cytiva

IN Carta analysis software

SINAP instructions

User Manual

cytiva.com

Table of Contents

1 Important user information

Glossary

The following table explains the meaning of the terms and abbreviations used in this manual.

Term	Definition
Annotation	User created labeling (target/background) of input data to define areas of interest
Background	Area of an image that is not of interest to the user
Base Model	One of the pre-trained neural network models provided with IN Carta SINAP for a range of segmentation applications
Epoch	Single iteration of learning algorithm working through entire training set in an attempt of creating a model
FOV	Field Of View
Ground Truth	User annotated input data
Input data	Images
Model	Trained Deep Learning neural network
ROI	Region of Interest
Target	Biological structure that is of interest to the user
Training Set	Set of annotated images used to train a model

Typographical conventions

Software items are identified in the text by ***bold italic*** text.

Hardware items are identified in the text by **bold** text.

In electronic format, references in *italics* are clickable hyperlinks.

Notes and tips

Note: *A note is used to indicate information that is important for trouble-free and optimal use of the product.*

Tip: *A tip contains useful information that can improve or optimize your procedures.*

2 Introduction

Introduction

Segmentation is the foundation of the image analysis pipeline, allowing researchers to identify regions of interest in tissue, whole organisms, individual cells, nuclei, and organelles. Through segmentation, researchers can extract information from images to quantitatively compare differences across diverse concentrations, treatments, time, genetics, etc.

Thus, it is crucial to get segmentation right, as it is the foundation for every analysis that follows. Accurate and reliable identification of structures, shapes, and sizes is vital for robust analyses. Errors in segmentation will be propagated and multiplied through the rest of the analysis, effectively lowering the assay robustness (less accurate dose curve, inability to detect small modulations of a phenotype, lower Z' for classification experiments). This leads to false-positives and false-negatives, and ultimately wastes time, money and resources.

Machine learning

Current segmentation algorithms often struggle to deal with non-optimal images where poor contrast, signal variability or high complexity of biological structures mean that users must compromise their analyses and work with a solution that accommodates some of their data, but not all.

This results in multiple, specialized tools, each suited for an application or data set, which often require training to be used effectively.

The advent of Artificial Intelligence (AI) has improved the situation. The development of computer systems able to perform tasks through machine- and deep-learning that would normally have required human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages, has addressed some of the intractable problems. These solutions still require a large amount of annotated *ground truth* data to create a reliable, accurate training set.

A learning algorithm capable of being taught to detect the biology in which a researcher is interested is required.

IN Carta SINAP is a trainable segmentation module that utilizes both classic machine learning and deep-learning in an intuitive format that produces robust, reliable segmentation across a wide variety of applications.

AI for image analysis

IN Carta SINAP uses machine learning as an aid for creating annotated ground truth training sets for the Deep Learning component of the software.

Deep Learning is an area of AI that uses multi layered neural networks to mimic the way that the human brain processes information. A network becomes more accurate as it is provided with more ground truth data from which to learn.

With Deep Learning, there is no requirement to define specific features that are of interest. A model learns in a similar way to humans, and as a result can outperform traditional segmentation methods in many applications.

IN Carta SINAP

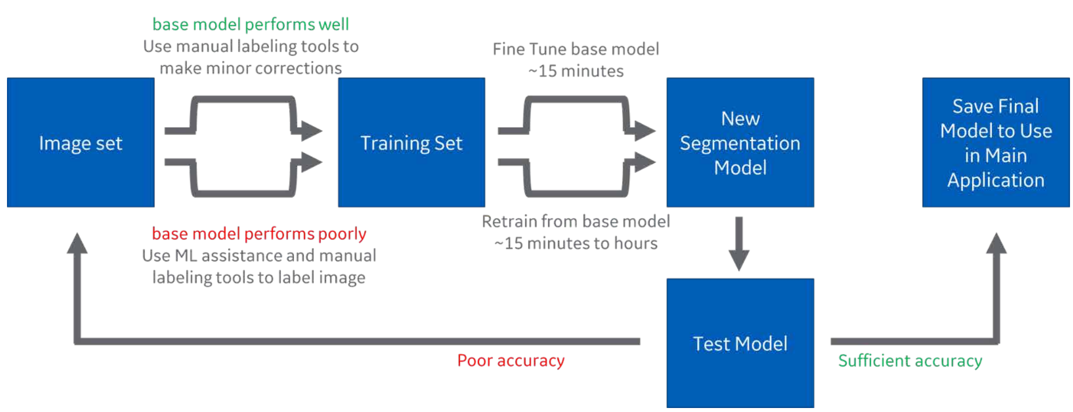
IN Carta SINAP uses pre-trained *base models* to reduce the requirement for large amounts of annotated ground truth data. This is to save time and reduce the amount of data that is required for a model to be accurate. If a model is a good match for the target of interest, the fine tuning will only require a small number of annotated images to be added to a training set.

