Celldetective: an AI-enhanced image analysis tool for unraveling dynamic cell interactions

Rémy Torro[1,2*], Beatriz Díaz-Bello[1], Dalia El Arawi[1], Ksenija Dervanova[1], Lorna Ammer[1,3], Florian Dupuy[1], Patrick Chames[3], Kheya Sengupta[2], Laurent Limozin[1*]

[1]Aix-Marseille Univ, CNRS, INSERM, Turing Centre for Living systems, LAI, Marseille, France.
[2]Aix-Marseille Univ, CNRS, Turing Centre for Living systems, CINAM, Marseille, France.
[3]Aix-Marseille Univ, CNRS, INSERM, Institut Paoli-Calmettes, CRCM, Marseille, France.

*Corresponding author(s). E-mail(s): remy.torro@inserm.fr; laurent.limozin@inserm.fr;

Abstract

A current challenge in bioimaging for immunology and immunotherapy research lies in analyzing multimodal and multidimensional data that capture dynamic interactions between diverse cell populations. Here, we introduce Celldetective, an open-source Python-based software designed for high-performance, end-to-end analysis of image-based *in vitro* immune and immunotherapy assays. Purpose-built for multicondition, 2D multichannel time-lapse microscopy of mixed cell populations, Celldetective is optimized for the needs of immunology assays. The software seamlessly integrates AI-based segmentation, Bayesian tracking, and automated single-cell event detection, all within an intuitive graphical interface that supports interactive visualization, annotation, and training capabilities. We demonstrate its utility with original data on immune effector cell interactions with an activating surface, mediated by bispecific antibodies, and further showcase its potential for analyzing extensive sets of pairwise interactions in antibody-dependent cell cytotoxicity events.

Keywords: bioimage analysis, multimodal microscopy, artificial intelligence, immune cell interactions

Introduction

An overarching goal of post-genomic biomedical research is to decipher cell states and transitions as well as cell interactions in functional dynamic assays [1]. This requires high throughput and high resolution experimental methods supported by advanced and powerful analyses [2, 3]. The field of immunology presents specific challenges, related to complex cell states characterized by transient and heterogeneous receptor expression, highly motile behavior and immune function relying on dynamic cell-cell interactions [4, 5]. These features are largely out of reach in ensemble biochemical measurements [6] as well as low time-resolution single cell multi-omic data [7].

Microscopy-based *in-vitro* assays can bridge temporal and spatial scales, contributing to decipher lymphocyte activation from molecular to cellular scales [8–10]. Combining high content and high spatial resolution, imaging flow cytometry was applied to reconstruct disease progression on single immune cell populations [11] or to functionally characterize therapeutic antibodies against liquid tumours [12]. The anlysis is however limited to single cell or cell doublets. Multichannel fluorescence microscopy was successfully applied to image-based profiling of human T lymphocytes and Natural Killer cells (NK) [13] or therapeutic targets in drug discovery [14], but it provided limited dynamic information on the system. In some cases, nanowell grids were used to organize cell contacts and facilitate image analysis [15, 16] at the expense of imposing less physiological conditions.

High content, multi-channel spatiotemporal imaging of living cells provides rich, dynamic and multiscale information, from molecular to cell population scale [17]. In single-cell image sequences, time-dependent signals contain functional information, traditionally collected or treated manually in immunological settings [18]. Recent methods performed profiling on time dependent data for cellular phenotyping, but without measuring single events in time [19]. Functional dynamic information like calcium imaging can also be extracted for immune cell classification [20, 21] or to interpret the the brain's neuronal activity [22]. Most recent works on cell dynamics, specially in the context of cancer immunotherapy, involve specialized microscopy techniques and complex analysis protocols [23–25], sometimes using proprietary software [26, 27].

In the last years, machine learning and deep learning (DL) have revolutionized the field of image analysis [11, 12, 28–32]. A critical step in the context of cell images is segmentation, for which DL tools are highly successful [33–38]. However, they often rely on pretrained models for which the original dataset is not always accessible or can be too remote from users' data [39], limiting adaptability. For dynamic data, cell tracking is also crucial and can be performed by available tools [40–42]. Recent methods that integrate profiling with time-dependent information significantly enhance cell tracking performance [43–45]. However, current software associating the two tasks are mostly applied to the analysis of cell lineage [41, 46–48]. Most popular free bioimage analysis software like Fiji [49], Icy [50] or Cellprofiler [51] were not originally designed for DL methods [52], relying on extensions and plugins to bring

in these new methods, or were not conceived for time-dependent data [53]. Finally, end-to-end analysis tools accessible to experimentalists, adapted for both images and time series are currently lacking, preventing dataset curation and DL based event-detection. With the exception of MIA [54], existing partial solutions require coding skills [55] or proficiency in integrating/mastering disparate tools [56].

We introduce Celldetective, an open-source software designed to tackle the challenge of analyzing image-based *in vitro* immune and immunotherapy assays. It addresses the problem of cell interactions by extracting and interpreting single-cell behaviors for up to two populations of interest in a 2D co-culture, with a cell pair description linking the populations. The software adapts to varying imaging conditions and cell types, incorporating dataset curation and deep learning model training features directly within the graphical user interface (GUI). Celldetective organizes and automates complicated and time-consuming processes while providing a user-friendly, no-code GUI, making it accessible to experimental scientists without programming expertise. We tested the software on two applications in immunology and immunotherapy: i) measuring immune cell behavior in contact with an antigen-presenting surface using a label-free microscopy technique, and ii) assessing dynamic cell-cell interactions in a multiwell co-culture of immune and target cells, an established cytotoxic assay, using multicolor fluorescence microscopy.

Results

Software analysis pipeline

Celldetective is implemented as a Python package with a PyQt5-based GUI, available through the PyPi package repository and regularly updated. The source code can be accessed on GitHub (https://github.com/celldetective/celldetective), while all models, datasets, and demos are hosted in a public Zenodo repository (https://zenodo.org/records/13907294). Extensive documentation is available online (https://celldetective.readthedocs.io). The software is designed to run locally on a single computer equipped with either a graphics processing unit (GPU) or multiple processors. Image processing is conducted frame by frame with garbage collection to significantly reduce memory requirements, allowing it to operate on most laboratory computers. Celldetective is compatible with multi-channel time-lapse microscopy image stack files in TIF format, which are typically generated by acquisition software such as Micro-Manager [57].

As illustrated in Fig. 1, the software input consists of an experimental project that organizes raw data into a two-level folder structure emulating a multiwell plate, commonly used in *in vitro* immune assays (startup window in Fig. S6A). At the top level, folders represent wells (or biological conditions), while at the second level folders represent positions (a field of view associated with the biological condition). Experiment metadata – including channel names and order, spatiotemporal calibration, and biological condition labels – are stored in a configuration file created during project setup (Fig. S6B). This structure organizes the experimental data by biological conditions and field of views, facilitating the processing of large data volumes.

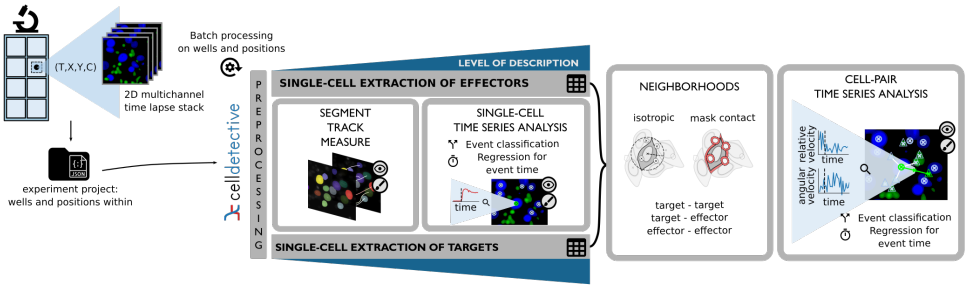


Fig. 1 Software Overview: an end-to-end GUI pipeline for studying interactions between target and effector cell pairs (from left to right). After loading an experiment project that mimics a multiwell plate structure, the user can apply preprocessing steps to the 2D time lapse microscopy images before segmentation. Target and effector cells are then segmented, tracked, and measured independently. Events are detected from the resulting time series, and the co-culture images are distilled into tables of single-cell measurements. The neighborhood module links cells in spatial proximity, and the cell-pair signal analysis framework facilitates the investigation of interactions between cell pairs. Eye and brush icons indicate steps where visual control and corrections are possible, with an appropriate viewer.
Fig. S6. Main GUI windows.
Fig. S7. Processing modules to extract single cells.
Fig. S8. Background correction methods.
Tab. S2. Generalist deep learning segmentation models.

