User Guide

ZOT Analysis Software

ZOTAS 1.0

Version 1.0.0 • March 5, 2025

Al Kassir S. *et al.* – Ongoing publication

https://www.zebrafish-research.com/zotas

Created by: Théo Mercé
Contact: theo.merce@u-bordeaux.fr

Git: https://github.com/TheoMrc

MRGM Rare Diseases Genetics and Metabolism

université de BORDEAUX

OBERON EDCs • TESTING • METABOLISM • IATA

Inserm

Table of contents

1. General considerations

In this chapter

About the Zebrafish Obesogenic Test (ZOT)

The Zebrafish Obesogenic Test (ZOT), which was developed by our team (Tingaud-Sequeira et al., 2011), assesses the effects of diet, drugs, and environmental contaminants, separately or in combination, on white adipose tissue (AT) dynamics in live larvae. The ZOT uses Nile Red as a lipophilic fluorescent probe that reveals adipocyte lipid droplets, the size (area on two-dimensional fluorescence images) of which is measured through image processing methods before and after exposure for 24 hours to test molecules or mixtures. The ZOT examines obesogenic or anti-obesogenic effects *via* a whole-organism short-term assay, hence providing relevant information for environmental and human risk assessments, a method for adverse outcome pathways mechanistic studies (Ouadah-Boussouf and Babin, 2016) and a screening tool (Kassir et al., ongoing submission).

The OBERON European project was launched in 2019 to improve premarketing regulatory testing of the metabolic effects of endocrine disruptors, including their obesogenic effects. Within the framework of OBERON, we optimized several aspects of the ZOT protocol (Kassir et al., ongoing submission) including staining, image acquisition, and image processing. The optimized ZOT protocol quantifies subcutaneous adipose tissue (SAT) in two regions of zebrafish larvae – anal fin ray SAT (AFRSAT) or ventral SAT (VSAT) – rather than the whole animal in order to facilitate image acquisition and enhance reproducibility (Kassir et al., ongoing submission).

About the ZOT Analysis Software (ZOTAS)

For protocol reproducibility and speed, we developed a fully automated deep-learning (DL) program that processes ZOT fluorescence images. This user guide presents an overview of the key aspects of image acquisition to take into consideration and gives systematic instructions to successfully complete image processing with the ZOT Analysis Software (ZOTAS).

Requirements

Precise descriptions of materials and methods of the optimized ZOT protocol are available in Kassir et al. (ongoing submission). The most critical points aspects of ZOTAS are indicated below.

Image acquisition and file format requirements

- Images – ideally squared – display a lateral view with the anteroposterior axis of the larva from left to right (see example) including the studied SAT (AFRSAT or VSAT) obtained using a fluorescence microscope with 4 times magnification and a 16-bit camera. Rectangle images are automatically cropped at a chosen position during preprocessing (the same for every image in the experiment).
- Acquired files are single-channel 16-bit fluorescence TIFF images – or – multi-channel TIFF image from which one channel of 16-bit depth is chosen and analyzed.

System requirements

Please ensure that your computer meets the following system requirements:

CPU	Tested on: Intel(R) Core(TM) i7-13700 CPU @ 2.10GHz
RAM	8 GB recommended
Available disk space	2 GB
Operating system	Windows 10/11 64 bits

Installation procedure

Download the ZOTAS installer from https://zebrafish-research.com/zotas/download, run it, and follow the instructions to install the software to the desired destination (available for Windows 10/11 64 bits systems for now).

