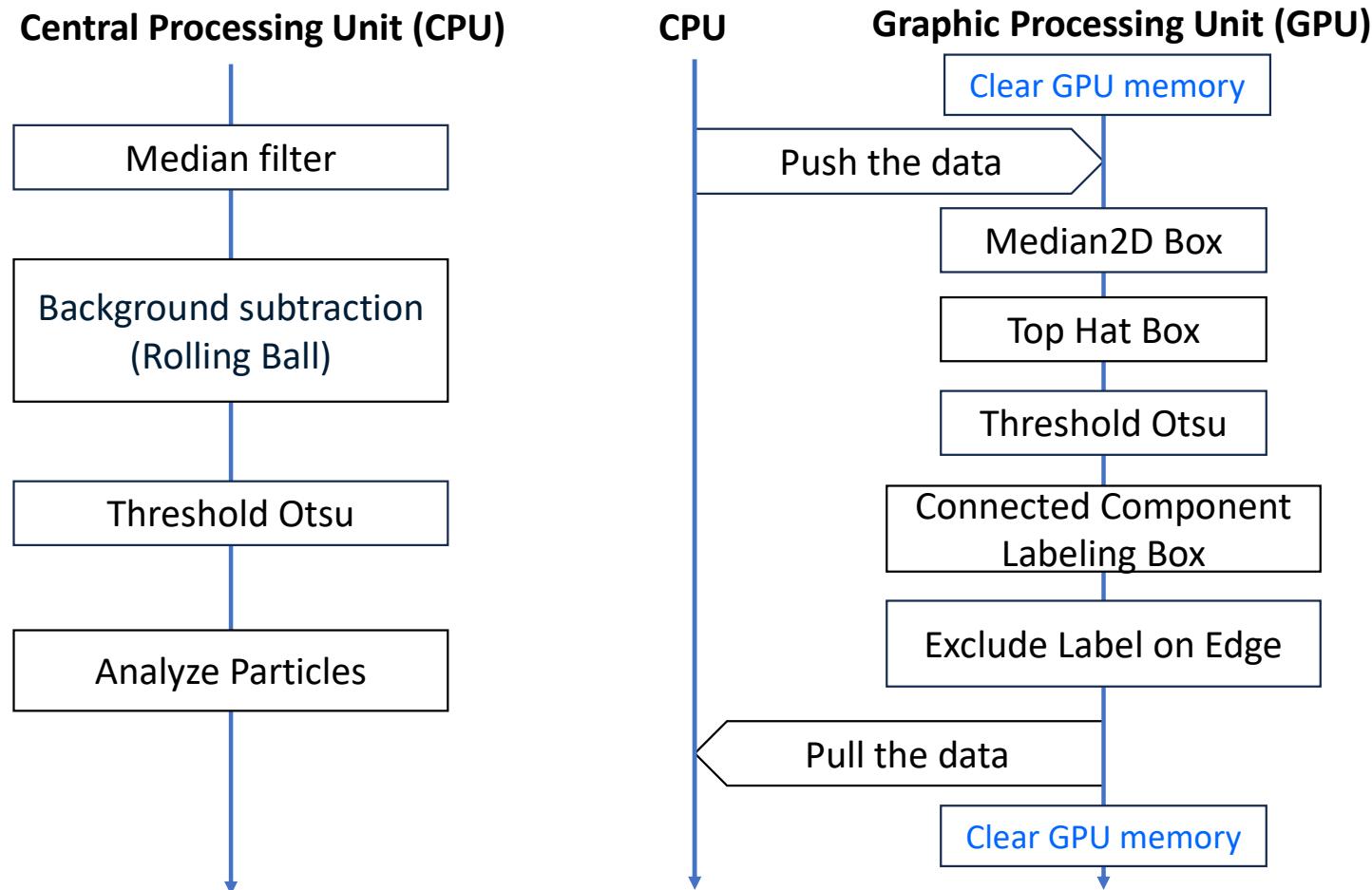
- GPUs are specialised in processing vectors and matrices – super fast



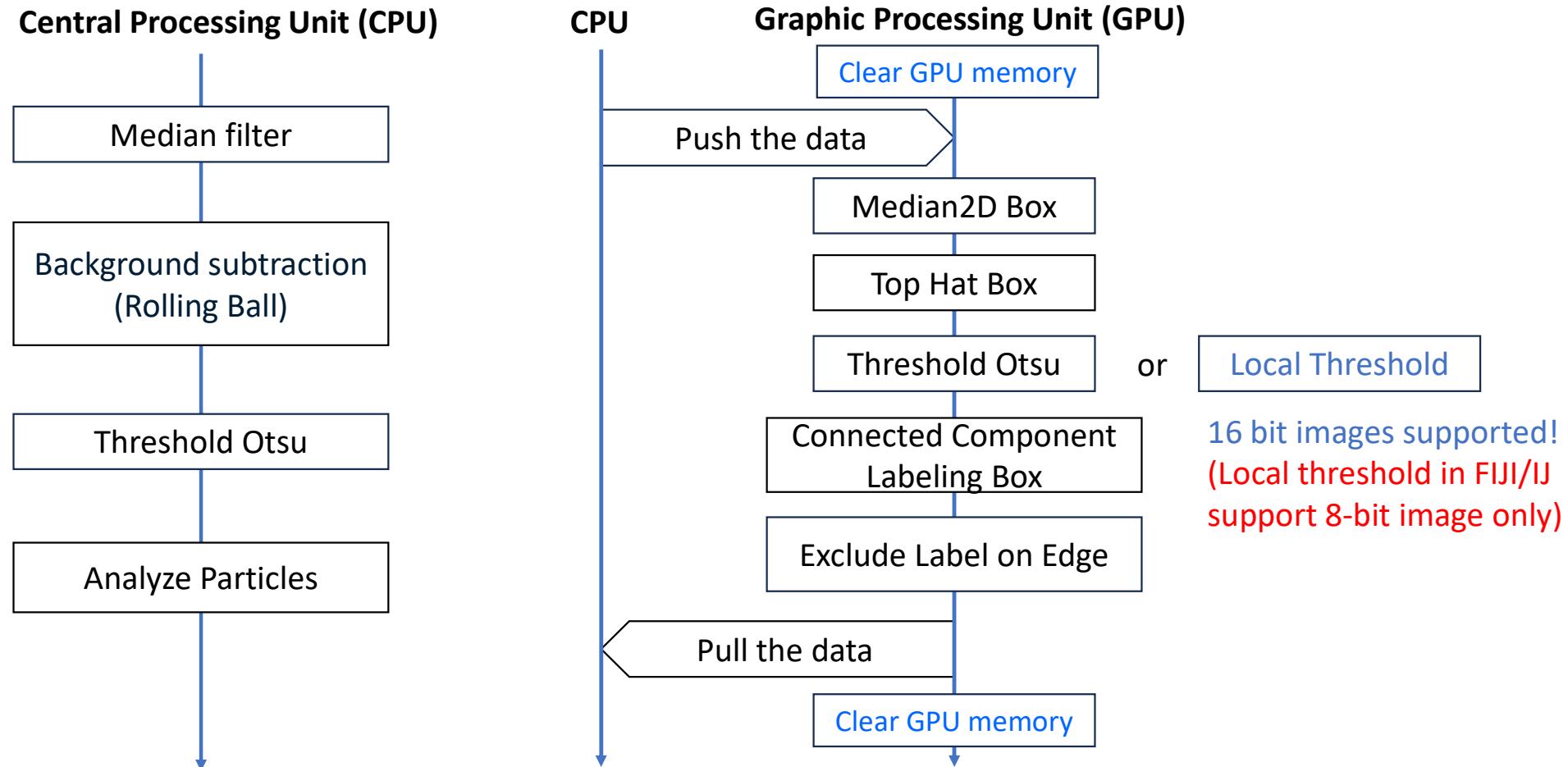
Workflow build (draft by CLIJ-assistant)



