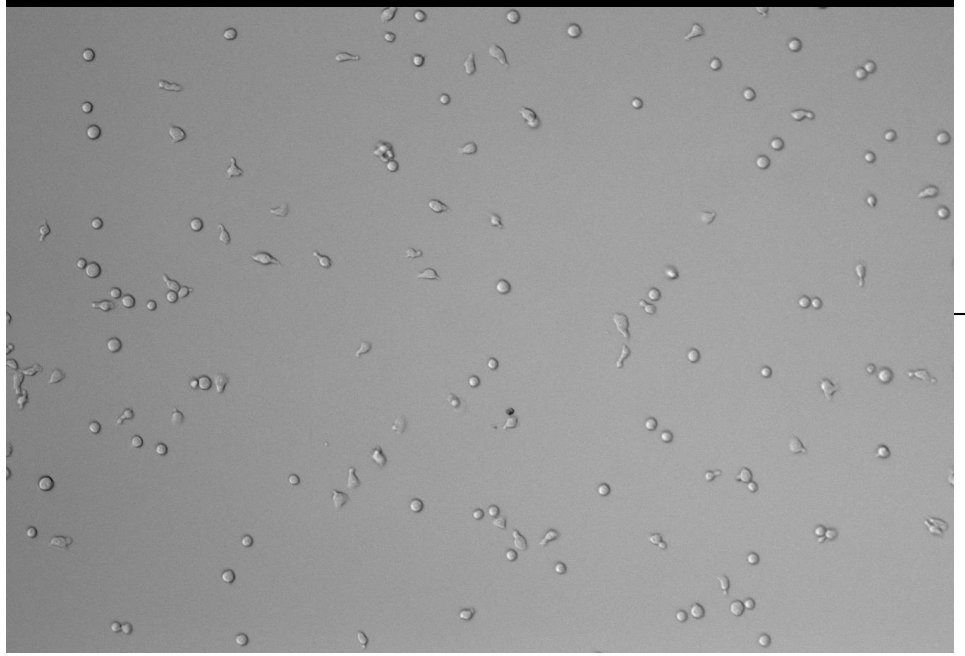
DEEP LEARNING USING BIAPY

Hands on training

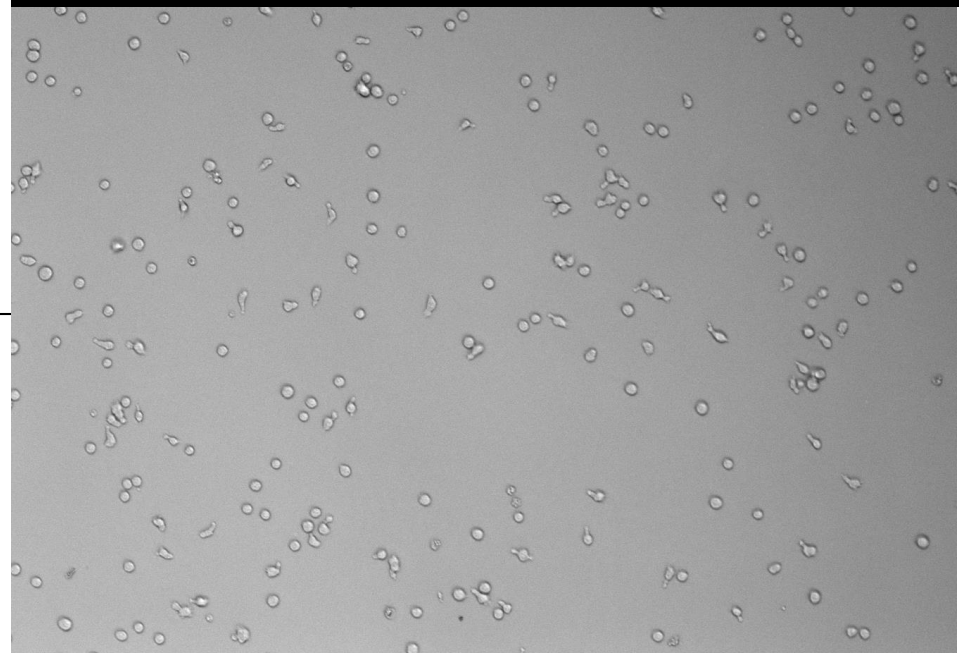
OUR TASK ON NEXT TWO DAYS

How does the coating on glass bottom dishes affect the migration of T-cells?

ICAM



VCAM



ICAM-1 (Intercellular Adhesion Molecule-1):
Binds to **LFA-1,** an integrin expressed on T cells.

- Typically mediates **firm adhesion and rapid crawling**.
- Promotes **polarized morphology** and fast, exploratory migration.
- Often associated with surveillance in non-inflamed tissues.

VCAM-1 (Vascular Cell Adhesion Molecule-1):
Binds to **VLA-4,** another integrin on T cells (also called $\alpha_4\beta_1$ integrin).

- Often leads to **slower, more stable interactions**.
- Induces **different signaling cascades,** sometimes associated with T cell retention or transmigration.
- More prominently expressed in **inflamed endothelium,** guiding T cells into sites of inflammation.

