Universiteit Leiden
The Netherlands

# Opleiding Informatica

A tool for manual 3D cell track annotation through time

Aaron Bos

Supervisors:
Lu Cao

BACHELOR THESIS

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

22/06/2023

**Abstract**

TLR2 and MyD88 are genes that have an effect on the regulation of leukocyte cell migration behaviour during the tail wounding in zebrafish larvae. Cell tracking can be used to analyse the influence of TLR2 and MyD88 on cell migration behaviour. This can be done by manual cell tracking, but this is time consuming and can not be used for high-throughput data. An alternative would be the usage of artificial intelligence for cell tracking. Deep learning has already been used to train an AI to track cells in live-cell imaging data. However, there are limitations to the usage of deep learning. Deep learning is a supervised learning method and thus requires labeled examples. The AI that had been trained to track cells in live-cell imaging data used a data set of over 11.000 trajectories to train. The creation of an annotation tool could help ease and speed up the creation of training data. This thesis contributed a tool to help ease the annotation process. These annotations can be used as training data. This training data can be used to train a deep learning model to do the cell tracking. The cell tracking data could help with the analyses of the influence of TLR2 and MyD88 on leukocyte cell migration behaviour.

# Contents

# 1 Introduction

## 1.1 Biological Background

TLR2 and MyD88 are involved in modulating neutrophil and macrophage cell migrating behaviour upon zebrafish larval tail wounding[HvSL+21]. A neutrophil is a type of white blood cell, these cells are one of the first responses to an infection. They destroy microorganisms by releasing enzymes after ingesting these microorganisms[nct]. TLR stands for Toll-like receptor, these receptors regulate inflammatory responses by acting as recognition factors for PAMPs and DAMPs [YWC10] [Vij18]. PAMPs are pathogen associated molecular patterns and Damps are damage associated molecular patterns. So to summarize, TLRs recognize molecular patterns associated with pathogens and damaged cells and can regulate inflammatory responses to this. TLR2 is, as the name suggests, a TLR. This one specifically senses bacterial lipoproteins[QNT+04]. MyD88 stands for myeloid differentiation factor 88 protein. MyD88 binds to the TIR domain of TLRs which activates downstream signaling [TA04];[CZCL20].
It is important that neutrophil migration is handled carefully. This is because of toxic granule contents that can be released by persisting neutrophil recruitment, and this will cause further damage to tissues [EW89]; [SBB+07]; [BLP13]. Both macrophages and neutrophils depend on PRRs or membrane localized pattern recognition receptors to detect invading microbes and their associated tissue damage [HD15]. PPRs play a crucial roll in recognizing PAMPs and DAMPs. One of the best known PRRs is TLR2. TLR2 acts as a hetrodimer with TLR1 or TLR6, it recognises gram positive bacteria. It is presumed that this recognising is based on specific binding to their cell wall components [QNT+04]; [ONMW12]. TLR2 expressing occurs after tissue injury. Infections aren't necessary for it's expression. An example of what could trigger it would be acute ischemic injury, which is an injury from reduced blood flow [SDC+07];[CBCC+12]; [XZL+13]; [MMW+14]. It is hypothesised that injury-induced TLR2 expression and activation is important to human health[SPF11];[MYvR+13];[MMW+14]. This hypotheses is based on a study that showed that TLR2 deficiency in mice had reduced macrophage infiltration into normal muscle following acute injury [MGL+16]. TLR2 deficiency in mice also causes a defective ability to recruit neutrophils to an injured liver [MMW+14].
There are studies that have shown that MyD88 expression changes after tissue injury. Just like TRL2, MyD88 has upregulated expression following ischemic injury in mice [WRP+20]. It has also been shown that both the expression of MyD88 and TLR2 are increased in diabetic wounded mice [DTB+10].
The paper studied leukocyte cell migration in zebrafish larvae after tail wounding. The reason the paper used the zebrafish model is due its small size and transparency of their larvae. It is possible to study the roles in leukocyte migratory behaviour after tail wounding in zebrafhis due to the availability of mutants of TLR2 and MyD88 [HLWR13]; [vdVvSSM13];[HYS+19]; [XTO+19]; [STK+19]. The paper used live fluorescent imaging to investigate the effect of the TLR2 and MyD88 mutations on leukocyte migration upon tail wounding. It was found that the mutants had reduced numbers of recruited neutrophils and macrophages at the wounding area compared to the control group. The migration in the TLR2 and MyD88 mutations was analysed upon the wounding. The results show that the directional persistence in the distant neutrophils was negatively affected in the mutations, but the migration speed was unaffected. For the distant macrophages both of these aspects where negatively affected. The paper has shown that the regulation of cell migration

behaviour of neutrophils and macrophages during wounding has TLR signaling as one of the involved factors[HvSL+21].

## 1.2 Cell tracking Background

Cell tracking is generally divided into two steps. The first step is cell segmentation. This is the splitting of images into segments. Here, each segment represents a cell. The second step is the usage of cell similarities to determine the trajectory of cells [HCH+18];[LLD20]; [TGS+19]; [MCSL+20];[LMK+21]. Algorithms for cell segmentation have gone through a long development process. Going from threshold segmentation and watershed-based segmentation, to the now popular deep learning segmentation methods.
Deep learning is a modern method of machine learning. It consists of a neural network with three or more layers. Neural networks are groups of nodes connected to each other by edges. These nodes send each other signals through these edges the in form of a number. The signals are modified by a weight attached to each edge. The sum of the signals a node receives is put in a non linear function. The nodes have a threshold. If the output of the function is higher than this threshold, the nodes will send a signal. These neural networks are trained for a task by providing an input and a result. During this training the weights of the edges are modified. This happens as reaction to the difference between the predicted result and the actual result. This is done using backpropagation. This attempts to adjust errors backwords. Meaning from output towards input. The additional layers of a deep learning network help to optimise and refine for accuracy [IBM]. A popular segmentation model in the biomedical field is 3D U-Net[RFB15];[KRT+21]. It is a deep learning model that reduces the amount of manual annotation that has to be done. It does this by only requiring a small amount of annotated data for training. 3D U-net has had a good performance on kidney embryo segmentation. However 3D U-net is not perfect for tracking neutrophils due to them colliding. This can however be solved by using the watershed method as a post processing [WMV+21].
Determining the trajectory of cells after segmentation is done by going frame by frame and looking at similarities. A graph-based tracking algorithm has shown to have great performance. This was demonstrated in the Cell tracking challenge in 2019 and 2020. This method utilises relative cell location information to associate the cell tracks[SJM21]. Another method is rule based methods ([SJM21];[PCGM22]). These methods are easier to tailor to the features of specific data sets. Deep learning tracking methods can be used to extract these features automatically. 3DeeCellTracker is a proposed deep learning-based software pipeline for cell tracking[WMV+21]. Data sets of cardiac cells in zebrafish and neurons in worm brains where used for this. Cells in these data sets could be simulated. This was due to the characteristic that these specific cells move in the same pattern. This sadly can not be done for neutrophils due their more complex and irregular cell movements. There is a 3D cell association learning network that was designated and used for the neutrophil tracking problem [RMC21]. While it did solve the cell collision problem and achieved significant performance, it has a problem. It required a large number of annotated examples for training.

The relevant paper can be spoken about now that its background has been discussed. The paper proposed a 3D U-Net segmentation model and a rule based tracking method that was tailored to the neutrophil data set it used. They started by improving the image quality. After this, additional layers where added to the raw 3D data. This was done using linear interpolation to enlarge the z-dimension from 8 to 29 layers. A 3D U-net segmentation model was trained. The watershed algorithm was also applied for colliding cells. Cell tracking was improved by adding additional features and calculating a similarity score using those features. The results have shown that the feature-based method that was created, preformed better than the graph-based method used in [SJM21]. [LYH+22]

## 1.3 Main Goal

To get more details regarding the change in cell migration behavior, a tool was asked to be made. The creation of this tool is the main goal of this thesis. The plan is to use this tool to annotate the 3D movement of neutrophils in multiple zebrafish larvea through time. These annotations can help show the involvement of TLR2 and MyD88 in modulating the migration behaviour of the looked at cells. This is done by using these annotations as examples for a machine learning algorithm. This machine learning algorithm will then try to learn how to annotate the movement of neutrophils. These annotations can then be used for further research. The 3D space in which the cells are annotated are created from multiple multi-layer 2D images. These multi-layer images are collections of 2D images which each represent a layer of a 3D image. The output of this tool will be an Excel file containing the annotations.