Workflow build (improve speed)



Workflow build (improve speed and save the GPU memory)



CLIJ2: Macro refine

```
1 // To make this script run in Fiji, please activate  
2 // the clij and clij2 update sites in your Fiji  
3 // installation. Read more: https://clij.github.io  
4  
5 // Generator version: 2.5.1.6  
6  
7 // Init GPU  
8 run("CLIJ2 Macro Extensions", "cl_device=");  
9  
10 // Load image from disc  
11 open("C:/Users/weich/AppData/Local/Temp/temp1740905840354.tif");  
12 image_1 = getTitle();  
13 Ext.CLIJ2_pushCurrentZStack(image_1);  
14  
15 // Copy  
16 Ext.CLIJ2_copy(image_1, image_2);  
17 Ext.CLIJ2_release(image_1);  
18  
19 Ext.CLIJ2_pull(image_2);  
20  
21 // Median2D Box  
22 radiusX = 2;  
23 radiusY = 2;  
24 Ext.CLIJ2_median2DBox(image_2, image_3, radiusX, radiusY);  
25 Ext.CLIJ2_release(image_2);  
26  
27 Ext.CLIJ2_pull(image_3);
```

```
1 // To make this script run in Fiji, please activate  
2 // the clij and clij2 update sites in your Fiji  
3 // installation. Read more: https://clij.github.io  
4  
5 // Generator version: 2.5.1.6  
6  
7 // Init GPU  
8 run("CLIJ2 Macro Extensions", "cl_device=");  
9 Ext.CLIJ2_clear();  
10 // Load image from disc  
11 //open("C:/Users/weich/AppData/Local/Temp/temp1740905840354.tif");  
12 image_1 = getTitle();  
13 Ext.CLIJ2_pushCurrentZStack(image_1);  
14  
15 // Copy  
16 Ext.CLIJ2_copy(image_1, image_2);  
17 Ext.CLIJ2_release(image_1);  
18  
19 //Ext.CLIJ2_pull(image_2);  
20  
21 // Median2D Box  
22 radiusX = 2;  
23 radiusY = 2;  
24 Ext.CLIJ2_median2DBox(image_2, image_3, radiusX, radiusY);  
25 Ext.CLIJ2_release(image_2);  
26  
27 //Ext.CLIJ2_pull(image_3);
```

Remove unnecessary open image

⋮
Ext.CLIJ2_clear();
(Clear the GPU memory)

Remove unnecessary pull()
calls by the end



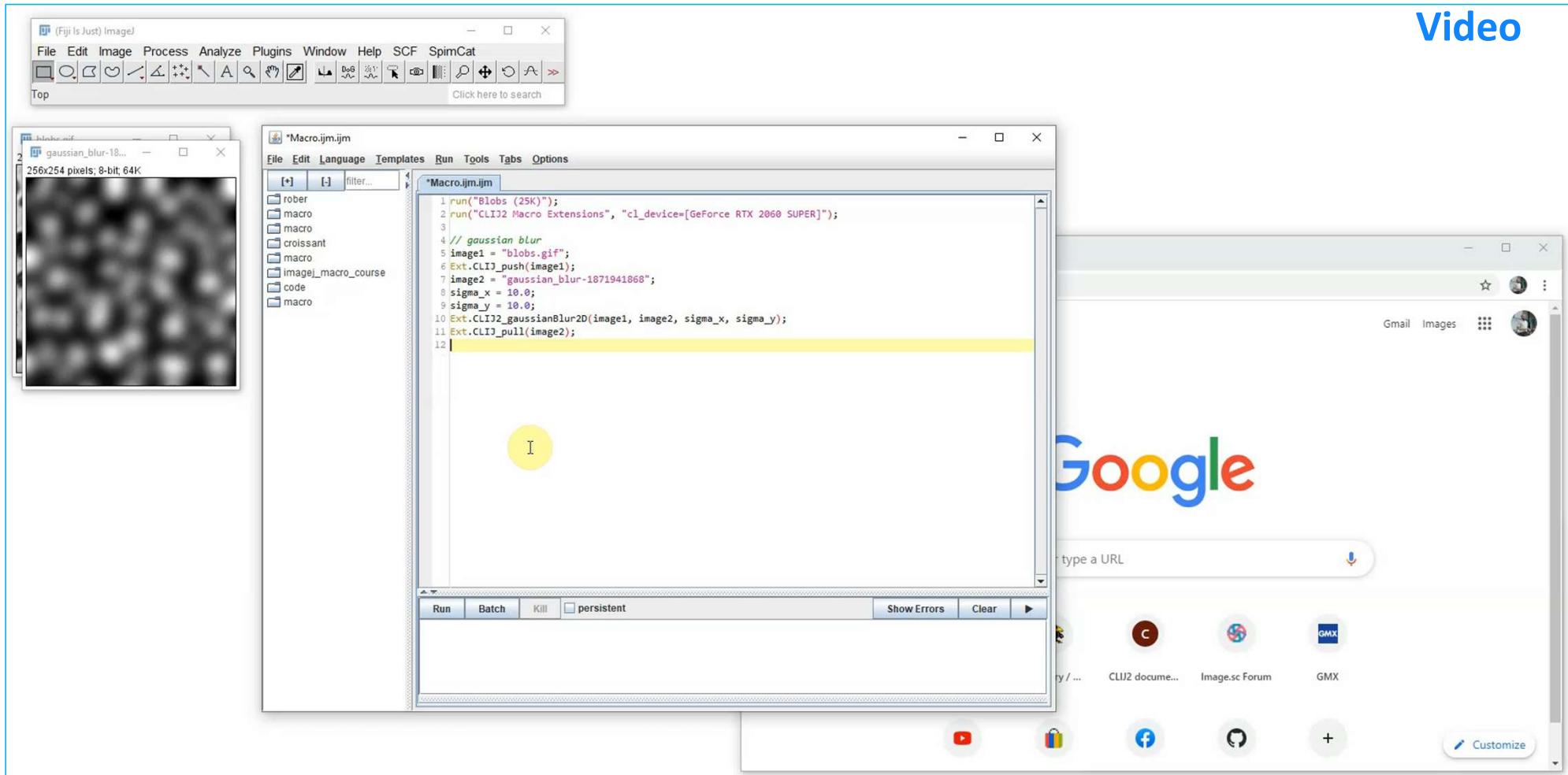
@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

