

Master Computer Science

Cell Trajectory studies

Exploring new cell trajectory tracking algorithms with neutrophil cells

Name: Wilson W.C. Yiu Student ID: s3120163 Date: 29/08/2022 Specialisation: Bioinformatics

1st supervisor: Dr. Lu Cao 2nd supervisor: Prof. Fons Verbeek

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS) Leiden University Niels Bohrweg 1 2333 CA Leiden The Netherlands

Abstract

Neutrophil migration has a remarkable effect in many research areas such as cancer cell studies [1], tumor invasion [5], response of wounding [10], etc. Despite numerous researches have been done on neutrophil cells in the past, a large part of it still remains unclear due to its complex nature. Cell trajectory tracking has been one of the common ways to understand neutrophil migration characteristics. Further understanding of cell movement behavior could potentially provide new insights to biologists on their research work. Thus, this research explores new tracking models and evaluates them to the existing ones based on the evaluation methods proposed by Smith Kevin, Gatica-Perez Daniel[9]. Four algorithms namely Viterbi, Viterbi with Multi-Layer Threshold (Viterbi-MLT), Viterbi with Single Layer Threshold (Viterbi-SLT) and Feature weighted routing are developed. Viterbi-MLT an enhanced Viterbi algorithm that is designed to put nearby cell's trajectory into account during route decision outperformed other algorithms in three out of four evaluation methods and achieved a comparable result on the fourth measurement. On the other hand, it is observed that the top performing models generally do better in either the first three or the last evaluation measurement. Thus, there is no absolutely best algorithm under current evaluation scheme but the individual algorithm has its own advantage. However, the Viterbi-MLT model is conceivably the overall best algorithm despite its slight underperformance in one of the indices.

Contents

1	Background	4				
2	Dataset 2.1 Neutrophil cell images 2.2 Cell movement prediction and the Score matrix	5 5 6				
3	Methodology 3.1 Viterbi 3.1.1 Routing phase 3.1.2 3.1.2 Retrieving the best trajectory 1 3.1.3 Problem in Viterbi routing 1 3.1.4 Viterbi with Multi-layer Threshold 1 3.1.5 Threshold adjustment 1 3.1.6 Alternative path routing 1 3.1.7 Reroute 1 3.1.8 Termination of a cell trajectory 1 3.1.9 New cells in latter frame 1 3.1.10 Additional occupation rules 1 3.2 Viterbi-MLT workflow diagram 1 3.3 Viterbi with Single Layer Threshold 1 3.4 Feature weighted prediction 1 3.4.1 Direction variation 1 3.4.2 Cell distance 1 3.4.3 Average cell movement distance 1 3.4.4 Feature score, weight and final score 2 3.5.1 Avoiding program deadlock 2 3.5.1 Avoiding program deadlock 2 3.5.2 Minimizing reroute events 2<	889900112334567799001223333455				
4	Result 2 4.1 Best hyperparameter value 2 4.2 Performance statistic 2 Extra section: Cell trajectories Classification 2 5.1 Background 2 5.2 Data 2 5.3 Methodology 2 5.4 Result 3	5 6 8 8 8 9 1				
6	Discussion and Conclusion 32					
7	Future work 3 7.0.1 Tracking with other models 3 7.0.2 Routing algorithm variations 3 7.0.3 State space exploration 3 7.0.4 Cell synthesis 3	3 33 33 34				

1 Background

Neutrophil research plays a crucial role in immune system studies. It serves as one of the first immune cells responding to microorganism invasion. It is the most abundant leukocyte in white blood cells in humans[17] and helps the body to fight infection. It performs its role by ingesting microorganisms[18] which has to be in close proximity. For this reason, their movement pattern is an important research area in the domain of immunology.

Neutrophil migration is involved in a wide range of research studies such as influence on cancer progression and metastasis, embryonic development, cell differentiation, immune response, regeneration, tumor invasion, response of wounding, etc[10, 11, 12, 13, 14, 15, 19]. It is a dynamic and complex process as seen from the related research paper and observed in the cell images data. In the past, there have been different studies done on this topic. Various sophisticated algorithms were built to predict cell trajectory. For instance, Delta in 2020[6], Viterbi with swap[8] and Hungarian[7] in 2015. Those research provided precious information and intuition in cell trajectory studies.

Neutrophil migration studies bring numerous benefits. It helps to improve the accuracy of trajectory prediction and to unravel more cell movement characteristics. This leads to an improved confidence of research outcome that relies on this data, or even makes certain kinds of essential research data available. For instance, when a reliable set of predicted trajectory data on a specimen dataset is available, we may be able to train a machine learning classifier with a specimen data set that is labeled with mutation type. After then, it can be applied to an unlabeled dataset to form a larger dataset.

There is conceivably room for exploration in the cell trajectory studies domain as technology of imaging advanced rapidly and the past research focused more on segmentation. Biological and medicinal microscopy technology medicine has advanced dramatically in the past decade. Those microscopes are capable of capturing better brightfield and fluorescence images which reveal the dynamic process of living cells[4]. Those images contain more details and may provide new insights about cell movements. Nevertheless, the cell tracking process is composed of at least two phases — segmentation and routing. In the past, more research effort was spent on segmentation(1.8m vs 1.4m for cell segmentation and cell trajectory respectively from Google Scholar search). However, the importance of cell routing knowledge can not be understated because it substantially affects cell migration judgment and could have a critical influence on the research result. Furthermore, cell tracking involves various challenging processes such as costing huge human resources in creating ground truth data, having lengthy ground truth data, having overlapped cell trajectories in 2D images, etc. We noticed that the overlapping problem has been a great challenge. Thus, it is one of the major focuses of this research. One of our goals is to see if new algorithms could handle overlapping cell trajectories better. The overlapping scenario is difficult in several aspects: the cells look like a single cell when they overlap in a 2D image, see figure 15 for an illustration; The overlapping event can not be predicted easily as it is affected by the surrounding cells. Those cells could lead to various possible routes options that increase the difficulty for an overlap event to be predicted correctly. More details are provided in the methodology section.