## 1.4 Research Questions

There are two research questions this thesis asks:

1. What method can be used to visualize 3D neutrophils through time from multi-layer 2D image data which shows neutrophils locations in a zebrafish larvae?

2. How can a 3D visualization of neutrophils through time be used for 3D cell track annotation of the movement of these neutrophils?

## 1.5 Related work

3DeeCellTracker is a related work that has been mentioned in the background [WMV+21]. It is a deep learning based software pipeline. 3DeeCelltracker used an alternative way to acquire more training data. This was done by creating synthetic training data. The synthetic training data was created by first transforming annotated data using an affine function, and adding random movement to each point after that. Figure1 illustrates this process. The tool this thesis made would have to generate less training data if additional synthetic training data could be created. But this isn't the case. This is due to the movement of neutrophils being more complex than movement of the cells 3DeeCellTracker used for its training data.

Another related work created a 2D cell tracking software called LIM Tracker[AOKA22]. It is implemented as a plugin for ImageJ/Fiji. LIM tracker has a conventional tracking function that consists of recognition processing and link processing. It has a sequential search-type tracking function that is based on pattern matching. LIM Tracker also has a manual tracking function. The software has a pen tool function. This tool can be used to create arbitrarily shaped regions. This tool can also be used to create training data. This can be used with the deep learning operation

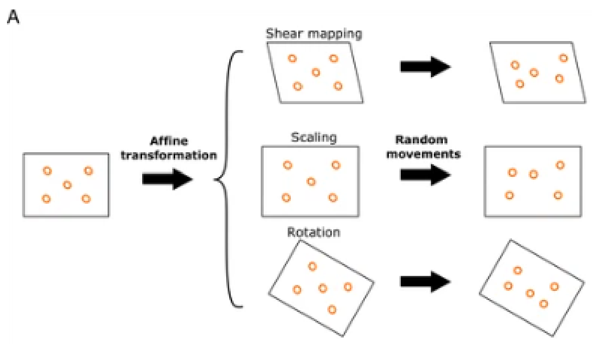

Figure 1: Generation of synthetic training data[WMV+21]

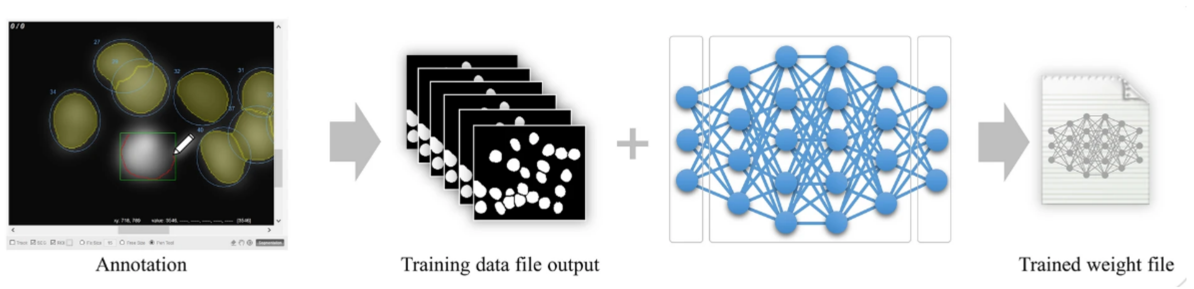

Figure 2: The deep learning training procedure in LIM Tracker[AOKA22]

tool that is build into the software. This allows the user to create a recognition process for their own data set. Figure 2 shows the procedure of the creating of a recognition process for a data set in LIM Tracker.

# 2 Materials and methods

## 2.1 Programming language and libraries

The program was written in Python, all code was executed in Python 3.10.10. The most important Python libraries used in the project were:

- PySide2
- Mayavi
- PIL
- AnnotationSpace3D

## 2.2 Image data

Raw 3D data samples in form of TIFF files were provided and used for the creation of the tool. All samples where series of 120 TIFF files. These files where used for the creation of the 3D projection and the testing of the annotation.

## 2.3 Forked code

The made tool was built upon an extension of annot3D, it was made by Zoraiz Qureshi from the University of Virginia. Annot3D is an interactive annotation tool for 3D TIFF volumes. It has an integrated UNET semantic segmentation for binary annotation along with embedded live volume rendering. Figure 3 shows annot3D. Annot3D was used as staring point of the tool due to the need of similar functions. It already had a function which would project a multi-layer TIFF file as a 3D projection. This function didn't work for the 3D data samples used during this project. This was due to some differences in the TIFF files used in annot3D and those used in this project. This could however be fixed by adjusting the function that handled the reading of the TIFF files.

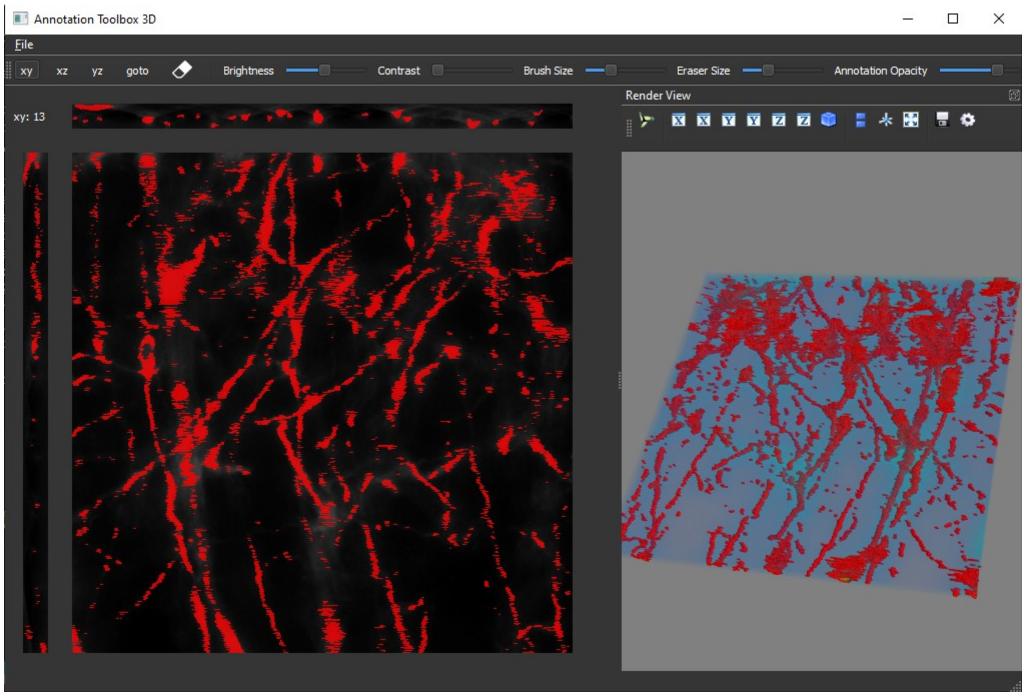

Figure 3: annotations made in annot3d

## 3 Main contributions

### 3.1 Transforming TIFF files to a matrix

Annot3D already started with a function that would transform a multi-layer TIFF file to a 3D matrix. This was done by going over every pixel in the image and looking at it's intensity. This would then be stored in a 2D array. After all images where transformed into a 2D array, these arrays would be combined to form a 3D array. However, this function could not be used for the planed tool in its original state. The first problem was the fact that it was hard coded which file would be read and transformed to a 3D matrix. This was solved creating a dictionary of the TIFF files in the data map upon startup of the tool. For this dictionary, the files in the data map are sorted and assigned a number. Dictionary entries are then added where each filename is assigned to its number.
There was still a problem with the current function. There was contrast a function that was meant to remove the outliers from the data. This was done by lowering the values of the 99th percentile. This worked fine for the TIFF files Annot3D originally used. But it caused problems for the TIFF files in the data set provided for the creation of this tool. Figure 4 shows the difference in data sets. As the figure shows, most space in our own data set is empty while the data Annot3D uses is far more dense. This difference caused problems with the contrast function. The contrast function would remove outliers. This was done by lowering all values higher than the 99th percentile. Due to the sparseness of the data, any value higher than 0 would be higher than the 99th percentile and thus be lowered to zero. This problem was solved by simply no longer removing the 99th percentile.
Another addition was to include some filters to reduce noise in the image. This was done using the Python Imaging Library or PIL for short. First, a contrast enhance function is used to make

