WORKING TITLE
Developing a pipeline for high-throughput analysis of dynamic and static T cell/tumor cell data

Masterarbeit
and der Medizinischen Fakultät
der Eberhard Karls Universität Tübingen

vorgelegt von

Quach, Julia

Tübingen, DATUM

Contents

1. Summary

It is widely known that cancer is one of the leading causes of death in Western society [citation]. Its high mortality is predominantly caused by tumor resistance against all available therapies [citation]. To combat this resistance, new therapies have been developed from which the most recent and promising therapeutic approach are immunotherapies which have shown great success in the past [citation]. Immunotherapies make use of engineered or innate immune cells, especially T cells, to eliminate malignant tumor cells. Still, many cancer types remain resistant. To overcome those therapy resistances, there follows a high need to understand the crucial factors behind efficient T cell killing.

To elucidate the mechanisms of efficient T cell killing, [Introduction on OVA tumor models, live cell imaging, staining, fix-while-filming]

To achieve meaningful results, large data sets and their statistical analysis are pivotal. To enable a large-scale analysis of tumor and T cell interaction, this thesis aims at developing an R package to automatically evaluate cell dynamics in live cell imaging movies. This would enable a quantitative analysis of cell-cell contacts on big sample sets. Besides, the R packages proposes to correlate dynamical cell data to immunological staining results. To put it in a nutshell, this thesis' goal is to heighten the explanatory power of available live cell imaging and immunological staining techniques by enabling large-scale analyses and correlation of dynamical to immunological staining methods.

To achieve an efficient analysis of live cell imaging data, [... More details on the R package and workflow].

We show that [...]

2. Introduction

- T cell activity against tumor cells
- Crucial factors for T cell killing and the relevance of cell dynamics
- Tools to investigate cell dynamics
- Software to analyze cell tracks

2.1 T cells against tumor cells

- Hallmarks of Cancer
- Escaping immune system
- T cell killing
- Tumor cell elimination by the body always by T cells (I think)

2.2 The relevance of cell dynamics

- Advantages of static methods (FACS, sequencing, immuno-stainings)
- Shortcomings of static methods
- Benefit of cell dynamics

2.3 Available tools to investigate cell dynamics

Requires recognizing the cell and then characterizing its movement.

2.3.1 Cell segmentation

- Stardist
- CellPose

2.3.2 Cell tracking

- LAP tracker
- Kalman tracker

2.4 Scope

3. Methods

- Imaging
- Segmentation
- Tracking
- Export and loading into RStudio
- Cell-cell contact computation
- Quality control (CelltrackR)

3.1 Extracting features

For a given set of cell tracks, we would like to observe meaningful features of the cell movement and activity. To acquire the cell motility data, tumor cells (B16F10-H2BmCherry-OVA) and T cells (OT1 GFP) were seeded into a 3D gel and imaged over time for several hours at a rate of 90 seconds per image. Afterwards, the images were segmented and cell tracks were obtained using an image analysis tool, e.g., the TrackMate Plugin for ImageJ or Imaris 3D.

The cell tracks have the following structure: Each .csv-file is a table with the columns "ID", "time point", "x-" , "y-" and "z"-coordinate. We have separate .csv-files for the tracks from tumor cells and the tracks from T cells.

- $X :=$ Tracks from tumor cells (tumorTracks.csv)
- $Y :=$ Tracks from T cells (tcellTracks.csv)

We would like to extract the following features from our data:

1. Identify cell-cell contacts between T cells and tumor cells
2. Number of T cell contacts per tumor cells
3. Number of contacts between one T cells and one tumor cells
4. Duration of a T cell contact
5. Correlation of T cell dynamics to tumor cell outcome (i.e., Fix-While-Filming method).

3.1.1 Identify cell-cell contacts between tumor and melanoma cells.

We implement a function *contacts*(X,Y, radDist, minTime) where $X :=$ tumorTracks.csv and $Y :=$ tcellTracks.csv are tracks objects, *radDist* is a distance threshold and *minTime* a minimum of consecutive timepoints. The function shall return an array (in Python it should be a list) of cell pairs that have a distance below radDist for more than minThresh timepoints with the start and end time of this "contact".

1. Calculate the Euclidean distance for every cell pair (x, y) where $x \in X$ and $y \in Y$ for every timepoint t.
2. Set threshold radDist for the radial distance.
3. Filter for cell pairs with a distance below radDist.
4. Set threshold minTime for the number of consecutive timepoints.
5. Filter for cell pairs with a distance below radDist for more than minTime timepoints.
6. Return a list with cell pairs and their start and end time for which their distance is below the given threshold.

Note: To code this, use the Matlab (and the R) implementation as pseudo code.

3.1.2 Number of T cell contacts per tumor cell

We implement a function *numContacts*(X,Y, radDist, minTime).

1. Apply *contacts*(X, Y, radDist, minTime)
2. Sort by elements from X (tracks of tumor cells).
3. For every tumor cell $x_i \in X$ that was involved in a contact, count the number of contacts and return it with the respective start and end time of contact (duration).

3.1.3 Number and duration of contacts for one T cell and one tumor cell

We implement a function *numContactsDetailed*(X,Y, radDist, minTime).

