

# **README: Code and Software Submission for Nature Research**

Alexander Jovanovic, Ahmad Sadeghi and Lucas Boeck

April 29, 2025

# Contents

<b>Acknowledgements</b>	<b>3</b>
<b>1 System Requirements</b>	<b>4</b>
1.1 ilastik . . . . .	4
Requirements . . . . .	4
Version . . . . .	4
1.2 Fiji . . . . .	4
Requirements . . . . .	4
Version . . . . .	4
Plugins . . . . .	4
1.3 R: The R Project for Statistical Computing . . . . .	4
Requirements . . . . .	4
Version . . . . .	5
1.4 MATLAB . . . . .	5
Requirements . . . . .	5
Version . . . . .	5
Packages / Toolboxes . . . . .	5
<b>2 Installation Guide</b>	<b>6</b>
2.1 ilastik . . . . .	6
macOS installation . . . . .	6
windows installation . . . . .	6
linux installation . . . . .	6
2.2 Fiji . . . . .	7
macOS installation . . . . .	7
windows installation . . . . .	7
linux installation . . . . .	7
Plugins installation . . . . .	7
2.3 R: The R Project for Statistical Computing . . . . .	8
macOS installation . . . . .	8
Windows installation . . . . .	8
2.4 MATLAB . . . . .	9
macOS installation . . . . .	9
Windows installation . . . . .	9
Linux installation . . . . .	9
Toolboxes . . . . .	9
<b>3 Demo and instructions for use</b>	<b>10</b>
3.1 ASCT Image Analysis Pipeline . . . . .	10
Step 1: Fiji preprocessing . . . . .	10
Step 2: ilstik Pixel Classification . . . . .	10
Step 3: ilastik Morphology Object Classification (MOC) . . . . .	11
Step 4: ilastik Propidium iodide Object Classification (POC) . . . . .	11
Step 5: MATLAB Drift Correction . . . . .	11
Step 6: MATLAB Single-Cell Tracking . . . . .	11
Step 7 (Optional): R Markdown Time-Kill Curve . . . . .	12
3.2 ASCT per well data Analysis . . . . .	13
ASCT Data Analysis in R . . . . .	13

## Acknowledgements

We would like to thank the following contributors for their support during the development of this image analysis pipeline: Dr. Ahmad Sadeghi for writing the MATLAB code for drift correction and single-cell tracking. Dr. Lucas Boeck for feedback and review of the overall pipeline design and implementation. Dr. Loïc Sauter for assistance with Fiji scripting and image analysis workflows.

# 1 System Requirements

The software has been tested on 64-bit Linux [sciCORE](#) High-performance computing (HPC) environment and macOS.

## 1.1 ilastik

ilastik the interactive learning and segmentation toolkit allows the user to leverage machine learning algorithms particularly for segmentation and classification to analyse image data. For more information on ilastik's system requirements, visit [ilastik](#).

### Requirements

64-bit machine, with  $\geq 8$  GB RAM

MacOS 10.9 and higher

Linux glibc 2.5 and higher

Windows 7 and later

### Version

ilastik-1.3.3post3-Linux

## 1.2 Fiji

For more information Fijis/Imagej's system requirements, visit [Fiji](#).

### Requirements

Windows XP, Vista, 7, 8, 10, 11, etc.

MacOS 10.8 "Mountain Lion" or later

Linux on amd64 and x86 architectures

### Version

HPC platform: Fiji/20201104-1356

local: Fiji/20230224-1206

### Plugins

BaSiC 2017-01-12

ilastik4ij 1.7.3

## 1.3 R: The R Project for Statistical Computing

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. For all platforms, having a modern CPU (2010 or later), at least 8 GB of RAM, and sufficient disk space (64 GB or more) is recommended for optimal performance, especially when working with large datasets or complex computations. For more information visit [R](#).

### Requirements

Windows 7 or later

macOS 11 (Big Sur) or higher. Both Intel-based and Apple silicon (M1, M2, etc.) Macs are supported.