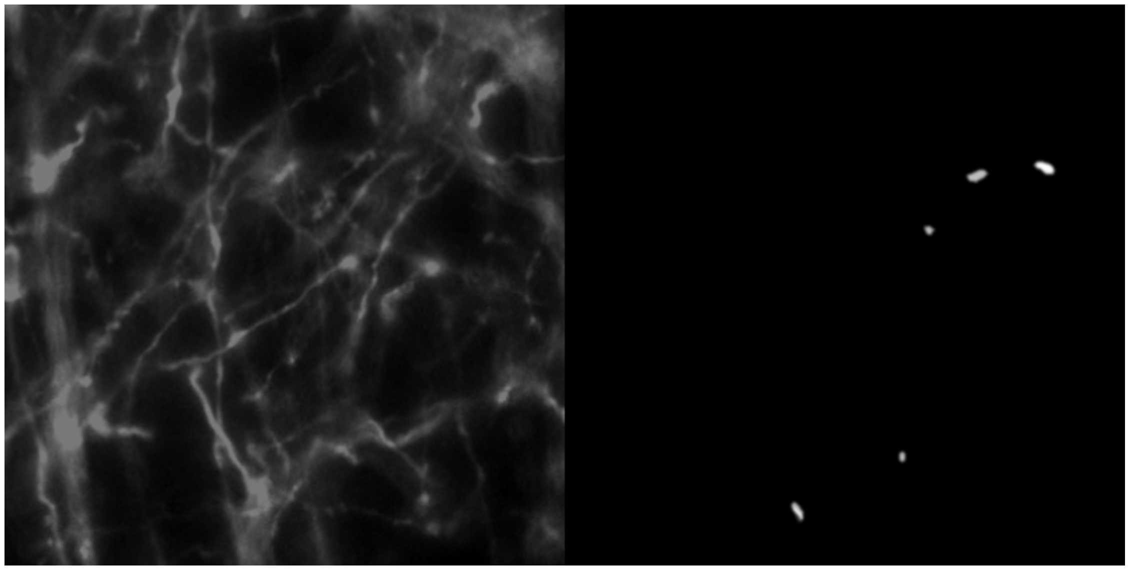
Figure 4: A comparison of Annot3D data on the left and our own data on the right

the less bright cells stand out more against the black background. After this, a median filter is used. The median filter goes through every pixel and replaces the value of it with the median of the pixels in the window. The so called window are the pixels in an $x$ range of the pixel. A filter with $x = 1$ would look at the median value of the neighbouring pixels. After this, a Gaussian blur filter is applied to the image. This filter smooths the image using a Gaussian function. A Gaussian function is known for expressing the normal distribution in statistics. The function differs depending on how many dimensions are there. In this case images are two dimensional, so the formula for the Gaussian function would be: $F(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$ where $\sigma$ is the standard deviation and x and y are the respective dimensional distance from the origin. The origin is usually at the center. This formula is applied to create a Gaussian distribution. This distribution is then used to build a convolution matrix that is applied to the original image. While there are algorithms that result in a more uniform smoothness, this one is better at preserving edges.

## 3.2 Projection to 3D

Annot3D already created a 3D projection from the 3D matrix. This was also the main reason the tool made in this thesis was build from Annot3D. The 3D projection was done using the Mayavi Python library. The Mayavi library is meant for 3D scientific data visualization and plotting. The specific function that was used for the projection was the volume rendering function. Volume rendering is a series of techniques used to create a projection of a 3D data set. It usually uses a scalar field as input. In this case the 3D matrix that was created from a single TIFF file is used with the volume rendering function of Mayavi. The volume rendering of Mayavi will render places with a low value with more transparency than places with a high value. An example of the result of rendering can be seen in figure 5.

The most significant change that is made to the function that handles the 3D projections, is the ability to change between different projections. This is done by removing the previous projection, and then reading the TIFF file of the new projection to obtain a matrix that can be projected.

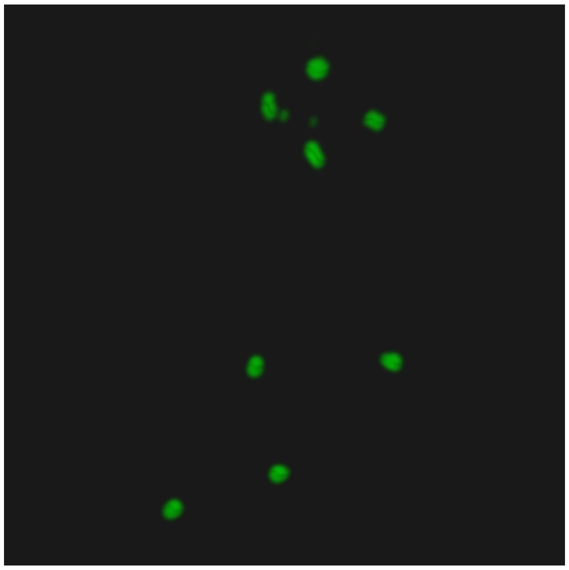

Figure 5: example of volume rendering of a TIFF file using Mayavi

The switching between projections allows the user to not only annotate the cell locations in a single image but annotate through time. There are three buttons that allow the user to change the projection. Two simple buttons that switch to the next or previous TIFF file in the dictionary and a goto button. The goto button allows the user to input a number of the file which they wish to see, which of course causes that file to be loaded.

## 3.3 Annotation

Annot3D preformed annotation differently than the tool made for this thesis. In Annot3D annotation was done by drawing on the 2D images. The user would have to go through all the layers and annotate. This would be a lot of work for only annotating a single cell. The tool does the annotation by allowing the user to click on the cell in 3D projection. The mayavi library has a picker function that can be used. This function will return information at the moment when the user clicks somewhere in the screen. One of the pieces of information it returns is the coordinates the user clicked on. This information is used for two things. First, it will be saved in an array to remember the location of that cell if it ever needs to be re-renderd . Second, it is used to render a sphere on that location using the points3d function of mayavi. Figure 6 shows an example of how this looks like.