Where the structure of interest is dissimilar to images used to train a model, the performance of the model may be sub-optimal. In this case, the user can re-train an existing model to suit their application.

Re-training a model using IN Carta SINAP adopts an iterative approach, where the user adds more annotated ground truth data until the model accurately differentiates the target from the background. This takes longer than fine tuning and requires more data in the training set. Timing and size of training set is highly variable and is dependent on how different the structures are to the base model, and how complex the problem is.

In either case, when the user is satisfied with the performance of the model, it can be saved and applied to the full data set.

The saved model will also be available as a pre-trained model for subsequent use in other experiments, as shown in the illustration.



3 Using IN Carta SINAP

About this chapter

This chapter contains instructions on how to run your methods using IN Carta SINAP.

In this chapter

3.1 General user information

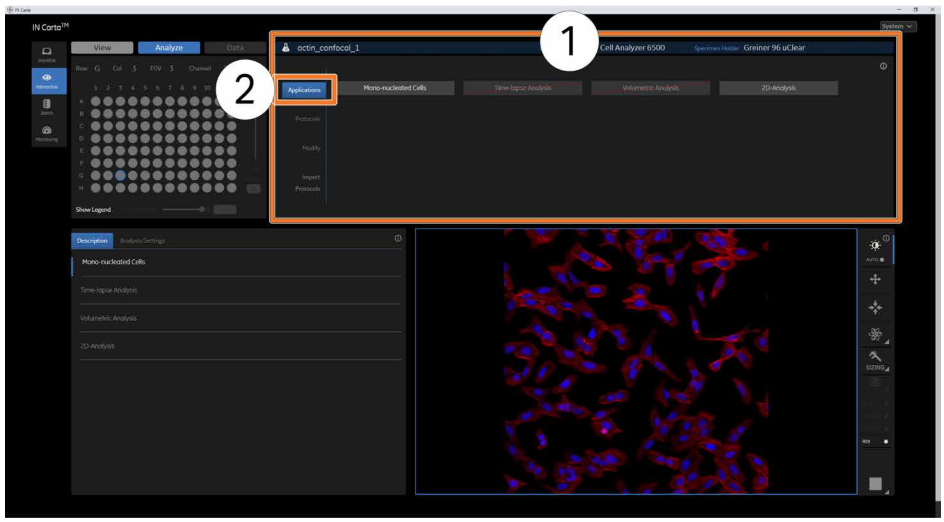
Saving data and data storage

- It is not possible to delete or overwrite a model once it has been saved.
- Upon editing and saving, a new model is created with a unique identifier.
- Training set data is stored in a temporary folder while the SINAP application is being used.
- Each time the SINAP application is started, all previously stored training-related data is deleted.
- Closing the SINAP application without saving a model means that any changes made to the model will be lost.

3.2 Launch the application

Follow the steps below to launch the application.

Step	Action
1	From the ***Applications Tab*** in the ***Protocol Editor Table***, click ***2D-Analysis*** to start the segmentation work flow.