Linux (e.g., Ubuntu, CentOS, or cluster environments such as SciCORE)

## Versions

HPC platform: R 4.1.2 (2021-11-01)

Local: R 4.4.0 (2024-04-24)

## R packages

The R packages that we used in ASCT data analysis and visualisation are listed in Table 1.

Data wrangling	Visualization	Utilities
dplyr	ggplot2	platetools
tidyr	ggpubr	gtools
stringr	ggforce	readxl
reshape2	ggprism	gridExtra
	ggrepel	GGally
	ggcorrplot	MESS
	ggExtra	RColorBrewer
	gghighlight	scales
	directlabels	
	wesanderson	
	plotrix	

Table 1: R packages used for data analysis and visualisation, grouped by purpose. *Note:* `plotly` enables interactive plots, extending beyond static visualisation.

## 1.4 MATLAB

### Requirements

64-bit system (macOS, Windows, or Linux)

Minimum 8 GB RAM (16 GB recommended for time-lapse processing)

At least 5 GB of disk space for installation, plus space for image data

macOS 11 or later, Windows 10 or later, or recent Linux distributions (Ubuntu, CentOS, etc.)

### Version

Local: MATLAB R2024b

### Packages / Toolboxes

Image Processing Toolbox

## 2 Installation Guide

### 2.1 ilastik

ilastik can be [downloaded](#) from the official website, and the corresponding installation guidelines can be found [here](#). You can also follow the simplified guide below to install ilastik.

#### macOS

1. Open the downloaded `.tar.bz2` file; it should extract automatically.
2. Move the `ilastik.app` to your Applications folder.
3. Control-click (right-click) on `ilastik.app` and select “Open” to launch it.

#### Windows Installation

1. Run the downloaded self-extracting installer.
2. Follow the on-screen instructions to complete the installation.
3. After installation, launch `ilastik.exe` from the Start Menu.

#### Linux Installation

1. Open a terminal.
2. Navigate to the directory containing the downloaded `.tar.bz2` file.
3. Extract the contents using: `tar xjf ilastik-1.*-Linux.tar.bz2`
4. Navigate into the extracted directory: `cd ilastik-1.*-Linux`
5. Run Ilastik: `./run.ilastik.sh`

## 2.2 Fiji

Fiji can be [downloaded](#) from the official website. Installation is straightforward as Fiji is distributed as a self-contained package for all major operating systems. The software comes bundled with key plugins and supports updating via its built-in updater.

### macOS

1. Download the .zip archive for macOS from the Fiji website.
2. Unzip the archive and move the `Fiji.app` to your Applications folder.
3. On first launch, right-click `Fiji.app` and select “Open” to bypass security prompts.

### Windows Installation

1. Download the Windows .zip archive from the Fiji website.
2. Unzip the contents to a preferred location (e.g., `C:\Fiji`).
3. Run `ImageJ-win64.exe` or `Fiji.exe` to launch the application.

### Linux Installation

1. Download the Linux archive (.tar.gz) from the Fiji website.
2. Extract the contents using: `tar -xvzf Fiji-linux64.tar.gz`
3. Move the extracted `Fiji.app` folder to a convenient directory.
4. Navigate to the folder in a terminal and launch Fiji with: `./ImageJ-linux64`

### Plugins installation

The necessary Fiji plugins to run the provided macro are included in the supplementary folders `goes_in_Fiji.app-jars` and `goes_in_Fiji.app-plugins`. Rather than using the Fiji updater or online plugin sites, you should directly copy these files into your Fiji installation:

1. Copy `BaSiC_.java` into your `Fiji.app/jars/` directory.
2. Copy the following .jar files into your `Fiji.app/plugins/` directory:
  - `BaSiC_.jar`
  - `ilastik4ij-1.7.3.jar`
  - `parallelcolt-0.10.1.jar`
3. Restart Fiji to load the plugins.