As you might see in the figure the sphere doesn't appear in the center of the cell. To allow the user to move the sphere to the center of the cell buttons are used. These can be seen in figure 7. Just using these buttons would be hard because the user wouldn't know which directions the axes are. Mayavi has a function to display axis indicators. This function is used to help the user orientate the axis. These indicators can be seen in figure 8. The user might wish to remove annotations due to various reasons. This can simply be done with the delete button. The delete button deletes the currently selected cell annotation. Another button with a somewhat related function is the continue button. This button will place an annotation on the exact same spot of the previous annotation of that cell if one exists. This will reduce the work of the user by allowing them to place a annotation

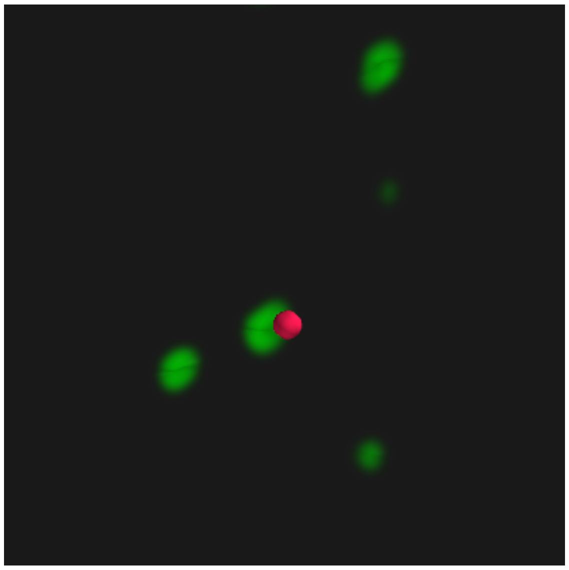

Figure 6: Example of the sphere used to annotate the location of a cell

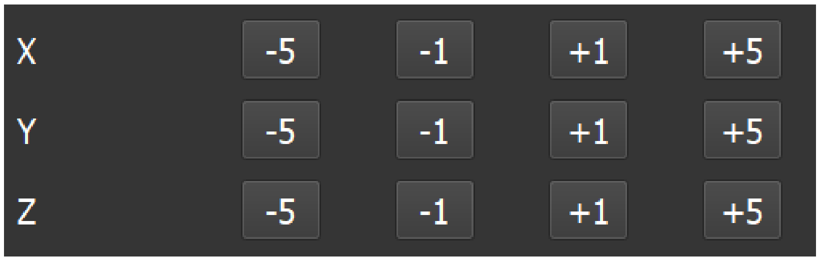

Figure 7: The buttons used to adjust the annotation sphere location

that will only need minor adjustments. The adjustments can be done with the adjustment buttons.
The user has to be able to annotate multiple cells while using the tool. For this reason a drop down menu was added to the interface. The drop down menu allows users to switch between the cell they are annotating. Changing to another cell also changes the color of the annotation sphere that is placed, to help the user differentiate between the different cells they could be annotating. Figure 9 shows what this looks like.
Sometimes cells are too bright while others are barely visible. To help users in these cases, a transparency slider was added. The slider reduces or increases transparency by changing the vmax value of the volume rendering function. The vmax variable is used to scale the colourmap. Vmax is the value that gains the colour associated with the highest value. Lowering it will give more parts of the projection the highest value colour. Making places with lower values less transparent in this case. Decreasing the vmax will decrease the size of the scale. This means that changes in value will

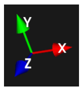

Figure 8: Axis indicators used to let the user orientate

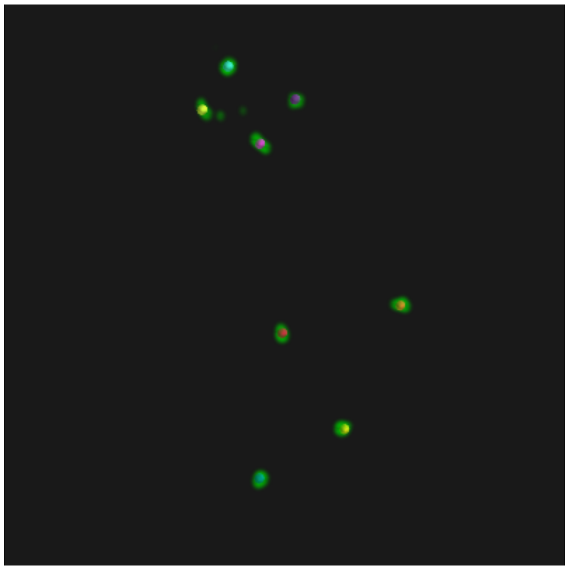

Figure 9: example of a frame with multiple annotations

have a more drastic effect on the transparency. Increasing the value will do the exact opposite. It will lower the change transparency that changes in values will cause. The difference can also be seen in Figure 10.
IN other situations, cells are very small or very big. This difference can be caused by the TIFF files that are used. Very detailed images of a only a couple cells will result in very large cells. Images showing large groups of cells are likely less detailed and will result in very small cells. It is hard to place an annotation in a small cell if the sphere is blocking most of it while placed. It can also be hard to find the center of a larger cell due to the amount of space that the annotation could be placed in. To solve this, a slider was added to increase or decrease the size of the sphere that is used to show the location of annotations. The slider simply changes the value of the sphere size and re-renders all annotation spheres to change their size. The difference between the minimal sphere size and the maximum sphere size can be seen in Figure 11. Figure 12 shows the whole interface to give context to the discussed features.

## 3.4 Annotation trajectory

A trajectory function was created to let the user see the path of the cell that they annotated. This function doesn't require the finalization of the annotation. The function allows the user to visibly see their progress which possibly helps them retain their motivation. The trajectory function draws spheres on the locations the selected cell has been. These locations are obtained from the users annotations. It also shows the movement of the annotated cell by drawing lines between locations that came after the previous cell. Figure 13 shows an example of how this trajectory looks like. In some cases, it might be hard to look at the trajectory due to cells being blocked by the annotation sphere or other cells. Because of this, the volume button was added. Pressing this button while the trajectory function is active will cause the the annotation spheres and the cell projections to be removed. An example of the effect of the button can be seen in figure 14.

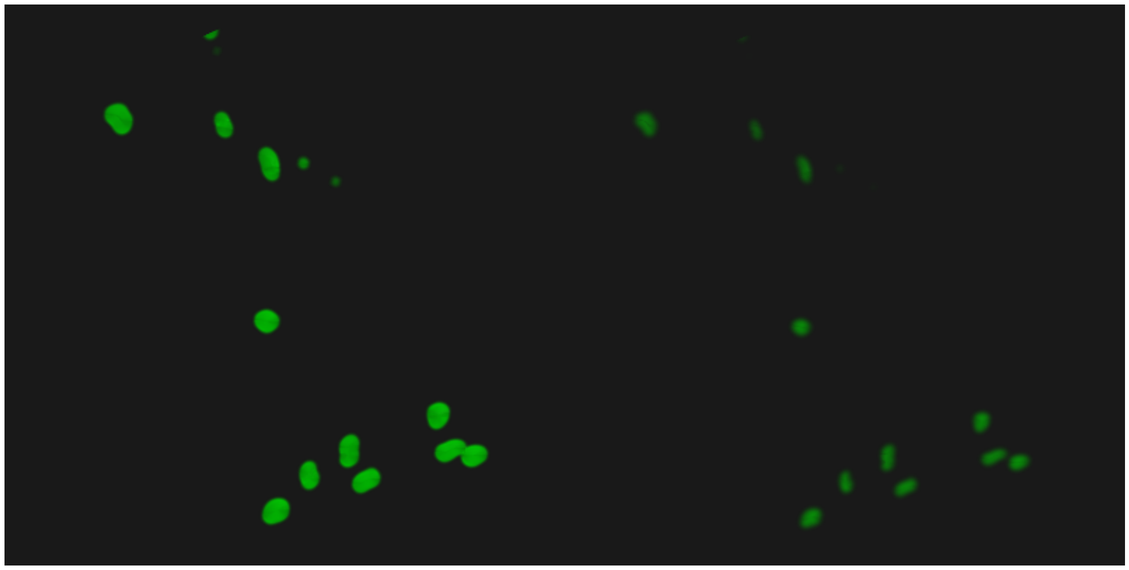

Figure 10: The difference between maximal transparency on the right and minimal transparency on the left

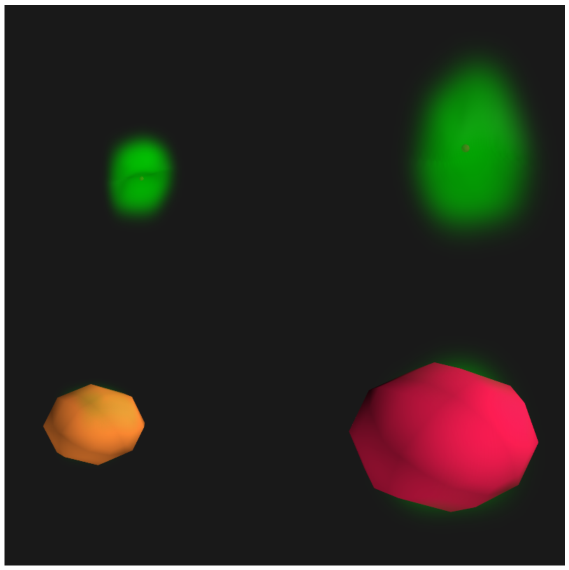

Figure 11: The difference between minimal annotation sphere size on top and maximal size on the bottom