CLIJ2: Macro editing

Video



Adapted from [Open-Access Training Materials of Dr. Robert Haase](#), licensed [CC-BY 4.0](#)



@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

Community-developed checklists for publishing images and image analysis

New workflows

| | | |
|---|-----------------------------------|--------------------------------------|
|  | Cite components and platform | <input type="checkbox"/> Minimal |
|  | Describe sequence | <input type="checkbox"/> |
|  | Key settings | <input type="checkbox"/> |
|  | Example data and code | <input type="checkbox"/> |
|  | Manual ROI | <input type="checkbox"/> |
| 193 | Exact versions | <input type="checkbox"/> |
|  | All settings | <input type="checkbox"/> Recommended |
|  | Public example data and code | <input type="checkbox"/> Recommended |
|  | Rationale | <input type="checkbox"/> Recommended |
|  | Limitations | <input type="checkbox"/> Recommended |
|  | Screen recording or tutorial | <input type="checkbox"/> Ideal |
|  | Easy install and usage, container | <input type="checkbox"/> Ideal |

FIJI

Schindelin, J., Arganda-Carreras, I., Frise, E. et al. Fiji: an open-source platform for biological-image analysis. *Nat Methods* **9**, 676–682 (2012).
<https://doi.org/10.1038/nmeth.2019>

CLIJ

- Robert Haase, Loic Alain Royer, Peter Steinbach, Deborah Schmidt, Alexandr Dibrov, Uwe Schmidt, Martin Weigert, Nicola Maghelli, Pavel Tomancak, Florian Jug, Eugene W Myers. *CLIJ: GPU-accelerated image processing for everyone.* [Nat Methods](#) **17**, 5–6 (2020) doi:10.1038/s41592-019-0650-1
- Daniela Vorkel, Robert Haase. *GPU-accelerating ImageJ Macro image processing workflows using CLIJ.* [arXiv preprint](#)
- Robert Haase, Akanksha Jain, Stéphane Rigaud, Daniela Vorkel, Pradeep Rajasekhar, Theresa Suckert, Talley J. Lambert, Juan Nunez-Iglesias, Daniel P. Poole, Pavel Tomancak, Eugene W. Myers. *Interactive design of GPU-accelerated Image Data Flow Graphs and cross-platform deployment using multi-lingual code generation.* [bioRxiv preprint](#)

BioVoxcel 3D box

Jan Brocher. (2024). biovoxxel/bv3dbox: BioVoxcel 3D Box - v1.22.3 (bv3dbox-1.22.3). Zenodo. <https://doi.org/10.5281/zenodo.12571844>



Why you should learn CLIJ

- Interactive workflows
- Automatic code generation (multiple languages)
- Easy data visualization
- Comprehensive documentation
- GPU acceleration
- Advanced image processing libraries
- Can be adapted to Python and Napari (with cI Esperanto)



@weichen01.bsky.social



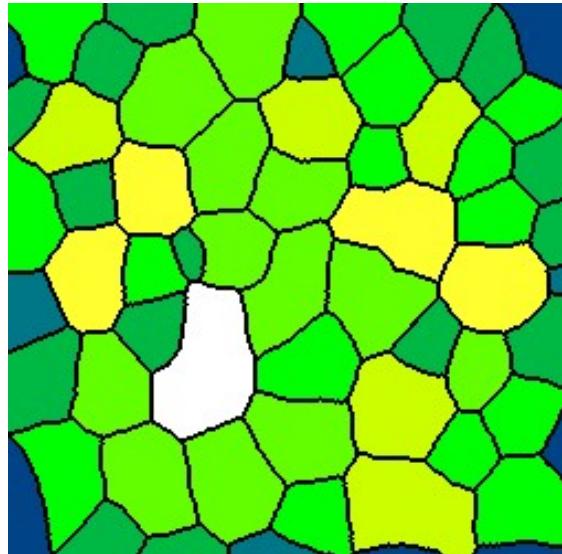
Wei-Chen Chu
ICOB Imaging Core

Advanced image processing libraries

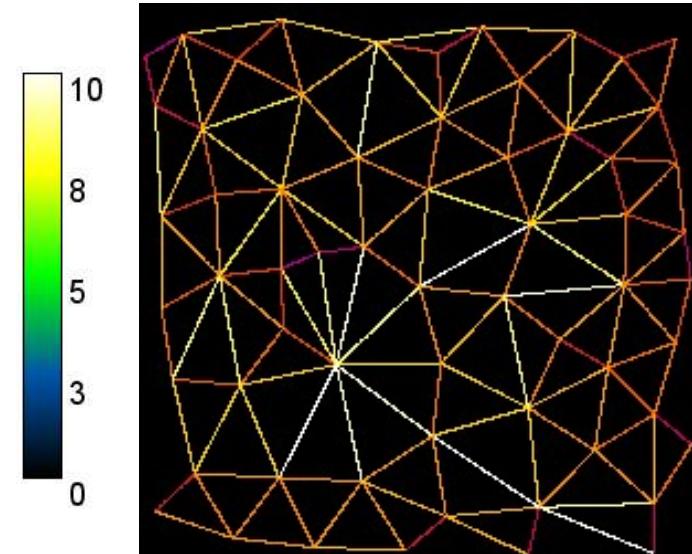
CLIJ2 is particularly effective for studying neighbors