Upon loading a project, a modular interface organizes all possible user actions (Fig. S6C with single-cell extraction modules illustrated in Fig. S7). Image preprocessing enables users to transform images prior to segmentation (Fig. S8). Two processing blocks drive the single-cell extraction for the effector and target populations, independently. Population-specific segmentation, optional tracking and measurements, reduce the image stacks into tables, where each row represents a single cell at a specific time point. With tracking information available, single-cell measurements are structured and interpreted as time series to detect events. Alternatively, users can classify cells at each time point from their measurements. A viewer is integrated at each step for quality control, error correction, or data annotation. The final module addresses cell interactions within or across populations, enabling users to describe and annotate cell pairs. Throughout the process, users can plot and compare single-cell or cell pair measurement distributions via a table exploration module designed for manipulating tracks. Additional analysis modules allow to perform survival studies and represent synchronized time series over cell populations.

Rather than reinventing efficient existing solutions, we incorporated established open-source deep learning segmentation methods StarDist [35] and Cellpose [36] (see model list in Tab. S2)—alongside the Bayesian tracker bTrack [41] into Celldetective. The software provides intuitive graphical options for tuning input and control parameters of these algorithms. The diverse software functionalities are listed and compared to existing software in Table 1. Notably, Celldetective stands out by offering training and transfer options in segmentation, enabling detection of a population of

interest within a mixed population, as well as time series annotation and survival analysis, not currently integrated in any other software. These features will be illustrated in the two biological applications described after.

Table 1 Comparative table of software functionalities with a selection of available solutions. A ✓is attributed if the task can be carried without coding. The use of an integrated solution or plugin is indicated in parentheses.

Software feature	ImageJ/Fiji	CellProfiler	CellACDC	MIA	Celldetective
Ref.	[49]	[51]	[47]	[54]	this work
Traditional segmentation	✓	✓	✓	✗	✓
DL segmentation	✓	✓	✓	✓	✓
Corrections, annotations	✓ (Labkit)	✓ (GIMP)	✓	✓	✓ (napari)
Training	✗	✗	✗	✓	✓
Transfer	✗	✗	✗	✓	✓
Tracking	✓ (TrackMate)	✓	✓	✓	✓ (bTrack)
Track Visualization	✓ (TrackMate)	✗	✓	✓	✓ (napari)
Classification from measurements	✗	✓	✗	✗	✓
Time series annotations	✗	✗	✓* *cell-cycle	✗	✓
Interaction analysis	✗	✓	✗	✗	✓
Experiment manager	✗	✗	✓	✗	✓
Multi-condition data exploration	✗	✓ (CP Analyst)	✗	✗	✓
Survival analysis	✗	✗	✗	✗	✓

High-throughput and time-resolved analysis of cell adhesion and spreading responses

In this first application, we investigated the dynamics of human primary Natural Killer (NK) immune cells as they sedimented and spread on cancer-cell-mimicking surfaces. Such assays offer rich spatiotemporal insights into interactions between lymphocytes and surrogate antigen-presenting cells [10, 58, 59]. The bispecific antibody (bsAb) studied here, with potential therapeutic applications, bridges the NK cell's activating receptor CD16 and the surface-bound antigen Human Epidermal Growth Factor Receptor 2 (HER-2), a breast cancer marker. Our objective was to evaluate the bsAb's

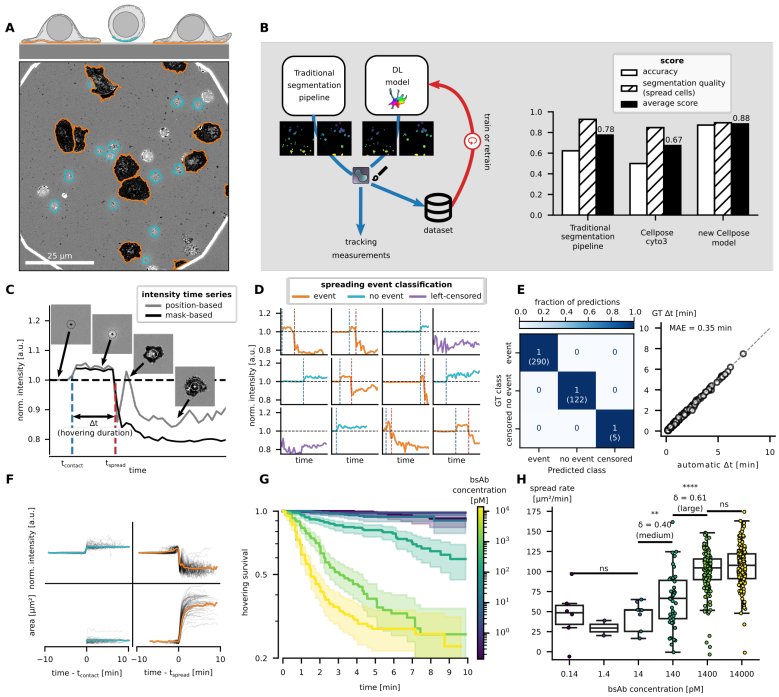


Fig. 2 Functional response of immune cells in a spreading assay. A) Schematic (top) and snapshot (bottom) of a spreading assay imaged by time-lapse RICM. Primary NK cells sediment on an HER2 coated surface in the presence of bispecific anti-CD16 × HER2 antibodies. Cells touch the surface in a desynchronized manner and may spread after a stochastic hovering duration. The contours of the cells are represented in orange for cells classified as spread on the surface and blue for hovering cells (corresponding colors are also used in the schematic). B) A traditional segmentation pipeline is used to generate a first instance segmentation of the cells imaged in RICM. Cell masks can be manually curated before training a new Cellpose [36] model starting from scratch or from an existing generalist model. Barplot comparing segmentation scores obtained for three different methods that can be called within the software. With the improved results yielded by the new model, the user can proceed to tracking and measurements. C) Intensity time series for a cell performing a contact and spreading sequence (see text for details). D) Single-cell intensity time series color-coded for three classes: 1) the cell spreads during the movie ('event', orange), 2) the cell is not observed to spread during the movie ('no-event', green) and 3) the cell is already spread at the beginning of the movie ('left-censored', purple). E) Benchmarking of cell class automatic determination reported by a confusion matrix, and comparison of spreading times determined automatically. F) Individual cell time series (tonal, morphological, etc) are synchronized with respect to a characteristic time to measure the average population response at the event time. G,H) Functional response for the cell population as a function of bsAb concentration: hovering survival or distribution of cell decision time before spreading (G), spreading rate (H).
Fig. S9. Designing a traditional segmentation pipeline.
Fig. S10. Segmentation correction and annotation.
Fig. S11. Overview of segmentation strategies.
Video: ricm_events.mp4

effectiveness in promoting cell spreading, used here as a proxy for antibody-mediated

cell-cell interaction [58, 60]. We imaged cell behavior at the surface interface using reflection interference contrast microscopy (RICM), a label-free technique sensitive to submicrometer distances between the cell membrane and the surface [61] (Fig. 2A).

Cell segmentation in RICM images is challenging because the cells can be locally brighter or darker than the background, causing direct thresholding techniques to fail [61]. To address this, previous approaches have adopted special techniques, such as background correction and applying a variance filter before thresholding [10, 62, 63], which, however was often inadequate and required expert intervention. Our goal here was to train a DL model to segment lymphocytes spreading or hovering on the surface, without need for expert corrections. The complete strategy – background subtraction (Supplementary Fig. S8B), generating an initial set of labels with the traditional segmentation pipeline (Supplementary Fig. S9), correcting errors in the viewer to build a training dataset (Supplementary Fig. S10), and training a DL model from this dataset – is illustrated in Figure 2A–B. Moving away from this application, Celldetective provides a whole ecosystem to perform segmentation and adapt deep learning methods (see Supplementary Fig. S11). Here, we trained a new Cellpose model on the RICM dataset (both the dataset and model are available on Zenodo). We benchmarked this model against the traditional segmentation pipeline and the *cyto3* generalist Cellpose model [37] using a test set, quantifying lymphocyte detection accuracy and segmentation quality separately (see Materials and Methods). Hovering cells, which appeared as bright blobs with interference patterns and had an ill-defined area in RICM, were excluded from the segmentation quality score. Our model achieved a higher detection accuracy (0.86), compared to the traditional pipeline (0.62) or the generalist model (0.5). The traditional pipeline tended to produce many false-positive detections, while the generalist model struggled to simultaneously detect spread and hovering cells. Segmentation quality, however, was higher for the pipeline (0.93) than for our model (0.89) or the generalist model (0.88). Since most spread cell labels in the test set required little manual correction (except for contiguous cells), we were not surprised to observe the highest segmentation quality with the pipeline method. The generalist model segmented spread cells well when it could detect them. Overall, the new Cellpose model achieved the best average performance across both tasks (0.88), making it the most effective approach for segmenting cells in this assay.