Steps for solving the task

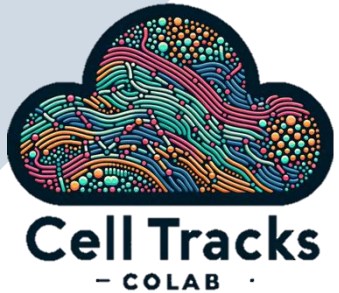
Train a deep learning model to segment cells

Cell tracking

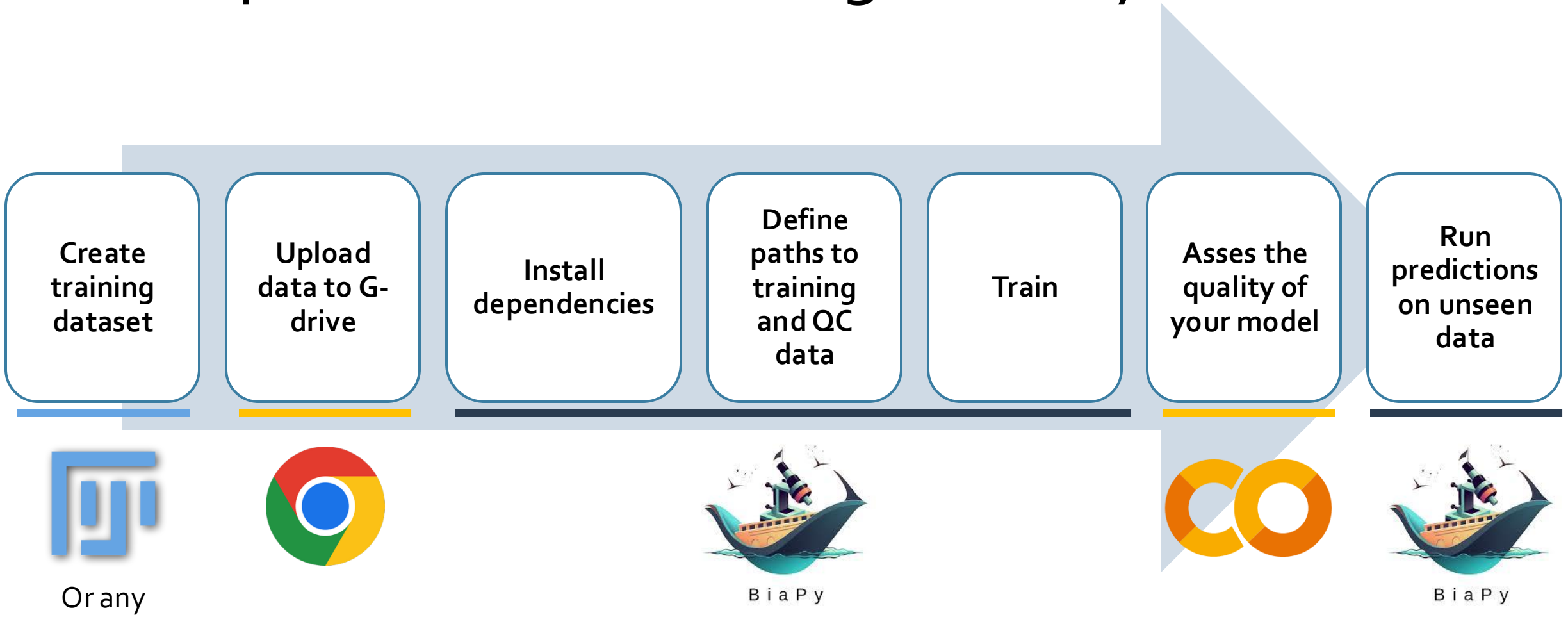
(tomorrow with Jean-Yves)

Track analysis

(tomorrow with Laura)



Steps for model training in BiaPy



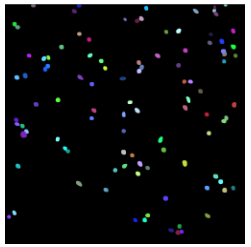
How to create a training dataset for StarDist?

- Open Fiji – activate LOCI update site
- Restart Fiji
- Open your image to be annotated *Image1-T-cells*
- Select the Oval  or Freehand selection tool 
- Start drawing selections around each (yes each!) cell
- After every cell press t -> the selection will be stored to the ROI manager
- Repeat until all cells are in the ROI manager
- When done *Plugins › LOCI › ROI Map*
- Save the generated ROI map with the same title as the original image
- At the end you have one folder with the original images, another with the ROI maps

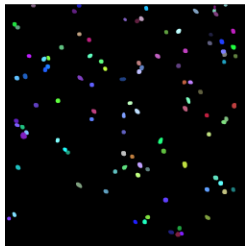
Modified from Fazeli et al., 2020

Suggested folder structure for BiaPy

- 1-train
  - 1-raw
  - 2-label
- 2_test
  - 1-raw
  - 2-label
- 3_model
- 4_own_data
- 5_own_data_results

200 image pairs, 10 % for validation

10 QC image pairs

Data to be segment

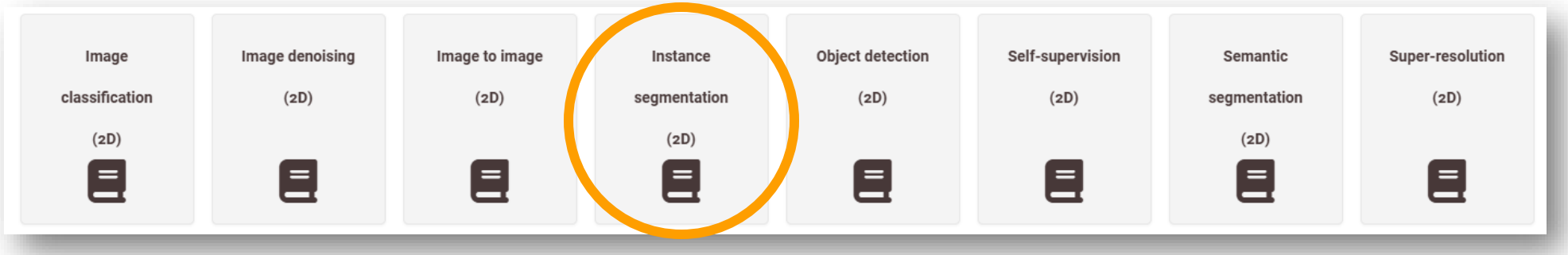
Your results will be here

Do you all have Google drives?