Labels



Touching Neighbor count Map



Distance Mesh between Touching Labels



@weichen01.bsky.social

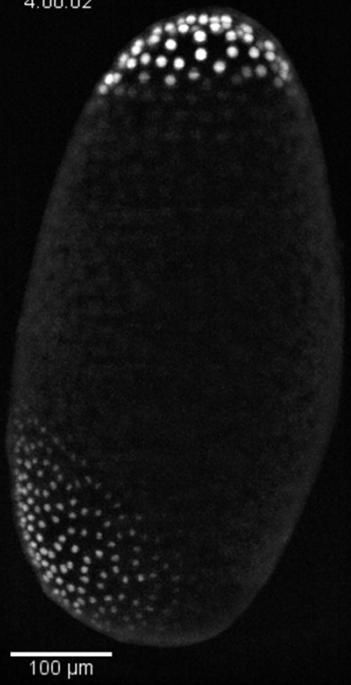


Wei-Chen Chu
ICOB Imaging Core

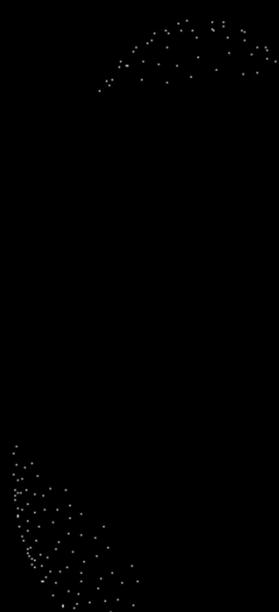
GPU accelerated image processing in life sciences

... to study embryo development

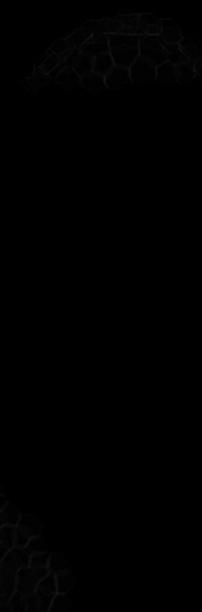
Tribolium castaneum nuclei-GFP,
Background subtracted
4:00:02



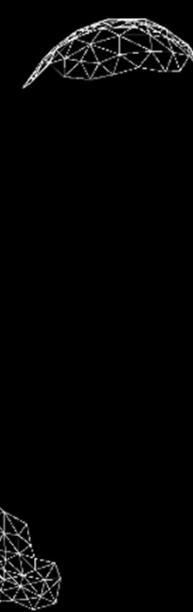
Spot detection (3D)



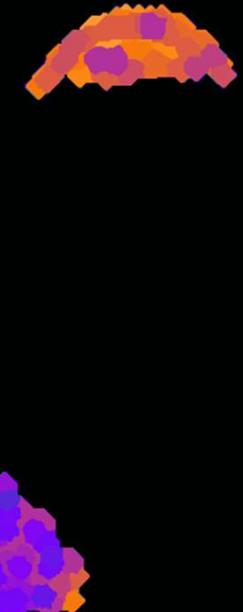
Theoretical membranes
(pseudo Voronoi map)



Neighbor mesh



Average centroid distance of
neighbors



Modified from [Open-Access Training Materials of Dr. Robert Haase](#), licensed [CC-BY 4.0](#)

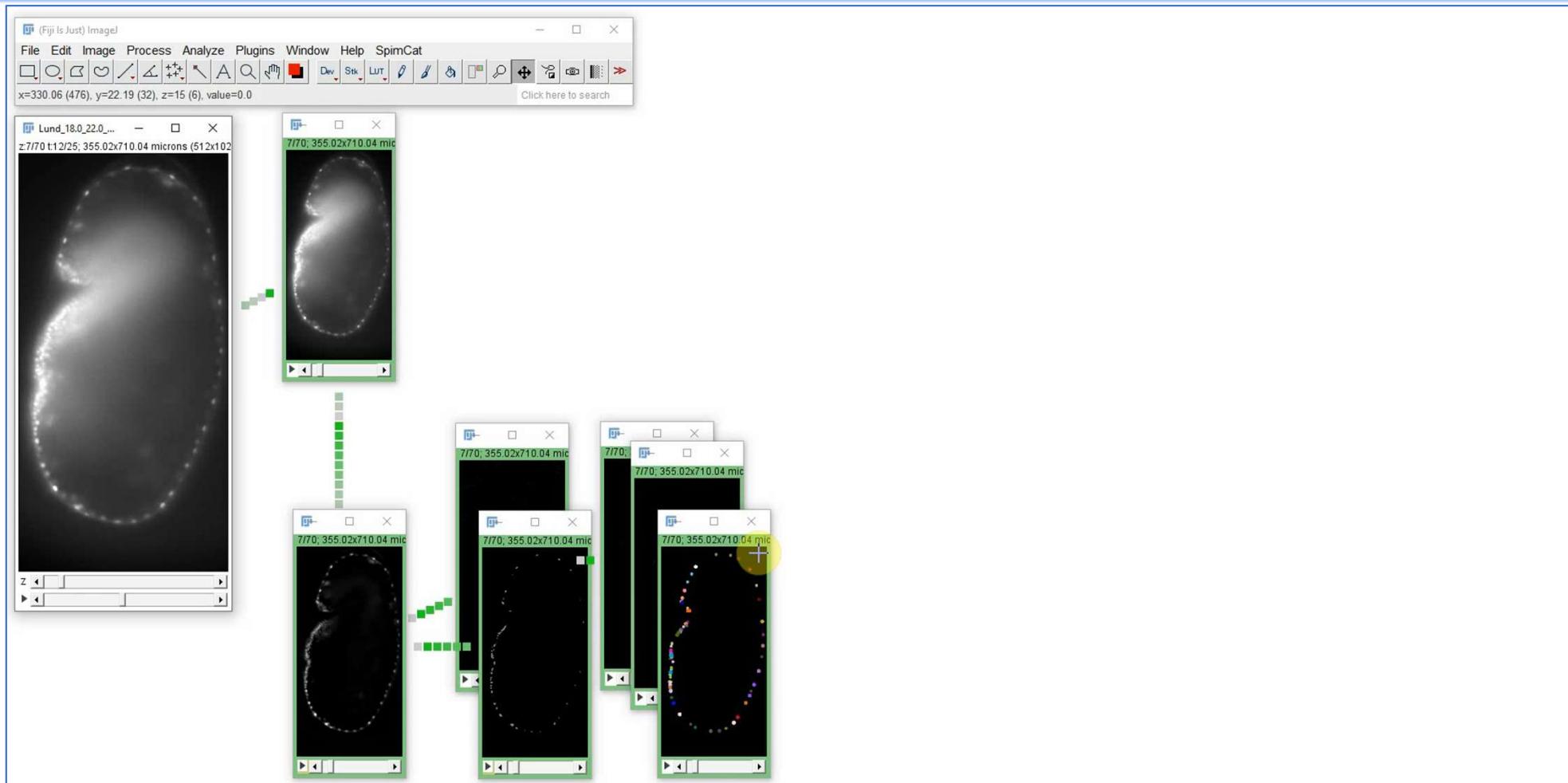


@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

From Fiji/ CLIJ to napari



Adapted from [Open-Access Training Materials of Dr. Robert Haase](#), licensed [CC-BY 4.0](#)