Next, tracking of cells — which move while hovering and are quasi-stationary while spreading — was performed using a versatile motion model (see Materials and Methods), without feature inputs to prevent sharp morpho-tonal transitions from disrupting tracking. We enabled track extrapolation from the beginning of the movies, allowing intensity measurements to start before the cells reached the surface (Figure 2C-D). With track post-processing, we could measure intensities prior to cell arrival, with a position-centered average intensity measurement, in a user-defined disk (position-based), which compared well with the average over the cell mask, when available (mask-based). Notably, the position-based intensity showed a spike during the spreading phase, which was not observed in the mask-based intensity.

We defined two characteristic event times (Fig. 2C) : contact time (t_{contact})—the moment when the cell first appeared in RICM, with an average intensity greater than 1; and the onset of spreading (t_{spread}) when the average intensity dropped below 1.

The hovering time ($t_{\text{spread}} - t_{\text{contact}}$), for spreading cells, is the time required to make a spreading decision. t_{contact} is automatically detected as the time of first segmentation. For t_{spread}, spreading events were identified through condition-based classification (mask-based RICM intensity < 1) and an irreversible event hypothesis (see Materials and Methods). We benchmarked classification performance and hovering duration estimates against a manually annotated test set (Fig. 2D). The set was annotated with a dedicated viewer (see video ricm_events.mp4). A confusion matrix indicates perfect agreement between automatic classification and ground truth, while a correlation plot comparing estimated hovering durations to the ground truth shows excellent agreement, with a mean absolute error of 0.35 minute (Fig. 2E).

After automatically extracting the event times, we examined average population response based on single-cell time series, by synchronizing each single-cell measurement to a specific event time to calculate the mean response across the population (Figure 2F). Next, we tested whether varying the concentration of bsAb on the surface impacts spreading. First, we estimated the "hovering survival" by plotting fraction of cells hovering at a given time (Figure 2G). We observe a higher fraction of spreading cells and shorter spreading decision times with increasing bsAb concentrations. The acceleration of the spreading decision time was particularly apparent in the 10-1000 pM bsAb concentration range, saturating beyond 100 pM. Next, Using the table exploration feature in Celldetective, we differentiated the area with respect to time for each cell track, extracting the spreading rate at the onset of spreading (t_{spread}), which increases in the 14-1400 pM range (Figure 2H). Overall, determining event times for individual cells, measured at high throughput, highlights Celldetective capabilities in evaluating cell-surface interactions.

Cytotoxic response in a co-culture with overlapping channels

In the second application, we studied antibody-dependent cell cytotoxicity (ADCC) in a high-density co-culture of target tumor cells (MCF-7) and effector cells (primary NK) for various biological conditions (concentration of bsAb anti-CD16 $\times$ HER2, tumor cell HER2 expression). A multi-color composite representation of a single position is shown at endpoints in Figure 3A, illustrating how the number of dead (red) nuclei increases over time.

First, we studied the lysis behavior of target cells over time, which entailed achieving a single-cell description for the target cells. Directly using generalist nuclear segmentation models (StarDist *versatile fluorescence* and Cellpose *nuclei* models) on the Hoechst channel, we could only achieve a poor accuracy, limited by effector cell screening of the underlying target cells. While it was possible to segment all cells and classify them either simultaneously [64] or in two steps [65], label contamination from overlapping populations remained an issue. We produced expert annotations of the MCF-7 nuclei using all available multi-channel and time information. Since the MCF-7 nuclei are larger than primary NK cell nuclei, we expected to be able to train an efficient model detecting the MCF-7 nuclei from the Hoechst channel only, failing only on blurry images, where the size separation was no longer obvious (Fig. 3B). We used Celldetective to retrain the StarDist *versatile fluorescence* model on our dataset using the Hoechst channel as its input and considerably improved the

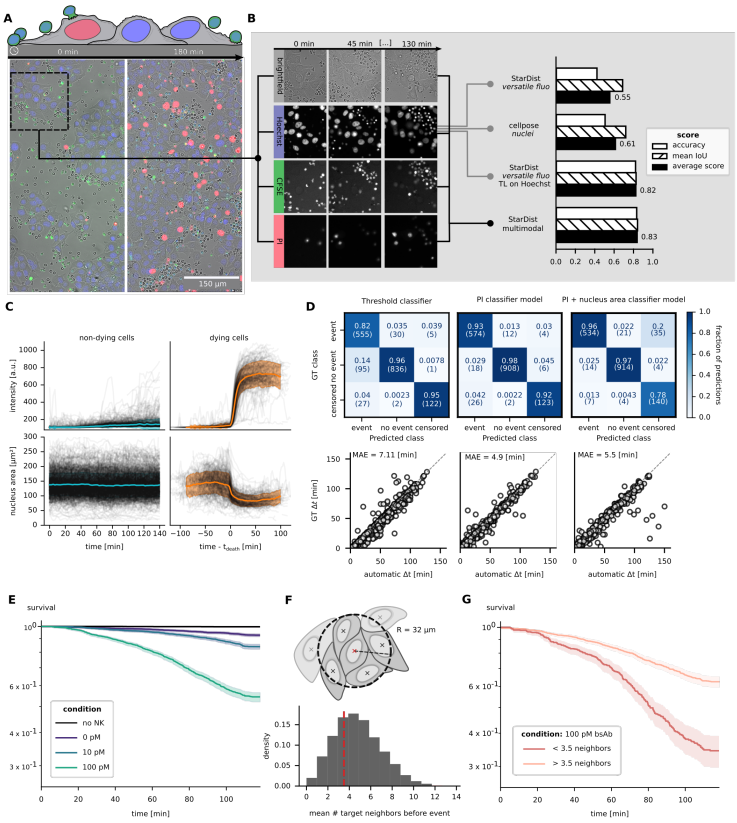


Fig. 3 High throughput cytotoxic response of cancer cells in co-culture with immune cells. A. Schematics side-view of target/NK cells co-culture assay for bispecific ADCC (top) and representative multimodal composite images, obtained at two different time points, with target nuclei labelled in blue, dying cells in red and NK cells in green (bottom). Corresponding colors are also used in the schematics. B. Decomposition of partly overlapping fluorescence channels and benchmark of segmentation DL models. Brightfield and fluorescence images at three different time points. Hoechst channel (target nucleus) is taken as input to the existing StarDist *versatile fluo* or Cellpose *nuclei* models (available directly in Celldetective). Two new models trained in Celldetective were benchmarked: a transfer of the StarDist *versatile fluo* model on our MCF-7 nuclei dataset, with Hoechst as input, and a StarDist multimodal modeln trained from scratch using 4 channels as its input. C. Time series of nuclear fluorescence intensities and nucleus apparent area for a set of non-dying target cells (left) and of dying cells (right); the reference time t_0 is the death time for the dying cells. D. Benchmark of three methods for event classification and regression (Threshold method on PI, DL classifier on PI, DL classifier on PI and nuclear area). Top row: confusion matrices with 3 classes (fraction of predictions displayed). Bottom row: correlation plots. E. Survival curves of target cells without NK cells or in the presence of NK cells for different bsAb concentrations. F. Schematics for neighbors counting and histogram of target neighbors count. G. Survival curves for two subpopulations of targets at 100 pM bsAb, splitted as a function of the local target cell density.
Fig. S12. Event detection model architectures.
Fig. S13. Neighbor counting methods.
Tab. S3. MCF7 nuclei segmentation models.
Video: adcc_rgb.mp4

performance, from 0.55 to 0.82. Last, we trained a completely multi-channel model on all four available channels to detect the MCF-7 nuclei and achieve a slightly higher performance (0.83). All of the models benchmarked are accessible directly in Celldetective (see Supplementary Tab. S2 and S3). Target cell tracking is detailed in Materials and Methods.

An expert annotated death events defined by a sharp increase in PI signal at t_{death}. Single-cell time series were then grouped by event class and synchronized to t_{death} (Figure 3C). The average PI intensity for cells undergoing death shows a distinctive step-like transition with low lateral variance, validating the annotation accuracy (Fig 3C, top-right). The PI signal in live cells, by contrast, rises slightly over time but remains well below the levels observed in dead cells (Fig 3C, top-left). Notably, the apparent nuclear area of dying cells contracts sharply at t_{death} (Fig 3C, bottom-right), while the nuclear area of non-dying cells remains constant (Fig 3C, bottom-left).