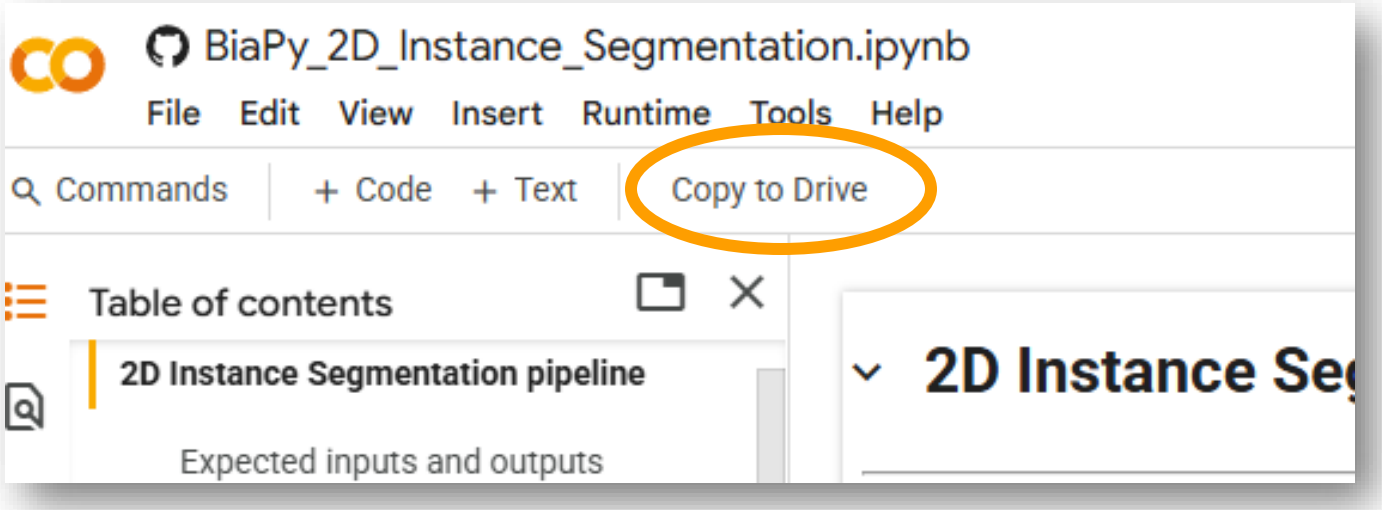
- Log in to your google account
- Make sure the workshop images are uploaded
- Using the same browser go to https://biapyx.github.io/

Open the notebook

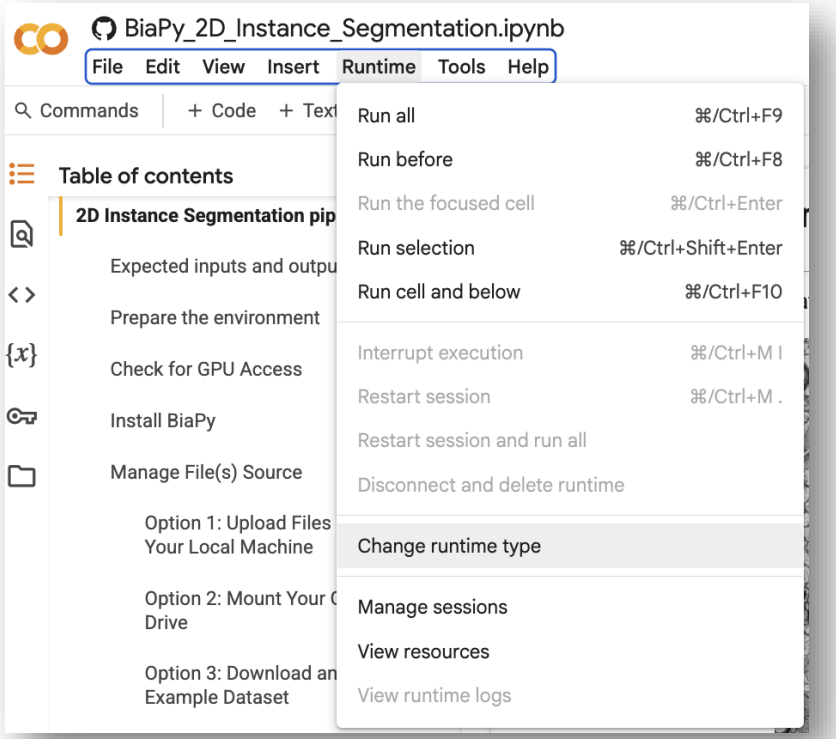
- Look for the notebook, called Instance segmentation (2D) and open it