This setup was tested on a clean Fiji installation on macOS and confirmed to work without requiring additional plugin installations. For reproducible analysis, users are encouraged to use the provided plugin files and installation method exactly as described. For more information or updates to specific plugins, refer to:

[BaSiC](#)

[ilastik Fiji plugin](#)

## 2.3 R: The R Project for Statistical Computing

Follow the link to [download R](#) and please choose a location close to you. If you have troubles follow [the how to documents](#) provided by the official website. Optionally, install [RStudio](#) as a graphical interface for a more user-friendly experience on a local device.

### macOS Installation

1. Go to [for macOS installation](#)
2. Download the latest R installer package (e.g., `R-4.4.0-arm64.pkg` for Apple Silicon or `R-4.4.0-x86_64.pkg` for Intel Macs).
3. Double-click the downloaded `.pkg` file to launch the installer.
4. Follow the on-screen instructions to complete the installation.
5. Once installed, R can be launched via the `R.app` in the `/Applications` folder.

### Windows Installation

1. Go to [for windows installation](#)
2. Download the latest R installer executable (e.g., `R-4.4.0-win.exe`).
3. Run the downloaded file and follow the installation wizard.
4. After installation, launch R from the Start Menu or desktop shortcut.



## 2.4 MATLAB

### macOS Installation

1. Go to [MathWorks MATLAB](#).
2. Sign in with your institutional or personal MathWorks account.
3. Download the latest MATLAB installer for macOS.
4. Launch the installer and follow the on-screen instructions.
5. During installation, select and install the Image Processing Toolbox.

### Windows Installation

1. Visit [MathWorks MATLAB](#).
2. Sign in and download the appropriate MATLAB installer for Windows.
3. Run the downloaded executable and follow the installation wizard.
4. Include the Image Processing Toolbox when selecting components.

### Linux Installation

1. Download the MATLAB Linux installer from your MathWorks account.
2. Extract the installer and run: `sudo ./install`
3. Follow the GUI-based installation steps.
4. Ensure that the Image Processing Toolbox is selected.

### Toolboxes

The MATLAB scripts used in the ASCT analysis pipeline require the Image Processing Toolbox, which provides functions for segmentation, region measurements, and object tracking. No other additional toolboxes are required.

### 3 Demo and instructions for use

The aim of this demo is to outline the core steps of the ASCT image analysis pipeline. The pipeline integrates four key tools: Fiji for image preprocessing, ilastik for segmentation and classification, MATLAB for single-cell tracking, and R for downstream quantitative analysis.

The demo dataset consists of a time-lapse sequence of a single field with 30 frames, captured in both brightfield and fluorescence channels. The workflow begins with image preprocessing and metadata extraction using a Groovy script in Fiji. This is followed by pixel classification in ilastik, which serves as input for both morphology and viability classification.

Next, object identities from the ilastik morphology classification are used in MATLAB for drift correction and single-cell tracking. Finally, the resulting tracking data are analyzed in R to generate time-resolved quantitative outputs, including time-kill curves and summary statistics. Each step in the pipeline produces intermediate files required for subsequent steps, culminating in per-cell tracking data suitable for downstream analysis. Note run time depends on the device used.

#### 3.1 ASCT Image Analysis Pipeline

##### Step 1: Fiji preprocessing

1. Open `Fiji`.
2. Drag and drop the script `ASCTimageAnalysis_1.Fijipreprocessing.groovy` onto Fiji.
3. When prompted, select the directory containing the raw image data.
4. Run the script.

##### Output:

- `*_BF.h5` – Brightfield time-lapse.
- `*_FLcorr.h5` – BaSiC background-corrected fluorescence.
- `*_Metadata.csv` – Extracted time metadata.

*Run time: 1 min.*

##### Step 2: ilstik Pixel Classification

1. Open `ASCTimageanalysis_2_ilastik_PC.ilp` in ilastik.
2. Select batch processing.
3. Load the `*_BF.h5` file as your raw input data.
4. Run the pipeline to generate pixel classification results.

##### Output:

- `PC-*.h5` – pixel probability map (cell / border / background).

