1. Experiments

We used five live-cell imaging datasets to build and to challenge the R package. All data sets were acquired using a Widefield Microscope DMI8 by Leica Microsystems (Wetzlar, GER) as imaging device equipped with a [...] objective with a resolution of [...] at 5% CO_2.

1.1 Data sets

To set the distance and duration thresholds for cell-cell contact computation, we used three data sets of live-cell imaging films with tumour and T cells. An overview of the data sets is given in Table 1.1.

Table 1.1: Overview of data sets that were used for building the R package

Tumour cell line	T cell line	Frame rate (sec/frame)	Frame size (µm × µm)	Duration (h)	Temperature (°C)
YUMM-OVA					
add cell concentration YUMM					
Ovarian					

To challenge the work flow for cell-cell contact detection, we applied the image analysis pipeline on three additional data sets (see Table 1.2).

Table 1.2: Overview of data sets that were used for challenging the R package

Tumour cell line	T cell line	Frame rate (sec/frame)	Frame size (µm × µm)	Duration (h)	Temperature (°C)
YUMM					
YUMM-OVA					
B16F10					

2. Method

The R package *cellcontacts* was built to process cell tracks acquired from live-cell imaging of tumour cells with T cells. It is meant to compute cell-cell contacts and characteristics of cell movement for large data sets. Moreover, it offers functions to connect the results on cell dynamics to findings from immunological staining. The package requires preprocessing of the live-cell imaging films with a cell tracking tool, e.g., TrackMate or Imaris. In this chapter, the algorithm to compute cell-cell contacts is described and the image analysis pipeline incorporating the cell tracking tool TrackMate is laid out.

2.1 Algorithm to compute cell-cell contacts

In this section, we formally define cell-cell contacts and describe the algorithm to compute cell-cell contacts.

Formalizing the data set and defining cell-cell contacts Let $n \in \mathbb{N}$ be the the last time point in the live-cell imaging film, equivalent to the maximum frame number. Let M_i be the set of all cells at one time point $i \in \{1, \dots, n\}$ with $0 \in M_i$ representing a non-registered cell. Let $U_i \subset M_i$ be the set of tumour cells and $V_i \subset M_i$ the set of T cells at a time point $i \in \{1, \dots, n\}$ such that $M_i = U_i \cup V_i$.

A tumour cell shall be represented as $u = (u_1, \dots, u_n) \in U_1 \times \cdots \times U_n$ such that for each time point $i \in \{1, \dots, n\}$, the element u_i either represents the tumour cell at this time point or equals 0 if the respective tumour cell was not registered at this time point. Analogously, we define T cells as $v = (v_1, \dots, v_n) \in V_1 \times \cdots \times V_n$.

Let $d_i : U_i \times V_i \to \mathbb{R}$ be a metric that describes the euclidean distances between the segmented area of tumour cell and a T cell for a given time point $i \in \{1, \dots, n\}$ and if either the tumour cell or the T cell was not registered during that time point, we define $d_i(u_i, v_i) = \infty$.

Let $x \in \mathbb{R}$ be a distance threshold in µm and $m \in \mathbb{N}$ a duration threshold given as number of frames. We define that a tumour cell $u \in U_1 \times \cdots \times U_n$ and a T cell $v \in V_1 \times \cdots \times V_n$ are *in contact* if and only if their distance is below the distance threshold for m consecutive frames:

$$u \text{ and } v \text{ in contact} \quad \Leftrightarrow \quad \exists\, j \in \{1, \dots, n\}\ \forall i \in \{j, \dots, j+m\} :\ d_i(u_i, v_i) \leq x$$

Algorithm (cell-cell contact computation) To compute cell-cell contacts, we need to find all tumour cell/T cell pairs (u, v) that maintain a distance below threshold for m consecutive frames. As the number of distance calculations grow quadratically with increasing frame size of the film, we add preprocessing steps to fasten the computation time.

Algorithm to compute cell-cell contacts

Input:

- Cell tracks and cell areas of tumour cells and T cells at all time points.
- A distance threshold x.
- A minimum duration threshold expressed as a number of frames m.

Output: Data frame with columns

- Names of cell pairs that maintain a distance below threshold for at least m consecutive frames.
- Time points during a contact.
- Cell-cell distances during a contact.