This research aims to examine how well could new algorithms with sophisticated rules handle the

dynamic cell movement process. Those algorithms have to be capable of handling overlapping trajectories scenarios. In addition, it shall be highly adaptive and reusable in the long term. On the other hand, we are also interested in exploring the feasibility of creating a machine learning classification task on zebrafish specimens with the existing data.

Several algorithms are built in this research. All algorithms are implemented with a recursive-like rerouting feature to tackle the overlapping trajectories observed in the images. Models with different routing focuses are built: the Viterbi group focuses on overall best trajectory; the feature weight model focuses on the influence from previous frames when predicting cell trajectory in a new frame. Moreover, adjustable hyperparameters are created for tuning and adapting to new datasets. A state space exploration with grid search is performed referring to the mentioned evaluation method. The best result of each algorithm is recorded and analyzed together with other existing algorithms. Finally, we developed a supervised learning classifier with the labeled zebrafish dataset and examined it with a different train and test ratio.

2 Dataset

2.1 Neutrophil cell images

The neutrophil images are the primary data source of this research. Those images are captured from Zebrafish specimens. There are several advantages of using zebrafish specimens for experiment: it is relatively cheaper to maintain than other specimens such as rodents; the cells are transparent during the early stage; and it has a short life cycle. However, it cannot replace certain experiments such as drug development[2]. The experiments done in this project emphasize on cell movement. The tail is being selected as the analyzing region. The zebrafish's tail is being trimmed and observed under a confocal laser scanning microscope(CLSM) as shown in figure 1[2]. A series of images are captured over a continuous period of time to obtain a video series that consist 120 to 130 frames.



Figure 1: Zebrafish cell image capturing process[10]. A specimen of the zebrafish tail is placed under the CLSM. A green light source is projected to the tail to obtain a microscopy image as shown in the right.

The experimental dataset contains twenty series of images from different specimens. Sixteen of the series consist of 120 timelapse images and the other four with 130. Figure 2 presents

one of the original images on the left and segmented images on the right. The cells in the original image express green fluorescent protein (GFP) obtained from CLSM. The segmented image is the result produced by the trained U-Net model[3]. The dimension of the images is 512 by 512. It is observed that the longest movement distance of a cell between each frame is approximately 53 m (or 35 pixels) from ground truth data. The cell distributions could vary substantially over a certain period of time (frames). Figure 3 reveals a segmented video sample of frame 1 and 60.



(a) Original image

(b) Segmented image from U-Net mode

Figure 2: Image source of Series 1 frame 16

The cell distributions could vary substantially over a certain period of time (frames). Figure 3 reveals a segmented video sample of frame 1 and 60.



(a) Frame 1

(b) Frame 60

Figure 3: Image source from Series 11. Several new cells entered the frame from the bottom edge of the inspection region.

2.2 Cell movement prediction and the Score matrix

In order to make cell trajectory prediction feasible, a linkage reference is required among all cells over each frame. We used the linkage data generated by the U-Net tracking model since

the focus of this paper is on cell track routing. This linkage score is derived using the original neutrophil image and segmented image in frame t and t-1. A cell 'seed' in frame t-1 will be provided for prediction, the U-Net model will then produce a predicted segmented parent and daughter cell image. Figure 4 is a showcase of the score derivation process from a U-Net Tracking model.

This image is then compared with all the cells in the existing frame t to generate a probability score between 0 to 1 for each cell. Since each connection is represented with a score, a frame with 30 cells followed with a frame with 35 cells would result in 1050 connections. To evaluate this in an organized and understandable way, a score matrix is created to store the information.



Figure 4: U-Net Tracking model: frame t and t-1 from original image, segmented frame t, and isolate 'seed' cell image to be predicted are served as an input of the U-Net model[3]. The U-Net model then produces the prediction result of the parent and daughter cell, which are labeled as (5) and (6) on the right.

The score matrix contains all cells' connections' score (or probability). It has a substantial effect on routing decisions, particularly the Viterbi group as it is the sole reference of cell linkage relationships. In contrast, it is less determining in the feature weighted model as described in the methodology section 3.4 because it only serves as one of the four features that influence the routing decision. Figure 5 demonstrates an example of the score matrix.

There are more cells in each image in the dataset, ranging from 6 to more than 30, see figure 2b and 3a. The number of cells in the current and next frame determines the number of rows and columns of the matrix respectively. A time step occurs in each incremental frame and will produce one score matrix. Thus, the series with 120 frames will produce 119 matrices. These matrices provide essential information for determining cell routes which is illustrated in the next section.



Figure 5: A score matrix that represents the connection probability among all the cells between two consecutive frames. Each row and column represents a cell in the current frame(t) and next frame(t + 1) respectively. The intersect point of a row and column represents the connection relationship and score of two particular cells.

3 Methodology

Four cell track routing methods are developed namely Viterbi, Viterbi with Multi-layer Threshold (Viterbi-MLT), Viterbi with Single layer Threshold (Viterbi-SLT) and Feature weighted algorithm. All algorithms are developed with various hyperparameters variables settings which are tunable for optimization. Numerous executions were done in each model with various hyperparameters combinations. The best result discovered in each model is compared together with three other existing cell trajectory tracking models namely KDE[5], DeLTA[6] and Hungarian[7].

3.1 Viterbi

Viterbi is originally used at global track linkage, MuSCs and myoblasts cell tracking problems[8] which achieved a superior performance. It searches for the overall best connection score by solving a dynamic programming problem over a state space diagram. This allows it to identify the optimal trajectory of each cell track. A simple Viterbi example is shown on figure 6.