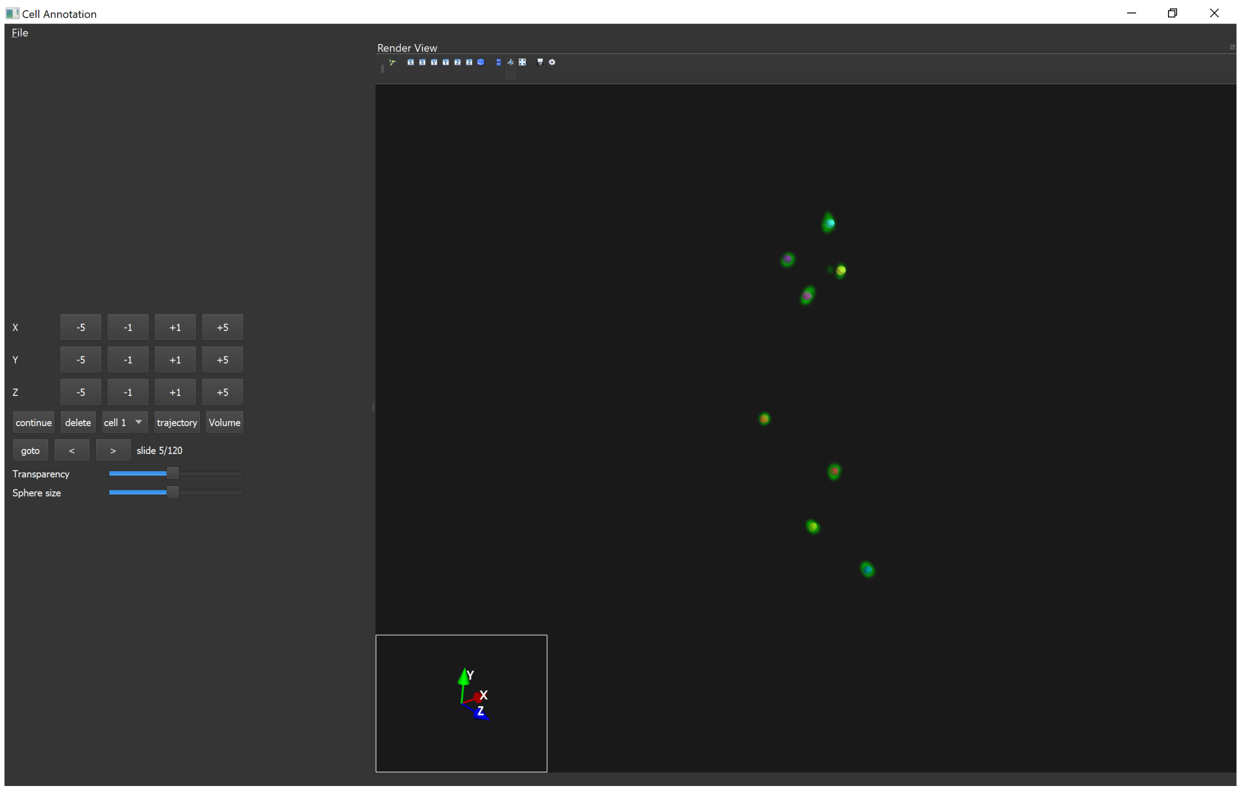

Figure 12: The interface of the created tool

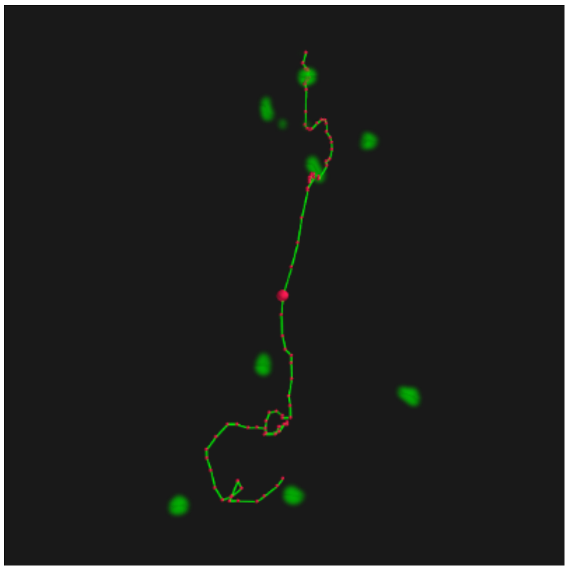

Figure 13: Example of trajectory displayed by the trajectory function

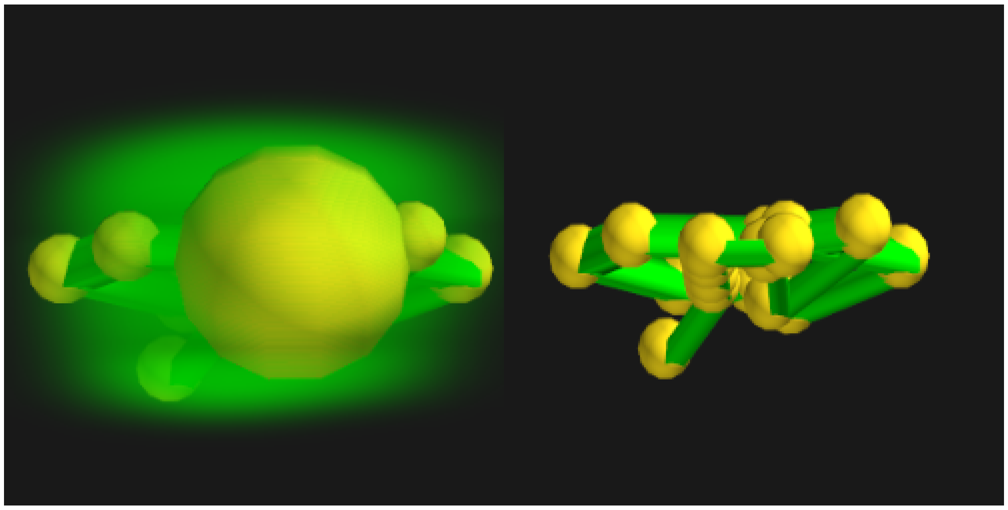

Figure 14: Example of the difference between the regular trajectory and one with the volume toggled

## 3.5 Output

An important note to make is how the coordinate system works. From the point of view of the individual images the origin is in the top left corner. The images used during testing are $512 \times 512$ so the top right corner would be $(512, 0)$ and the bottom left corner would be $(0, 512)$. This is an important detail when using the exported data since the assumption that the origin would be in the bottom left corner could be made. Figure 15 helps illustrate the coordinate system.

The tool created for this thesis currently has 2 kinds of output. Those outputs are intermediate saves and exports. Intermediate saves are meant to allow the user to save and load their data before they are done with all annotations. These saves work by creating an Excel file and putting in the information of the annotation data matrix. Every line of the Excel file has an annotation number, a slide number and the (x,y,x) coordinates of one of the annotated cell locations. Loading data works by reading every line of the Excel file and adding the coordinates to the annotation data matrix using the annotation number and the slide number.