Instructions:

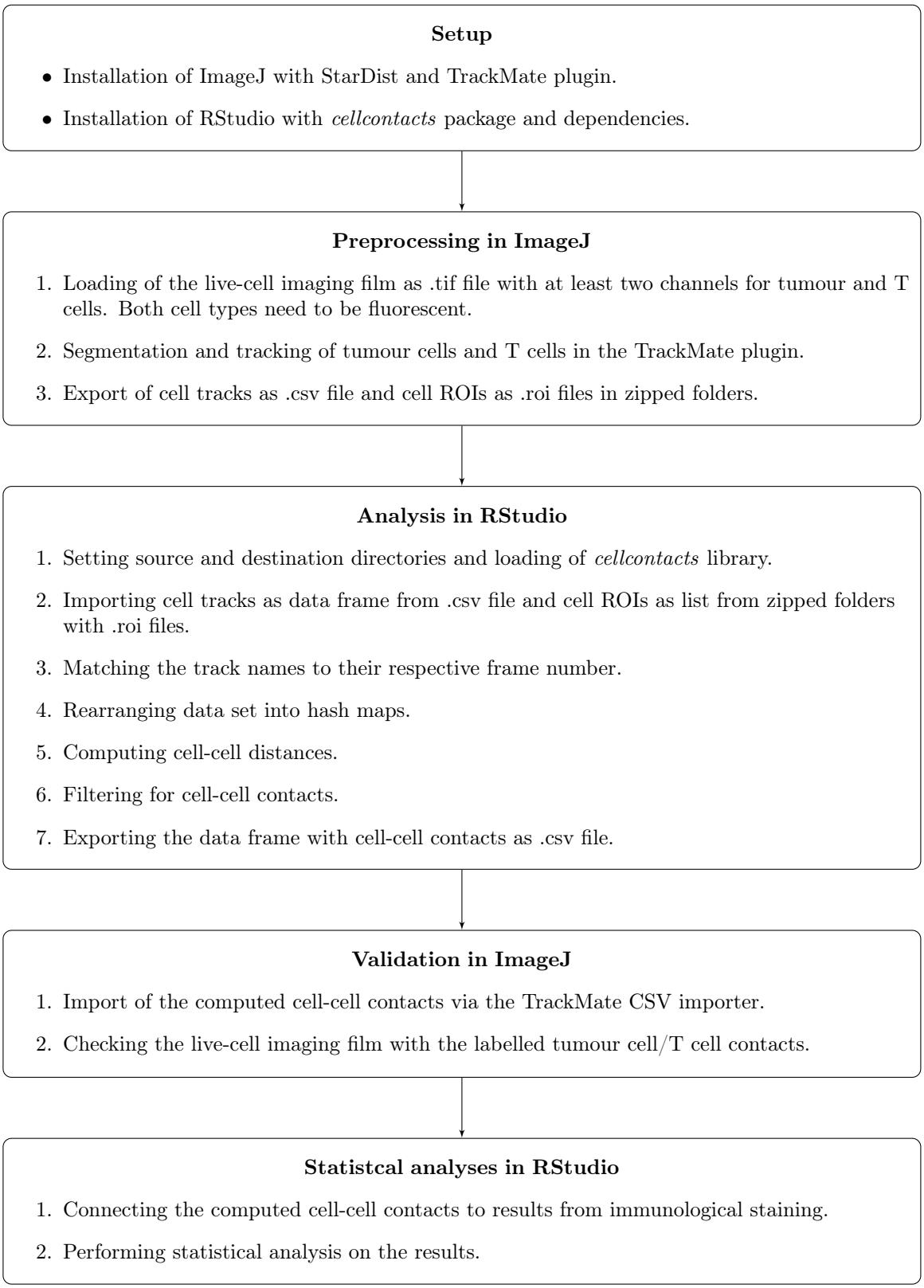
1. Preprocessing
   - 1.1 Compute the center points of all tumour and T cells.
   - 1.2 Create a grid in which columns and rows have a distance of 100 µm.
   - 1.3 Sort the computed cell center points into the grid.
2. Distance computation
   - 2.1 Compute all possible tumour cell/T cell pairs for each time point.
   - 2.2 For each pair, save the information whether the grid coordinates are farther away than two two rows or columns.
   - 2.3 If the grid coordinates are at least in neighbouring rows and columns, compute and save the distance of the segmented tumour and T cell area.
3. Filtering for contacts
   - 3.1 For all time points, discard cell pairs that have a distance above the threshold x.
   - 3.2 For the remaining pairs, discard cell pairs that maintain a distance below threshold for less than m frames.
   - 3.3 Return the pair names, their distance and their time point information as a data frame.