@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

Learn more

- [Introduction to Bioimage Analysis by Peter Bankhead](#)
- [Lecture BioImage Analysis 2020 by Robert Haase – YouTube, Slides](#)
- [Open-Access Training Materials by Robert Haase](#)
- [Bioimage Analysis with FIJI/ImageJ and Friends @ICOB \(2024, Chinese\), Slides](#)
- [Open source AI Tools for bioimage analysis workshop @ICOB \(2024, Chinese\), Slides](#)
- EABIAS: <https://eabias.github.io/>

The screenshot shows the EABIAS website's "Resources" page. The top navigation bar includes links for Home, About EABIAS, Activities, and Resources. On the right, there is a language dropdown menu currently set to English, with Chinese and Japanese options available. Below the navigation, there are two main sections: "Resources" and "Learning Materials". The "Resources" section contains links to the "Bioimage Analysis Text Book", "Bioimage Data Analysis Workflows", and "Bioimage Data Analysis Workflows – Advanced Components and Methods". The "Learning Materials" section contains links to the "EABIAS YouTube Channel", "NTU imaging core YouTube Channel", and "ICOB imaging core learning resource collection".



@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

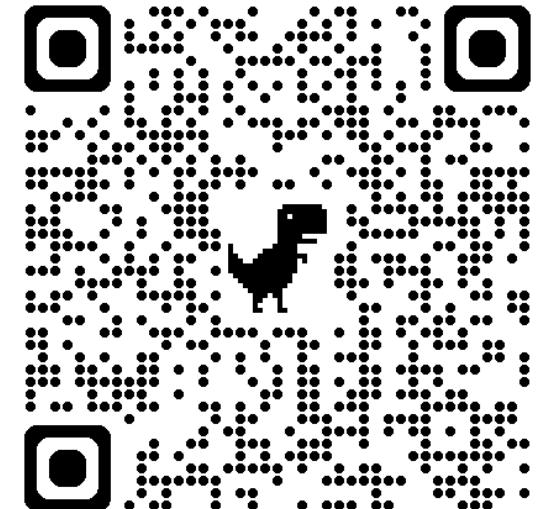
Acknowledgements



EABIAS (East Asia Bioimage Analysts' Society)



Dr. Robert Haase



課後意見調查



國立臺灣大學
National Taiwan University



合大醫學院
National Taiwan University
College of Medicine



NSTC 國家科學及技術委員會
National Science and Technology Council



IBC
ACADEMIA SINICA



neubias
network of european
bioimage analysts

GloBIAS
Global Bioimage
Analysts' Society



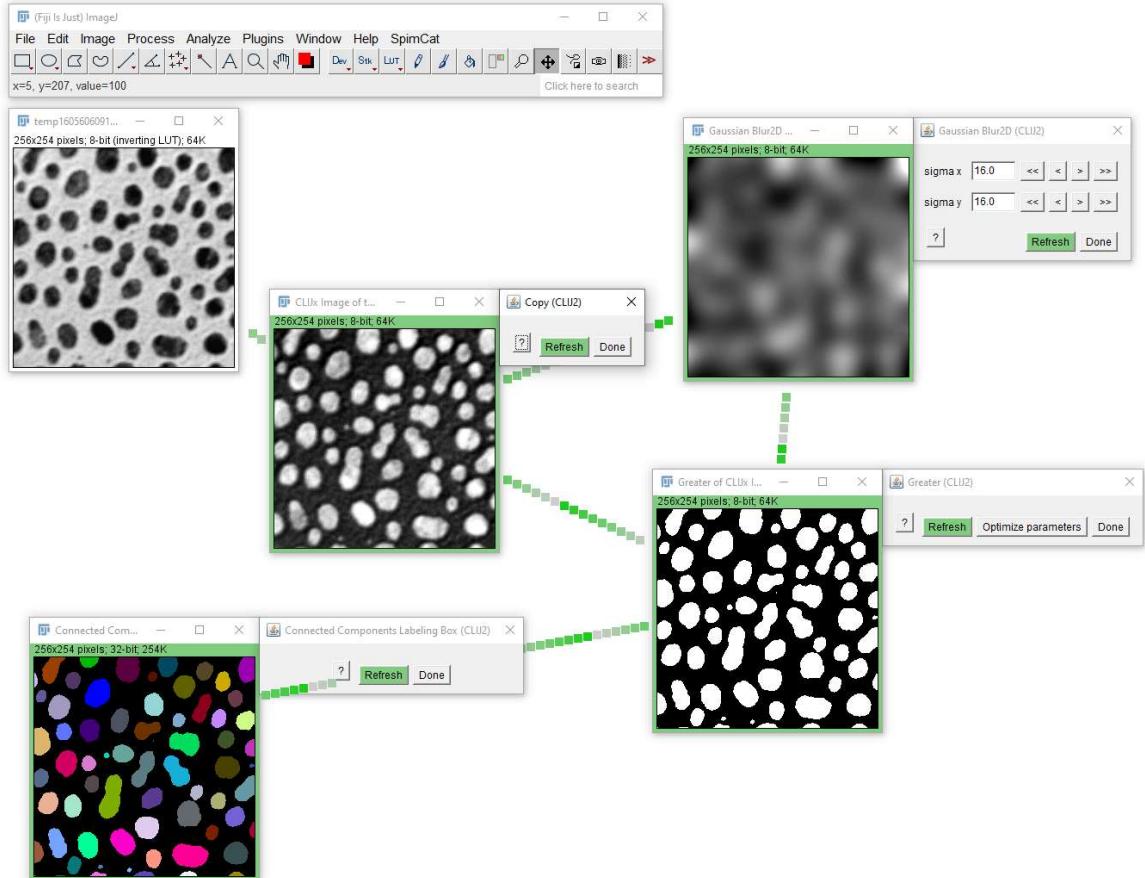
@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