Event detection was evaluated for three methods on this dataset: (1) a condition-based classification using the irreversible event hypothesis on the PI intensity time series, (2) a DL model trained to detect death events from PI alone, and (3) a DL model trained to detect events using both the PI intensity and nuclear area time series. Detailed information on the training procedures for these models is provided in the Materials and Methods section and the DL architecture on Fig. S12. Figure 3D presents the confusion matrices for each method, showing agreement between predicted and ground-truth event classes, along with a correlation plot evaluating the accuracy of death time extraction. Method 1 incorrectly predicted about 15 % of the events as death events when none had occurred. This method also had the highest mean absolute error (MAE) for death time estimates (7.1 minutes). Method 2 performed better, with only 3 % of its predicted death events corresponding to an absence of events. The multivariate model (Method 3), incorporating both PI and nuclear area data, achieved the highest precision in death event detection (96 % of predicted death events were true events). However, this model showed a reduced accuracy in identifying cells as already dead, with 20 % of these cells (2 % of the total number of cells) observed to die later. As before, and in practice, quality check and minor manual corrections were performed with the event annotation viewer, using this time a multichannel composite of the images (Video adcc_rgb.mp4).

We next investigated whether the probability of tumor cell lysis depended on bsAb concentration. Using $t = 0$ min as the synchronization time, we excluded cells already dead at the start (always less than 5%) and disregarded cells born from mitosis (during tracking post-processing). We represented the MCF-7 survival curve using t_{death} as the event time (Figure 3E), which shows an increase in the fraction of dead MCF-7 cells (endpoint) occurring at a faster rate with rising bsAb concentrations.

Our single-cell approach permits also to link cell survival with local context. To do this, we calculated a target-target neighborhood within a 32 μm radius (isotropic method, see Materials and Methods and FIg. Fig. S13) to count the local density experienced by each target cell until death (Figure 3F). We then applied a filter in Celldetective to compare survival outcomes for cells in low-density versus high-density environments (Fig 3G), observing a significant survival advantage for targets in high-density regions.

From a single time point to a dynamic readout of effector-target interactions

After describing the ADCC assay from the targets' perspective, we shifted our focus to the effectors. Effector cells were segmented using a custom multichannel Cellpose model that combines brightfield, Hoechst, and CFSE, a cytoplasmic fluorescence label (see Supplementary Tab. S4). Unlike targets, effectors exhibit high motility and can form dense clusters in the assay, limiting reliable tracking.

When effector tracking was unfeasible (e.g., at an effector-to-target ratio of 5:1 and frame interval greater than 2 minutes), we could still perform a multicondition analysis of the cells at each time point. We aimed to measure the signal of antibodies tagged with a single ATTO 647 dye. Specifically, we compared the CE4-21 bsAb with the CE4-X control antibody, which only binds to the HER2 antigen on targets, across two target cell lines with distinct antigen expression levels (WT and HER2+) [66]. Figure 4 presents the mean bsAb fluorescence intensity under effector cell masks at 1 hour, broken down by target cell type, antibody, and contact state (in or off contact with targets, see Materials and Methods and video single-timepoint-contact.mp4). For WT targets (lowest HER2 expression), the cells exhibited a signal in the CE4-21 bsAb condition, regardless of the contact state (Figure 4A). In contrast, no signal was measured with the CE4-X antibody condition, supporting that the bsAb-condition signal came from the effector cells (visual assessment in Figure 4B). With the HER2+ targets and CE4-X antibody, the signal measured off-contact was negligible, confirming that the antibody did not bind to the effectors (Fig 4A-B). The signal measured in contact did not originate from the effector cells but from the target cells below. With the CE4-21 bsAb, on the other hand, the signal measured in contact was a sum of contributions from both the target and effector cells.

Based on the measured signals, we sought to determine whether the added effector signal from the CE4-21/off-contact condition (without target contribution) and the effector signal from the CE4-X/in-contact condition (without effector contribution) could approximate the CE4-21/in-contact signal. Using the distributions generated by the software, we simulated the distribution of this summed signal with a Monte-Carlo approach, and compared it with the CE4-21/in-contact condition. Figure 4C reveals that the simulated signal distribution was significantly lower than that of the CE4-21/in-contact condition, with a small effect size (Cliff's Delta > 0.33). By calculating this effect size at each time point, we further confirmed the robustness of this finding, as the effect consistently maintained the same direction, with a medium effect size observed more often than not (Suppl. Fig. S14).

If effector tracking is achievable (e.g., at an effector-to-target ratio of 1:1 and frame interval lower than 1 minute), we can explore the measurement time series. We illustrate this software feature by adding a degranulation marker (fluorescent anti-LAMP1) to monitor the lytic activity of the effector cells in the assay. In this experiment, we compared two bsAbs, CE4-21 and CE4-28, formed of the same CE4 nanobody against HER2 and two different nanobodies against CD16 on the effector side.

First, we investigated whether LAMP1 expression was influenced by biological conditions, mirroring our approach with fluorescent antibodies. For each target cell

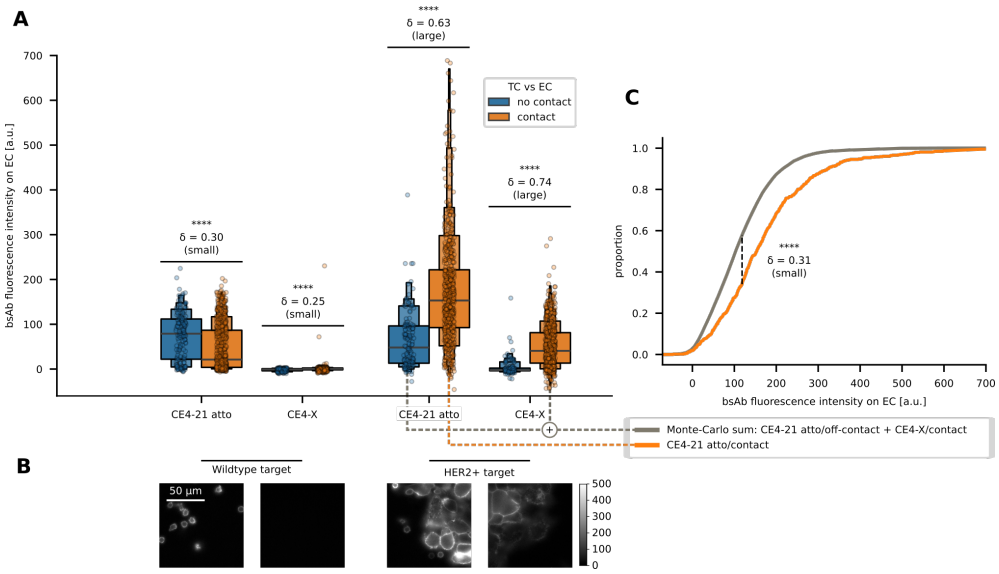


Fig. 4 Single time point analysis of effector-target interactions. In the absence of effector tracking, fluorescent bsAbs were used to study the distribution of Ab on effectors and targets. A. bsAb intensity measured on effector cells on targets with high or low antigen expression. bsAb CE4-21 binds to both HER2 and CD16, while CE4-X control antibody binds only to the HER2 antigen on target cells. B. Representative snapshot of fluorescence channel for each condition of A. C. Empirical cumulative distribution functions (ECDFs) for the simulated (gray, sum of CE4-21/no contact + CE4-X/contact) and observed (orange, CE4-21/contact) distributions. The dotted black line represents the Kolmogorov-Smirnov statistic, *i.e.* the maximum separation between the two ECDFs.
Tab. S4. Primary NK segmentation models.
Fig. S14. Effect size at each time point.
Video: single-timepoint-contact.mp4

type and antibody condition, we reported the fraction of LAMP1-positive effector, decomposed by contact state, across all time points (Fig. 5A). The results indicated that, except for the no-antibody controls, the fraction of LAMP1-positive effector cells was consistently higher in-contact with target cells.

From this time-integrated approach, we shifted our focus to the dynamics of the effector cells, specifically investigating whether their motility was influenced by target cell proximity. Given that velocity estimators can be noisy, we computed a time-averaged velocity estimate for each cell track, decomposed by contact state. The resulting velocities for all conditions are presented in Figure 5B. In each condition, the contact velocities were consistently lower than off-contact, regardless of antibody presence or type. This difference was always significant, with a small effect size for WT targets (Cliff's Delta > 0.147) and a large effect size for HER2+ targets (Cliff's Delta > 0.474). The lowest contact velocities were observed in the presence of antibodies, showing a significant drop from the target-specific control, with medium effect sizes for the WT bsAbs and HER2+ CE4-28, and a large effect size for HER2+ CE4-21. However, we did not detect any difference between the two antibodies, separately for each cell type.