2.2 Work flow

This section is a detailed guide through the image analysis work flow. An overview of the steps in the work flow is given in Figure 2.1. An exemplary R script are provided on the GitHub page of the R package and a detailed tutorial of the workflow is provided in the supplement ([...]).

Requirements To use this work flow on a live-cell imaging film, the imaged cells need to be fluorescent to enable their correct segmentation. To ensure correct tracking of fast-moving T cells, the frame rate needs to be set to 90 s/frame or faster.

Figure 2.1: This figure gives an overview how cell-cell contacts in live-cell imaging films are automatically detected and quantitatively analysed using the R package *cellcontacts*.



For the image processing, installation of the open source package Fiji, also known as ImageJ, with the plugins StarDist for cell segmentation and TrackMate for cell tracking is required. To use the *cellcontacts* package, the programming language R is needed, preferably in the RStudio environment.

Installation The latest development version of *cellcontacts* can be installed in the R environment after getting access to the *cellcontacts* project on GitHub:

```
devtools::install_github( "juliaquach02/cellcontacts" )
```

The package can be loaded via:

```
library( cellcontacts )
```

Preprocessing in ImageJ To extract the image features, we load the live-cell imaging film as .tif-file in Fiji to use the TrackMate plugin for segmentation and tracking. Two separate TrackMate session are needed to analyse T cells and tumour cells. During segmentation and tracking, the plugin offers the possibility to adjust parameters and to manually correct segmented cells or cell tracks if necessary. After segmentation and tracking, the cell areas of both data sets are renamed in the TrackMate GUI by their assigned track name and an unique index for the respective track.
We export cell tracks as .csv file. By default, the first three lines of the .csv file contain the header of its columns in three different text styles (capital letters, full name and abbreviated). To enable correct loading of the .csv files with R, two of the first three lines are deleted, e.g. via a text editor.
The renamed cells region of interests (ROIs) are exported from TrackMate and saved into several zipped folders with around thousand .roi-files per folder. Saving the ROIs into several zipped folders will enable parallel, hence faster, import to R. For the saving process, we provide an ImageJ macro on the package's GitHub page.
Apart from exporting cell tracks and cell ROIs, the TrackMate sessions can be saved as .xml file. This allows reloading the session with its segmentation and tracking parameters. At the end of the preprocessing, there are two .xml files, one for the tumour cell and one for the T cell analysis session. In addition, there are two .csv files with the track information and at least two zipped folders with .roi files representing the segmented tumour cell and T cell ROIs. An overview of the data structure after this preprocessing step is given in the Table 2.1.

Table 2.1: This is an overview of files that should be available for both, tumour and T cells, after preprocessing a live-cell imaging film.

Folder	File type	Content
Tumour/T cells	.zip-file or folder with .zip-files	.roi-files that represent the area of the segmented tumour/T cells with file names in the format Track_[id]_[index].roi
	.csv file	Data frame in which each row represents one cell in one frame. The data frame should contain columns for the track names (in the format Track_[id]_[index].roi), the x-, and y-coordinate and the frame number.
	.xml file	This file is optional. It contains information about the TrackMate session such that the session can be reloaded.

Setting directories and loading the track data After acquiring all necessary files, we switch to RStudio to use the *cellcontacts* package. For this, we set the directories to import the data. Cell tracks saved as .csv files are loaded as data frame using the base R function `read.table()`. To load the zip-files containing the ROIs, we use the `read.ijzip()` function from the `RImageJROI` package.

Putting data into hash maps For an efficient retrieval of our data during computation, we provide functions to add the data set into hash maps. To decrease the computation time of cell-cell distances, we sort the cell ROIs into a 2D grid to get rough estimates of their position. To create a 2D grid, we provide the function `create_grid()`.
Using the hash maps, we can easily access the track names of all cells at a given time point to compute possible tumour cell/T cell pairs. Moreover, we can retrieve the center points and approximate position of the cells before computing cell-cell distances. Calling the approximate cell position avoids the computation of euclidean distances for cell pairs that are clearly farther away than the distance threshold. Finally, we add a hash map to facilitate the access to ROI coordinates for each cell track name.