Figure 6: A Viterbi example of finding the best route for Cell 1. The dotted line and solid line represents all the possible connections of the network. There are various paths for reaching a particular node, but only the best connection score is shown in each node.

	Frame 3	Frame 4
Unnen nede	$0.95 \ge 0.95 = 0.903$	$0.903 \ge 0.7 = 0.632$
Opper node	$0.1 \ge 0.85 = 0.085$	$0.095 \ge 0.05 = 0.062$
Lower node	$0.95 \ge 0.1 = 0.095$	$0.903 \ge 0.5 = 0.452$
Lower node	$0.1 \ge 0.8 = 0.080$	$0.095 \ge 0.4 = 0.038$

Table 1: All possible route scores of each node in frame 3 and 4. The bolded text is being chosen for connection.

There are two major phases in the Viterbi algorithm named as routing and best track retrieval phase.

3.1.1 Routing phase

During the routing phase, the best connection of each layer is calculated from earlier frame to later frame. When connecting to a new frame, all connections from layer t to t - 1 of a node is calculated (table 1). Only one connection is selected from each of the nodes in the later frame to the previous frame. And this connection is the highest total multiplied score of all paths to that layer. An example is illustrated in figure 6, the selected connections are shown with solid lines, the calculation details are shown in table 1.

3.1.2 Retrieving the best trajectory

The process enters the best track retrieval phase when the routing process reaches the last frame (frame 4 in this case). In this phase, the node with the highest total score will be decided as the end node of the best track. The entire trajectory is then retrieved by following the single best connection established in the routing phase.

For instance, the upper node in frame 4 with value 0.632(figure 6) is selected as the end point of the trajectory. The trajectory of Cell 1 is retrieved by following the solid lines. Viterbi assures the best overall score track is being selected.

3.1.3 Problem in Viterbi routing

The Viterbi algorithm is effective in finding out the overall best path. However, it has a problematic pattern that is demonstrated in figure 7. When the next cell (orange cell) is being routed, one of the routes shares the same path with the red cell track and never splits because they use the same scores after the merge happened.



Figure 7: The orange route is being routed. The predicted track shares the same track with the red cell since frame 2 and is never splitted.

This circumstance occurs as Viterbi treats each track to be independent from one another (one cell's trajectory does not affect another cell), resulting in the shared tracks never split. In contrast to reality, the calculation of an individual cell trajectory is largely affected by the surrounding tracks. Thus, an alternation is made to the original Viterbi algorithm, namely Viterbi with Multi-layer Threshold.

3.1.4 Viterbi with Multi-layer Threshold

An extended Viterbi algorithm is developed for cell routing — an enhanced version of Viterbi. Viterbi-MLT has three extra features on top of the original Viterbi algorithm for managing the merge and split scenarios among cell tracks. The features are:

- A threshold adjustment feature which provides a reference for merge/split decisions.

- An alternative path routing feature which allows the algorithm to route to a different node in certain conditions.
- A reroute feature to reroute existing trajectories that are later found to be occupying a node that belongs to another cell track.

3.1.5 Threshold adjustment

A threshold is created to support the alternative path selection feature. It provides a reference for merge/ split decisions. The initial value of the threshold is an adjustable hyperparameter. This threshold value has to be discounted for every additional frame in Viterbi-MLT because the connection score has a decreasing trend from probability multiplication when routing to each new frame. The threshold discount rate is calculated with equation 1.

$$Adjusted threshold = threshold \ ^{(framenumber-1)} \tag{1}$$

The threshold value for each frame is indicated at the bottom region of each figure in the coming sections.

3.1.6 Alternative path routing

Alternative path routing feature enables the track routing to split and merge with more specific conditions. In the routing phase, the best node to frame t - 1 could possibly be occupied by one or more cell tracks. In such circumstance, connection to that node would only be allowed if:

- (1) The total score of the routing track is above threshold up to that layer, OR;
- (2) All occupied cell tracks' scores are below threshold up to that layer.

This rule is also applied to the last layer in the best track retrieval phase. Figure 8 exhibits the concept mentioned.



Figure 8: Alternative path selection example for t - 1 frames and the last frame.

When considering node connections from frame 3 to 2, the upper node in frame 2 is originally the best node to be connected for orange cell track, but it is occupied by the red cell track with a score higher than threshold. On the other hand, the routing track score of orange is 0.74 which is lower than threshold. Thus, connection to that node would be ruled out and a connection to frame 2 lower node (second best connection) is established, see the solid line from frame 3 to 2 in figure 8. In the best track retrieval phase which starts from frame 4 in figure 8, although the upper node has a higher overall score (0.412), it does not meet any of the conditions mentioned in the routing section. Thus, the lower node is chosen as the end node of the track (score 0.298) and a split event in frame 3 is predicted for the two cell tracks.

3.1.7 Reroute

The reroute feature allows an earlier routed cell track to be reset when an occupied node is found to be a better match for later routed track. To exemplify, let's consider the yellow cell is being routed after the orange cell, and the best track of the yellow cell is determined as shown in figure 8. It shares the lower node with the orange cell track in frame 2 at the beginning.



Figure 9: The orange route is being rerouted due to a collision with the yellow track. The new route connects to a different node in frame 2.

Nevertheless, only the yellow track score is above threshold. Therefore, the orange cell track is sent to reroute as it is occupying a node that is more suitable for the yellow track. While rerouting orange track, it is connected to node 3 in frame 2 since the first two nodes are unsuitable. The two tracks are sharing the same node in the next frame on the grounds that both of them are above threshold. In the last frame, they share the same node again because both of their scores are below threshold.

3.1.8 Termination of a cell trajectory

The cell trajectory terminates in three conditions. The first one is when it reaches the last frame of the video. The second condition is when the score in the next frame is all zero. The third condition is when the next frame does not have an available cell due to nodes owned by other trajectories and the merge criteria is not fulfilled.