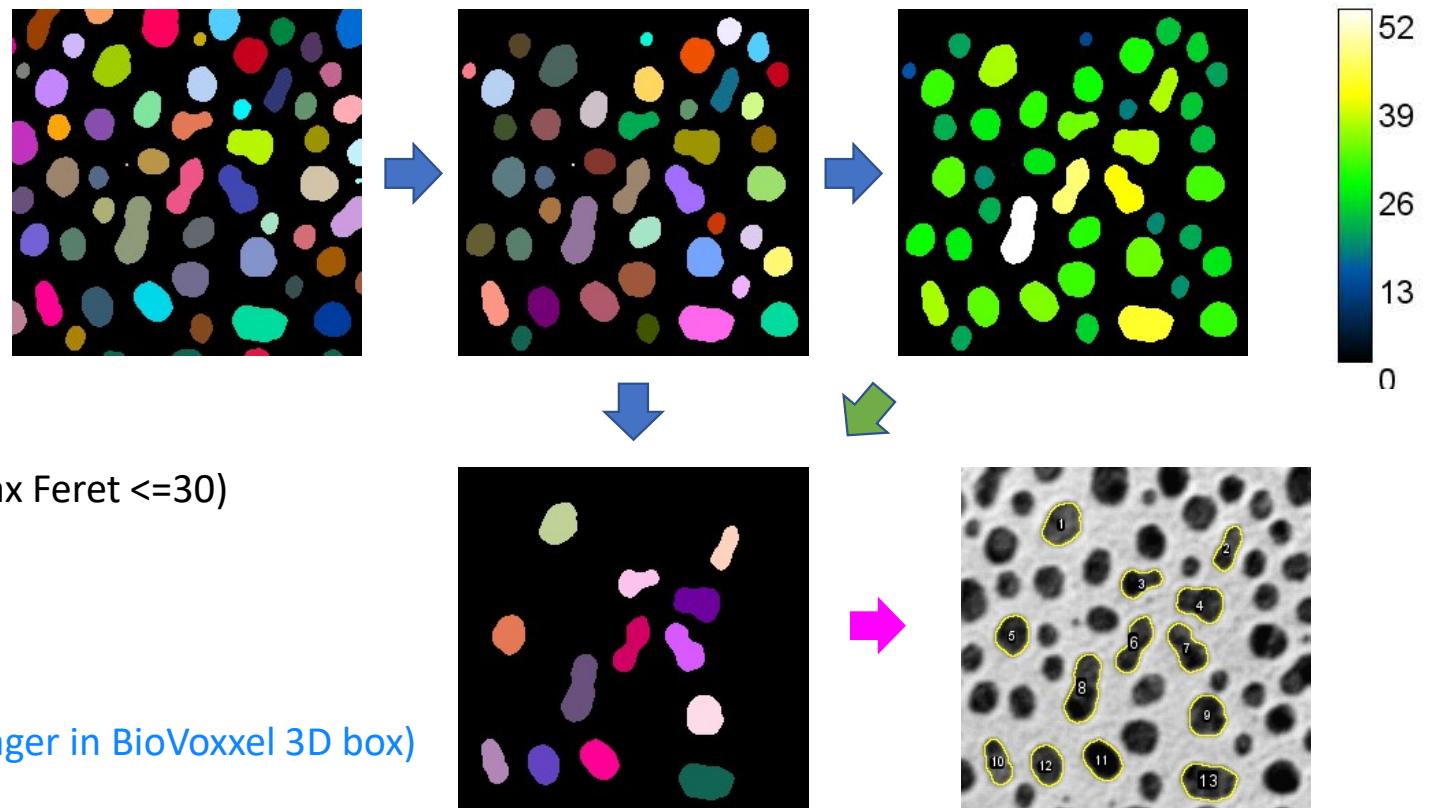
Excise 1: Segment blobs.gif

- Design a workflow for segmenting blobs.gif (File > Open Samples...)
- Export the workflow as ImageJ Macro script
- Refine the ImageJ Macro
- Optional: Export the workflow as Icy Javascript, as Icy protocol and for QuPath as groovy script. Feel free to generate a Fiji plugin.
- Furthermore, if you only used operations, which are "py" compatible, export a clEsperanto-based Jupyter notebook and a Python script that uses Napari.



Excise 2: Label processing

- Load the example label image
(or Segment blobs.gif using StarDist 2D)
- Exclude Labels On Edges
(Hint: using search bar)
- Generate “Maximum Feret Map”
- Exclude the shorter label (e.g Max Feret ≤ 30)
- Generate an ImageJ Macro
- Convert the final labels to ROIs
(Hint: using Label to 2d Roi manager in BioVoxcel 3D box)
- Refine the macro to allow you to repeat this workflow
(Hint: you can also combine the Macro recorder)