Part	Name
1	Protocol editor table
2	Applications tab

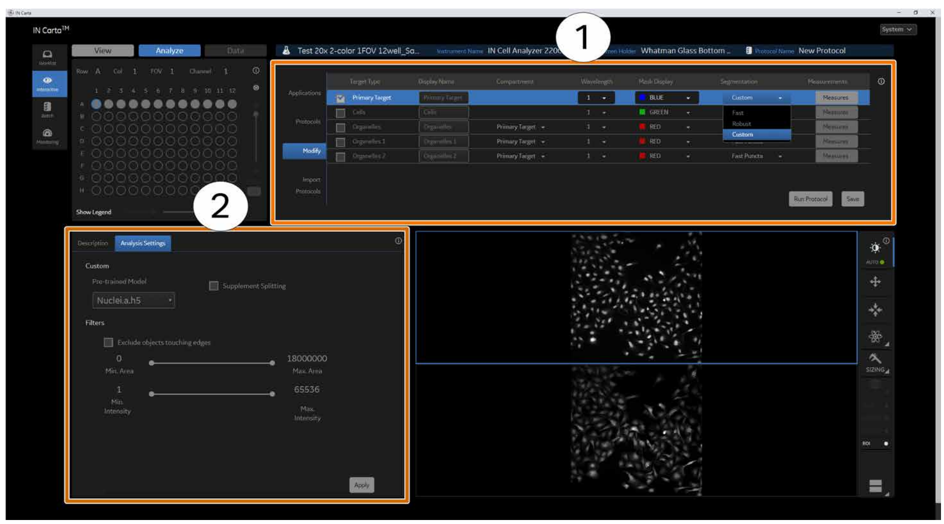
To use a pre-trained model with no further input or training, continue to *Section 3.3 Create new protocol with deep learning analysis, on page 9*.

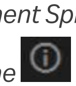
To refine or create a new model, proceed to *Section 3.4 Refining a pre-trained model, on page 10*.

3.3 Create new protocol with deep learning analysis

Follow the steps below to create a new protocol with deep learning analysis.

Step	Action
1	Create a new 2D Analysis protocol by clicking ***New***.
2	Select ***Custom*** in the ***Segmentation*** drop down list.
3	Define the ***Wavelength*** (channel) of interest.
4	Specify the ***Display Name*** (e.g. nuclei, puncta, mitochondria, other structure).
5	Select a ***Pre-trained Model*** from the drop down list in ***Analysis Settings***.



Tip:
More information on filters and 'Supplement Splitting' is accessible through the information panel. To access this, click the  *icon in the* ***Analysis Settings*** *panel.*

Step	Action
6	Click ***Apply*** .
7	If the segmentation is satisfactory, click ***Run Protocol*** to run the analysis on the data set, or proceed to *Section 3.4 Refining a pre-trained model, on page 10*.

Part	Name
1	Protocol editor table
2	Analysis settings

3.4 Refining a pre-trained model

General considerations for generating a training set

This section describes general considerations to be considered when generating a training set.

- Different methods of adding images to training set can be used to generate a training set.
- ROI can be used to specify a properly segmented region within an active image. When ROI is active and ***Add to Training Set*** is clicked, then only the ROI will be added to the training set. The ROI's position can be changed by holding Ctrl on the keyboard and dragging using the left mouse button
- Multiple ROIs can be added to a training set for any given FOV.
- Consistent labeling of an image is the most important factor. Poor quality ground truth is the biggest obstacle to quickly training a model.

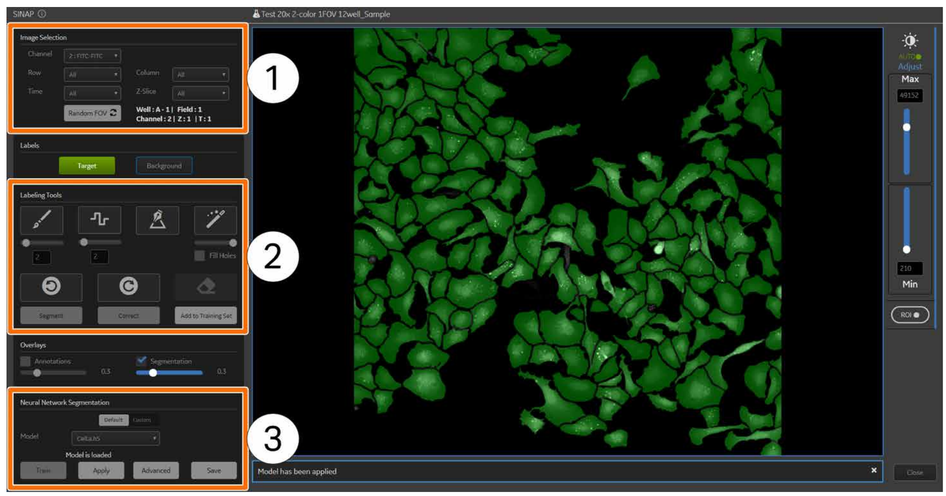
Tip: *It's more important to label the right pixels than a lot of pixels. The most useful information is found at boundaries between target and background. Use the brush tool to precisely annotate these areas.*