3.1.9 New cells in latter frame

New cells could appear in a later frame by entering the CLSM watch over region in the middle of the observation period, or moving in from an outside layer. Thus, a cell that does not belong to any existing tracks is treated as a new cell. An example is depicted in figure 10. A new cell is being detected at frame 3, it will be routed largely similar with the condition mentioned except with a start frame discount.



Figure 10: A new cell (green) is detected in frame 3. It goes through a start frame discount for its connection, then takes over the lower node from yellow in frame 4.

A start frame discount is applied during score calculation because cells starting in later frames have fewer multiplications than track begins at an earlier frame.

3.1.10 Additional occupation rules

In the process of rerouting, a track that starts in a later frame may occupy the rerouting path of a track beginning from an earlier frame as shown in figure 11. In such a case, the track that starts earlier would take over the later one even if it has a low connection score, the later track will be removed from track results. It is because the cell in the later trajectory no longer satisfies the definition of a new cell as it is owned by an existing trajectory. This situation only occurs in the reroute phase as the cells from earlier frames are first routed.



Figure 11: Cell 5 established a connection to the beginning of Cell 6 during reroute. Cell 6 is then being removed

3.2 Viterbi-MLT workflow diagram



Figure 12: High level program flow of Viterbi-MLT

3.3 Viterbi with Single Layer Threshold

Viterbi with Single Layer Threshold (Viterbi-SLT) works similarly to the Viterbi-MLT approach but with one major difference. In the routing phase, Viterbi-SLT only compares the connection score to the previous frame instead of the total multiplied score when resolving the best route backward (the solid lines). In the best track retrieval phase, the multiplied total score is still required for best track selection. Figure 13 illustrates the difference track result from both approaches.



Figure 13: The yellow trajectory is the final track from single layer scoring. Upper node is being selected as the best node in the best track retrieval stage by considering the total score. The blue trajectory is the final track from typical Viterbi algorithm

	Frame 3	Frame 4
Unner rede	$0.9 \ge 0.80 = 0.72$	$0.63 \ge 0.60 = 0.378$
Opper node	$0.4 \ge 0.85 = 0.34$	$0.30 \ge 0.65 = 0.195$
Lower node	$0.9 \ge 0.70 = 0.63$	$0.63 \ge 0.50 = 0.315$
Lower node	$0.4 \ge 0.75 = 0.30$	$0.30 \ge 0.55 = 0.165$

Table 2: All route scores of each node in frame 3 and 4. Bolded scores are being connected.

It can be seen that the lower node is being connected in frame 2 and 3 although the total score is lower than the upper node, see table 2. During the best track retrieval stage, the first node

in frame 4 is determined as the best track node as it scored 0.195 which is higher than the second one.

The threshold, alternative path selection and reroute features mentioned in Viterbi-MLT are all applicable Viterbi-SLT except that only the immediate layer score is used for determining cell occupation in collision events. Consequently, the threshold and start frame discounts are irrelevant in the routing phase as there is no multiplication and score decrement. However, both of the discount values are necessary reference in the best track routing phase owing to the fact that the multiplied score is used for best track selection.

The two approaches could end up in very different cell tracking results. In figure 13, the blue line represents the final result of the multi-layer algorithm which connects to the upper node in every frame, whereas the yellow line that stands for the single layer method anticipated a pathway going through the lower node in the middle frames.

3.4 Feature weighted prediction

Feature weighted algorithm uses several features and an interpretation of cell route options to predict cell trajectory. It takes into account the previous frame track data. For example, the direction variation and average cell distance feature can refer to data up to five or more previous frames. In addition, it looks into the route options available by itself and the surrounding tracks when making decisions for merge and split. Four features are built for linkage scoring namely score matrix, direction variation, cell distance and average cell movement distance. The score matrix is identical to the one used in the Viterbi group. The others are further elaborated in the upcoming section.

3.4.1 Direction variation

The direction of cell movement is an important factor that affects its next destination. Figure 14 is a ground truth sample which provides a clue that the cell movement tends to stick to a similar direction in a split event.



Figure 14: Two cells that are moving from different directions, merged and then split according to their trend.

The directional variation is more reliable when several prior frames are referenced. One of the reasons is that the cell may merge with other tracks up to four frames approximately before

splitting. Hence, this can produce a sufficient direction variation between the two tracks. The number of the former frames to be included is a hyperparameter which can be tuned. To calculate the direction variation, we first have to calculate the directional vector. Figure 15 displays how the directional vector is calculated.



Figure 15: Directional vector calculation from the last 5 steps of a routing cell track.

The start location of the cell from a specific step will be set to (0,0). The relative movement for each frame after is calculated and added. The end point of the last vector will form a direction vector from the start point. In this case, the relative coordinate of last five steps are $(1, 2) \Rightarrow (1, 1) \Rightarrow (2, 1) \Rightarrow (1, 0) \Rightarrow (2, -1)$. The result of the directional vector is (7, 3) computed by adding all the five relative vectors.

After then, the degree between the directional vector and each of the nodes in the next frame is calculated as shown in figure 16.



Figure 16: The degree between the directional vector and vector of other candidate cells are calculated

Nodes that has a smaller degree to the directional vector will receive a higher score that is calculated with equation 2:

Directional variation score =
$$\frac{\cos(degree) + 1}{2}$$
 (2)

The score will range between 0 (180°) to 1(0°).

3.4.2 Cell distance

Cell distance score is calculated with equation 3 using the pixel distance between the source and the candidate cell's center point coordinate. The distance is calculated based on Euclidean distance Maximum distance is a hyperparameter and is set to 35 based on observation from ground truth data.

$$Distance \ score = \begin{cases} \frac{\text{Distance}(n, n+1)}{\text{Given max distance}} & \text{for Distance } (n, n+1) <= \text{Given max distance} \\ 0 & \text{for Distance } (n, n+1) > \text{Given max distance} \end{cases}$$
(3)

3.4.3 Average cell movement distance

Average cell movement distance is also an important factor that affects track routing. Cells that are inactive tend to linger around the proximate region, while cells that are more active tend to travel at a similar distance over each frame. Figure 17 is a cell movement track extracted from a ground truth example which illustrates this scenario.