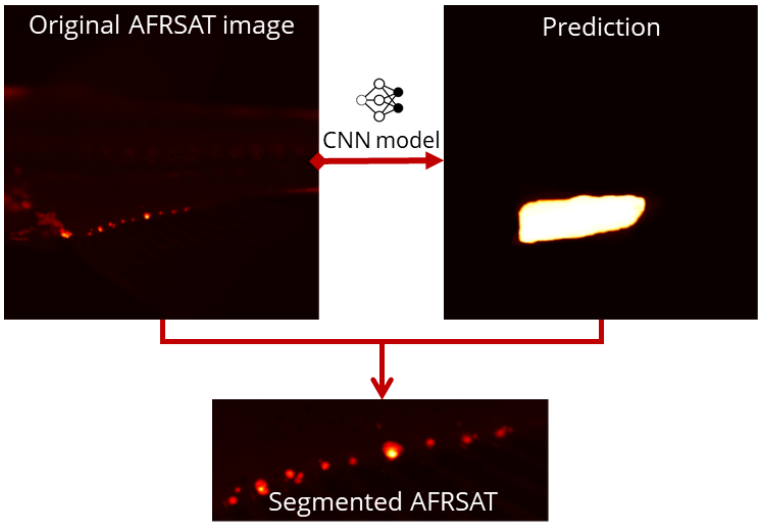
Hint The ZOTAS installer requires administrator level privilege. We recommend ticking the option to Add a Windows Defender exception to prevent recognition of ZOTAS as a malicious software.

Hint Shortcuts to the ZOTAS executable are created during installation.

General principles of ZOT image processing

Step one: Deep-learning image segmentation

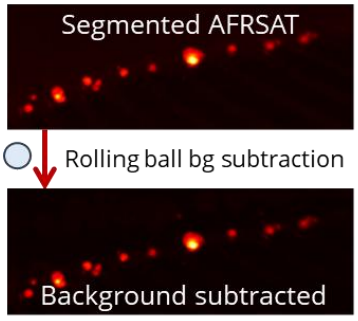
The image processing starts with a deep learning-based segmentation of the region of interest (AFRSAT or VSAT) from the original fluorescence image. The square image is processed by a U-Net model, i.e., a convolutional neural network (CNN), trained to recognize AFRSAT or VSAT in x4 Nile red fluorescence 16-bit TIFF images. During preprocessing, the rectangle images are cropped at a chosen position along the wider axis.



Deep learning segmentation of an AFRSAT image

Step two: Background subtraction

The background is then subtracted from cropped AFRSAT or VSAT region images using a rolling ball algorithm (Sternberg, 1983) implemented in the scikit-image 0.18.3 python module with a 25-ball radius. This algorithm performs background subtraction based on image object size and intensity values, a subtraction that preserves adipocytes. The image is smoothed with a 3x3 mean filter before computing background with the rolling ball algorithm. Then, the background is subtracted from the original (non-smoothed) image. This step facilitates and

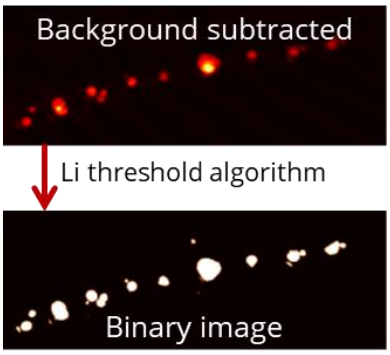


Rolling ball background (bg) subtraction

reduces mistakes in threshold calculation by the Li algorithm during the next step (See Step three below).

Step three: Thresholding algorithm and quantification

Following background subtraction, the larva region image is binarized by the Li thresholding algorithm based on gradient descent to the local histogram minima, initialized either with Otsu or Triangle thresholds for AFRSAT or VSAT images, respectively. The area covered by adipocytes is calculated as the number of pixels for which the intensity value is above the Li threshold (white pixels opposite).



Adipocyte segmentation by Li-threshold

The whole process of steps one to three is repeated before and after vesicle exposure to the tested conditions. The resulting value is the difference in adipocyte lipid droplets area before and after exposure either:

- non-normalized (recommended): $\Delta Area\ (mm^2) = Post\ Exp\ Area - Pre\ Exp\ Area$
- or normalized: $\Delta Area\ (\%) = \Delta Area\ (mm^2)\ /\ Pre\ Exp\ Area$

Normalization is discouraged as it biases results in cases of nonhomogeneous initial larvae adiposity between groups.

Getting additional help

To receive technical support and software assistance or for any related questions, please contact theo.merce@u-bordeaux.fr.

