Universiteit Leiden
The Netherlands

Bachelor Bioinformatics

ZebraVishualizer:

A Tool for 3D Cell Trajectory Annotation and Visualization

Luuk Deen

Supervisors:
Lu Cao & Katy Wolstencroft

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

16/08/2024

Abstract

Neutrophils are one of the first responders to tissue damage and are essential to the innate immune system. Using microscopy, it was found that the TLR3 and MyD88 genes play a role in the regulation of neutrophil migration in zebrafish. However, the exact underlying mechanisms of neutrophil movements are unknown.

In order to gain insights in the movements of neutrophils, it helps to visualize their trajectories. Several software tools exist for visualizing microscopy images. However, these tools are not perfect for the task of visualizing neutrophil trajectory over time in 3D space.

This thesis provides a tool for the 3D visualization and annotation of cell trajectories over time, called ZebraVishualizer. The tool helps to gain a better understanding of the movement of neutrophils in zebrafish by providing a clear visualization of their trajectory. In addition, ZebraVishualizer can be used for manually improving cell annotations. This improved trajectory could be used for further research, such as training a deep learning algorithm. The tool can also be used for research on other cell types.

Contents

1 Introduction

1.1 Zebrafish

The zebrafish (Danio Rerio) is a freshwater fish from the Cyprindae family and of the order Cypriniformes. The zebrafish is native to South Asia, where it lives in stagnant to moderately flowing clear water of shallow depth and near neutral pH [EPRP07]. Zebrafishes are often kept in aquariums, due to their aesthetics, wide availability, cheap price and playful nature [GC02].
Zebrafish is a popular vertebrate model organism used in a multitude of research fields, such as developmental biology, drug development and genetics [TZ02][VWKH+17][Par06]. It possesses many qualities that are useful for scientists.
First of all, zebrafish can reach a high population density of up to five fish per liter [MS11]. In addition, female zebrafish can lay up to 300 eggs per week, leading to a quick reproduction cycle. Because of this, it is easy and cheap to obtain the amount of fish needed.
Zebrafish embryos are large and robust and develop rapidly. They are also able to develop outside of the uterus and are transparent. In addition, their size remains nearly constant during early development [Dah06]. This makes it possible to use non-invasive imaging techniques from fertilization until later development, in order to obtain high-resolution views of cellular and subcellular processes.
Furthermore, zebrafish exhibit physological and genetical similarities to humans and other mammals. Approximately 70% of all human genes have at least one zebrafish orthologue [HCT+13]. The full zebrafish genome has been sequenced and several well-characterized zebrafish mutant strains are readily available, making them even more useful.
The combination of these unique attributes makes the zebrafish a valuable model for a broad range of scientific research.

1.2 Inflammation

Inflammation is the response of the body to pathogens, damaged cells or other harmful stimuli. It is part of the innate immune system. The goal of inflammation is to maintain homeostasis of the damaged tissue by eliminating the cause of the injury, repairing the damaged tissue and removing cell debris [SBB+07].

Usually, inflammation is acute. This means that the inflammation is temporary and ceases after the repair process has started. Acute inflammation starts shortly after an injury and doesn't last more than a few days.
Acute inflammation is started by immune cells that are already present in the damaged tissue. These cells use pattern recognition receptors (PRRs) for recognizing pathogen-associated molecular patterns (PAMPs) and damage-associated molecular patterns (DAMPs) [HD15]. PAMPs and DAMPs are molecules that are associated with pathogens or damaged cells respectively.
Toll-like receptors (TLRs) are a prominent type of PRR in regulating the inflammatory response [YWC10][Vij18]. TLRs trigger the production of cytokines after sensing PAMPs. Cytokines are a category of small proteins that attract immune cells. An example of a TLR is TLR2, which recognizes bacterial liposaccharide. All TLRs, except TLR3, use myeloid differentiation primary response 88 (MyD88) as an adapter to connect to proteins that relay signals inside of the immune cell [JM14][CZCL20].

By recognizing these PAMPs and DAMPs, the immune cells get activated. Upon activation, the immune cells will release variety of inflammatory mediators. These mediators will initialize the immune response, which contains a vascular and cellular component [RSAR+15].
The vascular component involves an alteration of the local vasculature [Med08]. Macrophages and mastocytes release mediators that increase the permeability of blood vessels, allowing blood plasma to flow from the vessels into the infected tissue. The plasma contains several antimicrobial mediators such as antibodies, which can immediately attack pathogens. In addition, several clotting mediators are released that will provide a structural scaffolding that aids in repairing the damage.
The cellular component is characterized by leukocytes, a type of white blood cells. These cells aid the inflammation in several different ways. Some leukocytes release enzymatic granules that damage pathogens. Many leukocytes ingest and destroy viruses, bacteria and cell debris by phagocytosis. Leukocytes also help develop and maintain the inflammation by releasing mediators.

There are 4 possible outcomes for acute inflammation [RCKC98]:

1. **Resolution**: The damaged tissue is fully restored to its original state. This is usually the outcome of a limited inflammation.

2. **Fibrosis**: The damaged tissue cannot be regenerated completely. Instead, scar tissue will be formed. This tissue does not contain specialized structures and thus may lead to functional impairment. This is often the outcome when there was damage in a tissues that can't regenerate or if the amount of damage was very large.

3. **Abscess formation**: A cavity is formed. This abscess contains dead bacteria and white blood cells and other cell debris.

4. **Chronic inflammation**: If the inflammation persists for to long, it can turn into a chronic inflammation. Chronic inflammation is caused by a large amount of macrophages. These macrophages try to remove pathogens by releasing toxins. However, in case of a chronic inflammation, these toxins also hurt the organisms own tissue. Because of this, chronic inflammation often leads to tissue destruction. Chronic inflammation can last many days or months and even multiple years.

1.3 Neutrophil Migration Behaviour

Neutrophils are the most abundant white blood cell in the human body, they make up approximately 50-70% of all white blood cells. They are one of the first types leukocyte to be recruited towards tissue damage [CB02].
At the site of injury, the neutrophils will release cytokines in order to recruit more immune cells [EM08]. In addition, neutrophils directly attack pathogens. They do this in three different ways [HK09]:

- Phagocytosis, the eliminating of microbes by ingestion.
- Degranulation, the release of several anti-microbial proteins.
- Generation of neutrophil extracellular traps (NETs) that trap and kill extracellular microbes [BRG+04].

Neutrophils are able to migrate towards the site of inflammation because of chemotaxis via amoeboid movement [SWG10]. Receptors on the cell surface of the neutrophils, such as PRRs, detect chemical gradients of PAMPs, DAMPs and cytokines. These gradients are used by the neutrophils to direct their path towards the injury.

Neutrophil movement is often studied in mice or in zebrafish larvae. Zebrafish are useful because of their transparency, making it convenient to use transgenic lines that fluorescently label neutrophils so that they can easily be tracked [HLWR13]. In some cases, neutrophils aggregate at the surface level of tissue, which makes the thin tissue of mouse ears a good option for research [CSH+08].