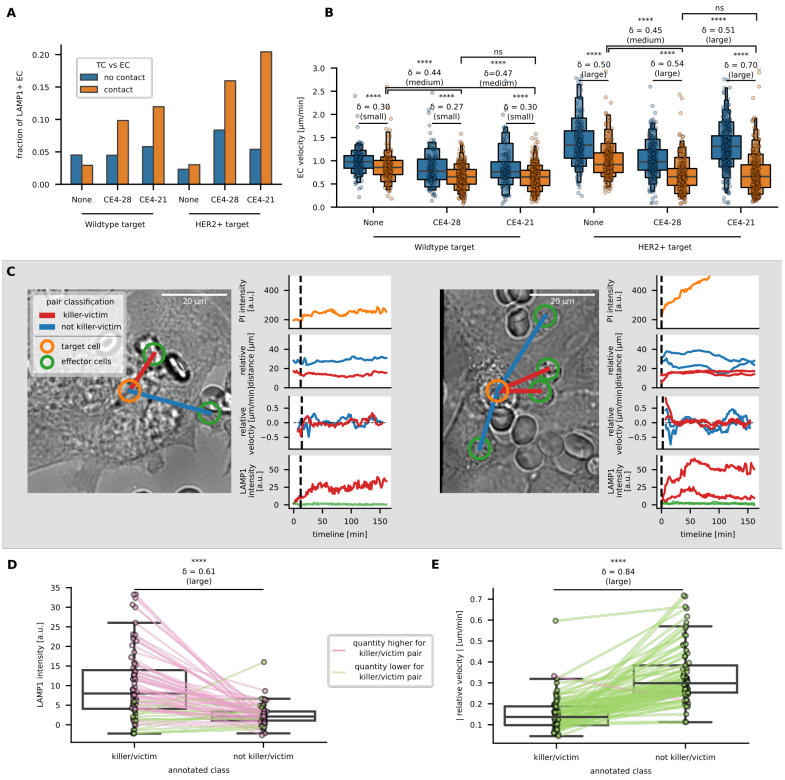


Fig. 5 Effector-target dynamic interactions. A. Fraction of LAMP1-positive effector cells for different bsAb conditions, decomposed by contact with the targets (over all time points). B. Average effector velocity for different bsAb conditions, decomposed by target contact (up to two estimates per effector). C. Examples of killer-victim identification with one (left) or two (right) killers. One target cell (center) is connected to effector neighbors via colored segments. Dynamic effector-target interaction monitoring includes PI signal of target cell, relative distance and relative velocity between target and effector, as well as LAMP1 signal in effector neighbors. D, E. Characteristic parameters for killer/victim and non-killer/victim pairs. Effector cells are manually annotated as potential killer or not. Each pair of points represents one victim target, with the average over all neighbors of the parameter decomposed by killer class. D. LAMP1 intensity. E. Relative effector-target velocity.
Video: cell_interactions.mp4

After computing effector-target neighborhoods, we measured features of cell pairs, including relative distance and angle, as well as their time derivatives. Figure 5C illustrates time series data for pairs, showing the PI intensity of a target cell that died at 5 minutes alongside the relative distance, velocity, and LAMP1 signal of neighboring effectors. Notably, only the selected neighbor (highlighted in red) exhibits a strong LAMP1 signal, which rises shortly after the target cell's death and remains elevated until the observation concludes.

Using the pair annotation feature, we aimed to identify "killer-victim" pairs. An expert annotated each neighborhood containing a dead target cell, identifying the

most probable effector cell killer to establish these pairs. This annotation was done via a cell pair viewer, similar to the viewer designed for single cells, which leveraged color cues for pre-determined target cell death events and immobilized effector cells (Video cell_interactions.mp4). The expert viewed only the brightfield movie, blinded to the LAMP1 signal, and identified effectors with prolonged, restricted movement in contact with the target cell. Effectors that arrived post-mortem were excluded, and in ambiguous cases, multiple pairs could be labeled as "killer-victim" within a neighborhood. Due to the complexity of this task, precise event times were not annotated. Figures 5D-E display average relative velocity and LAMP1 expression for each neighborhood, with values grouped by killer-victim status and shown in paired boxplots. Results indicate significantly higher LAMP1 expression and lower relative velocity in killer-victim pairs, with both metrics showing a large effect size (Cliff's Delta > 0.474).

Discussion

Here we developed a new software to study multichannel time-lapse microscopy data at scales ranging from single-cells to cell populations. It integrates original methods for event detection and cell-pair analysis with established methods for cell segmentation and tracking [35, 36, 41]. Although the pipeline steps are primarily sequential (see Fig. 1, S7) and could be performed independently with different tools and scripts, we demonstrate how integrating them within a unified, interactive software environment enhances data exploration and facilitates the analysis of dynamic cell interactions in large datasets.

Celldetective demonstrated its capacity to manage datasets of hundreds of gigabytes, generating high-quality plots that include hundreds of single-cell traces and data points. As a standalone application, it empowers biology experts to analyze single-cell data, annotate labels and events, and train models to automate tasks without requiring coding skills—similar to recent advances for fixed images [54]. Celldetective integrates the open-source annotation tool napari for label correction and introduces unique tools for event annotation and pair classification. The software combines traditional threshold-based image and signal analysis methods with deep learning-based approaches, which can be used complementarily for enhanced results.

We showed how recording multivariate time series may reveal multiple phenomena occuring during the same event. For example, cell death, monitored through an increase in PI fluorescence, was observed alongside apparent nuclear shrinkage (see Fig. 3C). We interpret this as cell de-adhesion upon death, releasing the pushing pressure on the nucleus and allowing its axial shape relaxation [67]. In immunotherapy assays involving human primary cells, we prioritized a fluorescence-based labeling strategy that avoids genetic modification. Due to the exchange of fluorescent dyes between cell populations and cell lysis during the assay, along with significant size disparities between effector and target cells, generalist deep learning segmentation models perform poorly. To address this, we trained custom models within the software, using StarDist for target nuclei segmentation (convex object) and Cellpose for effector cells (any shape). Performance improved slightly with the simultaneous use of multiple

input channels (see Fig. 3B). Looking up close, the improvement was more noticeable on difficult data (blurry, noisy). This approach also benefited event detection: training a model on both PI and nuclear size time series enhanced the event classification performance on the two most important classes for survival studies (see Fig. 3D). Our spatially resolved immunotherapy assay documents the effect of heterogeneity in target cell density. Indeed, we observed how the high density of target cells reduces their ADCC killing rate, suggesting an apparent protective effect of close target neighbors (see Fig. 3G).

Immune cells leverage transient and weak contacts to take rapid decisions [9]. We demonstrate the power of high throughput imaging and analysis to explore such mechanisms. In a cell-surface assay, we measured by interference microscopy the decision time for NK cells to spread on an antigen surface in the presence of bispecific antibody. Celldetective provides a higher statistical power than in earlier works [59]. Thus, we observed differences in NK response to bispecific antibodies concentration in the minute range, which could not be resolved yet for other systems like TCR-pMHC [68]. Here, the detection of sharp cell morphology changes during adhesion and spreading could be further enhanced by combining RICM with brightfield imaging, since each imaging modality exhibit a contrast specific to each cell state [58].

Proximity between effector and target provides a spatial gating out of reach for traditional flow cytometry, showing some clear differences in LAMP1 signal and relative velocity in neighboorhoods of target. We introduce a classification scheme, which allows quantification of a subpopulation based on arbitrary thresholds on any measurement. Additionally, instantaneous classification can be complemented by a time series analysis, with tracks available. To go further into the analysis of effector-target dynamic interactions, we addressed the question of killer:victim pair identification. The strategy consisted in collecting relative measurements like velocity and manually annotating the potential killer for each target from the images. Our data are consistent with one or several killers. The most prominent parameters for killer identification are relative velocity and LAMP1 signal. Since cell-pairs were implemented like single-cells, Celldetective gives the possibility to train an event detection model from pair, target and effector time series to automate killer:victim pair identification.

As a perspective, further work is required to analyze automatically the dynamic signaling and provide more robust prediction. Celldetective offers in principle the complete environment for such an advanced analysis, as demonstrated with the death time automatic detection.