Figure 17: The degree between the directional vector and vector of other candidate cells are calculated

The cell track labeled with yellow arrows has a distant moving distance, whereas the cell track circled with light blue lingered in the nearby region. It can be seen that both cells maintained a similar movement distance over the 4 frames. The average movement score is derived with equations 4 and 5.

$$Average \ distance = \frac{\sum_{n=1}^{\text{total steps}} \text{Distance}(n, n-1)}{\text{Total steps - 1}}$$
(4)

$$Distance \ score = \begin{cases} \frac{\text{Average distance}(n, n+1)}{\text{Given max distance}} & \text{for Distance } (n, n+1) <= \text{Given max distance} \\ 0 & \text{for Distance } (n, n+1) > \text{Given max distance} \end{cases}$$
(5)

The score is normalized to unity with a given max distance.

3.4.4 Feature score, weight and final score

All feature scores are designed to generate a value between 0 to 1 for easy comparison. The influence of each feature can be adjusted through a 4 by 1 weight matrix. The weight matrix consists of 4 positive rational numbers, [0.3, 0..4, 0.2, 0.1] for example. It represents the weight of each feature and the final score is calculated by equation 6:

Final score =
$$\sum_{n=1}^{4}$$
 Feature score(n) * Feature weight(n) (6)

The sum of the weights should be one so it will produce a unity final score.

3.4.5 Track options analysis

Track options analysis is inspired by observing the score matrix. From the observation it is found that cells tend to merge regardless of the score if they have no other option or they would split otherwise. Thus, a strategy is created to review all the cell tracks' options to estimate a merge/split event at a collision node.

Figure 18 revealed an example of three major circumstances. In figure 18a, the cells are allowed to share node 0 in frame t + 1 regardless of their connection score because all of them have no other choice.

However, in figure 18b, every cell has more than one route option. In such a case, each node in frame t + 1 can only have one connection to the previous frame, and the connection establishment starts from the highest to lowest score. If all the nodes are occupied in the latter frame, the remaining unconnected tracks end at that point. To give an example, a connection is established from frame t node 0 to frame t + 1 node 0 in figure 18b as it scored 0.9 which



Figure 18: Three major examples involved in merge and splits in a three cells to two cells routing circumstance. The bolded score represents the effective connection in the routing state.

is the highest. Node 0 in frame t + 1 then becomes unavailable to other tracks, the next unfinished track with the highest score will be routed to the next best available path. The node with value 0.5 that belongs to cell track 1 is the next highest value among the available paths. Thus, it will connect to the upper node in frame t + 1. At this point, frame t + 1 has no other available node, cell track 2 will be terminated at frame t. In figure 18c, cell track 0 in frame t has no other option and the other two tracks have more than one option. Thus, cell 1 will own node 0 and the other cells have to find the second best connection available. Cell 2 then won the connection to the upper node with a slightly higher score than Cell 0. Since all nodes are exhausted in frame + 1, cell 2 will not be further connected to any node.

The reroute feature is applied in this model as the result of the earlier cell tracks can not be finalized until all other tracks routing are completed.

3.5 Implementation specifications

The algorithm of the models expressed in this paper has a recursive-like pattern. Thus, special attention is needed to avoid deadlock and ineffective performance.

3.5.1 Avoiding program deadlock

A programming deadlock may occur when the track has an circulating dependency displayed in figure 19.



Figure 19: Circular cell tracks dependency relationship among cell track 1, 2 and 3

However, this would only occur if the cell route is always executed in sequential order. For this reason, the track to reroute is always handled first to avoid this circumstance.

3.5.2 Minimizing reroute events

The track of a cell track could go over 100 frames, and the intricate relationship between cell tracks could lead to the amount of reroute growth drastically. Particularly when the later tracks frequently induce reroutes to former ones. An example is shown in table 3.

Frame	Cell	Cause reroute to Trajectory
1	1	-
1	2	1
1	3	2, 1
1	4	3, 2, 1
2	5	4,3,2,1

Table 3: An example of track dependency that would cause exponential growth of cell reroute as it progresses further.

The reroute count rises in exponential scale since the latter track always causes the former ones to reroute. To tackle this, a dependence tree is constructed to store the dependency relationship among cells. A deadlock relationship may occur as the tracks' dependency keeps changing from rerouting. Hence, a recursive checking, which removes any old relationship when

recursive is detected, is built to ensure no deadlock relationship is inserted to the map. The dependency map would greatly reduce the number of reroutes in a situation where cells have a lot of dependency.

3.6 Ground truth data

The ground truth data used in this paper are obtained by manual labeling. An expert has to go through frame by frame to select the point that is believed to be the next node of the trajectory. To reduce bias, the selected point is validated by comparing with the segmented image and original image. The number of selections an expert has to do with one cell track is equal to the number of images in the series. For instance, a series with 20 tracks with average track length of 80 requires 20x80=1600 labels. This ground truth serves as the standard for performance evaluation.

3.7 Evaluation method

3.7.1 Falsely Identified Tracker (FIT)

FIT is a measurement of the number of incorrect in the prediction track to the best mapped ground truth track. The ground truth to be paired is illustrated in figure 20.

Identification Map w.r.t. estimate ${\cal E}$

- for each estimate, \mathcal{E}_i
 - collect \mathcal{GT} s which pass the coverage test into a function $n_{ij} = \sum_t \sum_j \mathbb{1}(F_{i,j}^t > t_C])$
 - determine the majority rules winning estimate $\hat{j}_i = \arg \max_j n_{ij}$
 - map $\mathcal{E}_i \to \mathcal{GT}_{\hat{j}_i}$
- end

Figure 20: Pseudocode of finding the best mapping ground truth for each predicted tracks[9]

After then, the number of incorrect predictions (green region in figure 21) will be counted and become the FIT score.