*Run time: 5 min.*

**Step 3: ilastik Morphology Object Classification (MOC)**

1. Open `ASCTimageanalysis_3-ilastik_MOC.ilp` in ilastik.
2. Select batch processing.
3. Set `*_BF.h5` as the raw data input and `PC-*.h5` as the prediction map.
4. Run the classification to label cells based on morphology.

**Output:**

- `MOC-*Object_identities.h5` – identity map used for tracking.
- `MOC-*.csv` – table of morphology classifications.

*Run time: 2 min.*

**Step 4: ilastik Propidium iodide Object Classification (POC)**

The Propidium iodide Object Classification is the viability classification.

1. Open `ASCTimageanalysis_4-Ilastik_POC.ilp` in ilastik.
2. Select batch processing.
3. Use `*_FLcorr.h5` as the raw data and `PC-*.h5` as the prediction map.
4. Run the object classification.

**Output:**

- `POC-*.csv` – table of viability classifications (piPOS / dead and piNEG / live).

*Run time: 2 min.*

**Step 5: MATLAB Drift Correction**

1. Open MATLAB and the `ASCTimageanalysis_5-MATLAB-drift.m` script.
2. Update the variable `parentPath` to point to the parent directory containing the `Data/` folder.
3. Run the script.

**Output:**

- `Drift-*.csv` – table containing time-resolved positional drift estimates.

*Run time: 10 min.*

**Step 6: MATLAB Single-Cell Tracking**

1. Open MATLAB and the `ASCTimageanalysis_6-MATLAB-SingleCellTracking.m` script.
2. Update the variable `parentPath` to point to the parent directory containing the `Data/` folder.
3. Run the script.

**Output:**

- `RawTrackTable-*.csv` – initial unfiltered object linkages
- `CorrectedTrackTable-*.csv` – refined single-cell tracking table.
- `ScatterPlot-*.pdf` – area variation between frames.
- `TrackPopulationPlot-*.pdf` – tracking completeness over time.
- Subdirectory containing overlay images - with tracked cell IDs displayed on brightfield images across all timepoints.

*Run time: 5 min.*

### Step 7 (Optional): R Markdown per field Time-Kill Curve

An optional R Markdown script is provided to generate a time-kill curve based on the tracking and viability output from the image analysis pipeline. The script automatically detects output files (such as `CorrectedTrackTable_*.csv` and `POC-*.csv`) and processes data for each analysed field.

1. Open RStudio or your preferred R environment.
2. Load and run the script `ASCTdata_analysis_singlefield.Rmd`.
3. The script will:
  - Locate the relevant result folders from `1_ASCT_image_analysis/Data/`.
  - Merge metadata, tracking, and viability data.
  - Apply missing-value filtering and track smoothing logic.
  - Generate and save a time-kill curve as `_SingleFieldResults.plot.pdf` for each field.

#### Output:

- `*_SingleFieldResults.plot.pdf` – time-kill curve showing the fraction of viable cells over time for each analysed field.