For its applicability, Celldetective does not require coding skills. It was tested by graduate students from our labs with diverse backgrounds in biophysics, cell biology and immunology. Its release is accompanied by examples and a complete documentation. Software maintenance and user assistance are offered via github platform and a professional development team (https://github.com/centuri-engineering). One highlight of Celldetective is to offer alternative methods, in particular for cell segmentation, so that the user can maximize the outcome. In order to guide the user for software applicability and workflow, we propose a decision tree,

visible as question marks in the interface, to assist in choosing between the different options.

The experimental systems analyzed here are largely two-dimensional, but at the cellular scale, exhibit a three-dimensional aspect. Future software expansions could incorporate 2.5D or 3D sequence analysis. Currently, to our knowledge, only proprietary solutions are offered and exploited in immunotherapy assays [26]. Notably, cell-interaction studies benefit from higher time resolution, which may limit compatibility with extensive 3D analysis.

The methodologies we demonstrate here should adapt well to commercial high-throughput automated microscopes for long-term cell co-culture studies [39], for which no efficient analysis solutions are available, to our knowledge. A distinctive aspect of our approach is the use of compact, efficient models that are user-trainable on a single GPU setup. This contrasts with approaches relying on large-scale databases [39] and extensive DL models [69]. Moreover, the software's integrated visualization and annotation features encourage meticulous review of segmentation and tracking results, enhancing confidence in the analysis.

Convolutional models used by the software, such as U-Net and ResNet, are prone to forgetting previously learned tasks upon transfer [70]. To address this, Celldetective allows the addition of the original dataset during transfer from a model we trained. Future extensions could include foundation models like SAM [71] for broader adaptability.

Conclusion

In conclusion, Celldetective is a user-friendly software designed for non-specialists, providing advanced cell-scale analysis for large cell populations. Combining high time-resolution with robust statistical capabilities, we show the potential to uncover new fundamental parameters characterizing dynamic interactions between immune and tumor cells. We demonstrated the efficiency of Celldetective through two applications in the context of immunotherapy by isolating time-dependent cell-scale parameters and paired-interactions, in high cell-density environment. We believe that such approach holds great potential for enhancing the design and discovery of drugs in a more powerful and rational manner, while also finding broader applications in fundamental studies of cell-cell interactions in developmental biology, microbiology or neurosciences.

Materials and Methods

Software and hardware

The software, implemented as a Python package with a PyQt5 graphical user interface (GUI), incorporates Material Design for styling and integrates Matplotlib canvases for displaying images, animations, and plots. Each processing module—segmentation, tracking, measurement, event detection, and pair analysis—operates on a single movie at a time within an independent subprocess. Upon completion, CPU and GPU memory are fully released, enabling seamless iteration on the next movie. Where applicable, multi-threading has been implemented and can be customized within the

software. Each module generates outputs (images or tables) saved directly to the experiment folder, ensuring data preservation and modularity. This design allows users to import data flexibly from external tools, including ImageJ and Cellpose UI. Detailed outputs include mask images from segmentation, trajectory tables from tracking, and additional columns for measurements and event detection. The neighborhood module further extends the trajectory tables, producing a pickle file with a nested-dictionary structure that logs neighbor information at each time point.

Development and testing were conducted on various hardware configurations: an Intel Core i9 CPU, NVIDIA RTX 3070 GPU, and 16 GB RAM desktop running Ubuntu 20.04 for primary development; extensive testing was completed on both Windows 11 (Intel Core i9 and Core i7 laptops) and an older Ubuntu 20.04 desktop with an Intel Core i5 CPU. Celldetective is routinely validated across Python versions 3.9 to 3.11 on both Windows and Linux, with comprehensive procedural and graphical unit tests.

Antibody design and production

The bispecific antibody (bsAb) C7b-21 (or CE4-21, CE4-28) is a fusion of two single domain antibodies (sdAb, also called nanobodies), sdAb C7b (or CE4) targeting against Human epidermal growth factor receptor-2 (HER-2/neu or ErbB2) [72] and sdAb C21 (or C28) targeting human CD16 (FcγRIII), using the human CH1/Ck heterodimerization motif, corresponding to the bispecific Fab (bsFab) format [73]. bsFab CE4-28 was produced by co-transfection of two plasmids using the mammalian transient system Expi293 (Thermofisher) and purified as previously described [74]. For experiments in Fig. 4 bsAb CE4-21 and control antibody CE4-X where specifically labelled with Atto647 dye using transglutaminase reaction (Zedira, Darmstadt, Germany).

Cell lines culture, effector extraction and phenotyping

Modified MCF7-HER2+ cells were stably transfected from MCF7 cell line to overexpress HER2 receptors. Determination of HER-2 levels on the cells was performed by flow cytometry with Herceptin antibody and a secondary fluorescent anti-human antibody. Fluorescence intensity was determined and correlated with HER-2 levels. Cells were maintained in RPMI 1640 media (Gibco, Life Technologies), supplemented with 10% of Fetal Bovine Serum, FBS (Gibco, Life Technologies), 50 mg/mL of hygromycin as an antibiotic resistance selection. Cells were amplified three times per week and keep in the incubator at 37°C under 5% CO_2 atmosphere.

Primary human NK cells. NK cells were isolated as described in [58]. Briefly, blood samples were obtained from the Etablissement Francais du Sang (Marseille, France), using the MACSxpress whole-blood human NK cell isolation kit (Miltenyi Biotec, Bergisch Gladbach, Germany) a negative selection was performed. The characterization of the sorted cells was determined by flow cytometry with anti-CD16, anti-CD3 and anti-CD56 antibodies. Cells were maintained in RPMI 1640 medium and 10% FBS at 37°C, 5% CO_2 and used in the following 24 hours.

Cell-surface assay

Primary human NK cells, freshly isolated and prepared at a concentration of 200,000 cells/mL, were introduced to an uncoated eight-well chambered polymer coverslip-bottom (Ibidi, 80821). For surface preparation, each well was initially rinsed with PBS, then incubated with 100 μg/mL Biotin-labeled bovine serum albumin (BSA; Sigma A8549-10MG) in PBS for 30 minutes at room temperature (RT) with agitation. Non-adsorbed BSA-biotin was removed by four PBS rinses. Next, a 10 μg/mL Neutravidin solution (Thermo Scientific, 31000) in PBS was added for 30 minutes at RT with agitation, followed by another four PBS rinses. The wells were then incubated with a 10 nM/mL solution of Her2/ERBB2 protein (Sinobiological, 10004-H27H-B) in 0.2% BSA for 30 minutes at RT with agitation and subsequently washed four times with PBS. After this surface preparation, the Ibidi chamber was placed under a microscope at 37°C. Prior to NK cell introduction, the cells were pre-incubated with bsFab (C7b-21) at the desired concentration for 30 minutes at 37°C. They were then injected into the prepared sample.

Cell-surface contact and spreading dynamics were monitored using reflection interference contrast microscopy (RICM), which is sensitive to variations in cell-surface distance. The RICM configuration, described in detail by Limozin et al. [61], employed an antiflex Zeiss objective (NA = 1.25) and a green LED light source (λ = 546 nm) with a 14-bit CCD detector (Andor iXonEM, Oxford Instruments). This setup allowed for live cell observation at 37°C. Images were taken after cell deposition in the chamber, with sixteen fields of view selected per condition and monitored through cyclic imaging over a 10-minute interval. The motorized stage (Physik Instrumente, Germany) facilitated sequential imaging, with temporal intervals between images in each field ranging from 15 to 20 seconds, enabling analysis of decision time between cell landing and spreading as well as NK cell spreading kinetics.

Cell-cell assay

MCF7-Mod cells, expressing HER2 at a surface density of approximately 500 molecules/μm^2 [66], were used as targets. 80,000 cells per well were seeded into an 8-well chamber μ-Slide (Ibidi—Munich, Germany, polymer bottom, TC treated) and cultured overnight in RPMI complemented media under standard conditions (37°C, 5% CO_2) to reach exponential growth. The next day, media was removed, and cells were incubated with Hoechst 33342 (5 μg/mL) in colorless RPMI at 37°C for 10 minutes. After three rinses in warm RPMI with 10% FBS, cells were returned to the incubator.

Primary NK cells were labeled with CFSE dye following manufacturer instructions (CFSE CellTrace, ThermoFisher). 2.5 million cells were centrifuged at 1500 rpm for 5 minutes, supernatant was discarded, and cells were incubated with 2.5 mL CFSE in PBS for 20 minutes at 37°C. Following a brief incubation in RPMI with 10% FBS (12.5 mL added to the cells for 5 minutes), cells were centrifuged, resuspended in complete media, and incubated.