When aggregated near the surface level, neutrophils are often migrating in a highly coordinated way, called neutrophil swarming [TW17]. In order to be able to move coordinated, neutrophils communicate between each other using homotypic communication, as well as with other leukocytes using heterotypic communication. Neutrophil swarming is categorized in five different phases:

1. **Initiation**: Neutrophils near the injury switch from random to chemotactic movement and use chemical gradients to move towards the injury [PCPH18].

2. **Amplification**: These first "pioneer" neutrophils rectruit more neutrophils by releasing cytokines [LG14].

3. **Additional amplification through intercellular signalling**: Even more neutrophils are attracted towards the site of inflammation by intercellular communication using leukotriene B4 (LTB4) [LAA+13].

4. **Aggregation and tissue remodelling**: The neutrophils form aggregates that help in creating a wound seal that isolates the inflammation site from other tissue [PS18]. The neutrophils in these aggregations show less movement than in the first three phases. The exact mechanisms of this phase are unknown.

5. **Recruitment of myeloid cells and resolution**: With the resolution of the inflammation, the aggregates resolve. Myeloid cells move towards the inflammation site and release signals attracting neutrophils, leading to less aggregation. Not much is known about how this process is regulated.

1.4 TLR2 and MyD88 involvement in Neutrophil Migration

Hu et al. [HvSL+21] found that TLR2 and MyD88 are involved in regulating the behaviour and speed of neutrophils and macrophages as a response to tail wounding in zebrafish. This conclusion is the result of the following experiment.

First, the $tlr2^{sa19423}$ and $myd88^{hu3568}$ zebrafish mutant lines were identified [HYS+19][vdVVSSM13]. These mutants have truncated versions of the TLR2 and MyD88 genes respectively. The mutants were crossed with the double transgenic line *Tg(mpeg1:mCherry-F);TgBAC(mpx: EGFP)*, so that they have green GFP fluorescence in neutrophils (due to *TgBAC(mpx: EGFP)*) and red mCherry fluorescence in macrophages (due to *Tg(mpeg1:mCherry-F)*) [BHK+14][RLT+06].

This resulted in zebrafish that have either a non-functional TLR2 gene ($tlr2^{-/-}$) or MyD88 gene ($myd88^{-/-}$) and express fluorescence in neutrophils and macrophages. Wild-type zebrafish of the double transgenic line were used as a control group ($tlr2^{+/+}$, $myd88^{+/+}$). The caudal fins of these zebrafish were wounded three days post fertilization (dpf) at the site indicated in Figure 1.

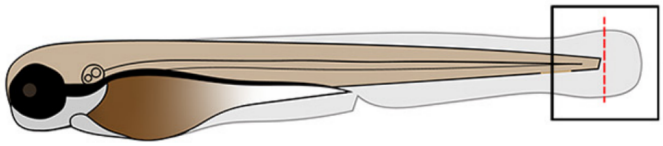

Figure 1: Drawing of a zebrafish. The red dashed line indicates the wounding site. The black square is approximately the imaged area. [HvSL+21]

Time-lapse imaging was performed on wounded and unchallenged 2dpf larvae of both groups using a Leica MZ16FA fluorescence stereo microscope equipped with a DFC420C color camera using a 20× objective (N.A. 0.75). During 120 minutes, images were taken each minute at 8 different depths in the sample.

The images were used for cell tracking. Cell tracking was either done manually by using a plugin from Fiji [MDS12][TCB+15] or automatically with the Viterbi Algorithm [MJGB15]. A number of variables were calculated using the manual tracking data, such as the distance to the wound and the mean speed. A statistical analysis was performed on these variable. This analysis showed that TLR2 and MyD88 are both essential for the directed migration of neutrophils and macrophages towards the wounded tissue.

1.5 Related Work on Tools for Visualization

In order to better understand the migration of neutrophils, research is done towards their trajectories in response to wounding, such as in the research of Hu et al. [HvSL+21]. Experiments like these often yield information about the neutrophils in the form of microscopy imaging.
These images are challenging to interpret without any processing, because of several reasons. First of all, neutrophil migration is a process that takes place in three dimensional space over a certain time period (3D + Time space). Therefore, images are made of different layers in the sample as well as at different time intervals. It is hard to get a clear overview over all these dimensions without any processing of the images. Secondly, there are often multiple cells visible at the same time, which makes it difficult to keep track of a singular cell over all the different images.

In order to make it easier to understand data like this, a multitude of software tools was made.
One such tool was made by Zhe Deng [Den20]. This tool creates a 2D visualization of neutrophil and macrophage trajectory, using data from an experiment similar to that of Hu et al. [HvSL+21]. This visualization shows the trajectories as lines and cells as dots. In addition, there is a static module. This module contains static images of each path and statistics about the paths.
This visualization gives a good overview of the trajectories of the cells and makes it easier to interpret the data. However, the visualization is in 2D, while the cells move in 3D space. Therefore,

it does not give a complete representation of the trajectories.

Another tool for visualizing neutrophil trajectory is Zebrafishualizer by Robert Gijsbers [GD24]. Zebrafishualizer is an interactive 3D video of neutrophils from the microscopy images of Hu et al. [HvSL+21]. Zebrafishualizer first creates object meshes of the cells in the images, which are then rendered in an interactive environment. The visualization allows the user to move freely between the cells so that they can be viewed from any angle.
The tool makes it easy to view the neutrophils close-up from many different sides. However, the animation does not include a visualization of the trajectories of the neutrophils. The object-based method that was used results in a poor space complexity, because all object-files need to be saved. In addition, it makes it difficult and time consuming to create new animations of other data.

There also exist more generalized software for visualizing microscopy imaging. A notable example of such a program is ImageJ [SRE12]. ImageJ is a tool for the processing and analyzing of scientific images. The tool provides many techniques, under which segmentation and tracking. There are many plugins which can be used that provide even more options behond the basic functionality. ImageJ is open source, which makes it easy to create new plugins.
ImageJ is also able to visualize data in both 2D and 3D. However, since ImageJ is mostly meant for processing, it is difficult to create an interactive animation. In addition, ImageJ can be overwhelming when needed for only one specfic purpose, because of its large amount of options.

1.6 Goal of the thesis

The goal of this thesis is to create a software tool that is able to visualize the trajectories of neutrophils in 3D space. The trajectories are visualized in an interactive 3D animation. The user is able to view the cells from all possible angles in order to get a good overview of their locations and shapes. The visualization is animated so that the movement of the neutrophils through time can be observed. The tool can import and show trajectories from an external source. The accuracy of these trajectories can be improved by adjusting the annotated locations of cells. In addition, new trajectory data can be created for unannotated neutrophils. Despite the tool being built for specifically data about neutrophils, data in the same format about other cell types can be used as well.
This tool will help biologists in understanding the movement of neutrophils and other types of cells by giving a clear view of their trajectories. In addition, the tool can create and improve trajectory data which can be used in future research, for example for training a machine learning based tracking method.
The tool is named ZebraVishualizer, as a combination of the model organism that was used for obtaining the data and its use case of visualizing this data.