Computing cell-cell distances Cell-cell distances need to be computed to filter for cell-cell contacts. For this, we provide the function `compute_distTimepoint_wGrid()`. This function computes for a given time point all possible tumour cell/T cell pairs. For each pair, the function checks the rough position of both cells using their position in the 2D grid. If the cells are at least in neighboring columns and rows of the grid, the euclidean distance between the cell ROI is computed. The output of the function is a data frame listing all possible cell pairs with either the distance of the pair or a remark that the distance is substantially larger than the distance threshold for a cell-cell contact.

Filtering cell-cell contacts To filter the cell pairs involved in a cell-cell contact, we only keep pairs that maintain a distance below a distance threshold for a minimum duration of frames. The distance and duration threshold can be set manually. As result, we obtain a list of tumor cell/T cell pairs and each list entry represents one pair and contains the columns *time point* and *distances*.

Exporting and visualization of results To check of the computed cell-cell contacts, we export the track information of the T cells that were in contact as .csv-file using `prepareExportContacts()` and `write.csv()`. The exported files are loaded into TrackMate via ImageJ > Plugins > Tracking > TrackMate CSV Importer. Using the .csv file, TrackMate labels the T cells in the .tif-file only during a contact (see Figure 2.2). This allows us to revisit the live-cell imaging film and to check whether cell-cell contacts are correctly computed.

Correlating results to immunological staining results We often want to connect the dynamic information of the cell tracks to the static information of the same cells from immunological staining. For this, we provide the function `match_to_endpoint_ROIs()` to match the cell tracks to the corresponding cell ROIs at the end point. The function `add_meas_to_matches()` can add the measurements of the signal intensity to the mapping of cell tracks to endpoint ROIs.

Identifying dying tumour cells [...]

Computing characteristics of cell tracks [...]

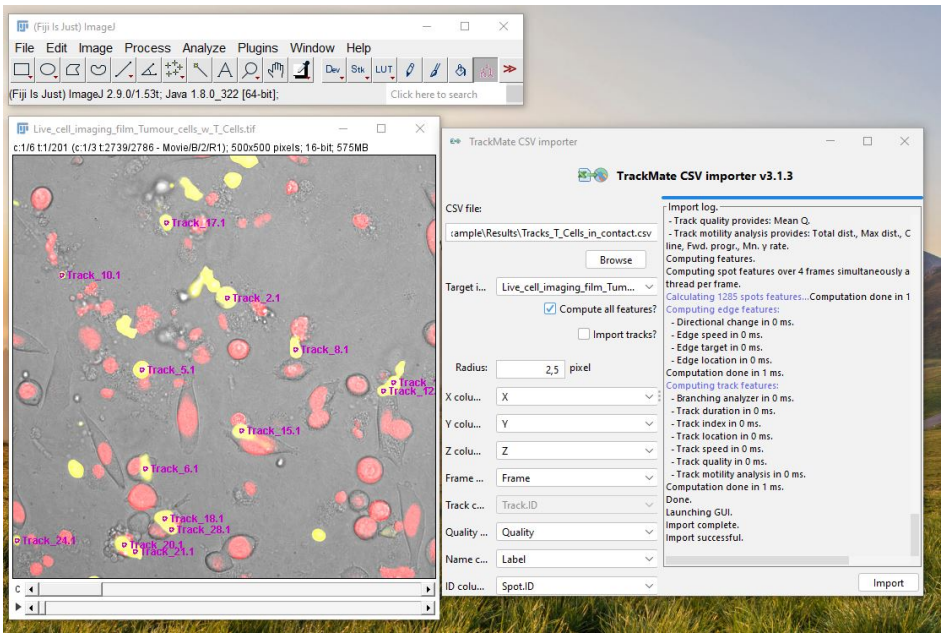


Figure 2.2: Via the TrackMate CSV Importer GUI, we can load our results back into a TrackMate session.

2.3 Quality control

Evaluation metric Our efforts to assess the reliability of our work flow were two-fold. On the one hand, we imported the computed results back into TrackMate such that T cells are labelled during a cell-cell contact for a visual validation. On the other hand, we manually counted cell-cell contacts in live cell imaging methods and evaluated the accordance of the computed cell-cell contacts to the manually obtained results.

To-do: Paragraph *Examining track quality*