Dilutions of the bispecific antibody (bsAb) CE4-28 were prepared for final concentrations of 1, 10, and 100 pM. The bsAb solution (2 μg/mL) was added to

wells with Hoechst-labeled target cells, incubated at 37°C for 20 minutes, followed by Propidium Iodide (PI; Sigma). NK cells were then added at an effector-to-target (E:T) ratio of 2.5, and the slide was placed on a Zeiss AxioObserver microscope with temperature control at 37°C. Imaging was performed with a 20x/0.4 objective (pixel size 0.31 μm) in transmitted light (brightfield) and epifluorescence channels (CFSE, Hoechst, PI), capturing 5-9 fields per well over 2-3 hours with 3-minute intervals between frames.

For NK cell tracking, observations were conducted using a 40x/1.3 Oil DIC (UV) objective (pixel size 0.157 μm) with a 1.73-minute interval per frame, and the E:T ratio was adjusted to 1:1. To monitor degranulation, anti-LAMP1 APC-labeled antibodies (BioLegend, Clone H4A3, Cat. 328619) were added at 5 μL per 1 million NK cells.

Software methods

Tracking tuning: Celldetective provides a graphical interface for the open-source Python package bTrack [41], integrating its Bayesian tracker, which, after tuning, proved well-suited for tracking immune and cancer cells in our applications. We developed a configuration interface that enables users to view and adjust the bTrack configuration file (defining the motion model and tracklet reconnection parameters) or to import configurations from the bTrack-napari plugin. Typical bTrack configuration files are provided for the applications we present. Additionally, users can incorporate mask-based measurements to enhance the Bayesian update, with feature normalization handled automatically by the software. Finally, users can set options for track post-processing (interpolation, extrapolation, filtering) to refine and correct the tracking data. A tracking quality control module visualizes the raw bTrack output (before post-processing) in napari, recoloring labels to preserve cell identity.

Segmentation model training: To train a model, users can select any combination of channels as the input. Each channel's normalization can be independently adjusted using either percentile or absolute normalization. For RICM images, which are background-corrected with intensities centered around 1, an absolute normalization range of 0.75 to 1.25 was adopted. In contrast, fluorescence images—where light source power and exposure often vary across experiments—use percentile normalization, typically set between the 0.1-th and 99.99-th percentile for each image. During training, images in the training set are randomly augmented with operations such as flips and lateral shifts. Each input channel can undergo independent intensity adjustments, including blurring (random sigma between 0 and 4 px), physical noise addition (Gaussian, local variance, Poisson, and speckle), and random deactivation of channels (excluding the primary input channel), encouraging the model to learn redundancy across channels. StarDist models are trained for around 300 epochs with a learning rate of approximately 10^{-4} and a typical batch size of 8, while Cellpose models are trained over roughly 3000 epochs with a learning rate near 10^{-2} and a similar batch size.

Annotations for segmentation: To improve the annotation experience in napari, we developed a plugin that addresses common annotation errors identified with our expert annotators. The plugin detects repeated label values by comparing cell bounding boxes with mask areas, interpreting significant differences as multiple cells sharing the same label. Noncontiguous objects are automatically separated and relabeled, while small objects (less than 9 px^2) are filtered out. The plugin also reassigns label values sequentially from 1 to the total number of objects, manages annotation naming and storage, and appends experiment metadata to each annotation, enabling precise tracking of annotation origin and context.

Classification and condition-based event detection: We developed a module to classify cells by writing conditions on the measurements with AND and OR logic gates. A flow-cytometry-like scatter plot showing the classification result at each time point for any combination of measurements assists the user in writing arbitrarily complicated conditions. This classification is instantaneous, as each cell from each time point is classified independently. With cell tracks available, we can perform a time-correlated analysis of the binary condition response for each cell. We implemented two particular descriptions: the first is that the cell has a unique state (condition always true or false): the median response determines the cell class; the second is the irreversible event, introduced to describe the cellular events in our applications. It implied three possible states for each cell: always false (no event observed), always true (the event happened before the beginning of observation), or transitioning from false to true (the event occurred during observation). For transitioning events, the software catches the event time by fitting a sigmoid over the binary response for each cell.

Deep learning event detection: To automate event detection, we implemented a deep learning approach with two convolutional models: (1) a classifier that assigns each cell to one of three event classes based on selected measurement time series, and (2) a regressor that estimates the event time for cells exhibiting the event, using the same time series input. Both models share a common backbone, a modified 1D ResNet, designed to process multivariate time series (Supplementary Fig. S12). To match the user-defined input tensor size, time series endpoints are extrapolated by repeating the edge values. In the training set, the time series are augmented with random noise and time shifts to balance the classes and generate artificially a quasi uniform event time distribution. Users can opt to use our pre-trained event-detection models for specific applications (Tab. S5) or easily train new models on their own datasets generated within the associated viewer, with only a few clicks. The event detection system comprises two sequential ResNet-like models with a shared backbone but distinct heads tailored to different tasks: classification and regression (see Fig. S12). Input time series are structured as tensors of shape $(T \times n_{\text{channels}})$, where n_{channels} represents the number of coupled time series (or channels) and T is an arbitrary maximum time series length, typically set to 128 frames. The input tensor is initially processed by a 1D convolution layer with a $(1,)$ kernel to expand it into 64 filters. It then passes through two 1D-ResBlocks, each with a $(8,)$ kernel and 64 filters,

followed by a max-pooling layer (size 2) that doubles the number of filters to 128. The tensor continues through two additional 1D-ResBlocks before undergoing a global average pooling operation. A dense layer with 512 neurons collects the convolutional information, followed by a dropout layer with a 50 % dropout rate. The final layers differ by model task: for classification, a layer with three neurons and softmax activation distinguishes between three classes; by convention "event observed", "no event observed", and "event left censored"; for regression, a single neuron with linear activation predicts t_{event}. The models are trained for 300 and 600 epochs, respectively, using the Adam optimizer with a user-controlled learning rate (typically 10^{-4}, unless mentioned otherwise). For classification, categorical cross-entropy loss is minimized, with class weights introduced to mitigate class imbalance. For regression, the mean squared error is minimized. The batch size, usually set to 64 or 128 depending on the dataset size, is user-controlled. Only cells classified as the first class ("event observed") proceed to the regression model during both training and inference.

Neighborhoods: Single cells from two populations can be linked through a neighborhood scheme, which can either define relationships within a single cell population (e.g., targets-targets) or across two distinct populations (e.g., effectors-targets). To accommodate specific application requirements, we developed two distinct neighborhood methods. The mask contact method identifies touching cell masks at any time point, using mask dilation to allow a tolerance for cell contact. The isotropic method, by contrast, disregards cell masks and instead uses a circular region of a selected radius, centered on a reference cell's centroid, to determine neighboring cells. This approach is particularly useful for quantifying cell co-presence in cases with nuclear segmentation or unreliable masks. Neighbor counts extend the single-cell data tables in the same way as other single-cell measurements (more details in Supplementary Fig. S13). The reference-neighbor pairs identified at least once are then measured (e.g., relative distance, angle, velocity) and stored as rows in a cell-pair table. Similar to single cells, these pairs can be classified and annotated through an interactive cell pair viewer, which will be demonstrated in the second application.

Details on the analyis of the cell-surface assay

Image preprocessing: A RICM background for each well of the cell-surface assay was reconstructed using the model-free background estimate method in Celldetective (described in Supplementary Fig. S8B). The reconstructed backgrounds were optimized and divided to the RICM images, yielding float intensities centered around 1.

Single-cell extraction: The traditional segmentation pipeline consisted of two mathematical operations on the pixels (subtracting one and taking the absolute value) and a Gaussian filter (kernel of 0.8 pixels), followed by thresholding, watershed and automatic removal of non-lymphocyte objects. The pipeline was defined with the dedicated Celldetective module (Supplementary Fig. S9). To track the cells, the bTrack configuration was tuned to increase the isotropic spatial bin size to consider hypotheses (`dist_thresh` = 99.99), so that hypothesese could be generated for each

tracklet. The branching, apoptosis and merging tracklet reconnection hypotheses were disabled and the segmentation miss rate reduced to 1 %. The position-based intensity measurements were performed in a circle of 2 μm diameter centered around the cell positions.

Spreading rate: For each spread cell, the area was differentiated as a function of timeusing a sliding window of 5 frames (1 min 18 s) looking forward in time: from t_i to t_{i+4} (this was configured graphically in the table exploration feature of Celldetective). The value at the onset of spreading t_{spread} was then extracted for each cell (with a track collapse option in the software) and plotted.