1.7 Thesis overview

First, chapter 2 describes the materials and methods that were used to create a tool for visualizing and annotating cell trajectories. Then, chapter 3 shows and explains the workings of the resulting software ZebraVishualizer. Finally, chapter 4 concludes the thesis by discussing the advantages and shortcomings of the tool and provides suggestions for further enhancements and research.

The code and data that was used for and created by this thesis can be found at https://github.com/luckey-Luuk/ZebraVishualizer_annot3d

2 Materials and Methods

2.1 Data

The data that is used in this thesis was obtained by Li et. al [CWW+22]. Tail wounding was conducted on 3dpf zebrafish and images were taken using the same method that Hu et. al [HvSL+21] used. However, only the neutrophils are shown in green GFP fluorescence, the macrophages are not visible. A feature weighted tracking method was used on this data.

The data consists of 120 tagged image file format (TIFF) files, a flexible format for the storage of images. Each TIFF file contains an eight-layer 3D stack of images taken from different depths in the zebrafish sample at a certain time step. The interval between layers is $5\sim6\mu m$, the thickness of the zebrafish tails was $35\sim42\mu m$. The images were taken at a time interval of one minute over the course of 120 minutes. This results in data of the size (512, 512, 8, 120), corresponding to the axis (x, y, z, t). Nine of these time-lapse series were taken.
Several pre-processing methods were used in order to improve the image quality. First, the contrast was enhanced in order to highlight the cells. However, background noise was enhanced as well and thus a median filter was used for denoising. Subsequently, the cell surface was smoothed by applying a Gaussian blur. The effects of these pre-processing steps can be seen in Figure 2. Because the data has only 8 layers on the Z-axis, it is difficult to observe movement along this axis. Therefore, linear interpolation was used twice to increase the amount of layers on the Z-axis to 29, resulting in a size of (512, 512, 29, 120) [JSJ+18]

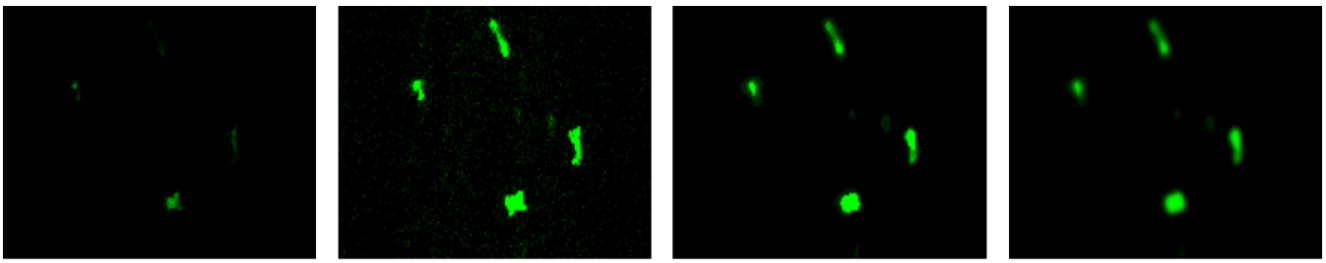

Figure 2: The image preprocessing steps. (a) An unprocessed image. (b) Image with enhanced contrast. (c) Denoised image. (d) Image smoothed with Gaussian blur. [CWW+22]

In order to identify the positions of the different cells, a 3D U-Net segmentation model was used [WMVY21]. The 3D U-Net generates new TIFF files in which each cell has been given a cell ID, by giving all pixels that belong to the same cell the same colour value. Because of this, the image loses its original colours and instead becomes black-and-white as seen in Figure 3. The new files are called masked TIFF files.

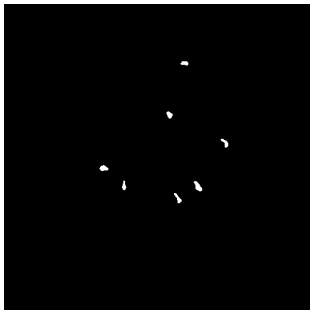

Figure 3: The 15th layer of the first TIFF file from time lapse `20190701--2` after preprocessing and segmentation. The cells are not colored green anymore, instead they each have a unique integer value as their cell ID. This is the result of the segmentation.

The cells were tracked by a feature weighted algorithm. This algorithm uses three different cell features in order to calculate which cell on the next frame is most likely to be the same as a cell on the current frame. The trajectories are stored by linking the cell indexes of similar cells frame by frame. The trajectories of all nine time-lapses are saved in this way in a .pkl file and can be indexed by the interval name. The trajectories are saved as a list of tuples. Each tuple contains three values: the cell ID on the current frame, the current cell ID and the cell ID that this cell was linked to on the previous frame. This .pkl file can be used in combination with the masked TIFF files in order to obtain the actual trajectories.

2.2 Forked Code

A tool for manual 3D cell tracking that was made by Aaron Bos [Bos23] was used as the basis of the code for ZebraVishualizer.
This tool is in turn based on Annot3D [Qur21]. Annot3D is an interactive tool for annotation of 3D TIFF volumes. TIFF files are visualized in an isometric view using volume rendering, as well as in a planar view. Annot3D is supposed to generate semi-automated binary annotations using U-NET predictions. However, its development has not finished, so this is not yet fully functional.
The tool made by Bos [Bos23] is meant for annotation of cell trajectories in 3D TIFF files. The tool uses TIFF files similar to the ones used by ZebraVishualizer. Trajectories are annotated by clicking on cells in the "Render View" and can be easily saved for later use or exported. However, annotation can only be done manually and the only trajectories supported for import are created by the tool itself. The tool shows the time steps of the timelapse one-by-one, there is no animation. Only the trajectory of one cell can be edited or shown at the same time. The tool offers a good starting point for ZebraVishualizer, because it already provides many functionalities that are needed for ZebraVishualizer. However, there are still several functions missing.

3 Results

The result of this thesis is a software tool called ZebraVishualizer. This section will explain all the features that is has to offer.

When opening ZebraVishualizer, first a file selection menu will pop-up. In this menu, the user can select a folder containing TIFF files the user wishes to visualize. Only folders can be selected and only folders containing TIFF files will work. These TIFF files should be named in the format `s++t` or `s--t`, where `s` is the sample name and `t` contains the time step at which the image stored in the TIFF file was taken. ZebraVishualizer will order the files according to `t` and also play the animation in this order. If the cancel button is pressed instead of choosing a folder, ZebraVishualizer will load the folder containing masked TIFF files of neutrophils from Li et al. [CWW+22] by default. This folder is named "`20190701--2_inter_29layers_mask_3a`". This default was put in place for convenience during development and because the main goal of the thesis was to specifically visualize this data.
After picking the data to visualize, the main interface of ZebraVishualizer will open. The interface is shown in Figures 4 and 5. This interface consists of three sections: the toolbar, the render view and the options panel.