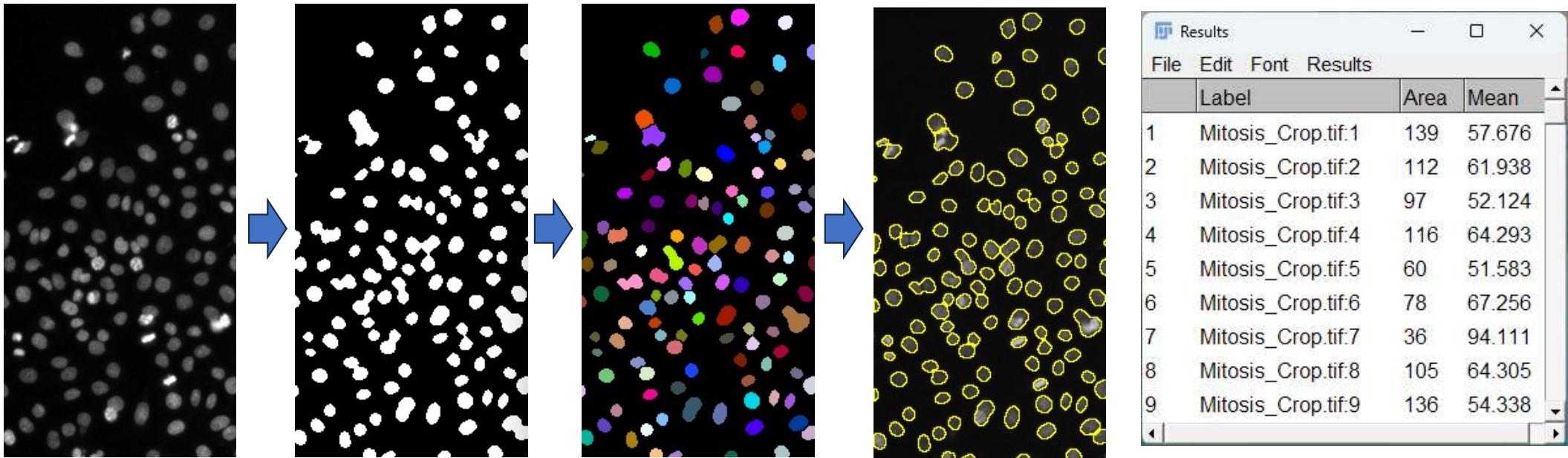


@weichen01.bsky.social



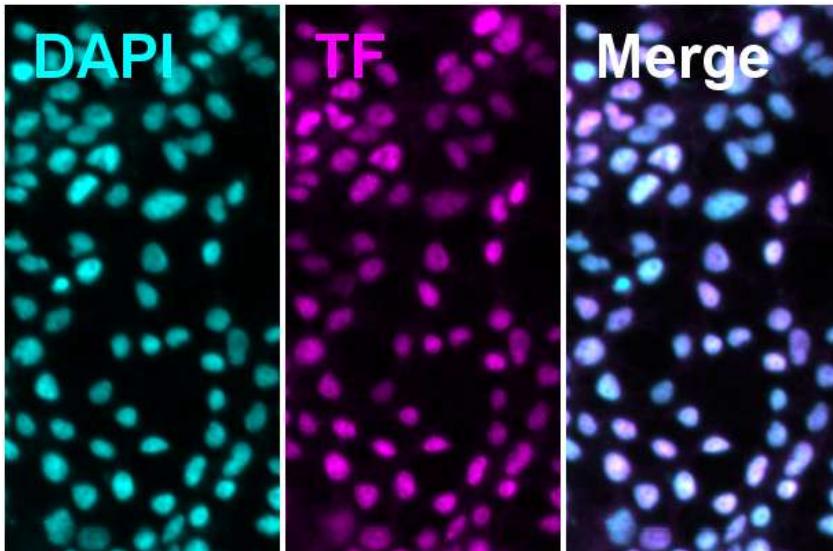
Wei-Chen Chu
ICOB Imaging Core

Single channel thresholding issue on intensity measurement



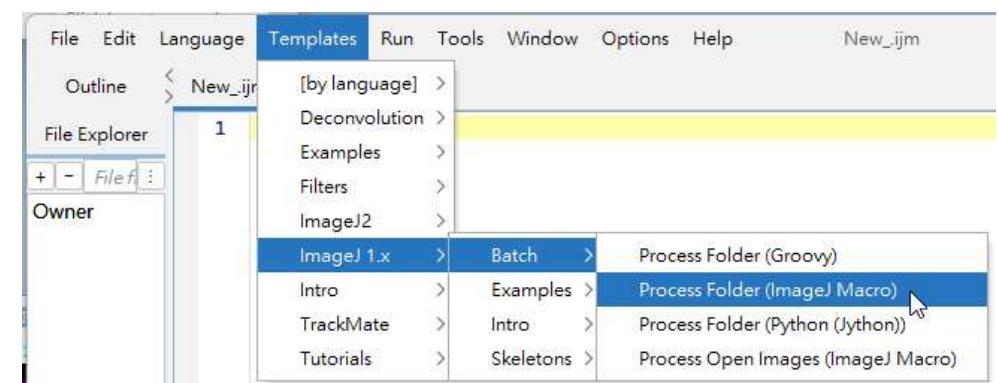
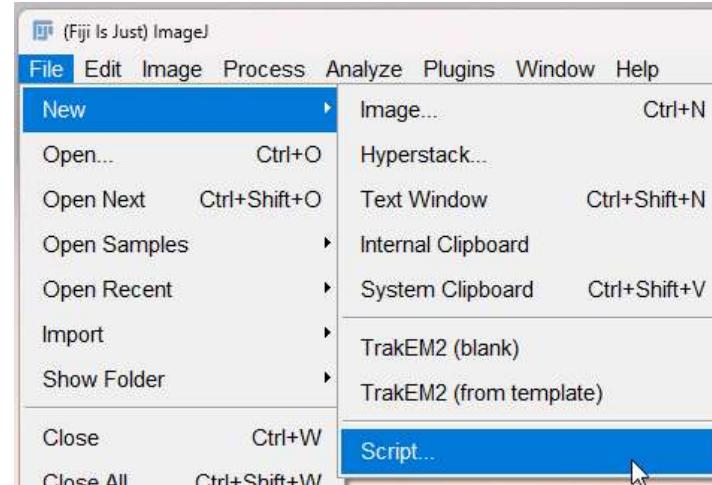
- The segmented area is influenced by intensity variations.
How can we accurately quantify "intensity" in this context?
- To ensure precise measurement of the target signal's intensity, the region of interest (ROI) should be segmented based on a **separate channel**.

Exercise3: Full Workflow and Batch Processing



Goal :

- Measure the area and transcription factor (TF) signal intensity based on DAPI segmentation.
- Export an ImageJ macro using CLIJ, then refine it to ensure reusability across different image files.
- **Integrate your code into batch processing templates.**
- **(Hint: Split channels first! You can also use the Macro Recorder and refer to the ImageJ Cheat Sheet.)**



ImageJ macro cheat sheet

Macro language elements

```
// comments for code documentation
numericVariable = 5;
stringVariable = "text value";
builtInCommand();
```

Switch between image windows

```
titleOfCurrentImage = getTitle();
selectWindow(titleOfAnyImage);

Navigation in image stacks
Stack.getDimensions(width, height,
channels, slices, frames);
Stack.setSlice(slice);
Stack.setChannel(channel);
Stack setFrame(frame);
Stack.setDisplayMode("color");
Stack.setDisplayMode("composite");
Stack.setDisplayMode("grayscale");
```

Keep in mind:

- Only one active window!
- One activate channel, slices!
- One ROI list
- One result table!

Handle image files and folders

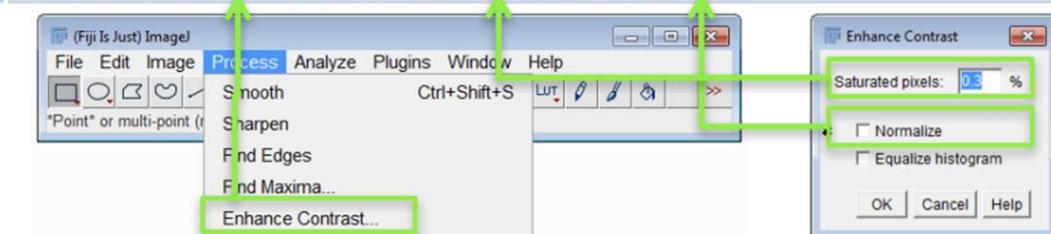
```
open(folder+imageFilename);
close();
fileList = getFileList(folder);
numFiles = lengthOf(fileList);
for (i=0;i<lengthOf(fileList);i++){
    file = fileList[i];
    open(file);
    // actual image processing...
    close();
}
```

Reading image calibration

```
getPixelSize(unit, pWidth, pHeight);
getVoxelSize(vWidth, vHeight,
vDepth, unit);
```

Calling any ImageJ/FIJI menu

```
run("Enhance Contrast...", "saturated=0.3 normalize")
```



Result tables

```
run("Set Measurements...", "area
mean standard min centroid");
corresp. to this:
    Area          Mean gray value
    Standard deviation  Modal gray value
    Min & max gray value Centroid
run("Analyze Particles...", "add clear display");
roiManager("Measure");
rowCount = nResults();
value = getResult("column title",
rowNumber);
setResult("column title",
rowNumber, newValue);
saveAs("Results", "myResults.xls");
run("Clear results");
```