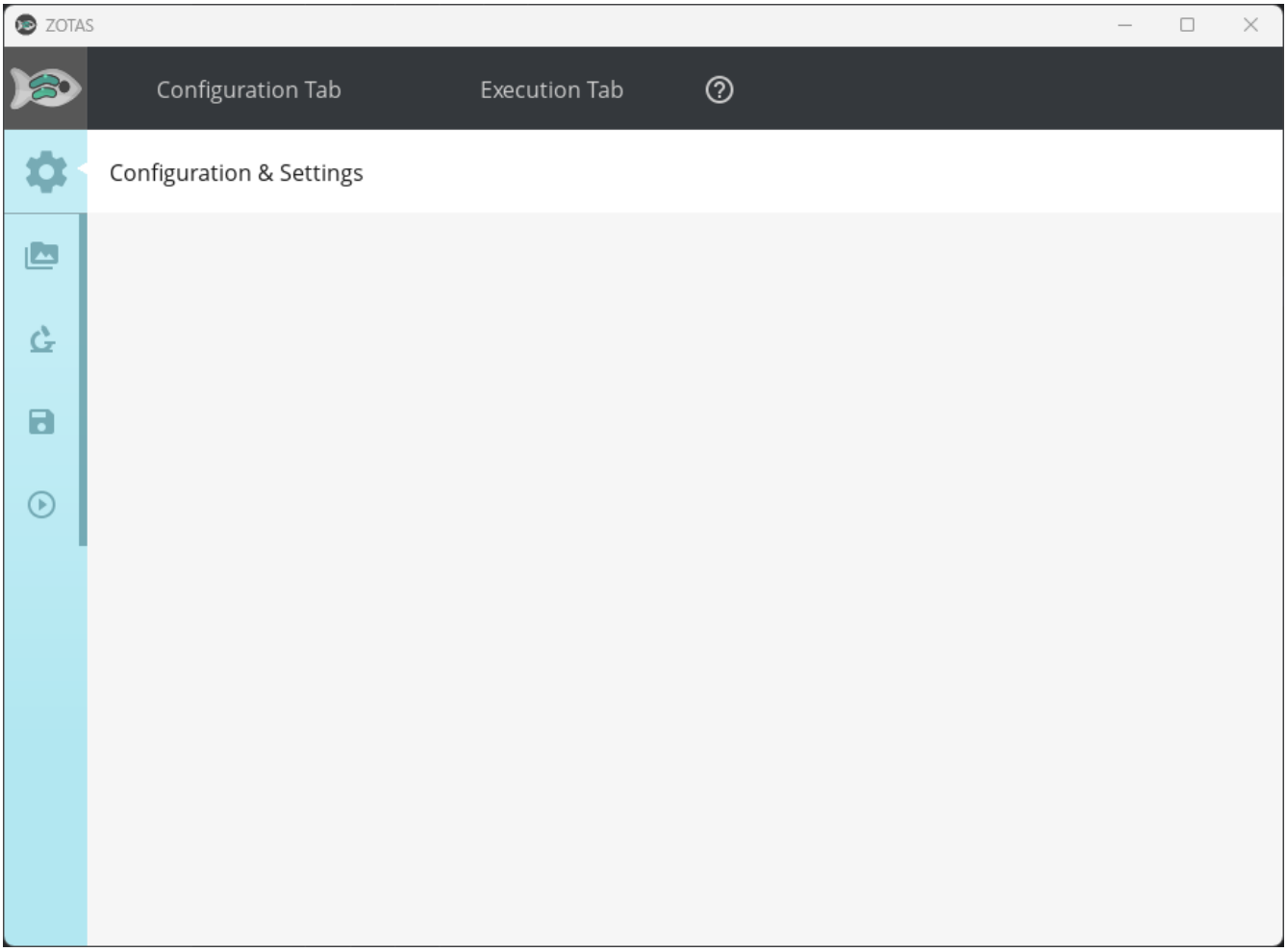
If you need help developing a similar AI-based software for you own scientific application, please contact theo.merce@u-bordeaux.fr.