3.1 Toolbar: load, save and export

The toolbar is located at the very top of the window and contains only one drop-down menu "File". This drop-down menu contains four options:

- **Load trajectories**: used to load trajectories for the currently visualized data that were generated by either ZebraVishualizer or Li et al.'s algorithm. Only *.xlsx* or *.pkl* files can be selected. The files should contain the right information.
  Trajectories saved in an excel file were generated by ZebraVishualizer and can immediately be imported.
  However, trajectories saved in a pkl file are originated from Li et al.'s algorithm and therefore need more pre-processing. The pkl files store trajectories by linking cell numbers on a certain frame to cell numbers on the next frame. Therefore, pkl files can only be used if the currently visualized data contains cell numbers, this is the case for masked TIFF files. ZebraVishualizer will find the centroid of each numbered cell and use the data from the pkl file to determine the trajectories. Because of this, it will take longer to load trajectories from a pkl file then from an excel file.
  This option can also be called with the hotkey Ctrl+O.

- **Save trajectories**: used to intermediately save the currently shown trajectories in an excel file so that they can be loaded again later. The saved excel file contains five columns: time step, dot (cell number) and the x, y and z coordinates. This way the location of each annotated cell at each time step is saved. This save file can be loaded into ZebraVishualizer.
  This option can also be used with the hotkey Ctrl+S.

- **Export dataset**: used to save the finalized trajectories in an excel file with extra information. The exported excel file contains the same columns as an intermediate save, but with one extra column for the distance that a cell travelled compared to the previous frame. This distance is calculated using the formula $d = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2 + (z_t - z_{t-1})^2}$, where t is the time step and x_t, y_t and z_t are the coordinates of the cell at time step t. If a cell was not visible on the previous slide, the distance is set to -1.
  In addition, it is possible to change the scale of the data when exporting, because the

dimensions of the TIFF file often do not correspond with the actual size of the pictured sample.
This option can also be called by using the hotkey Ctrl+E.

- **Exit**: closes ZebraVishualizer, the same as closing the window.

3.2 Render View

The render view is located at the right side of the window when starting ZebraVishualizer as seen in Figure 4. However, its place can be swapped with the options panel on the left, as well as popped out for it to become its own window. This can be done by pulling on the top bar of the render view to pop it out, after which it can be moved to the left of the options panel.
The render view uses Mayavi in order to show a scene that can include a 3D volume rendering of the TIFF file on the current frame and the annotations and trajectories of annotated cells. In Figure 4, only the 3D volume is shown. Figure 5 also shows the annotations of cells as colored dots and their trajectories as colored lines.

Aside from showing the animation, the main use of the render view is to annotate the cells or adjust already made annotations. An annotation can be set by simply clicking with the left mouse button anywhere on the scene. If there already was an annotation on the current frame for the cell that is being annotated, then the previous annotation on the current frame will be overwritten.

The view of the scene can be changed by using various mouse actions:

left mouse button: Holding down the left mouse button and dragging will rotate the view around the centre. Some buttons can be held in order to change the view in a different way:

+**Shift**: Holding shift while dragging with the left mouse button will pan the scene.
+**Ctrl**: Holding control rotates around the camera's axis (roll).
+**Shift**+**Ctrl**: Holding both shift and control while dragging up or down will respectively zoom in or out.

middle mouse button: Holding the middle mouse button will pan the scene, in the same way as left+Shift.
right mouse button: Holding the right mouse button and dragging up or down will zoom in or out, in the same way as left+Shift+Ctrl.
mouse wheel: Scrolling the mouse wheel up zooms in and scrolling downwards zooms out.

In addition, there are some useful keyboard shortcuts that can be used:

+: Zoom in.
-: Zoom out.
F: Set the camera's focal point to the current mouse position.
↑/↓: Rotate the camera up or down respectively.
Shift+↑/↓/←/→: Pan the camera in the direction of the arrow key.
←/→: Go to the previous or next slide respectively.
P: Play/pause the animation.

T: Change mouse interactions to trackball mode, this is the default mode and works as explained above.
J: Change mouse interactions to joystick mode. This will change the mouse interactions to be similar to a joystick. Holding the left mouse button will rotate the view towards the location of the mouse.
L: Opens a ui for adjusting the light configuration of the scene.
E/Q/Esc: Exit full-screen mode.
**spacebar**: Use the last pressed button from the options panel again.

Located at the top of the render view, there are some built in buttons from Mayavi:

- The first button from the left opens the Mayavi pipeline. The Mayavi pipeline can be used to interactively change the displayed scene and inspect the objects that make up the scene. It is not recommended to use this button, as the scene already gets updated by the Python code and making changes can result in unintended behaviour.
- The next eight buttons **X**, **X**, **Y**, **Y**, **Z** and **Z** are used to quickly change the perspective to view along the negative or positive x-, y- or z-axis respectively.
- The next button, with the cube symbol, changes the perspective to the default isometric view.
- The button with the two rectangles turns on parallel projection, instead of the default projection. This option projects the rendered objects as if they are on a fixed plane.
- The eleventh button toggles the visibility of an indicator for the axes. This is very useful for orientating from which side the animation is being viewed. In Figure 4 this button is toggled on, thus the axes indicator can be seen in the bottom left of the render view. Figure 5 has the button switched off and therefore the axes indicator is hidden.
- Pressing the next button switches to full-screen mode. If the render view is attached to the rest of the application, the entirety of ZebraVishualizer will be shown in full screen. However, if the render screen has been popped out as its own window, then only the render view will be in full screen. Full-screen mode can be exited by either using the button again or by pressing the Esc, E or Q key.
- The second to last button saves the current view as a 2D image. A variety of image formats can be chosen.
- The last button provides a UI in which the scene properties can be edited. It is not recommended to use this button for the same reasons as with the first button.

3.3 Options Panel

The options panel is located at the left of the render view when starting ZebraVishualizer. The option panel consists of three parts: navigation/animation buttons, the “Options” tab and the “Cell Selection” tab. The two tabs are located at the top of the panel and can be switched between. The navigation/animation buttons are located at the bottom of the options panel and will always be visible.

3.3.1 Navigation/Animation Buttons

This section of the UI contains buttons that help with navigating to the frame and playing the animation.
The "play/pause" button will start the animation if it is not running and will pause it otherwise. The animation works by automatically going to the next slide at a regular interval. The field at the left contains the delay in milliseconds between each frame while playing the animation. Increasing this value will make the animation slower, while decreasing it will speed up the animation.
The buttons "<" and ">" are used to navigate to the previous and next slide respectively. Pressing the "go to frame" button opens a small window in which a frame number can be entered. After selecting "OK", the render view will show the entered frame.
Next to the "go to frame", the current frame number and total amount of frames are shown. In the UI, the first frame gets number 1 in order to make it easy to understand for the user. However, internally the frame numbers start at 0, because this is more convenient for coding. This is mainly important to keep in mind when inspecting saved or exported files.