The exports are very similar to the intermediate saves. The main difference is that additional information is added. The distance is added of annotations that have the same number and have neighbouring slides. The distance is $-1$ if there is no annotation right before it, and else it is calculated using the formula $d = \sqrt{(x_1 - x_2)^2 (y_1 - y_2)^2 (z_1 - z_2)^2}$. Another difference is that the user is able to alter the dimensions of the data upon clicking the export button. Normally the program will take the size of the images and the number of layers to determine the dimensions. This might not always correspond to the actual dimensions. The coordinates of the annotations are adjusted to the new dimensions if a change is made. Another difference is that the x and z coordinates are switched. This is because those coordinates are switched during import. Because of that, it needs to be corrected.

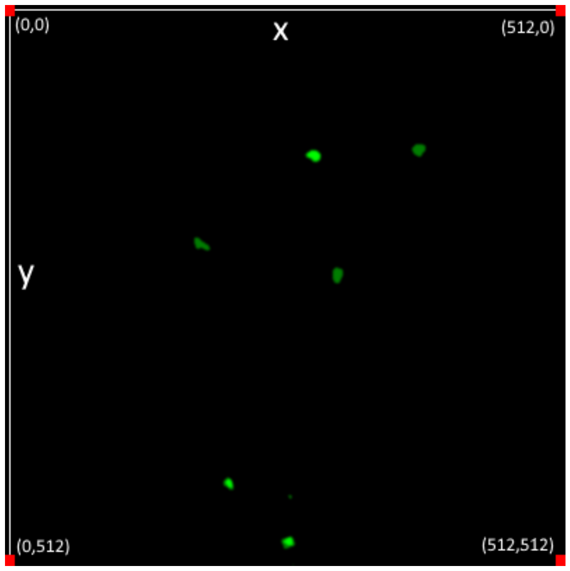

Figure 15: Illustration of the coordinate system used for exported data

## 4 Evaluation

The evaluation was done by looking at the notes made during the making of the tool. These notes contained planned features that aren't in the final product. It also contained comments about features that still needed improvement. To get more insight, a small survey was also done.

### 4.1 Features in need of improvement

One features that could have been done better is creating the annotation by clicking onto the 3D projection. While this function works, it could be better. There are occasions when the location where an annotation is placed might look fine at first sight. But when you turn the camera it can be observed that the sphere doesn't have the right depth. The sphere isn't touching the annotated cell. Figure 16 shows an example of this issue.

Another feature that didn't end up being fixed was the loading icon that would replace the mouse pointer while moving it over the 3D projection. It would return to being a normal pointer once movement stopped but would return to an loading icon once it was being moved again. The issue was looked into due to some of the testers expressing their annoyance and confusion. The cause of this issue hasn't been found yet.

A mistake that was made is the switching of the x and z coordinates during import. This was found out quite late and due to that not much time was left to solve the issue. Fixing the switching of the coordinates during import might cause some problems with functions that modify or use those coordinates. For this reason the problem was fixed by switching the x and z coordinates again during export. Figure 17 helps illustrate this.

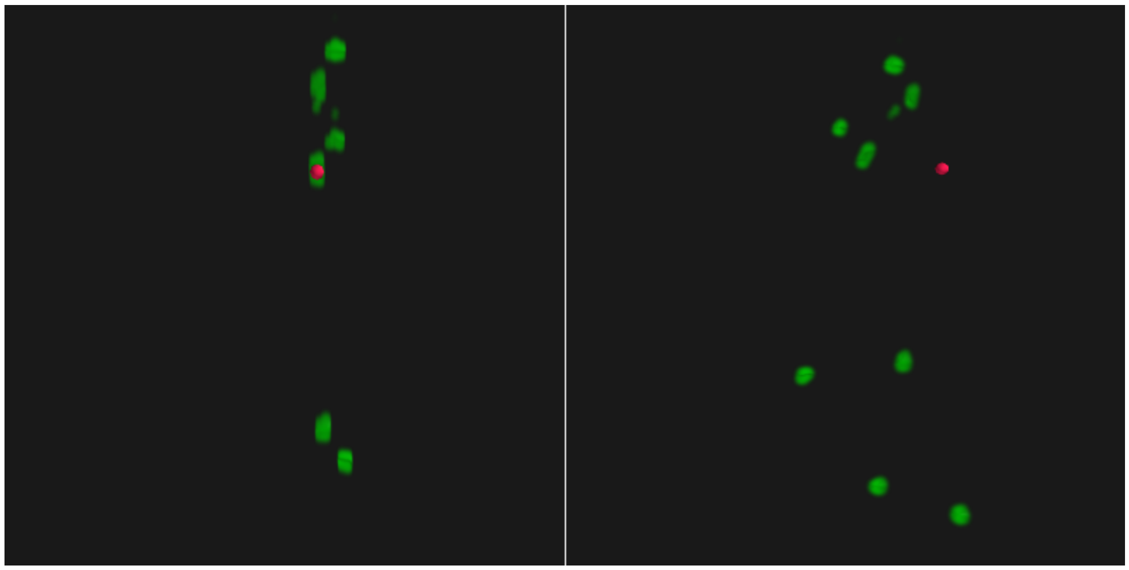

Figure 16: Example of an issue with the clicking on the projection to annotate, the left image shows the original camera position while the right shows an rotated camera position

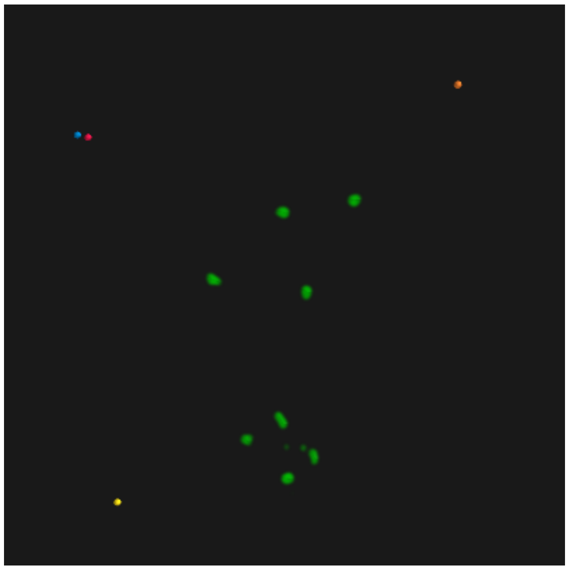

Figure 17: Illustration of result of switching x and z coordinates during import. For regular saves: red is (0,0,0), blue is (29,0,0), yellow is (0,512,0) and orange is (0,0,512). For export saves: red is (0,0,0), blue is (0,0,29), yellow is (0,512,0) and orange is (512,0,0)

| Question | mean score |
|---|---|
| How difficult is it to understand the interface? | 3.4/10 |
| How hard is it to annotate a single cell? | 2.8/10 |

Table 1: Mean score of the numerical survey questions

## 4.2 Missing features

One of the functions that didn't make it to the final version was a function that would show a time lapse of the movement of the cells through the 3D projection. In essence, it would load all projections in order. The current problem with the function is that the loading of a projection takes too much time. A potential solution to make it faster would be to save all the matrices that are created from the multi-layer images instead of discarding them and recreating them once the same projection is needed again. A downside of this approach would be slower startup time for the program due to the large amount of matrices that would need to be created upon startup. This proposed solution wasn't applied due to time constraints.
One last aspect that didn't make it to the final version was the creation of an executable. An executable could be created. But it would give an error when used. It is suspected that this error has to do with the projection created by mayavi. A solution for this error was not found. Due to this, no executable version was created.
The main reason an executable would haven been useful is that it would make it easier to use the tool. To use the current tool an user would need to be able to run Python files on their pc. They would specifically need to use Python 3.10.10. This is due to multiple reasons. The most important is that mayavi and PySide2 are not up to date with the most recent Python versions. It has to be noted that this could change in the future. Another reason is that the tool has not been tested for other versions of Python. The usage of another version could result in problems. Having a working executable would solve these difficulties a user might encounter.