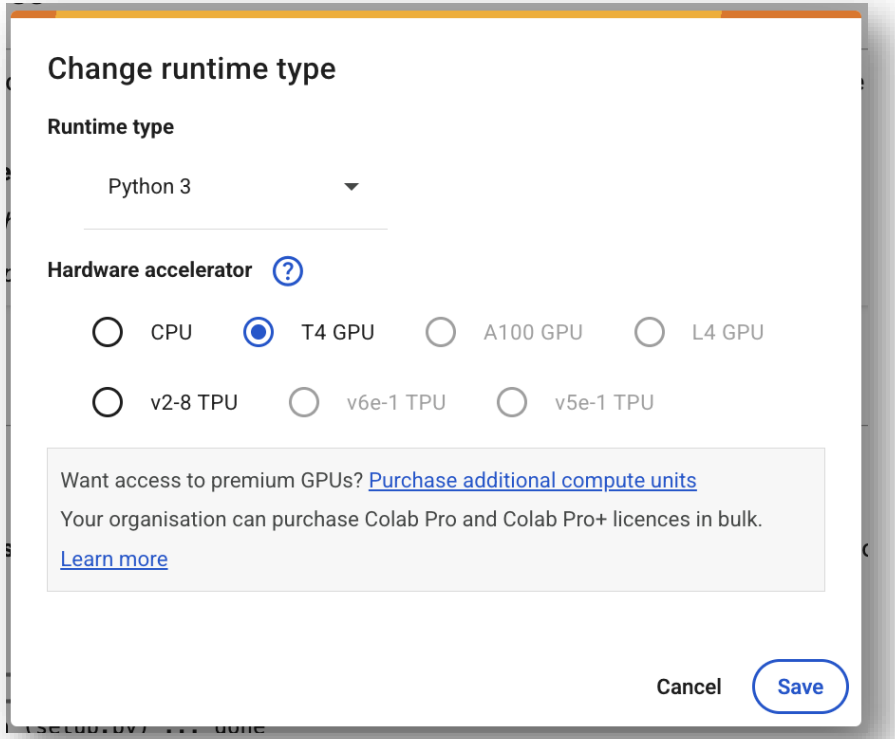


- Copy it to your drive and give it a name – close the original notebook



Prepare the session





Check for GPU access!!!
Go to **Runtime -> Change the Runtime type to T4 GPU**

- Play to reinstall Colab libraries so they are compatible wiht BiaPy
- Play to install BiaPy and its dependences

Connect to Google drive and load data

Manage File(s) Source

The input folder can be provided using three different options:

1. **Direct Upload**: Directly upload the desired folder.
2. **Google Drive**: Use a folder stored in your Google Drive.
3. **Sample Data**: Use a sample dataset provided by us.

Play the cell to connect your Google Drive to Colab

- Click on the URL.
- Sign in your Google Account.
- Copy the authorization code.
- Enter the authorization code.
- Click on "Files" site on the right. Refresh the site. Your Google Drive folder should now be available here as

Path to train raw images

`train_data_path:` /content/gdrive/MyDrive/Pasteur_Image_analysis_course_2025_BiaPy/1-train/1-raw

Path to train label images

`train_data_gt_path:` /content/gdrive/MyDrive/Pasteur_Image_analysis_course_2025_BiaPy/1-train/2-label

Path to test raw images

`test_data_path:` /content/gdrive/MyDrive/Pasteur_Image_analysis_course_2025_BiaPy/2_test/1-raw

Path to test label images (optional)

`test_data_gt_path:` /content/gdrive/MyDrive/Pasteur_Image_analysis_course_2025_BiaPy/2_test/2-label

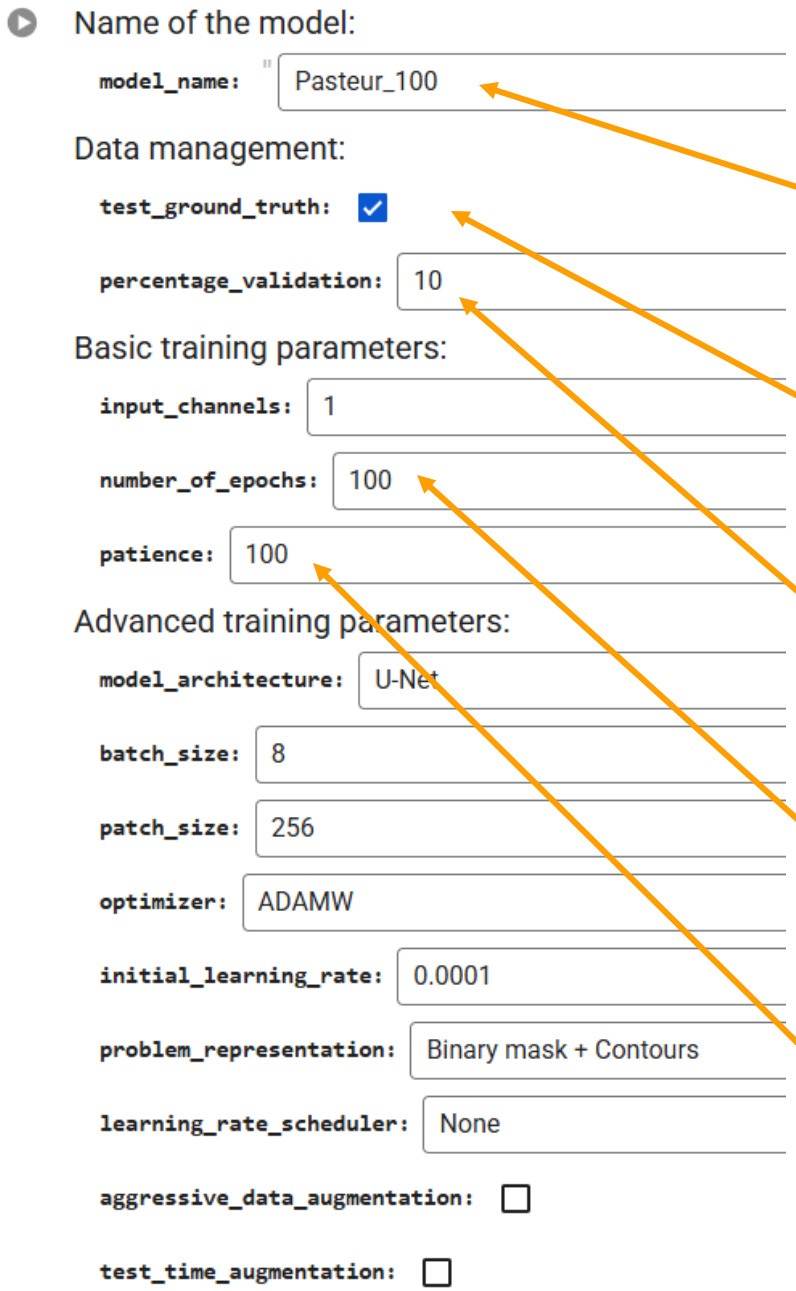
Path to store the resulting images (it'll be created if not existing):