3.3.2 Options Tab

The first tab on the options panel is the "Options" tab. This tab is opened in Figure 4. The tab has two different sections.

The first section provides various options that help with annotating the trajectories of the selected cell. The "copy last frame" button will place an annotation for the selected cell at the same location it was annotated at on the previous frame. "delete annotation" will remove the annotation for the selected cell on the current frame. The grid of buttons below these first two buttons are used to move the annotation slightly in the positive or negative X, Y or Z direction. This can be useful for improving the accuracy of annotations.

The second section contains a few options to change the look of the rendered scene. "show image" toggles showing the volume rendering of the TIFF files. The annotated cells and trajectories can still be viewed when the volume rendering is hidden. Hiding the volume rendering will significantly reduce the amount of computations that ZebraVishualizer has to make for going to the next slide. The animation will therefore be smoother and faster, at the cost of providing less details.
The "transparency" slider is used to increase or decrease the transparency of the volume rendering. The slider for "sphere size" alters the size of the spheres that annotate cells. These two sliders can be used if the volume rendering or spheres obstruct each other from view or simply to adjust the animation according to preference.

3.3.3 Cell Selection Tab

The second tab makes it possible to add new cells for annotation and allows to select which cell trajectories are visible and which trajectory is being edited. The "Cell Selection" tab is opened in Figure 5.
To select a cell for editing its annotations, the checkbox before "edit" needs to be selected. When clicking with the left mouse button in the render view, a new annotation will be made for the cell that is selected in this tab. The annotation options on the "Options" tab will also apply to only

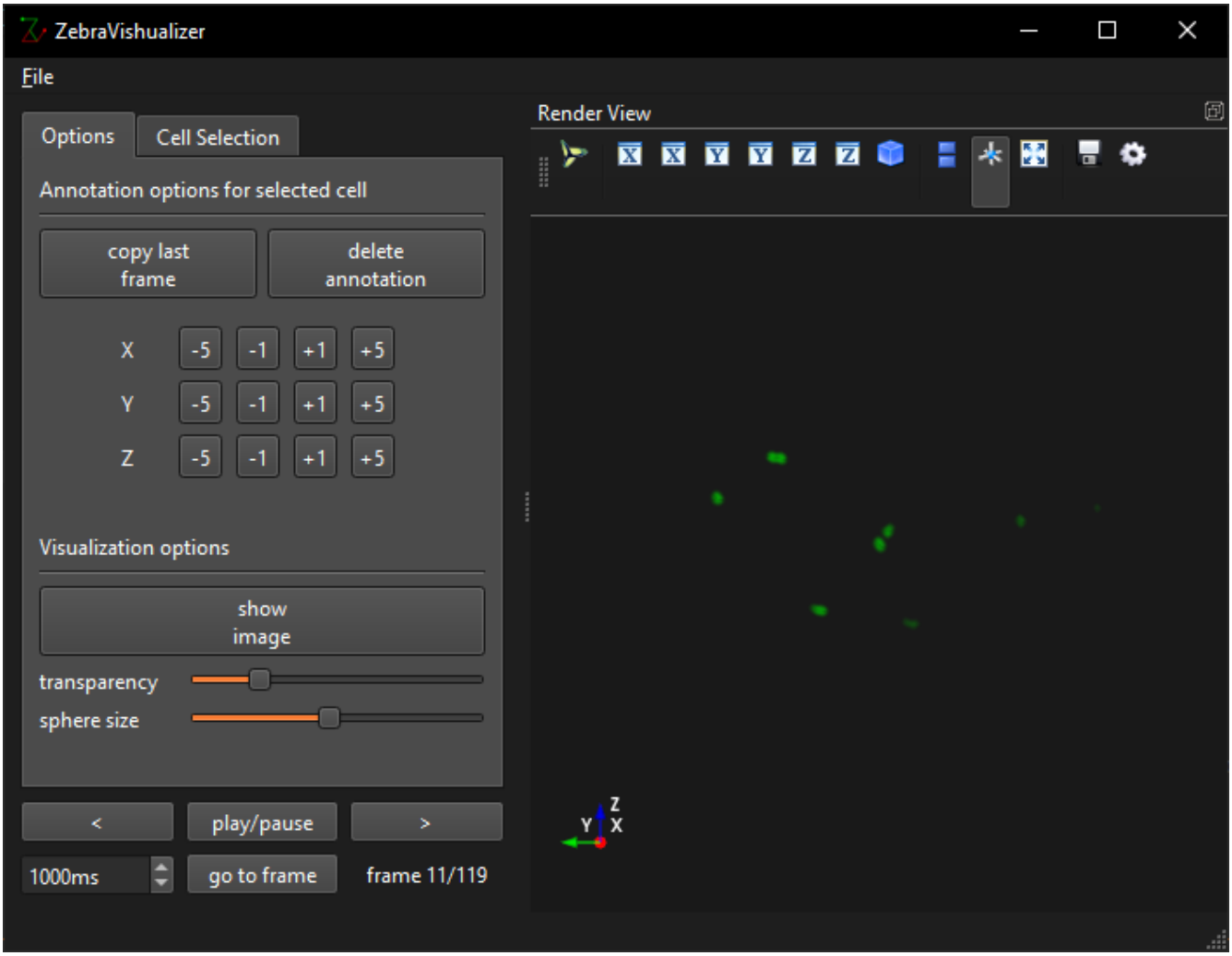

Figure 4: The UI of ZebraVishualizer. The "Options" tab is selected and the axis indicator is shown.