## 4.3 Survey

A small survey of five people has been done. This is a small number of respondents, but it might still be useful to show some criticism users have for the tool. Another important note is that the users are all acquainted with the author. This means they might give a more favorable response than they would give if a stranger asked. The survey started with a short explanation. This told them what their goal was and what the buttons did. The users would make a short series of annotations. Next, they would take a look at a longer annotation trajectory and the buttons involved with its display. After this, the users would fill in a short five question survey. Table 1 shows an overview of the mean score from the numerical survey questions.
The first question asked how difficult the interface was to understand on a scale from one to ten, with ten being difficult. The answers ranged between two and five. The average was a score of 3.4. This shows that user think that the interface was not difficult, but there is still room for improvement.
The second question was an open quesiton, asking for comments regarding the interface, in order to get user insight. Two of the users noted that they didn't think they would be able to understand the interface without the in-person explanation that was given. Both users wished for a written

explanation. Giving a good explanation of the interface in the README could help. Another user comments on the names of buttons. They stated that adding an on/off to the trajectory and volume button would make it easier to understand. Additionally they stated that The "Continue" button could be renamed to "reapply tracker" to make its use more clear. Another user found the move buttons confusing to use. While this is not a comment regarding the interface it is still useful to know. It shows that some users had difficulty with movement in the 3D projection. The final comment that was made stated that the buttons could be more spread out with all the unused space that is present.
The third questions asked the user to rate how hard it was to annotate a single cell on a scale frome one to ten, witht en being very difficult. The given scores ranged between one and four. The average score was 2.8. This seems to show that most users where able to annotate without too much difficulty.
The fourth question asked if users had any suggestions to increase ease of use. Only two responses where given to this question. The first asked for the feature of being able to scroll with the mouse wheel. This would indeed be useful and allow the user to have a better look at the cells. The other user asked for the "continue" button to be automatically activated when going to the next slide. While this is useful, in some cases it might not always be what the users wants to do.
The last question asked if the users had any other remarks that didn't fit the previous questions. Only a single answer was given. They stated that the interface functions but isn't that nice to look at. They stated that this could use work. This is an improvement that could be done but is not high priority compared to other things that can still be done.

## 5 Conclusions and Further Works

The creation of the annotation tool was successful. The tool can load the TIFF files. The tool can use these TIFF files to create a 3D matrix. A 3D projection can be created using using the 3D matrix . The user can annotate in the projection and switch between projections to allow them to annotate the whole trajectory. The tool allows for the display of the annotated trajectories. This allows users to view their progress. This trajectory can be saved and exported as an Excel file.
The tool was created to help with the manual annotation of neutrophil cell migration upon zebrafish larval tail wounding. This information is relevant for a paper that looked into TLR2 and MyD88[HvSL+21]. The paper specifically looked into how TLR2 and MyD88 are involved in modulating neutrophil cell migration behaviour upon zebrafish larval tail wounding. Experiments where done which used normal and mutated variants of these genes. The data from these experiments could be annotated using the tool that was created during this thesis. The cell trajectories created from this could then be used for a supervised learning algorithm. The data would be used as training data for deep learning. This would train the algorithm to preform the cell tracking. The cell tracking algorithm could then be used on all data that was created during the experiments with TLR2 and MyD88. The data created from this would helpful for comparing the mutated and normal versions of TLR2 and MyD88. It could give more insight in how the TLR2 and MyD88 genes are involved in the modulation of neutrophil cell migration.
There are functions that could be done better or could be added to the tool, as stated in section 4 Evaluation. Further works focused on the tool itself could look at doing multiple additions. An useful improvement would make annotation by hand more accurate. This could be done by solving the problem with the depth at which annotations are sometimes placed. Another useful addition would be allowing the user to see a time lapse of the cell movement. This feature didn't make it to the final product. This has to do with how projections are created. Matrices used for projection aren't saved. By saving these matrices, this feature could potentially be implemented in the future. A very useful thing future works could look into is the creation of an executable version of the tool. This would make it a lot easier for users to use the tool. A smaller addition that could be done would be improving the interface. A comment was made about this in the survey. Improving the interface could make it easier for users to use the tool. The involvement of more user feedbacks could help in this endeavor. The survey done in this thesis had only a few participants, which limited the information gained from it. Surveying more participants could help to gather more detailed information about what could be improved in the interface. However the improvements could go further. While an executable version of the tool would make it easier to use the tool, it would still need to be distributed. A version of the tool that would run through a web page would be useful. It would mean that no executable would need to be distributed. It would also make it even easier to use the tool. It would also make it easier to possibly do crowd sourcing to obtain annotations.
The most important future work that could be done is the creation of the training data for deep learning using this tool. This can be done by people familiar with the subject. However, this will limit the people who are able to do it. Thus the speed at which annotations are done would be slower. Alternatively, crowd sourcing can be used to obtain the annotations. This could, however lessen the quality of the annotations.

## References

[AOKA22] Hideya Aragaki, Katsunori Ogoh, Yohei Kondo, and Kazuhiro Aoki. Lim tracker: A software package for cell tracking and analysis with advanced interactivity. *Scientific Reports*, 12(1), 2022.

[BLP13] Jennifer C. Brazil, Nancy A. Louis, and Charles A. Parkos. The role of Polymorphonuclear leukocyte trafficking in the perpetuation of inflammation during inflammatory bowel disease. *Inflammatory Bowel Diseases*, 19(7):1556–1565, 2013.

[CBCC$^+$12] Angela Castoldi, Tárcio Teodoro Braga, Matheus Correa-Costa, Cristhiane Fávero Aguiar, Ênio José Bassi, Reinaldo Correa-Silva, Rosa Maria Elias, Fábia Salvador, Pedro Manoel Moraes-Vieira, Marcos Antônio Cenedeze, and et al. Tlr2, tlr4 and the myd88 signaling pathway are crucial for neutrophil migration in acute kidney injury induced by sepsis. *PLoS ONE*, 7(5), 2012.

[CZCL20] Lingfeng Chen, Lulu Zheng, Pengqin Chen, and Guang Liang. Myeloid differentiation primary response protein 88 (myd88): The central hub of tlr/il-1r signaling. *Journal of Medicinal Chemistry*, 63(22):13316–13329, 2020.

[DTB$^+$10] Mohan R Dasu, Ravi K Thangappan, Alika Bourgette, Luisa A DiPietro, Rivkah Isseroff, and Ishwarlal Jialal. Tlr2 expression and signaling-dependent inflammation impair wound healing in diabetic mice. *Laboratory Investigation*, 90(11):1628–1636, 2010.

[EW89] Franklin H. Epstein and Stephen J. Weiss. Tissue destruction by neutrophils. *New England Journal of Medicine*, 320(6):365–376, 1989.

[HCH$^+$18] David E. Hernandez, Steven W. Chen, Elizabeth E. Hunter, Edward B. Steager, and Vijay Kumar. Cell tracking with deep learning and the viterbi algorithm. *2018 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS)*, 2018.

[HD15] Takashi Hato and Pierre C. Dagher. How the innate immune system senses trouble and causes trouble. *Clinical Journal of the American Society of Nephrology*, 10(8):1459–1469, 2015.

[HLWR13] Katherine M Henry, Catherine A Loynes, Moira K Whyte, and Stephen A Renshaw. Zebrafish as a model for the study of neutrophil biology. *Journal of Leukocyte Biology*, 94(4):633–642, 2013.

[HvSL$^+$21] Wanbin Hu, Leonie van Steijn, Chen Li, Fons J. Verbeek, Lu Cao, Roeland M. 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.

[HYS$^+$19] Wanbin Hu, Shuxin Yang, Yasuhito Shimada, Magnus Münch, Rubén Marín-Juez, Annemarie H. Meijer, and Herman P. Spaink. 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 Genomics*, 20(1), 2019.

[IBM] The International Business Machines Corporation what is deep learning? https://www.ibm.com/topics/deep-learning. Accessed: 2023-06-01.

[KRT+21] Seifedine Kadry, Venkatesan Rajinikanth, David Taniar, Robertas Damaševičius, and Xiomara Patricia Valencia. Automated segmentation of leukocyte from hematological images—a study using various cnn schemes. *The Journal of Supercomputing*, 78(5):6974–6994, 2021.