*Run time: 1 min.*

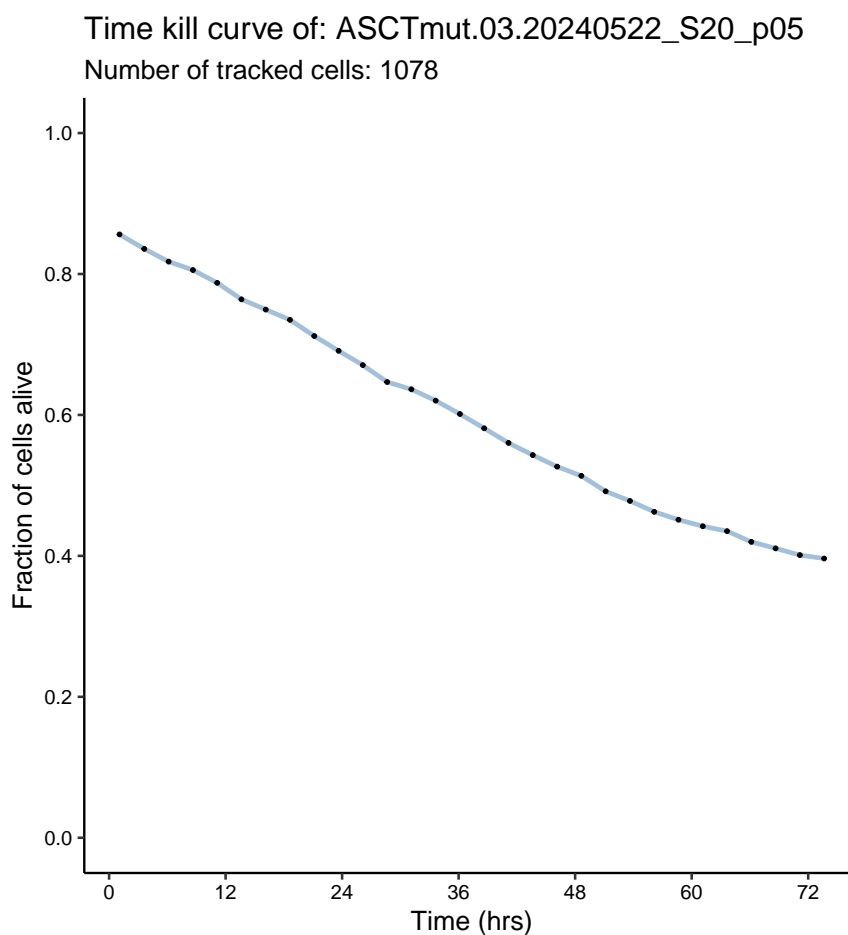


Figure 1: Representative output from the ASCT demo image: a raw time-kill curve of a single field, showing the fraction of cells alive over time and the total number of tracked cells.

## 3.2 ASCT per well data Analysis

### ASCT per well data analysis in R

The final step in the ASCT pipeline aggregates results across all nine imaging fields of a single well to derive time-resolved single-cell metrics. This analysis makes use of the output from previous drift correction and tracking steps, as well as the viability classification from ilastik.

To perform the analysis, open the R Markdown script `ASCT_data_analysis.Rmd` and run it directly. The script is designed to work with the expected folder structure and automatically locates the required files if organized correctly (see Figure 2). This analysis calculates the fraction of live cells over time using the viability maps, generates time-kill curves, estimates the area under these curves (AUC), and evaluates object-level changes in size (as a proxy for growth). These features are extracted per field and summarized at the well level.

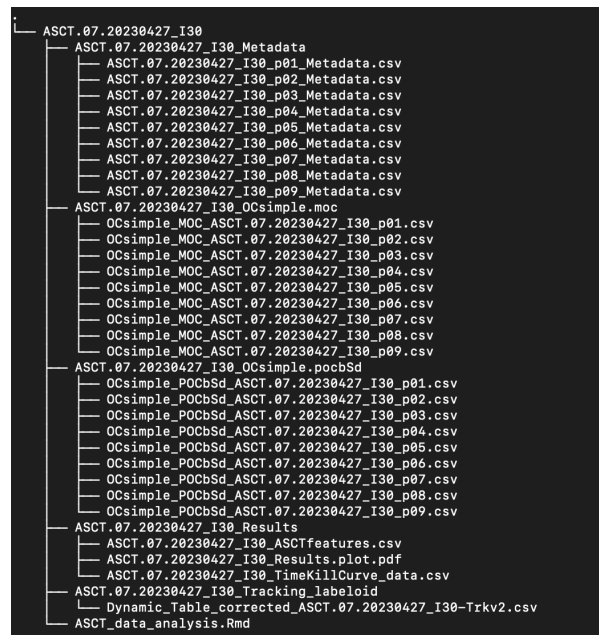


Figure 2: Directory layout for a single well (well I30), showing per-position metadata, morphology (moc), and viability (poc) outputs, along with downstream analysis results generated via R.

#### Output:

- `ASCTfeatures.csv` – summary features including AUC and growth dynamics
- `TimeKillCurve_data.csv` – fitted values and interpolated time-kill curve results
- `Results.plot.pdf` – visual summary of cell viability, growth, and tracked cell numbers

*Run time: under 2 min.*

The output shown in Figure 3 provides a compact visualization of viability dynamics, tracking quality, and cell growth behaviour over time.