1. Apply *numContacts*(X, Y, radDist, minTime)
2. Return two lists:
   (a) A list of the tumor cells and for each tumor cell, the number and respective duration of contacts with all its contact T cells.
   (b) A list of the T cells and for each T cell, the number and respective duration of contacts with all its contact tumor cells.

3.1.4 Correlation of T cell dynamics to tumor cell outcome

Analysis of the tracks object:

1. Apply *contacts*(X, Y, radDist, minTime) = result.
2. For every tumor cell $x \in$ result, save its ID, its last timepoint and its xyz-coordinate.

3. Exclude the cells which did not last until the last timepoint.
4. Return a list *Candidates* with the cell IDs, its last timepoint and its xyz-coordinate.

Analysis of the fixed sample:

1. For every parameter (p21, etc.), set a ROI around the positive cells.
2. Convert each ROI to a set $A_i \subseteq \mathbb{R}^2$ and label the ROIs uniquely.
3. For each ROI A_i, check whether a cell $x \in \mathit{Candidates}$ is also in our ROI $x \in A_i$.
4. Allow only one cell per ROI.
5. For each parameter, return a list of cell IDs that matched a ROI.

3.2 Data for network training

Our feature vector is: For each tumor cells in $x_0 \in X$.

$$\begin{pmatrix} \text{T cell contacts to } x_0 \text{ in total (integer)} \\ \text{Array of contacts of } x_0 \text{ to each T cells (boolean array)} \\ \text{Array of contact duration to each T cell (float array)} \end{pmatrix}$$

The length of the latter two arrays is the total number of T cells.
Our outcome vector is: For each tumor cells in $x_0 \in X$.

$$\begin{pmatrix} \text{dead (yes/no)} \\ \text{p16 (yes/no)} \\ \text{p21 (yes/no)} \\ \text{End nucleus size} \\ \text{Ratio start and end nucleus size} \\ \text{Parameter [\ldots]} \end{pmatrix}$$

where the parameter *dead*, *p16* and *p21* is a boolean, and the other parameters can be of the float type (e.g., representing fluorescence intensity of a marker).

3.3 Improvements

3.3.1 Improvement of tracking

Instead of calculating the distance between xyz-coordinates where one xyz-coordinate represents one cell, calculate the distance between ROIs.

3.3.2 Extracting senescence-relevant information (Oli)

If we can extend our data:

- Tracks from tumor cells (tumorTracks.csv)
- Tracks from T cells (tcellTracks.csv)
- Tracking of nucleus size: Time points of cell tracks for which an enlarged nucleus is detected

Then, we would like to extract the following additional features from our data:

1. For each tumor cell, Number of T cell contacts before and after change of nucleus size.
2. For each tumor cell, duration between last CTL contact and nucleus change.
3. Behavior of surrounding cells when senescent cell is present.

Number of T cell contacts before and after change of nucleus size Under the assumption that cells do not switch from a senescent state back to a normal state: For each tracks object t_j, $j \in J = Alltracks$:

1. Split the tracks object t_j at the first time point when an enlarged nucleus is detected into two tracks objects t_{j1} and t_{j2}
2. Calculate the number of T cell contacts with *contacts*(X,Y, radDist, minTime) for t_{j1} and t_{j2}.
3. Compare the number of T cell contacts for t_{j1} and t_{j2}.

Duration between last CTL contact and nucleus change In our data set, we should have the time point of the nucleus change. Now, we have to find the time point of the last CTL contact before nucleus change.

For each tracks object t_j, $j \in J = Alltracks$:

1. Split the tracks object t_j at the first time point when an enlarged nucleus is detected into two tracks objects t_{j1} and t_{j2}
2. For t_{j1} (which is the track before the nucleus size changes), use the function to calculate the duration of T cell contact per tumor cell (see **??**).
3. For the last contact in t_{j1}, get the last time point.
4. Return the difference between the time point of the nucleus change and the last time point of the last contact.

Behavior of surrounding cells when senescent cell is present Apply the following function for conditions with and without CTL:

1. For a tumor cell $x \in X$, consider its surrounding cells within a radius r.
2. Count the number of surrounding tumor cells that turn senescent.

3.3.3 Additional data for network training

Our additional features for each tumor cells in $x_0 \in X$.

$$\begin{pmatrix} \text{Number of T cell contacts before change of nucleus} \\ \text{Number of T cell contacts after change of nucleus} \\ \text{Ratio of T cell contacts before and after change of nucleus} \\ \text{Duration between last T cell contact and nucleus change} \\ \text{Cumulative number of surrounding cells with enlarged nucleus} \end{pmatrix}$$

4. Results

An R package was developed with the following dependencies and functions:

Dependencies:

- celltrackR, dplyr, ggplot2, keep, pracma, stringr

Functions:

- cellContactsByTracks
- cellContactsByROIs

Überarbeiten für ROI und Track output:

- contactByPairs
- displayCellDuringContact
- displayContacts
- matchTrackNamesWithIDs
- numContactsPerCell
- numContactsperPair
- plotTracksWithROIs
- ROIsToTracks
- speedDuringContact