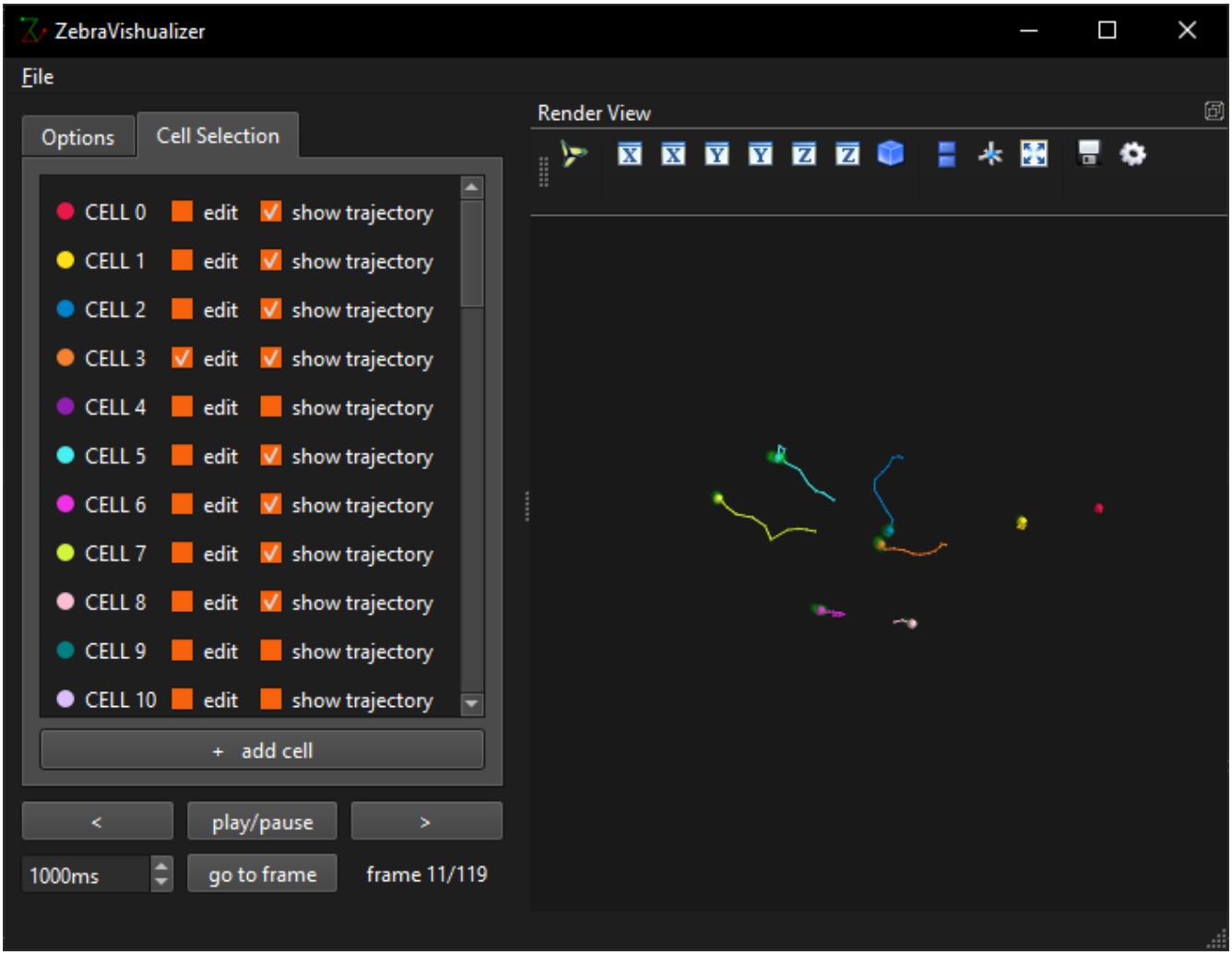

Figure 5: The UI of ZebraVishualizer. The "Cell Selection" tab is opened and the trajectories of multiple annotated cells are shown. The cell numbered with 3 is selected for editing. The annotations for this cell are shown in orange.

the selected cell. At all times exactly one cell is selected for editing. When selecting another cell, the previously selected cell will be deselected.
By checking the checkbox before "show trajectory" after a certain cell, the trajectory of that cell will be shown as a line in the color of the cells annotations. The trajectory is drawn by connecting the annotations of the specific cells on each consecutive frame. The trajectory is only shown up to the current frame and will be removed if the cell doesn't have an annotation on the current frame. If one or more frames is missing an annotation for the selected cell, then there will be a gap in the trajectory. The trajectory can be shown for multiple cells at the same time, as well as for only one cell or no cells at all. The spheres indicating annotations on the current frame will always be shown. When importing trajectories, one row of options will be added to the "Cell Selection" tab for each cell that has annotated trajectories in the imported file. If more cells need to be manually annotated, these can be added by using the "+ add cell" button. Pressing this button will add a new row to the tab, allowing for another cell to be manually annotated. Rows can not be removed from the "Cell Selection" tab.

4 Conclusion and Discussion

The goal of this thesis was to create a software tool for visualizing the three dimensional movement of neutrophils in zebrafish. This goal was accomplished by the creation of the application ZebraVishualizer. ZebraVishualizer uses volume rendering in order to create 3D projections of the original neutrophil data. Other data in a similar format can be used as well. It is also possible to import trajectory data generated by the algorithm of Li et al. [CWW+22] or by ZebraVishualizer itself. The trajectories will then be visualized as colored lines. It is also possible to adjust these trajectories or to create new ones by annotating the cells. The volume renderings and trajectories can be played as an animation by automatically showing each frame in order. The scene in which all of this is rendered can easily be navigated so that the cells and trajectories can be viewed from all possible angles and distances. ZebraVishualizer provides several options to adjust the visualization for optimal vision.
Despite all these features, there is still a lot that can be improved about ZebraVishualizer.

First of all, there is a bug which in some cases causes the cameras location to automatically change so that all elements in the scene are be visible in the view. This bug occurs when something in the scene changes, for example when going to another frame or when making an annotation. After a change like this happens, the camera will automatically move to a position from where all elements of the scene are visible and centered in the view. This will change panning and zooming of the camera. The rotation of the camera will remain the same.
This makes it more difficult to get a close up view of only one cell or a few specific cells while playing the animation. If a cell moves off the edge of the view, this bug can sometimes cause the camera to unexpectedly move when going to another frame, because then the center off all visible cells changes. This is especially a problem when the volume rendering is hidden, because then only the annotations are visible and not always all cells are annotated, causing a "shaky" animation.
Despite the severity of this bug and the understanding of what it does, the underlying cause of the bug could not be found and thus the bug remains a problem.

There are no other major bugs in the software other than the one mentioned above. However, there is still another important feature that could be improved. The rendering of the volume and cell trajectories is very slow. Because of this, the maximum speed at which the animation can played is severely limited. If the delay time is set to a very low value, the animation will sometimes not show frames in order to try and keep up.
This may be improved by saving the matrices that are used for creating the volume rendering. However, this may increase the memory that is used by the software. It may also reduce the loading time upon opening or during the first playback of the animation, since the matrices will still need to be created at some point.

Furthermore, there are many useful features which are not implemented yet.
For example, selecting many cells for showing their trajectory is currently very tedious. Adding a button to select all cells would make this a lot quicker.
In addition there isn't a way to delete a cell from the "Cell Selection" tab. Being able to remove unannotated cells would remove unnecessary clutter from this tab. The "Cell Selection" tab could also be improved by allowing the user to change the name and colour of cells. This would make it easier to distinguish between different cells.

At the moment, the volume rendering is colored green by the code, not because the original image was this colour. This was implemented in this way, because the masked TIFF files from Li et al. lost their colour during segmentation in order to label the cells, see Figure 3. The volume is then colored green in order to make it recognizable as GFP to biologists. If the imported data was not using GFP, it would be better if the user could choose the colour of the volume rendering themselves. If the data sample contains different cell types that were made visible with multiple fluorescent markers, then it would be important that ZebraVishualizer can assign multiple different colours to each of these cell types.