ROI manager

```
roiManager("add");
roiManager("split");
roiManager("delete");
roiManager("reset");
roiManager("measure");
roiManager("count");
roiManager("open", filename);
roiManager("save", filename);
roiManager("save selected", filename);
roiManager("select", index);
roiManager("select", newArray(index1,
index2, ...));
roiManager("deselect");
roiManager("show all");
roiManager("show all with labels");
roiManager("show none");
roiManager("and");
roiManager("combine");
```



CLIJ2 cheat sheet

CLIJ2 cheat sheet: ImageJ macro I GPU-accelerated image processing in Fiji

CLIJ2 cheat sheet: ImageJ macro II GPU-accelerated image processing in Fiji

CLIJ2 cheat sheet: ImageJ macro III GPU-accelerated image processing in Fiji

CLIJ2 cheat sheet: ImageJ macro IV GPU-accelerated image processing in Fiji

CLIJ2 cheat sheet: ImageJ macro II
GPU-accelerated image processing in Fiji

Operation **Parameters** **Result** **Dim**

| Operation | Parameters | Result | Dim |
|----------------------|--|----------|-----|
| Initialize CLIJ | [] HD, GPF or CPU | | 4D |
| Push | | 2D 3D | 4D |
| Pull | | 2D 3D | 4D |
| Create | 1024, 1024, 8 | 2D 3D | 4D |
| Convert | | 2D 3D | 4D |
| Copy | | 2D 3D | 4D |
| Copy slice | 50 | 2D 3D | 4D |
| Crop | 20, 20 | 2D 3D | 4D |
| Paste | 9, 9 | 2D 3D | 4D |
| Release | | 2D 3D | 4D |
| Clear | | 2D 3D | 4D |
| Rotate by 90 degrees | | 2D 3D | 4D |
| Rotate | , 45, true | 2D 3D | 4D |
| Flip | , true, false | 2D 3D | 4D |
| Translate | , 20, 20 | 2D 3D | 4D |
| Affine transform | "center", "center", "center", "center" | 2D 3D | 4D |
| Deform / warp | | 2D 3D | 4D |
| Projections | | 2D 3D | 4D |

<https://clij.github.io/clij2-docs> [@haesleinhuepf](#)

Filters

| Operation | Parameters | Result | Dim | Examples |
|------------------------|----------------|----------|--|----------|
| Gaussian blur | , 10, 10 | 2D 3D | Ext.CLIJ2_gaussianBlur2D(input, result, sigmaX, sigmaY); | |
| Difference of Gaussian | , 2, 2, 20, 20 | 2D 3D | Ext.CLIJ2_differenceOfGaussian2D(input, result, sigmaX, sigmaY, sigma2X, sigma2Y); | |
| Invert | | 2D 3D | Ext.CLIJ2_invert(input, result); | |
| Laplace | | 2D 3D | Ext.CLIJ2_laplaceBox(input, result); | |
| Mean | , 5, 5 | 2D 3D | Ext.CLIJ2_mean2DBox(input, result, radiusX, radiusY); | |

Result **Dim** **Examples**

| | |
|---|----------|
| Ext.CLIJ2_spotsToPointList(binary_spots, pointlist); Ext.CLIJ2_labelledSpotsToPointList(labelled_spots, pointlist); | 2D 3D |
| Ext.CLIJ2_generateDistanceMatrix(pointlist1, pointlist2, distance_matrix); | 2D 3D |
| Ext.CLIJ2_generateTouchMatrix(label_map, touch_matrix); | 2D 3D |
| Ext.CLIJ2_touchMatrixToLabelMap(pointlist, touch_matrix, mask); | 2D 3D |
| Ext.CLIJ2_meanFromTouchingNeighbors(values, touch_matrix, mean_value); | 2D 3D |
| Ext.CLIJ2_countTouchingNeighbors(touch_matrix, count_vector); | 2D 3D |
| Ext.CLIJ2_statisticsOfBackgroundAndLabeledPixels(image, labelmap); Ext.CLIJ2_statisticsOfLabeledPixels(input, labelmap); | 2D 3D |
| Ext.CLIJ2_pushResultsTable(image_name); | 2D 3D |
| Ext.CLIJ2_pushResultsTableColumn(image_name, column_name); | 2D 3D |
| Ext.CLIJ2_pullResultsTable(image_name); | 2D 3D |
| CLIJ2_pushArray(image_name, array, width, height, depth); | 2D 3D |

Installation instructions

- Install CLIJ2 by activating the "clij" and "clij2" update sites in Fiji.
- Commands listed as "CLIJ" or "CLIJ2" can be handled with either. They may change or disappear at any point. To build reliable, reproducible workflows use CLIJ or CLIJ2 commands only.

docs [@haesleinhuepf](#) #clij cheat sheet 2020-04-20

Adapted from [Open-Access Training Materials of Dr. Robert Haase](https://clij.github.io/clij2-docs/CLIJ2-cheatsheet_V3.pdf), licensed CC-BY 4.0
https://clij.github.io/clij2-docs/CLIJ2-cheatsheet_V3.pdf



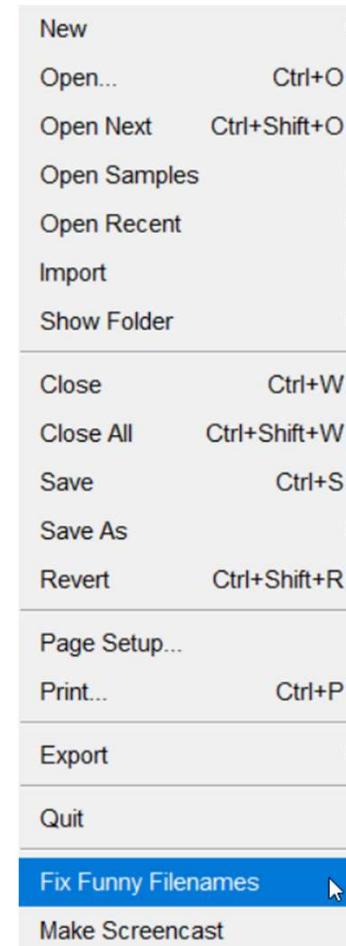
@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

Bonus: Fix funny filename

- File -> Fix Funny Filenames
- Rename files with problematic file or folder names automatically
image 001.tif → image_001.tif
- Try it if your ImageJ macro can not find you file or folder!



@weichen01.bsky.social



Wei-Chen Chu
ICOB Imaging Core

Bonus: How to name files

good file names are ...

- **machine readable**
- **human readable**
- **sorted in a useful way**

 pos.it/how-to-name-files

 @JennyBryan

 @jennybc

 @jennybryan@fosstodon.org