`output_path:` /content/gdrive/MyDrive/Pasteur_Image_analysis_course_2025_BiaPy/3_model

```
Number of training raw images: 200
Number of training label images: 200
Number of test raw images: 10
Number of test label images: 10
```

Play to visualize some data samples

Training parameters



model_name: Use only my_model -style, not my-model (Use "_" not "-"). Do not use spaces in the name. Avoid using the name of an existing model as it will be overwritten.

test_ground_truth: Select to use test data ground truth to measure the performance of the model's result. If selected, test_data_gt_path variable path set above will be used. Default value:

percentage_validation: Input the percentage of your training dataset you want to use to validate the network during the training. Default value: 10

number_of_epochs: Input how many epochs (rounds) the network will be trained. For the example dataset, reasonable results can already be observed after 100 epochs. **Default value: 100**

patience: Input how many epochs you want to wait without the model improving its results in the validation set to stop training. Default value: 20

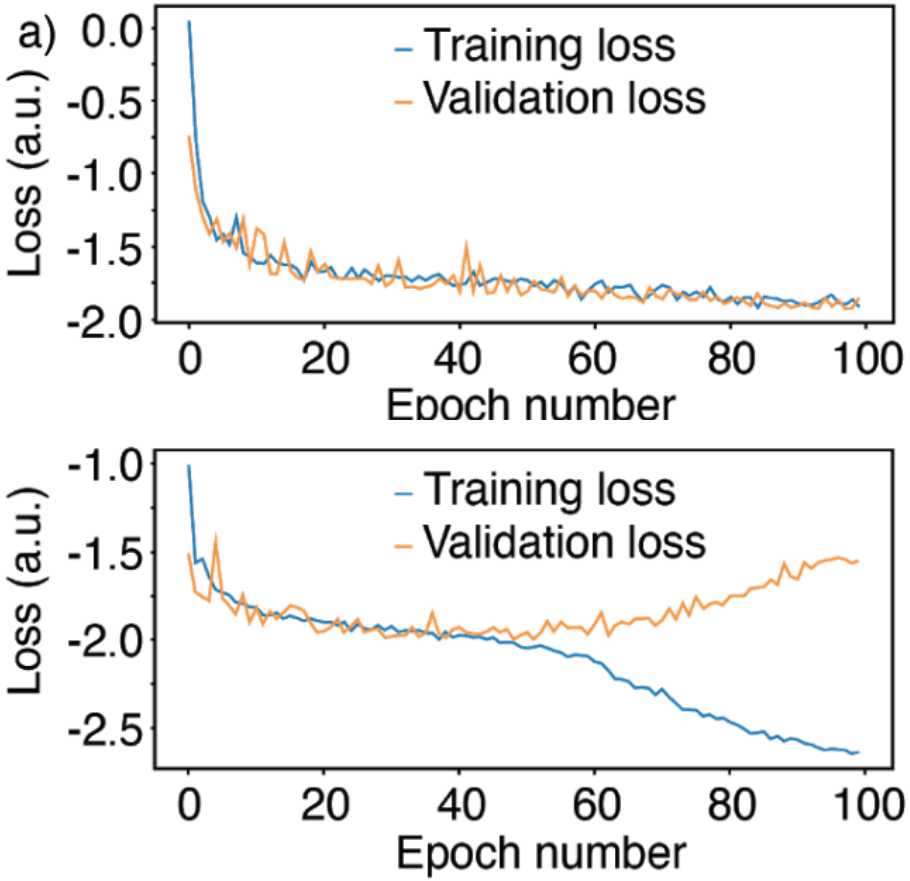
Train the model

Play to train the model

RUN QUALITY CONTROL ON YOUR MODEL