- Poor image quality, densely packed and overlapping objects, and rare object examples are tougher and typically require more annotated ground truth to be added to a training set.
- The same input images can be annotated differently to train a model to recognize a new target. In this way, models can be trained to recognize multiple targets from the same input images.

Refining a pre-trained model

Follow the steps below to refine a pre-trained model.

Step	Action
1	Click on ***SINAP*** to launch the IN Carta SINAP application window.
2	Select ***Channel***, ***Row***, ***Column***, ***Time*** and ***Z-slice*** from the drop down lists in the ***Image Selection*** panel.



Part	Name
1	Image selection panel
2	Labeling tools panel
3	Neural network segmentation panel

3.5 AI-assisted segmentation

Fine tuning

When a pre-trained model produces results that are close to expected and only minor corrections are required, it is simply a matter of fine tuning the model by correcting areas that have been mislabelled as target or background. This can be done using Workflow 1: Minor corrections below.

In cases where a model produces sub-optimal segmentation and requires training, refer to *Workflow 2: Major corrections, on page 13*.

Workflow 1: Minor corrections

Use this workflow if a model produces optimal segmentation and requires only minor corrections.

Step	Action
1	Click ***Random FOV*** to select a field of view to work on.
2	Adjust brightness/contrast using the control in the right panel to suit your preferences.
3	(Optional) Define a region of interest to work on by clicking ***ROI*** to speed up image segmentation
4	Select the desired ***Model*** from the drop down list in the ***Neural Network Segmentation*** panel.
5	Click ***Apply***.
6	Adjust the opacity of the ***Annotation*** and ***Segmentation*** masks using the sliders in the ***Overlays*** panel.
7	Select the ***Label*** to be applied (Target or Background) in the ***Labeling Tools*** panel.
8	Use the ***Background*** label to train the model to remove false-positive regions.
9	Use the ***Target*** label to train the model to remove false-negative regions

Step	Action
10	Select the labeling tool that best fits the objects/areas to be labeled. The tool options are described in the table below.

Labeling tool	Description
Brush	Allows free-hand annotation – the circle size of the cursor corresponds to the size of the brush stroke.
Line	Allows annotating a series of lines using mouse clicks.
Polygon	Allows for annotation of larger regions.
Connected component	Labels image regions with intensities like selected pixels. It can be used for quick annotation of structures of interest. 1. Adjust the slider bar to change the sensitivity. 2. Use the ***Fill Holes*** check box to include gaps in the annotated area. 3. Add more annotations to correct the segmentation of desired structures.

Step	Action
11	Click ***Correct*** to apply the annotations to the segmented image.
12	When the segmentation of the image is satisfactory, the FOV or ROI can be added to the training set by clicking ***Add to Training Set*** in the ***Labeling Tools*** panel.

Workflow 2: Major corrections

Use this workflow if a model produces sub-optimal segmentation and requires more ground truth images to be added for training.

Step	Action
1	Click ***Random FOV*** to select a field of view to work on.
2	Adjust brightness/contrast using the control in the right panel to suit your preferences.
3	Using the ***Labeling Tools*** described in step *9* of *Workflow 1: Minor corrections, on page 12*, annotate example regions for Target and Background. ROI may be used to speed up segmentation. ROI must include annotations for both Target and Background.

Step	Action
4	Click ***Segment*** in the ***Labeling Tools*** panel to segment the current FOV. ***Note:*** *A machine learning model is used to predict all unlabeled pixels in an image, assigning them as either Target or Background. This model can be re-used for any new FOV that is loaded or when the ROI is moved to a new position.* **a.** If the resulting segmentation mask is not satisfactory, more annotations can be added, followed by clicking ***Segment***. **b.** (Optional) Corrections can be added to a segmentation mask as described in steps *6 to 10* of *Workflow 1: Minor corrections, on page 12*.
5	When the segmentation of the image is satisfactory, the FOV or ROI can be added to the training set by clicking ***Add to Training Set*** in the ***Labeling Tools*** panel.