Figure 21: FIT. The mispredicted track points are colored in green. The numbers of mispredicted points of all tracks are accumulated to form the final FIT.

3.7.2 Falsely Identified Object (FIO)

FIO counts the number of mismatches in ground truth that are mapped to the best prediction track. The best prediction track to be mapped is shown in figure 22.



Figure 22: Pseudocode of finding the best mapping predicted tracks for each ground truth[9]

The region that is mistakenly routed to another region is checked as shown in figure 23. This region forms the score of FIO.



Figure 23: FIO. The mispredicted track points are colored in light green. The numbers of mispredicted points of all tracks are accumulated to form the final FIT.

3.7.3 Track Purity (TP)

TP is a measurement of the rate of the predicted tracks that matches the ground truth objects over the total number of frames exists.

$$TP^{i} = \frac{\max_{j} \, \mathsf{n}_{ij}}{\mathsf{n}_{i}} \tag{7}$$

3.7.4 Object Purity (OP)

OP is a measurement of the rate at which ground truth objects that match the predicted tracks over the total number of frames in ground truth.

$$TP^{i} = \frac{\max_{j} \, \mathsf{n}_{ji}}{\mathsf{n}_{j}} \tag{8}$$

4 Result

A simple state space search is executed to each of the algorithms mentioned in the methodology section. The best result is extracted and compared altogether with three other existing algorithms namely KDE [5], DeLTA[6] and Hungarian [7].

The four measurements proposed by Smith Kevin, Gatica-Perez Daniel[9] namely FIT, FIO, TP and OP are chosen for performance evaluation. In terms of performance, the lower FIT, FIO and higher TP, IO value the better. The details of the four measurements are given below.

4.1 Best hyperparameter value

The hyperparameters available and the best value found for each algorithm is shown in figure 24. The values are found by a grid search.



Figure 24: Best hyperparameter value discovered for each algorithm

4.2 Performance statistic

The performance chart of each chosen evaluation method is shown in figure 25. All the performance values are normalized to ease comparison. The components in the plots include: the green triangle that shows the mean values; the orange line that represents the median value; the upper and lower line of the box displays the first and third quartile of the data respectively; the whiskers tells the lower and upper extreme border (1.5 times the interquartile range); outliers are labeled with the black circles.

It is seen that the measurement from FIT has the highest variance among all of the models. The highest value recorded is 0.122 which is 13 times higher than the lowest value at 0.009. The performance data distribution of DeLTA, Viterbi-MLT, Viterbi-SLT and Feature weighted model has a very narrow range and are very close to zero, unlike the two other models that have a much wider distributed range. This can be seen by the quartile box and indicator of Hungarian and KDE that goes up to 0.2. FIO and TP on the other hand have a rather evenly distributed pattern: the quartile box and whiskers are located in similar regions. However, FIO and TP have a wider distribution as seen from the longer quartile box.

The outliner distributes various charts among the charts. With FIT having the most outliners and occurring in almost all models, in contrast to TP that has no outliers at all. FIO and OP have a few outliers.

All of the models have an unstable TP performance, almost all of them have a wide data distribution from 0.1 to 1. Only Viterbi and Viterbi-MLT did slightly better with a lower range at approximately 0.2.

On the other hand, all the models are not performing well in OP but having a much stabler performance. Most of them have their whiskers ranging in around 0.2. With at most two outliers from each of them. Feature weighted algorithm has the most uneven data distribution among



Figure 25: Performance chart of each measurements

all the models.	The first	and third	quartile	ranged	more	than	0.2	and	the	whiskers	ranged
around 0.4.											

Algorithm	FIT	FIO	TP	OP
KDE [5]	0.1224(7)	0.3993(6)	0.5366(7)	0.2682(5)
DeLTA [6]	0.0195(2)	0.3650(5)	0.6940(2)	0.2850(4)
Hungarian [7]	0.0870~(6)	0.4370(7)	0.6680(6)	0.3140(1)
Viterbi	0.0594(5)	0.2171(2)	0.6730(5)	0.2392(7)
Viterbi-MLT	0.0099(1)	0.1730(1)	0.7634(1)	0.2992(3)
Viterbi-SLT	0.0425(3)	0.2985(3)	0.6886(4)	0.2411(6)
Feature weighted	0.0507~(4)	0.3285(4)	0.6902(3)	0.3127~(2)

Table 4: The mean performance of each tracking model. The rank of each measurement is indicated in the parentheses besides the score.

The performance result in table 4 reveals that DeLTA is doing quite well among the existing methods. It managed to outperform other models in FIT, FIO and TP, particularly in FIT that achieved a mucher better score at 0.0195; It also managed to secure a second position in OP with value of 0.2850 which is approximately 10% lower than the best model which scored 0.3140.

The Viterbi-MLT stands out to be the best model among others. However, the second best model is rather difficult to be determined. The DeLTA algorithm did very well in FIT and TP ranked second but has a lower rank in FIO. Hungarian won the first place for OP but it ranked second last or worse in all other measurements. It did particularly badly in FIO which is almost 10

Among the four new algorithms introduced, the Viterbi-MLT model scored the best among all algorithms in FIT, FIO and TP; FIT, FIO and TP scored 49%, 20% and 10% better respectively than the second highest models. OP secured the third place which scored approximately 5% (or 0.047) lower than the best model. Viterbi-SLT has a mediocre score in most of the attributes, ranking around fourth place.

All of the models did not do well in OP, even the best model could only achieve 0.3140.

5 Extra section: Cell trajectories Classification

5.1 Background

One of the potentials of a good cell tracking algorithm is to support research studies by classifying specimens with machine learning methods. Thus, an experiment on this is held with given ground truth.