Quality Control1:
Inspection of the loss function

- Train model for hundreds of EPOCH

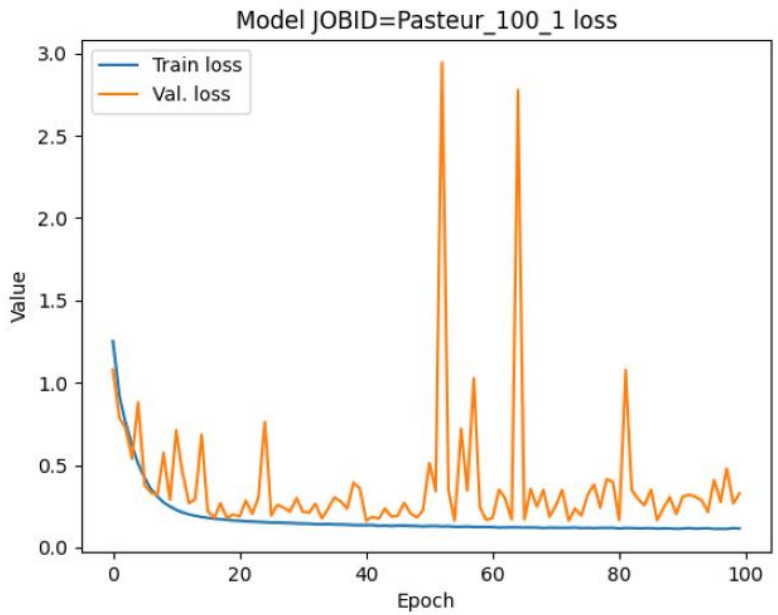


- Always evaluate the training progress, compare the training loss with the validation loss.
- **Training loss** describes an error value after each epoch for the difference between the model's prediction and its ground-truth target.
- **Validation loss** shows how well the network performs on the validation data.
- **Actions:**
- Decreasing Training loss and Validation loss
  - train for more epocs
- Curves are flattened out:
  - no more training is required
- Validation loss increases while Training loss simultaneously decreases:
  - network is overfitting = remembering the exact patterns from the training data and no longer generalizes well to unseen data.
  - training dataset should be increased.

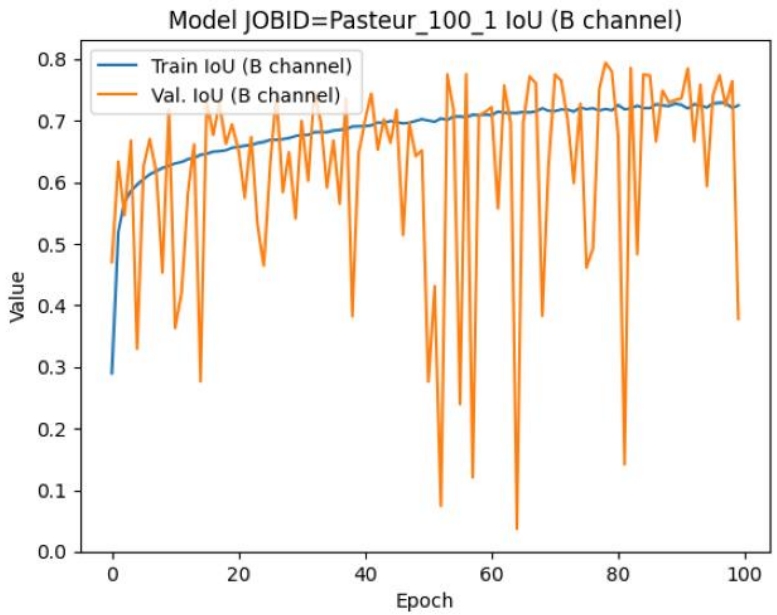
Inspection of the Loss Function and the Intersection over Union (IoU)

Play the cell to show a plot of training error vs. epoch number and IoU vs epoch number

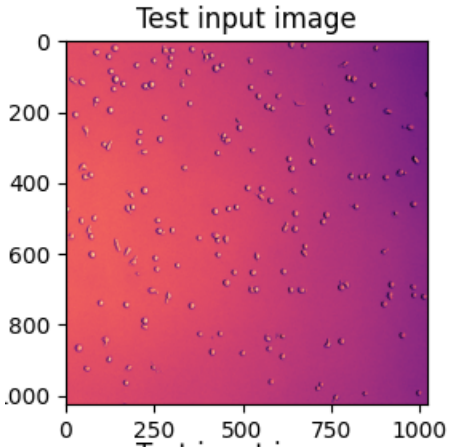
Training error vs epoch number

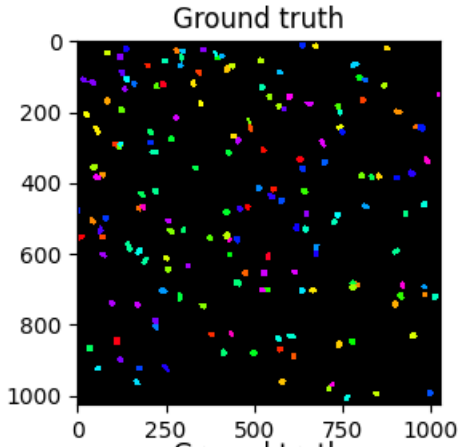


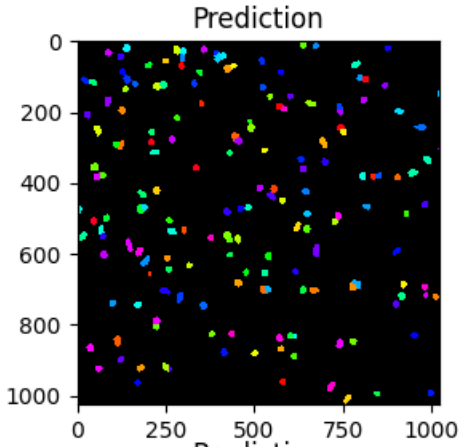
Intersection over Union (IoU) vs epoch number