2. Using the ZOTAS application

In this chapter

Exploring the start page

Upon launching the application, the home screen is displayed:



Every panel of the application is structured as follows:

- A horizontal black banner at the top including two text buttons used to switch between Configuration and Execution modes (on the left), and a help button on the right
- A vertical blue banner on the left side comprising:
  - An upper icon: either when the Configuration Tab is activated, or when the Execution Tab is activated.
  - Three icon buttons that launch three different configuration panels: Source and destination folder paths, Image processing and Data export settings.
  - (Optional) – For the Configuration Tab, a launching button.
- A light grey main panel (empty for home screen)

Launching an experiment analysis

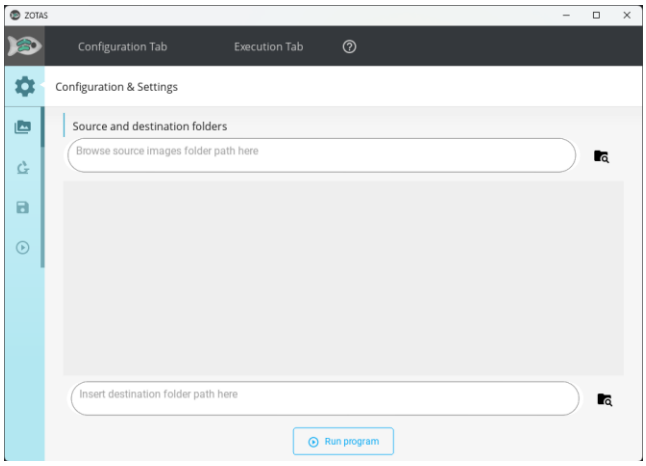
In order to launch the processing of an experiment, please follow the steps below:

1. Go to the Source and destination folder parameters on the side banner
    a. Browse to the source experiment folder
    b. Assert automated segmentation of larvae was successful (see below. *Automated detection of treatment, larvae and larvae region images*)
    c. Browse to the results target path (see below. *Destination folder*)
2. Go to the Image processing parameters on the side banner
    a. Assert detected images are 16-bit TIFF files (see above. *Image acquisition and file format requirements*)
    b. If there is more than one channel per image, indicate which channel to retain for further processing (see below. *Image processing parameters*)
    c. (Optional) If images are not square, set-up for rectangle image cropping
3. (Optional) Go to the Data export settings on the side banner
    a. Choose whether to save analysis summary data table and per treatment column data tables for each data type
    b. Choose data table files format
4. Launch the program (will take you to the Execution tab)
    a. Information about processing will be displayed in real time
    b. The processing log will be saved at <target folder>/log.txt
5. (Highly Recommended) Go to <target folder>/processing_plots
    a. Manually check all processing images for processing errors
    b. Exclude larvae with substantial segmentation or thresholding errors (usually ~5% of larvae) from results data tables in <target folder>

A more in-depth illustrated explanation is available for each step below.

Source and destination folder parameters

Clicking the icon on the side banner opens the Source and destination folder settings panel. The main panel is composed of two folder-browsing buttons corresponding to the experiment's image source folder path (top) and destination folder path (bottom), as well as a 'Run program' button. Images located inside your source folder are not modified during processing.

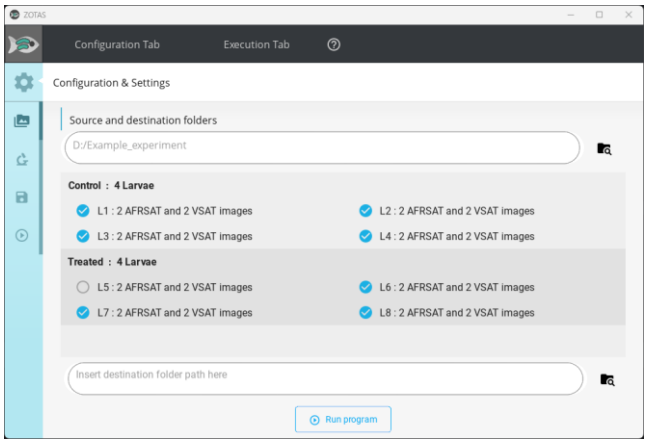


Screenshot – Folder path settings panel

Automated detection of treatment, larvae and larvae region images

Upon browsing the source folder path, treatment, larvae names as well as pre-exposure and post-exposure larvae region images are detected automatically from the path arborescence of all TIFF images found.