Another useful feature would be the ability to obtain information about certain aspects of the trajectories, such as their velocity, acceleration or direction. An option to compare different trajectories with each other would be useful as well. At the moment, ZebraVishualizer can solely be used for annotating the trajectory and viewing it. Adding calculations on the trajectory would allow ZebraVishualizer to be used as an analytics tool as well. In this way, the tool could provide even more new insights. A feature like this could probably be best added into a new tab next to the existing "Options" and "Cell Selection" tabs.

It would also be useful if there was an undo-button, in case an annotation is accidentally changed. This can easily happen while trying to adjust the camera. However, such a feature will probably be difficult to implement.

Currently, it can be difficult to know in which direction the scene is rotated, even when the axis indicator is shown, because it is not easily possible to tell which axis corresponds to which side of the sample. This could be made easier by adding the outline of the zebrafish tail into the scene. However, in order to implement this, it needs to be clear from the imported data which side of the data corresponds to which side of the zebrafish, which is not the case. The addition of a zebrafish tail object would also make ZebraVishualizer specific for experiments on zebrafish

tails, while one of its advantages is that it could also be used for similar data from other experiments.

Finally, an executable version of ZebraVishualizer could be made. Right now, ZebraVishualizer is a Python project and thus it is needed to install Python and all the needed libraries in order to run the software. This can be challenging for less tech savvy users. Releasing ZebraVishualizer as an executable would make it easier and quicker to install and would take less storage space.
A web version of the tool would be even easier to use, since it wouldn't need any installation. However, this will probably be more difficult to realize then an executable version.

For further research, ZebraVishualizer could be used to generate trajectory data and improve data obtained by tracking algorithms. The resulting trajectories can then be used as training data for a machine learning algorithm. Because the trajectory data is improved by ZebraVishualizer, the machine learning algorithm will be able to learn well and provide more accurate results. In the future, this machine learning algorithm can then be used for finding the trajectories of cells, which may be faster and easier than other existing methods.
Aside from this, ZebraVishualizer can already be used itself for research on the movement of neutrophils in zebrafish, especially if the above mentioned improvements are made. By using the animation and trajectory data of multiple samples, new insights can be gained about for example the speed, acceleration and direction in which the neutrophils move. The effect of certain genes, such as TLRs and MyD88 or of different types of injuries and pathogens can be investigated as well by using different datasets. ZebraVishualizer can also be used to research the movement of other cell types, such as macrophages.

In conclusion, ZebraVishualizer is a useful tool for annotating or improving cell trajectory data and visualizing this data. The tool can still be improved in several ways. However, it already provides multiple functions that will be valuable for research on cell movements.

References

[BHK+14] A. Bernut, J. L. Herrmann, Dubremetz-J. F. Kissa, K., J. L. Gaillard, G. Lutfalla, and et al. Mycobacterium abscessus cording prevents phagocytosis and promotes abscess formation. *Proc. Natl. Acad. Sci. U S A.*, 111:E943–E952, 2014. doi:10.1073/pnas.1321390111.

[Bos23] A Bos. A tool for manual 3d cell track annotation through time. Bachelor's thesis, Leiden University, 2023.

[BRG+04] V. Brinkmann, U. Reichard, C. Goosmann, B. Fauler, Y. Uhlemann, D. S. Weiss, and et al. Neutrophil extracellular traps kill bacteria". *Science*, 303 (5663):1532–1535, 2004. doi:10.1126/science.

[CB02] S. Cohen and R. C. Burns. *Pathways of the Pulp*, page 465. St. Louis: Mosby, 8 edition, 2002.

[CSH+08] T. Chtanova, M. Schaeffer, S. J. Han, G. G. van Dooren, M. Nollmann, P. Herzmark, and et al. Dynamics of neutrophil migration in lymph nodes during infection. *Immunity*, 29 (3):487–96, 2008. doi:10.1016/j.immuni.2008.07.012.

[CWW+22] Li. C, W.C. Yiu. W, Hu. W, Cao. L, and J. Verbeek. F. A feature weighted tracking method for 3d neutrophils in time-lapse microscopy. *BIBM*, 2022.

[CZCL20] L. Chen, L. Zheng, P. Chen, and G. Liang. Myeloid differentiation primary response protein 88 (myd88): the central hub of tlr/il-1r signaling. *J. Med. Chem.*, 63:13316–13329, 2020. doi:10.1021/acs.jmedchem.0c00884.

[Dah06] R Dahm. The zebrafish exposed. *American Scientist*, 94 (5):446–53, 2006. doi:10.1511/2006.61.446.

[Den20] Z. Deng. Cell trajectory visualisation in zebrafish. Bachelor's thesis, Leiden University, 2020.

[EM08] T. Ear and P. P. McDonald. Cytokine generation, promoter activation, and oxidant-independent nf-kappab activation in a transfectable human neutrophilic cellular model. *BMC Immunology*, 9:14, 2008. doi:10.1186/1471-2172-9-14.

[EPRP07] R. E. Engeszer, L. B. Patterson, A. A. Rao, and D. M. Parichy. Zebrafish in the wild: a review of natural history and new notes from the field. *Zebrafish*, 4 (1):21–40, 2007. doi:10.1089/zeb.2006.9997.

[GC02] G. S. Gerhard and K. C. Cheng. A call to fins! zebrafish as a gerontological model. *Aging Cell*, 1 (2):104–111, 2002. doi:10.1046/j.1474-9728.2002.00012.x.

[GD24] R. Gijsbers and Luuk D. Zebrafishualizer. Bachelor's thesis, Leiden University, 2024.

[HCT+13] K. Howe, M. D. Clark, C. F. Torroja, J. Torrance, C. Berthelot, M. Muffato, and et al. The zebrafish reference genome sequence and its relationship to the human genome. *Nature*, 496 (7446):498–503, 2013. doi:10.1038/nature12111.

[HD15] T. Hato and P. C. Dagher. How the innate immune system senses trouble and causes trouble. *Clin. J. Am. Soc. Nephrol.*, 10:1459–1469, 2015. doi:10.2215/CJN.04680514.

[HK09] M. J. Hickey and P. Kubes. Intravascular immunity: the host-pathogen encounter in blood vessels. *Nature Reviews Immunology*, 9 (5):364–375, 2009. doi:10.1038/nri2532.

[HLWR13] K. M. Henry, C. A. Loynes, M. K. Whyte, and S. A. Renshaw. Zebrafish as a model for the study of neutrophil biology. *Journal of Leukocyte Biology*, 94 (4):633–42, 2013. doi:10.1189/jlb.1112594.

[HvSL+21] Wanbin Hu, Leonie van Steijn, Chen Li, Fons J. Verbeek, Lu Cao, Roeland M. H. Merks, and Herman P. Spaink. A novel function of tlr2 and myd88 in the regulation of leukocyte cell migration behavior during wounding in zebrafish larvae. *frontiers in Cell and Developmental Biology*, 9, 2021. doi:10.3389/fcell.2021.624571.