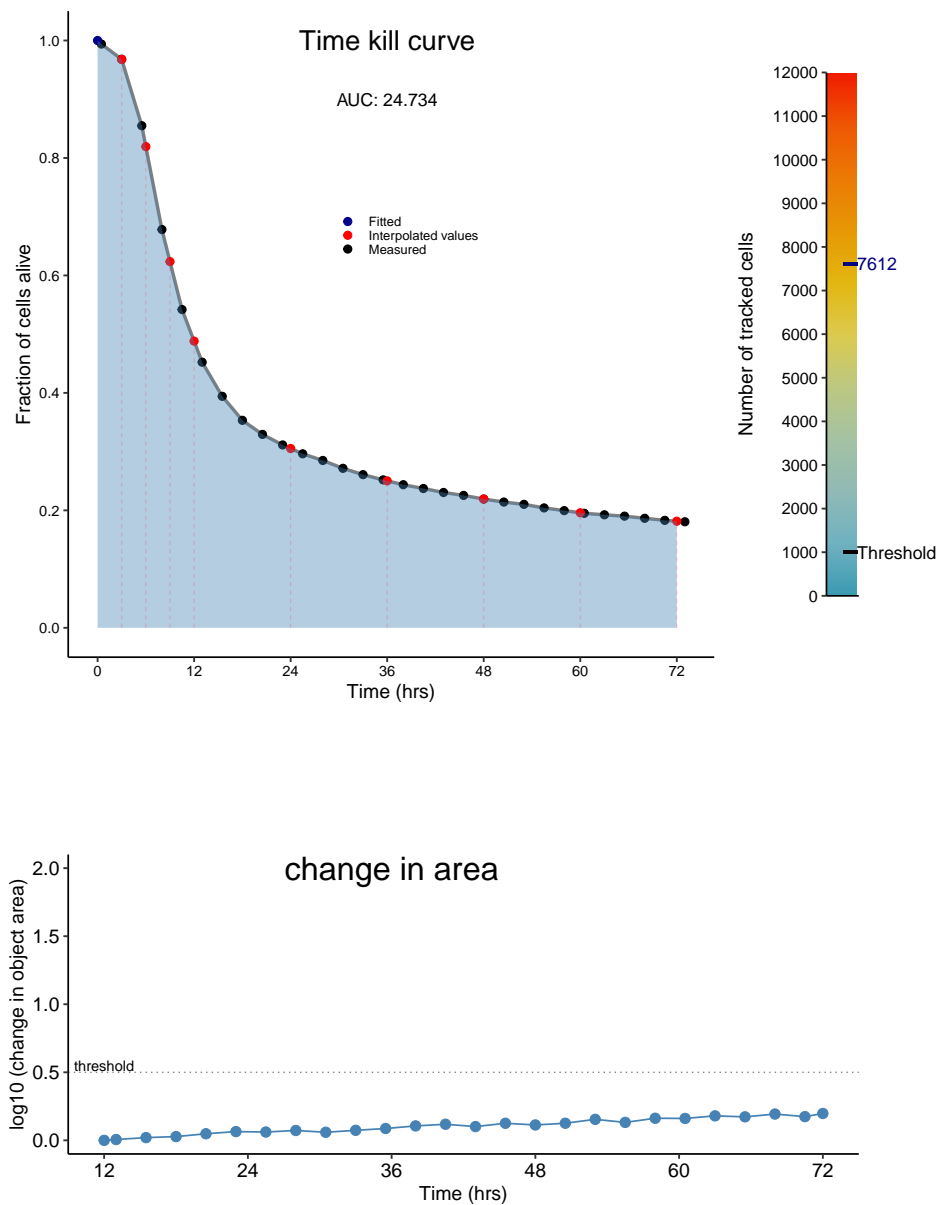


Figure 3: R-generated summary of ASCT data for well I30, including time-kill curves, number of tracked cells, and log-transformed object area changes.