Larvae for each treatment or condition are displayed with corresponding check-boxes. In the Image processing settings panel, the first image found is displayed for potential rectangle image cropping and image format compatibility is checked.



Screenshot – Automated detection of experiment images information

Critical This step necessitates a precise folder structure in the source experiment folder containing one or several treatment folders, each containing larvae subfolders that contain precisely named TIFF images as follows:

<experiment name> / <treatment name> / <larva name> / <pre or post><AFRSAT or VSAT>.tiff

Please note that the image file name itself must contain the larva region name and an indication of whether the images were taken before or after exposure ("pre", "before" or "BE", and "post", "after" or "AE", respectively, for pre- and post-exposure images).

Critical In order to be analyzed, a larva folder must contain two properly named images per larva region studied. The user can confirm this through the displayed larva

information: there must be at least two images associated with the larva region or regions studied.

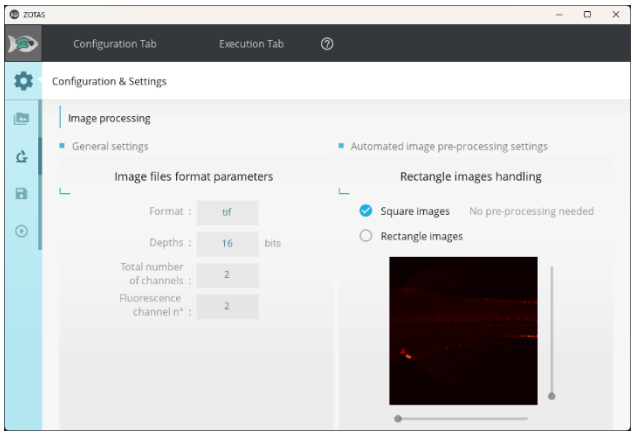
Destination folder

Destination can be set by browsing the target folder path or by writing / pasting the desired location.

Hint A sub-folder with the name of the experiment source folder is created inside the target location. This folder will contain the results data frames (usually .xlsx files) and a /processing_plots subfolder containing all of the processed images (.png format), which can be verified to assert the validity of the results obtained.

Image processing parameters

Clicking the icon on the side banner opens the Image processing settings panel. The main panel has two parts: an image formatting parameters section (left side) and an automated rectangle image cropping setting section (right side).



Screenshot - Image processing settings panel

Both sections are updated upon browsing the source folder path in the Source and destination folder settings panel.

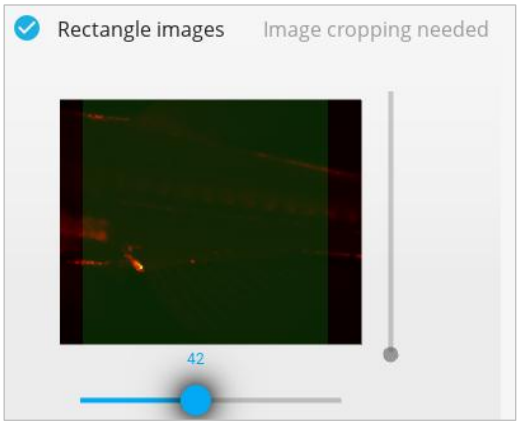
File format, number of channels and depths are extracted from the first image found.

Critical In order to be analyzed, data must consist of 16-bit TIFF files (see *Requirements* on page 2 for a more detailed description of image format requirements).

Critical If the image consists of more than one channel, the user must indicate which channel is relevant for Nile red fluorescence (from one to total number of channels). This channel will be extracted from the image to go through the analysis process.