5.2 Data

The ground truth data is in excel format which contains coordinates, labels, distance, etc as shown in figure 26. There are 105 trajectories in total in the dataset.

Track no.	Slice no.	x	Y	Distance	Velocity	Pixel Value
2	1	158	33	-1	-1	72
2	2	156	45	1.569	0.785	189
2	3	157	46	0.182	0.091	63
2	4	156	52	0.785	0.392	8
2	5	160	56	0.73	0.365	56
2	6	155	69	1.797	0.898	3
2	7	156	69	0.129	0.065	255
2	8	156	79	1.29	0.645	255
2	9	156	88	1.161	0.581	255
2	10	156	93	0.645	0.323	68
2	11	149	91	0.939	0.47	129
2	12	147	109	2.336	1.168	6
2	13	142	110	0.658	0.329	16
2	14	151	114	1.271	0.635	9
2	15	151	124	1.29	0.645	32
2	16	150	139	1.939	0.97	19
2	17	157	144	1.11	0.555	48

Figure 26: Sample ground truth data of labeled trajectory

The specimens are classified with two labels namely trajectory distance and mutation status. The class details of the two labels are shown in table 5 and 6.

Class	Description
Local	Total distance of the cells traveled is more than 200m
Distant	Total distance of the cells traveled is NOT more than 200m

Table 5: Trajectory distance

Class	Description
Plus	Representing $+/+$. It is wild type or unmodified allele of the Zebrafish gene
Minus	Representing -/-, which means both genes in the Zebrafish are missing

Table 6: Mutation status

5.3 Methodology

Binary images with dimension 512x512 are generated from the dataset, a sample is shown in figure 27.



Figure 27: Binary image sample generated from ground truth

Convolutional Neural Network (CNN) is widely used for classifying image data. However, the sample size of the dataset is too small to train a CNN. Thus, the number of images is increased by rotating and shifting the trajectory. A few samples of the extra image generated is shown in figure 28.

The number of images is increased to around 3,000 after this process, and it can be increased further by adjusting the parameter in generate_extra_binary_tracks.py.

Two other problems to handle are the high memory consumption and long execution time that is caused by the high dimension of the image.



Figure 28: New images generated from seed by rotating and shifting

To handle this, the images are being rescaled to 256×256 from dividing the coordinates by 2. A sample is shown in figure 29.



Figure 29: Image rescaled from 512x512 to 256x256

This saves 75% of the memory and also drastically decreases the execution time as the size of the images are reduced to 25Different training set ratios are applied to see if there is a convergence point on the accuracy.

5.4 Result

Figure 30 indicates the result of the classification accuracy in different training set percentage.



Accuracy of training and test set

Figure 30: Accuracy of the classification model in different training set percentage

The trend of the training and test set are very different. The training set has a very fluctuated performance in the beginning, the accuracy ranges from 55 to 80 percent when training set ratio is below 0.6. The accuracy increased rapidly above that point. In contrast, the test accuracy has a more stable upward trend, it gradually increases from 35 to 45 percent from the beginning to the end. There is no obvious convergence point observed from the result. On the other hand, there seems to be overfitting, especially starting from 60% when the training and test set accuracy gap keeps increasing.

The confusion matrix of the test set at 90% training set ratio is captured and shown in figure 31. It can be seen from the matrix that the prediction is relatively better for Distant +/+ and Local +/+, with an accuracy of approximately 71% and 52% respectively. The prediction accuracy for distant -/- and local -/- is far from ideal, the accuracy is worse than a wild guess.



Figure 31: Confusion matrix of test set at 90% training set ratio

6 Discussion and Conclusion

In this paper, new algorithms are being built by extending existing ones (The Viterbi group) or by observing the labeled data(Feature weighted). Those algorithms are built based on different inspirations and are optimized with hyperparameter tuning. From the experiment results obtained, it is noticed that the original Viterbi algorithm and Viterbi-SLT underperformed in most measurements. Feature weighted model generally has a reasonable ranking over the four measurements ranging from second to fourth. The performance indices of it in fact looks very similar to an enhanced version of Hungarian — there is a significant increase in all measurements except having a very comparative score in OP. In fact, the two algorithms have very similar concepts except more conditions are considered in the Feature weighted algorithm. However, it is just good enough to secure 2nd place in most of the evaluation indices. Viterbi-MLT managed to outperform other algorithms remarkable in FIO and FIT; it also did very well in OP. Thus, it is considerably the best model out of all models.

A common characteristic can be observed from the first two algorithms which did not do as good — they do not consider other trajectories or include fewer frames when making cell trajectory predictions. This provides a hint that including more frames and nearby trajectories into route prediction would lead to a better result.

On the other hand, two major performance patterns can be seen among the top algorithms. The first pattern is seen from Viterbi-MLT/SLT and DeLTA methods which tend to do better in FIT, FIO and TP. The second pattern is found from Feature weighted and Hungarian algorithms that have a better performance in OP but did much worse in the other 3 measurements.

Among the numerous experiments done for cell trajectory prediction, we did not find an

algorithm that can outperform the others in all measurements. Different algorithms could have very different approaches which leads to very different results.

From the experience of this project, we observed an intricate relationship between cell trajectories, and the huge amount of data makes it infeasible to evaluate and discover rules by solely eye observation. In addition, the cell trajectory rules could be changing ongoingingly due to ever growing advancement of the cell tracking technology. The routing rules may require an adjustment when more ground truth data is made available, working on different types of cell, etc. For these reasons, having a highly tunable and reusable algorithm is conceivably beneficial for realizing appropriate cell movement rules.

7 Future work

7.0.1 Tracking with other models

Cell track behavior is highly related to historical trajectory. (direction variation, distances, etc) The current U-Net Model attempts to predict the outcome with two frames (t and t - 1). The U-Net could be further enhanced to include multiple previous and following frames so the data used for prediction can cover more data for analysis.