3.6 Manual segmentation

In cases where applying a segmentation model or using the ***Segment*** button in the ***Labeling Tools*** panel yields unsatisfactory results, it may be desirable to train a model by annotating only target structures in a single or multiple fields of view. In this case, all unlabeled pixels are assigned as background.

Follow the steps below to perform manual segmentation.

Step	Action
1	Click ***Random FOV*** to select a field of view to work on.
2	Adjust brightness/contrast using the control in the right panel to suit your preferences.
3	Using the ***Labeling Tools***, annotate only the structures of interest as ***Target***. All unlabeled pixels will be assigned as ***Background***.
4	Click ***Correct***.
5	Repeat steps *3* and *4* above, if needed.
6	When the segmentation of the image is satisfactory, the FOV or ROI can be added to the training set by clicking ***Add to Training Set*** in the ***Labeling Tools*** panel.

3.7 Using a training set to train a model

After adding images to the training set, they can be used to train a model.

Step	Action
1	Select either a ***Default*** model from the drop-down list or a ***Custom*** model using file browse dialogue in the ***Neural Network Segmentation*** panel.

Tip:
*Consider adjusting **Advanced** parameters using the options seen in the table below. (More information on advanced parameters is accessible through the information panel – to access click the icon.*

Advanced parameters	Functionality
Fine tune	Use when the base model is closely related to images in the training set.
Retrain	Use when creating a model for a distinct biological structure.
Number of epochs	Increase to allocate more time to obtain a potentially more robust model.

Step	Action
2	Click ***Train*** to train the model based on the images in the training set.
3	Test the segmentation on the new FOVs to check performance. If it is satisfactory, click ***Save*** in the ***Neural Network Segmentation*** panel. Enter the name of the new model and click ***OK***.

Note:

If the segmentation results are not robust enough, then more images need to be added to training set.

Note:

Saved models are stored in c:\ProgramData\TSModels

Result:
The model can now be accessed from ***2D Analysis*** in the ***Analyze*** dashboard.

4 Troubleshooting

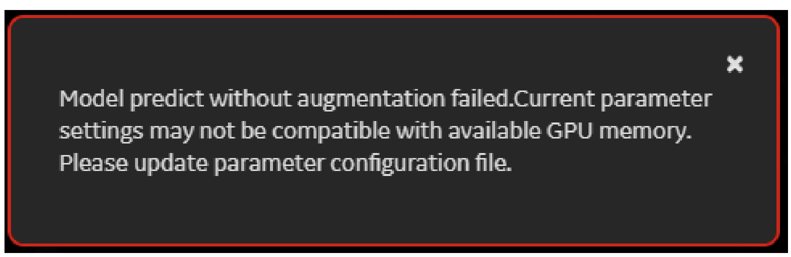
Update parameters for non-optimal GPUs (if required)

IN Carta SINAP requires an NVIDA™ GPU card with CUDA® compute capability of at least 3.5.

For more information on NVIDA GPUs see *https://developer.nvidia.com/cuda-gpus#compute*.

Note: *Ensure that the latest GPU drivers are installed on the computer.*

When using a computer equipped with an NVIDA GPU card with less than 24Gb memory, IN Carta will attempt to auto adjust GPU related settings. However, in some cases it will be necessary to update the neural net parameters for SINAP to run. The error message shown below might appear if parameters have not been properly updated.



To update the parameters for deep learning segmentation/training, run the *params_config.bat* file located in *c:\Program Files\INCarta*.

The table below shows default and recommended settings for parameter configuration. Follow the values in the table to adjust parameters for smaller GPU cards. Upon completion, SINAP should now run without displaying an error code.

Parameter	Default value	Recommended setting for smaller GPU
Train Crop Size	1024	512 (if fails, set to 256)
Train Batch Size	2	1
Number of blocks X	1	2 (if fails, set to 4)
Number of blocks Y	1	2 (if fails, set to 4)

Index

cytiva.com

For local office contact information, visit cytiva.com/contact

29616581 AA V:2 08/2020