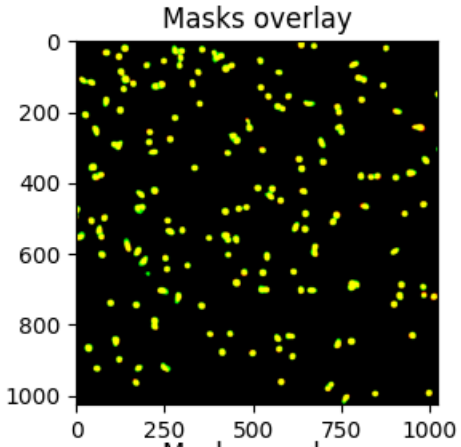


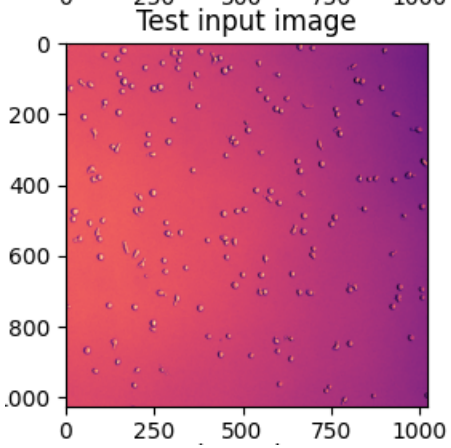
Quality Control2:
Visual inspection

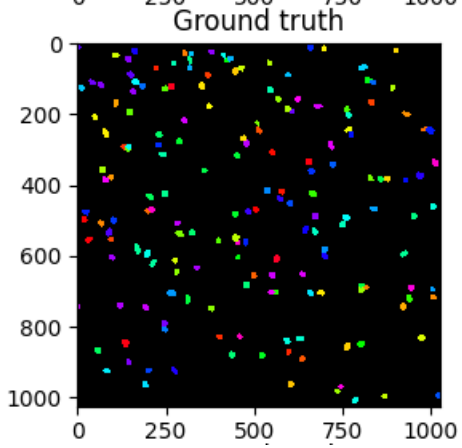


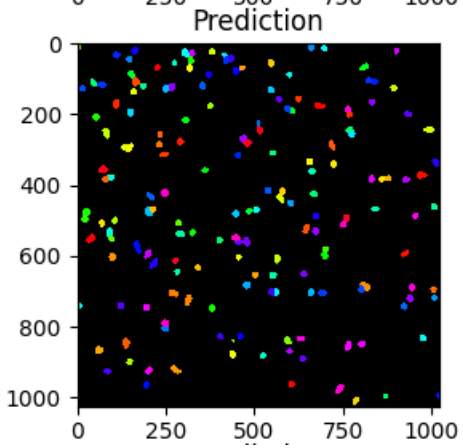


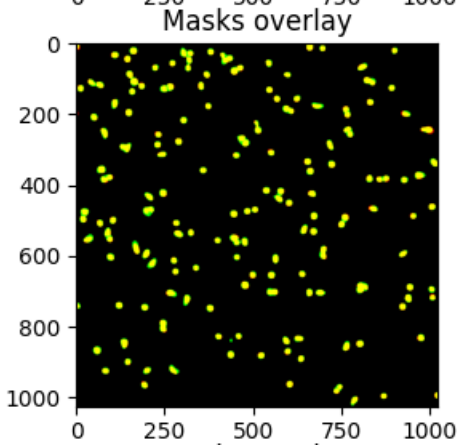












Quality Control3:
Error mapping and quality metrics estimation

GitHub

Link to Quality conrol notebook

Open in Colab

Quality control notebook for Pasteur 2025 course

Play the cell to connect your Google Drive to Colab

- Click on the URL.
- Sign in your Google Account.
- Copy the authorization code.
- Enter the authorization code.
- Click on "Files" site on the right. Refresh the site. Your Google Drive folder should now be available here as "drive".

Show code

Install dependencies

Show code

Load dependencies

Show code

Conduct QC on test data

```
gt_labels: "/content/gdrive/MyDrive/Pasteur_Image_analysis_course_2025_BiaPy/2_test/2-label"
predicted_labels: "/content/gdrive/MyDrive/Pasteur_Image_analysis_course_2025_BiaPy/3_model/Pasteur_100/results/Pasteur_100_1/per_image_instances"
QC_folder: "/content/gdrive/MyDrive/Pasteur_Image_analysis_course_2025_BiaPy/5_own_data_results/Pasteur_100/QC"
```

Show code

Quality Control3:
Error mapping and quality metrics estimation

image name	Prediction v. GT Intersection over Union	false positive	true positive	false negative	precision	recall	accuracy	f1 score	n_true	n_pred	mean_true_score	mean_matched_score	panoptic_quality
Training_source_ICAM 1-1.tif	0.89414007	2	153	2	0.987096	0.98709	0.974522	0.987096	155	155	0.873075	0.884487	0.873075

IoU: percent overlap between the target mask and your prediction output. **The closer to 1, the better the performance.** This metric can be used to assess the quality of your model to accurately predict nuclei. (whole image)

- "true positive" = When a segmented object has an IuO above 0.5 (compared to the corresponding ground truth)
- "**false positive**" = "n_pred" - "true positive"
- "false negative" = "n_true" - "true positive"

The mean_matched_score is the mean IoUs of matched true positives.