In the CNN classification experiment, an overfit pattern is observed due to the fact that the sample size of the data source is too small. A much larger sample size is essential for further feasibility exploration. From the experiment done, it appears that it is easier to identify between Local and Distant than +/+ and -/-. It remains uncertain whether a larger sample size would help with identifying the mutation status effectively.

7.0.2 Routing algorithm variations

Exploring the right routing rules with wider variations of routing algorithms would be beneficial due to the fact that it is difficult to rely purely on observing the segmented images to identify routing rules. We can try out rules with different calculation methods such as using average or accumulated connection scores. The dependency among cell tracks can also be further increased by applying a relative threshold during merge/split events. For example, the cells have to be higher than a certain percentage of the highest score cell in order to share the same node. In some occasions, determining cell merge is uneasy solely on the score function. For this reason, judgment on handling those events could potentially be tackled better if a feature is developed to predict the number of cells in each node. That might possibly greatly simplify the complexity for multiple track collisions at a certain node.

7.0.3 State space exploration

The Viterbi-MLT, Viterbi-SLT and Feature weighted model contain several positive rational hyperparameter numbers that form a multidimensional tunable state space. These parameters (such as merge threshold, discount rate per layer, feature weights, etc shown in figure 24) could lead to a great improvement in tracking when optimized properly. However, optimizing a state space in this size is infeasible with brutal-force exploration. Some state space exploration theories such as Evolution Strategy[16] can be implemented to search for better hyperparameters.

7.0.4 Cell synthesis

The ground truth from artificial or synthetic images provide an unbiased ground-truth[10], and it would also greatly increase the efficiency from creating one. The existing ground truth of cell tracks are labeled one by one manually. This is a very time consuming process as the track is very lengthy in general. In addition, this process could be very error-proning in certain circumstances such as multiple track collison.

Developing a cell synthesis program could provide a huge amount of error-free ground truth data for large scale training and testing. A simple version of cell synthesis can be found at cell_synthesis.py. The behavior of the synthesis cells can be controlled by parameters: move distance range, direction variation range, Initial total cell range, mitosis probability range and apoptosis probability range. All those parameters take two values that define the minimum and maximum value.

References

- [1] Automated cell tracking using StarDist and TrackMate https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7670479/pdf/f1000research-9-29845.pdf
- [2] Zebrafish development and regeneration: new tools for biomedical research
- [3] Ronneberger Olaf and Fischer Philipp and Brox Thomas. U-Net: Convolutional Networks for Biomedical Image Segmentation. CVPR, 2015.
- [4] R. M. Marzieh, and Y. P. P. Chen, "Tracking leukocytes in intravital time lapse images using 3D cell association learning network," Artif Intell Med. vol. 118, 2021.
- [5] Yan Kuan, Verbeek J. Fons. Cell tracking and data analysis of in vitro tumour cells from time-lapse image sequences. VISAPP 2009 - Proceedings of the 4th International Conference on Computer Vision Theory and Applications. 2009. Vol, 1. pp. 281-286.
- [6] Lugagne J-B, Lin H, Dunlop MJ. DeLTA: Automated cell segmentation, tracking, and lineage reconstruction using deep learning. PLoS Comput Biol. 2020. Vol, 16(4), pp. 1-18.
- [7] Tashita A. et al. Macrophage Tracking Using the Hungarian Algorithm in Time Lapse MR Images International Conference on Emerging Trends in Engineering and Technology, ICETET. 2016. pp. 169–173.
- [8] Magnusson E. G. Klas, Jalden Joakim, Gilbert M Penney, Blau M Helen. Global linking of cell tracks using the Viterbi algorithm. IEEE Transactions on Medical Imaging. 2015. Vol, 34(4), pp. 911-929.
- [9] Smith Kevin, Gatica-Perez Daniel, etc. Evaluating Multi-Object Tracking, IEEE. 2005.
- [10] A Novel Function of TLR2 and MyD88 in the Regulation of Leukocyte Cell Migration Behavior During Wounding in Zebrafish Larvae

- [11] Ekpenyong, A. E. et al. Viscoelastic properties of differentiating blood cells are fate- and function-dependent. PLoS ONE 7, e45237. https://doi.org/10.1371/journal.pone.0045237 (2012).
- [12] Chalut, K. J. et al. Chromatin decondensation and nuclear softening accompany Nanog downregulation in embryonic stem cells. Biophys. J. 103, 2060–2070. https://doi.org/10.1016/j.bpj.2012.10.015 (2012).
- [13] Pagliara, S. et al. Auxetic nuclei in embryonic stem cells exiting pluripotency. Nat. Mater. 13, 638–644. https://doi.org/10.1038/nmat3943 (2014).
- [14] Hodgson, A. C. et al. A microfluidic device for characterizing nuclear deformations. Lab. Chip 17, 805–813. https://doi.org/10.1039/c6lc01308b (2017).
- [15] Morrish, R. B. et al. Single Cell Imaging of Nuclear Architecture Changes. Front Cell Dev Biol 7, 141. https://doi.org/10.3389/fcell.2019.00141 (2019).
- [16] Evolution strategy. https://en.wikipedia.org/wiki/Evolution_strategy
- [17] Actor J (2012). Elsevier's Integrated Review Immunology and Microbiology (Second ed.). doi:10.1016/B978-0-323-07447-6.00002-8.
- [18] Dictionary of Cancer Terms. Neutrophil. https://www.cancer.gov/publications/dictionaries/cancer-terms/def/neutrophil
- [19] SenGupta Shuvasree, Hein Lauren E., Parent Carole A.. The Recruitment of Neutrophils to the Tumor Microenvironment Is Regulated by Multiple Mediators. https://www.frontiersin.org/articles/10.3389/fimmu.2021.734188/full