The chosen channel of the first image found is also displayed on the panel section on the right. Image size is verified by the user.

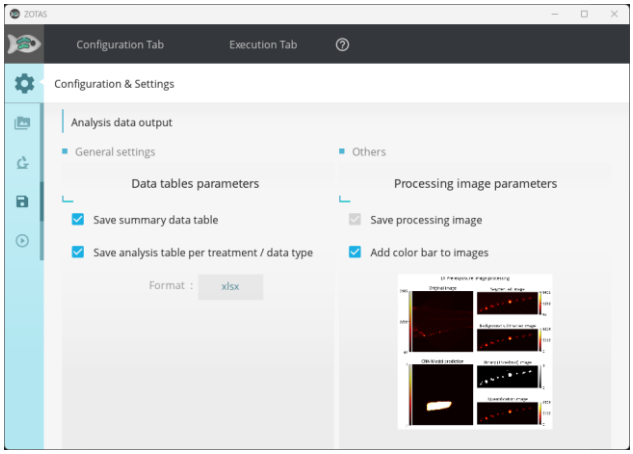
Critical For non-square images, using the sliders located around the displayed image, the user can define which square part of the image (green square) will be retained throughout the analysis process. Every image in the experiment will be cropped the same way.



Screenshot – Rectangle image cropping

Output images and data tables

The icon leads to the Data output settings panel. (Displayed below)



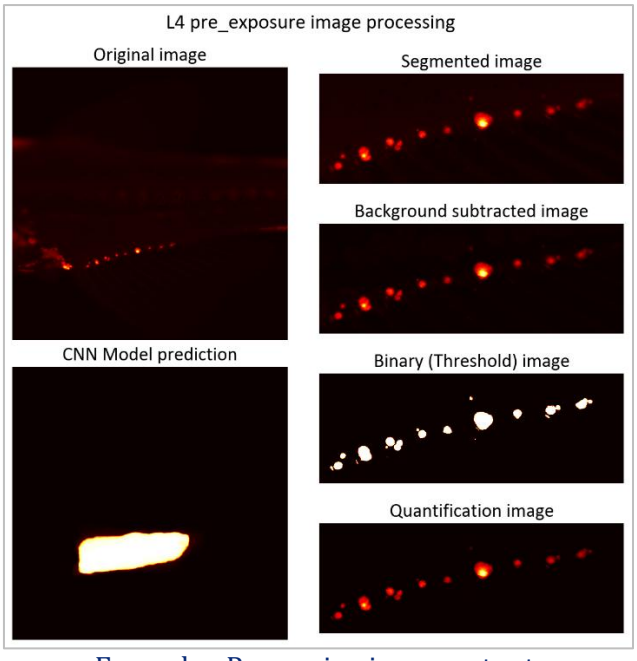
Screenshot - Data output settings panel

Two types of data tables are proposed:

- Analysis summary data table: one line per larva including all data obtained, i.e., one column for treatment, pre- and post-exposure areas, normalized and non-normalized area differences.
- Treatment / data type tables: one column per treatment, with values for larvae that received the same treatment stacked within these columns. One file is created for each data type (pre- and post-exposure areas, non-normalized and normalized area differences). Treatment / data type tables are more convenient for comparisons and statistics between treatment groups performed using other software.
- The user can chose between the formats .xlsx and .csv for the data table files.

A processing image is also saved for every analyzed image (two per larva) including:

- a square original fluorescence image (after pre-processing steps, i.e., extraction of the chosen image channel and cropping for rectangle images)
- Corresponding DL zone detection (see Step one above. *Step one:* Deep-learning image segmentation)
- Resulting segmented AFRSAT or VSAT image
- Image after background subtraction (see Step two above. *Step two*: Background subtraction)
- Binary image after Li-threshold for area quantification (see Step three above. *Step three: Thresholding* algorithm and quantification)
- Intensity quantification image (not used in the optimized ZOT protocol)



Example - Processing image output