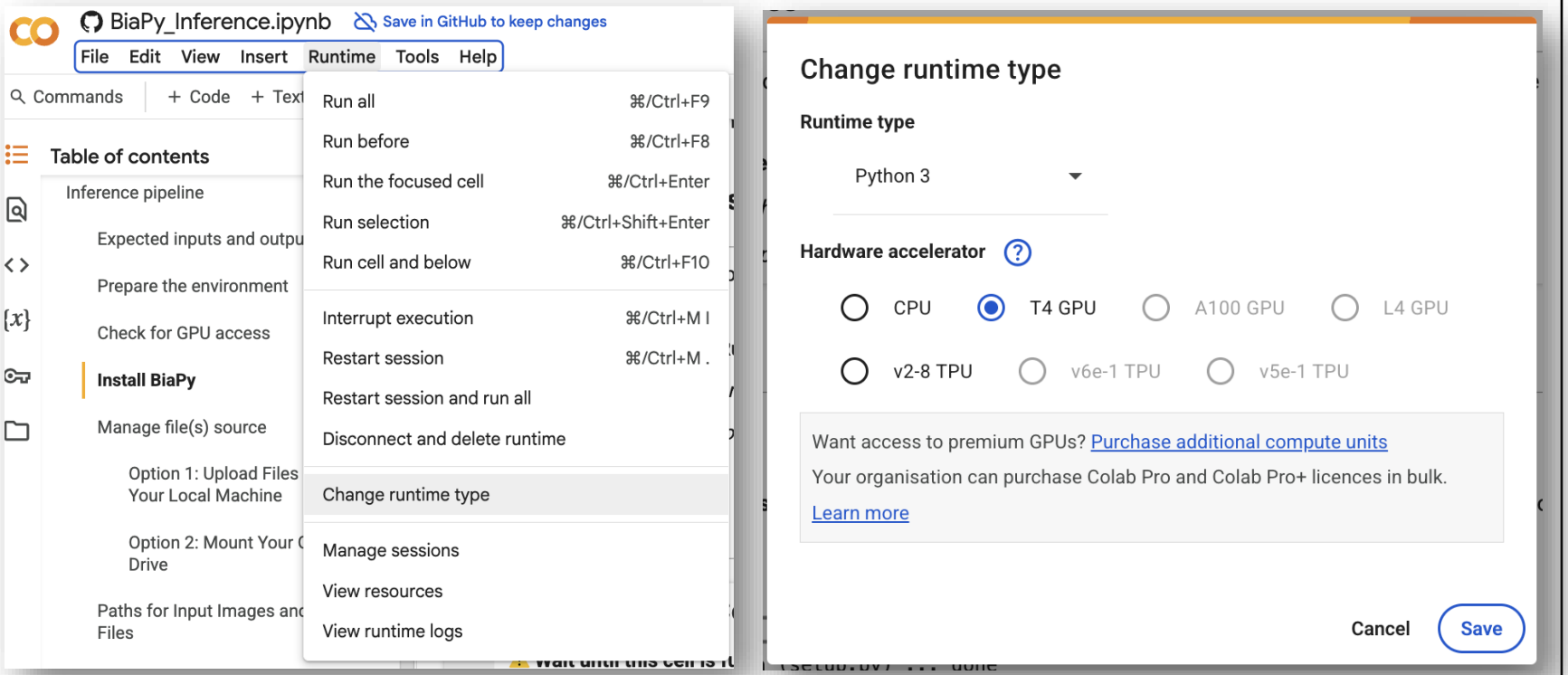
f1 score = 2 * the Area of Overlap divided by the total number of pixels in both images.

Choose a model you want to use for unseen data

- Your own model
- Pasteur_100

Apply to your own data

Using the same browser go to
https://biapyx.github.io/



Check for GPU access!!!
Go to **Runtime -> Change the Runtime type to T4 GPU**

Working with timelapse data

Optional: Converting 3D data into 2D

Optionally, you can use this cell to convert your 3D images or videos into 2D images by extracting individual slices or frames. This enables compatibility with a trained 2D model. If this cell is run the paths to these new data will be used automatically.

Path to store the 2D images generated from existing 3D

`output_2D_slice_folder:` /content/3D_image_slices

Path to store the 2D ground truth images generated from existing 3D (if exists)

`output_gt_2D_slice_folder:` /content/3D_gt_image_slices

Play to generate 2D images from 3D

Optional: Converting 2D slices into 3D again

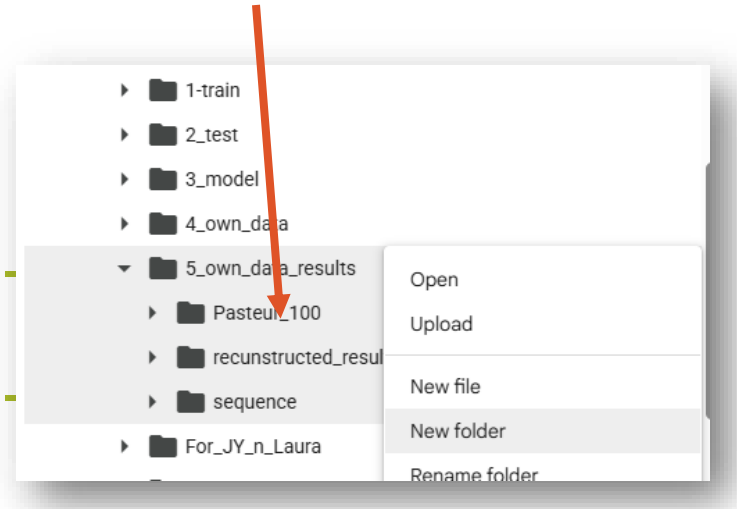
If you used the optional cell above to generate 2D images from 3D data, you can use this cell to reassemble the predicted 2D slices back into the original 3D image.

Path to store the reconstructed 3D images:

`reconstructed_stack_folder:` /content/reconstructed_3D_stacks

Play to generate 2D images from 3D

Create two new folders inside the 5_own_data_results folder_
- sequence
- reconstructed_results



AND WE ARE DONE ☺