[HYS+19] W. Hu, S. Yang, Y. Shimada, R. Munch, M. adn Marin-Juez, A. H. Meijer, and et al. Infection and rna-seq analysis of a zebrafish tlr2 mutant shows a broad function of this toll-like receptor in transcriptional and metabolic control and defense to mycobacterium marinum infection. *BMC Genom.*, 20:878, 2019. doi:10.1186/s12864-019-6265-1.

[JM14] Deguine J. and Barton G. M. Myd88: a central player in innate immune signaling. *F1000Prime Reports*, 6:97, 2014. doi:10.12703/P6-97.

[JSJ+18] H. Jiang, D. Sun, V. Jampani, M. Yang, E. G. Learned-Miller, and J. Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. *CVPR*, 2018.

[LAA+13] T. Lämmermann, P. V. Afonso, B. R. Angermann, J. M. Wang, W. Kastenmüller, C. A. Parent, and et al. Neutrophil swarms require ltb4 and integrins at sites of cell death in vivo. *Nature*, 498 (7454):371–5, 2013. doi:10.1038/nature12175.

[LG14] T. Lämmermann and R. N. Germain. The multiple faces of leukocyte interstitial migration. *Seminars in Immunopathology*, 36 (2):227–51, 2014. doi:10.1007/s00281-014-0418-8.

[MDS12] E. Meijering, O. Dzyubachyk, and I. Smal. Methods for cell and particle tracking. *Methods Enzymol.*, 504:183–200, 2012. doi:10.1016/B978-0-12-391857-4.00009-4.

[Med08] R. Medzhitov. Origin and physiological roles of inflammation". *Nature*, 454 (7203):428–435, 2008. doi:10.1038/nature07201.

[MJGB15] K. E. G. Magnusson, J. Jalden, P. M. Gilbert, and H. M. Blau. Global linking of cell tracks using the viterbi algorithm. *Ieee Trans. Med. Imag.*, 34:911–929, 2015. doi:10.1109/TMI.2014.2370951.

[MS11] A. H. Meijer and H. P. Spaink. Host-pathogen interactions made transparent with the zebrafish model. *Current drug targets*, 12 (7):1000–1017, 2011.

[Par06] D. M. Parichy. Evolution of danio pigment pattern development. *Heredity*, 97 (3):200–210, 2006. doi:10.1038/sj.hdy.6800867.

[PCPH18] S. A. Park, Y. H. Choe, E. Park, and Y. M. Hyun. Real-time dynamics of neutrophil clustering in response to phototoxicity-induced cell death and tissue damage in mouse ear dermis. *Cell Adhesion Migration*, 12 (5):424–431, 2018. doi:10.1080/19336918.2018.1471322.

[PS18] Park E Hyun YM Park SA, Choe YH. Real-time dynamics of neutrophil clustering in response to phototoxicity-induced cell death and tissue damage in mouse ear dermis. *Cell Adhesion Migration*, 12 (5):424–431, 2018. doi:10.1080/19336918.2018.1471322.

[Qur21] Z. Qureshi. annot3d. https://github.com/ZoraizQ/annot3d, 2021.

[RCKC98] S. L. Robbins, R. S. Cotran, V. Kumar, and T. Collins. Robbins pathologic basis of disease. *Philadelphia: W.B Saunders Company*, 1998.

[RLT+06] S. A. Renshaw, C. A. Loynes, D. M. Trushell, S. Elworthy, P. W. Ingham, and M. K. Whyte. A transgenic zebrafish model of neutrophilic inflammation. *Blood*, 108:3976–3978, 2006. doi:10.1182/blood-2006-05-024075.

[RSAR+15] D. J. Raiten, F. A. Sakr Ashour, A. C. Ross, S. N. Meydani, H. D. Dawson, C. B. Stephensen, and et al. Inflammation and nutritional science for programs/policies and interpretation of research evidence (inspire). *The Journal of Nutrition*, 145 (5):1039S–1108S, 2015. doi:10.3945/jn.114.194571.

[SBB+07] C. N. Serhan, S. D. Brain, C. D. Buckley, D. W. Gilroy, C. Haslett, L. A. O'neill, et al. Resolution of inflammation: state of the art, definitions and terms. *FASEB J*, 21:325–332, 2007. doi:10.1096/fj.06-7227rev.

[SRE12] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri. Nih image to imagej: 25 years of image analysis. *Nature Methods*, 9 (7):671–675, 2012. doi:10.1038/nmeth.2089.

[SWG10] C. N. Serhan, P. A Ward, and D. W. Gilroy. *Fundamentals of Inflammation*. Cambridge University Press, 2010.

[TCB+15] V. Torraca, R. Cui, C.and Boland, J. P. Bebelman, A. M. Van Der Sar, M. J. Smit, and et al. The cxcr3-cxcl11 signaling axis mediates macrophage recruitment and dissemination of mycobacterial infection. *Dis. Model. Mech.*, 8:253–269, 2015. doi:10.1242/dmm.017756.

[TW17] S. Y. Tan and W. Weninger. Neutrophil migration in inflammation: intercellular signal relay and crosstalk. *Current Opinion in Immunology*, 44:34–42, 2017. doi:10.1016/j.coi.2016.11.002.

[TZ02] C. Thisse and L. I. Zon. Organogenesis–heart and blood formation from the zebrafish point of view. *Science*, 295 (5554):457–462, 2002. doi:10.1126/science.1063654.

[vdVVSSM13] M. van der Vaart, J. J. Van Soest, H. P. Spaink, and A. H. Meijer. Functional analysis of a zebrafish myd88 mutant identifies key transcriptional components of the innate immune system. *Dis. Model. Mech.*, 6:841–854, 2013. doi:10.1242/dmm.010843.

[Vij18] K. Vijay. Toll-like receptors in immunity and inflammatory diseases: past, present, and future. *Int. Immunopharmacol*, 59:391–412, 2018. doi:10.1016/j.intimp.2018.03.002.

[VWKH+17] R. C. Van Wijk, E. H. Krekels, T Hankemeier, H. P. Spaink, and P. H. Van der Graaf. Systems pharmacology of hepatic metabolism in zebrafish larvae. *Drug Discovery Today: Disease Models*, 22:27–34, 2017. doi:10.1016/j.ddmod.2017.04.003.hdl:1887/49224.

[WMVY21] C. Wen, T. Miura, V. Voleti, and K. Yamaguchi. 3deecelltracker, a deep learning-based pipeline for segmenting and tracking cells in 3d time lapse images. *Elife. Sci.*, page 1–36, 2021.

[YWC10] L. Yu, L. Wang, and S. Chen. Endogenous toll-like receptor ligands and their biological significance. *J. Cell Mol. Med.*, 14:2059–2603, 2010. doi:10.1111/j.1582-4934.2010.01127.x.