[LLD20] Jean-Baptiste Lugagne, Haonan Lin, and Mary J. Dunlop. Delta: Automated cell segmentation, tracking, and lineage reconstruction using deep learning. *PLOS Computational Biology*, 16(4), 2020.

[LMK+21] Austin E. Lefebvre, Dennis Ma, Kai Kessenbrock, Devon A. Lawson, and Michelle A. Digman. Automated segmentation and tracking of mitochondria in live-cell time-lapse images. *Nature Methods*, 18(9):1091–1102, 2021.

[LYH+22] Chen Li, Wilson W.C. Yiu, Wanbin Hu, Lu Cao, and Fons J. Verbeek. A feature weighted tracking method for 3d neutrophils in time-lapse microscopy. *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2022.

[MCSL+20] Claire Mitchell, Lauryanne Caroff, Jose Alonso Solis-Lemus, Constantino Carlos Reyes-Aldasoro, Alessandra Vigilante, Fiona Warburton, Fabrice de Chaumont, Alexandre Dufour, Stephane Dallongeville, Jean-Christophe Olivo-Marin, and et al. Cell tracking profiler: A user-driven analysis framework for evaluating 4d live cell imaging data. *Journal of Cell Science*, 2020.

[MGL+16] Kamalika Mojumdar, Christian Giordano, Christian Lemaire, Feng Liang, Maziar Divangahi, Salman T Qureshi, and Basil J Petrof. Divergent impact of toll-like receptor 2 deficiency on repair mechanisms in healthy muscle versus duchenne muscular dystrophy. *The Journal of Pathology*, 239(1):10–22, 2016.

[MMW+14] Anna Moles, Lindsay Murphy, Caroline L. Wilson, Jayashree Bagchi Chakraborty, Christopher Fox, Eek Joong Park, Jelena Mann, Fiona Oakley, Rachel Howarth, John Brain, and et al. A tlr2/s100a9/cxcl-2 signaling network is necessary for neutrophil recruitment in acute and chronic liver injury in the mouse. *Journal of Hepatology*, 60(4):782–791, 2014.

[MYvR+13] Kouichi Miura, Ling Yang, Nico van Rooijen, David A. Brenner, Hirohide Ohnishi, and Ekihiro Seki. Toll-like receptor 2 and palmitic acid cooperatively contribute to the development of nonalcoholic steatohepatitis through inflammasome activation in mice. *Hepatology*, 57(2):577–589, 2013.

[nct] National Cancer Institute nci dictionary of cancer terms. https://www.cancer.gov/publications/dictionaries/cancer-terms/def/neutrophil. Accessed: 2023-04-28.

[ONMW12] Laura Oliveira-Nascimento, Paola Massari, and Lee M. Wetzler. The role of tlr2 in infection and immunity. *Frontiers in Immunology*, 3, Apr 2012.

[PCGM22] Manuel Petit, Guillaume Cerutti, Christophe Godin, and Gregoire Malandain. Robust plant cell tracking in fluorescence microscopy 3d+t series. *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, 2022.

[QNT+04] Valerie J. Quesniaux, Delphine M. Nicolle, David Torres, Laurent Kremer, Yann Guerardel, Jerome Nigou, Germain Puzo, Francois Erard, and Bernhard Ryffel. Toll-like receptor 2 (tlr2)-dependent-positive and tlr2-independent-negative regulation of proinflammatory cytokines by mycobacterial lipomannans. *The Journal of Immunology*, 172(7):4425–4434, 2004.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[RMC21] Marzieh R. Moghadam and Yi-Ping Phoebe Chen. Tracking leukocytes in intravital time lapse images using 3d cell association learning network. *Artificial Intelligence in Medicine*, 118:102129, 2021.

[SBB+07] Charles N. Serhan, Sue D. Brain, Christopher D. Buckley, Derek W. Gilroy, Christopher Haslett, Luke A. O'Neill, Mauro Perretti, Adriano G. Rossi, and John L. Wallace. Resolution of in flammation: State of the art, definitions and terms. *The FASEB Journal*, 21(2):325–332, 2007.

[SDC+07] Jürgen Schauber, Robert A. Dorschner, Alvin B. Coda, Amanda S. Büchau, Philip T. Liu, David Kiken, Yolanda R. Helfrich, Sewon Kang, Hashem Z. Elalieh, Andreas Steinmeyer, and et al. Injury enhances tlr2 function and antimicrobial peptide expression through a vitamin d–dependent mechanism. *Journal of Clinical Investigation*, 117(3):803–811, 2007.

[SJM21] S. Shailja, Jiaxiang Jiang, and B.S. Manjunath. Semi supervised segmentation and graph-based tracking of 3d nuclei in time-lapse microscopy. *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, 2021.

[SPF11] Ekihiro Seki, EekJoong Park, and Jiro Fujimoto. Toll-like receptor signaling in liver regeneration, fibrosis and carcinogenesis. *Hepatology Research*, 41(7):597–610, 2011.

[STK+19] Frida Sommer, Vincenzo Torraca, Sarah M. Kamel, Amber Lombardi, and Annemarie H. Meijer. Frontline science: Antagonism between regular and atypical cxcr3 receptors regulates macrophage migration during infection and injury in zebrafish. *Journal of Leukocyte Biology*, 107(2):185–203, 2019.

[TA04] Kiyoshi Takeda and Shizuo Akira. Microbial recognition by toll-like receptors. *Journal of Dermatological Science*, 34(2):73–82, 2004.

[TGS$^+$19] Hsieh-Fu Tsai, Joanna Gajda, Tyler F.W. Sloan, Andrei Rares, and Amy Q. Shen. Usiigaci: Instance-aware cell tracking in stain-free phase contrast microscopy enabled by machine learning. *SoftwareX*, 9:230–237, 2019.

[vdVvSSM13] Michiel van der Vaart, Joost J. van Soest, Herman P. Spaink, and Annemarie H. Meijer. Functional analysis of a zebrafishmyd88mutant identifies key transcriptional components of the innate immune system. *Disease Models amp; Mechanisms*, 2013.

[Vij18] Kumar Vijay. Toll-like receptors in immunity and inflammatory diseases: Past, present, and future. *International Immunopharmacology*, 59:391–412, 2018.

[WMV$^+$21] Chentao Wen, Takuya Miura, Venkatakaushik Voleti, Kazushi Yamaguchi, Motosuke Tsutsumi, Kei Yamamoto, Kohei Otomo, Yukako Fujie, Takayuki Teramoto, Takeshi Ishihara, and et al. 3deecelltracker, a deep learning-based pipeline for segmenting and tracking cells in 3d time lapse images. *eLife*, 10, 2021.

[WRP$^+$20] Natalie Wagner, Sabrina Reinehr, Marina Palmhof, David Schuschel, Teresa Tsai, Emely Sommer, Viktoria Frank, Gesa Stute, H. Burkhard Dick, and Stephanie C. Joachim. Microglia activation in retinal ischemia triggers cytokine and toll-like receptor response. *Journal of Molecular Neuroscience*, 71(3):527–544, 2020.

[XTO$^+$19] Yufei Xie, Sofie Tolmeijer, Jelle M. Oskam, Tijs Tonkens, Annemarie H. Meijer, and Marcel J. Schaaf. Glucocorticoids inhibit macrophage differentiation towards a pro-inflammatory phenotype upon wounding without affecting their migration. *Disease Models amp; Mechanisms*, 2019.

[XZL$^+$13] Yifei Xu, Ying Zhou, Haiyan Lin, Haiyang Hu, Yuxing Wang, and Geng Xu. Toll-like receptor 2 in promoting angiogenesis after acute ischemic injury. *International Journal of Molecular Medicine*, 31(3):555–560, 2013.

[YWC10] Li Yu, Liantang Wang, and Shangwu Chen. Endogenous toll-like receptor ligands and their biological significance. *Journal of Cellular and Molecular Medicine*, 14(11):2592–2603, 2010.