Segmentation benchmark: To use the Cellpose *cyto3* model, we passed a diameter value of 15 μm, a `cellprob_threshold` of 0 and `flow_threshold` of 0.4. For each sample in the test set, the lymphocyte detection accuracy was defined as: $DA = \text{TP}/(\text{TP} + \text{FP} + \text{FN})$, with TP the number of lymphocytes predicted, FP the number of non-lymphocyte objects detected and FN the number of lymphocytes missed. A detection was accepted when the intersection over union (IoU) reached at least 0.1. The segmentation quality was defined as the IoU between the ground truth and predicted masks when they could be matched (TP detections). Since hovering cells had an ill-defined area, the segmentation quality was only computed for the subset of spread cells. The scores reported in Fig. 2 are the average over all test samples of the lymphocyte detection accuracy and the segmentation quality. The average score was defined as the average of these two scores for each segmentation method. The benchmark was done in a Jupyter Notebook.

Details on the analyis of the cell-cell assay

Image preprocessing: The fluorescent bsAb and LAMP1 channels were corrected for the background using the paraboloid fit technique described in Supplementary Fig. S8A. The fitted background for each image was subtracted to the original image, without clipping. The movies from the LAMP1 experiment were aligned prior to analysis with the SIFT multichannel registration plugin in Fiji (to be able to investigate effector cell velocities). The macro used to align the movies in a Celldetective experiment project is available in the documentation.

Tracking of target cells: The MCF-7 nuclei were tracked with a quasi-stationary motion model, allowing for small displacements and potentially long-time gaps. We passed the Hoechst and PI intensities and the nuclear apparent area to the tracker which helped in experiments where the live/dead transitions were not too sudden. Cells not present in the first frame of the movie were filtered out to limit the number of false positive detections and exclude daughter cells from our study. Tracks were extrapolated until the end of the observation to measure intensities at the last detected position, which worked well in this system since the nuclei were mostly stationary.

Effector cell extraction: For the dynamical study, some manual corrections of target and effector segmentation results had to be done to achieve an excellent tracking quality for both populations and study their interactions. Effector cell velocity was computed over a sliding window of 3 frames (5.19 min) centered on the current time (from t_{i-1} to t_{i+1}).

bsAb fluorescence measurements: The mean bsAb fluorescence intensity over each NK cell mask was measured from the background corrected images.

LAMP1 measurement: The mean LAMP1 intensity over each NK cell mask was measured from the corrected images. We classified cells as LAMP1-positive or LAMP1-negative at each time point using an intensity threshold ($I_{\mathrm{LAMP1}} > 5$ a.u.).

Contact classification: Target cell contact with NK cells was inferred using a radial neighborhood calculation, as only target cell nuclei masks were available, and was further classified by the number of neighboring target cells: an NK cell in contact was defined as having at least one target neighbor (inclusive counting convention, see Supplementary Fig. S13). A quality check of this contact classification was performed in-situ on the images using a dedicated viewer that displays cells directly on microscopy images (Video single-timepoint-contact.mp4).

Simulation: The Monte-Carlo simulation for the sum of bsAb fluorescence signals and its comparison with the other conditions was performed in a Jupyter Notebook with in-house code and functions from the Celldetective package. This notebook allowed us to produce figures 4D and Figure Supp S14.

Segmentation benchmark: Nuclear detection accuracy was defined in the same way as the lymphocyte detection accuracy in the cell-surface assay. To use the Cellpose *nuclei* model, we set the diameter to 22 μm, the `cellprob_threshold` parameter to 0.0 and the `flow_threshold` parameter to 0.4. The segmentation quality is computed over all nuclei. As before, the scores reported in Fig. 3 are the average over all test samples of the detection accuracy and segmentation quality. The average score was defined as the average of these two scores for each segmentation method.

Death event detection benchmark: the event detection models we present were trained on the `db-si-NucPI` dataset available in Zenodo with a 20 % validation split and 10 % test split, randomized at each training. For each tested combination of batch size (BS $\in [64, 128, 256]$) and learning rate ($\alpha \in [10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}]$), we trained 10 event detection models for 300 epochs with the two different input approaches (PI time series alone or PI + nuclear area time series). The time series were normalized with a percentile normalization (0.1 % and 99.9 %) and clipping. The model input tensor length was set to 128 frames. For the hyperparameters that achieved the best average performance (here BS = 64 and $\alpha = 10^{-4}$), we picked the best event detection model produced out of the 10 for the benchmark of Fig 3D. This benchmark was performed on a separate test set, identical for all methods. The

annotations came from 3 movies, two of which belonged to experiments never seen by the models. The third is from a position adjacent to a position seen by the models. It was selected due to its over-representation of cells that are already dead from the beginning, or dying at the very beginning, which are very rare in most experiments. The benchmark was done in a Jupyter notebook.

Statistics

Statistical tests: Since for all multi-condition comparisons at least one distribution was not normally distributed, we assessed statistical difference using the nonparametric two-sample Kolmogorov-Smirnov one-sided test (Python package `scipy.stats.ks_2samp`). P-values were converted into star notations following the GraphPad convention: $P > 0.05$: ns, $P < 0.05$: *, $P < 0.01$: **, $P < 0.001$: ***, $P < 0.0001$: ****.

Effect sizes: For statistically significant differences, we computed Cliff's Delta to report the effect size (Python package `cliffs-delta`). We followed the conventions of the package to convert the effect size value into a qualitative effect: $\delta < 0.147$: negligible, $\delta < 0.33$: small, $\delta < 0.474$: medium, $\delta > 0.474$: large.

Plots

Most plots presented in this article were generated within Celldetective using the `matplotlib` and `seaborn` packages. Notable exceptions include the benchmarks (segmentation scores, confusion matrix and correlation plots). Composition was done on Inkscape to add statistical tests and effect sizes or merge separate plot layers.

Acknowledgments

We thank Cristina Gonzalez and Maddy Messa N'Dong for preliminary experiments; Juliette Prothon for producing and labelling the bispecific antibodies; Adrien Aimard, Dominique Touchard, Martine Biarnes for excellent technical support; Pierre-Henri Puech and Thierry Galliano for fruitful discussions. The project leading to this publication has received funding from France 2030, the French Government program managed by the French National Research Agency (ANR-16-CONV-0001) and from Excellence Initiative of Aix-Marseille University - A*MIDEX AMX-21-IET-017, as well as AMIDEX Emergence Innovation (project ForSelecAntibody), Plan Cancer PhysCancer program (project ComPhysAb) and ANR-22-CE09-0014. .

References

[1] Rafelski, S. M. & Theriot, J. A. Establishing a conceptual framework for holistic cell states and state transitions. *Cell* **187**, 2633–2651 (2024).

[2] Kramer, B. A., Sarabia del Castillo, J. & Pelkmans, L. Multimodal perception links cellular state to decision-making in single cells. *Science* **377**, 642–648 (2022).

[3] Bagheri, N., Carpenter, A. E., Lundberg, E., Plant, A. L. & Horwitz, R. The new era of quantitative cell imaging—challenges and opportunities. *Molecular Cell* **82**, 241–247 (2022).

[4] Malissen, B. & Bongrand, P. Early T cell activation: Integrating biochemical, structural, and biophysical cues. *Annual review of immunology* **33**, 539–561 (2015).

[5] Belardi, B., Son, S., Felce, J. H., Dustin, M. L. & Fletcher, D. A. Cell–cell interfaces as specialized compartments directing cell function. *Nature Reviews Molecular Cell Biology* **21**, 750–764 (2020).

[6] Kiesgen, S., Messinger, J. C., Chintala, N. K., Tano, Z. & Adusumilli, P. S. Comparative analysis of assays to measure CAR T-cell-mediated cytotoxicity. *Nature Protocols* **16**, 1331–1342 (2021).

[7] Ginhoux, F., Yalin, A., Dutertre, C. A. & Amit, I. Single-cell immunology: Past, present, and future. *Immunity* **55**, 393–404 (2022).

[8] Limozin, L. & Puech, P.-H. Membrane Organization and Physical Regulation of Lymphocyte Antigen Receptors: A Biophysicist's Perspective. *The Journal of Membrane Biology* **252**, 397–412 (2019).

[9] Bongrand, P. Understanding How Cells Probe the World: A Preliminary Step towards Modeling Cell Behavior? *International Journal of Molecular Sciences* **24**, 2266 (2023).

[10] Sengupta, K., Dillard, P. & Limozin, L. Morphodynamics of T-lymphocytes: Scanning to spreading. *Biophysical Journal* S0006349524001577 (2024).

[11] Eulenberg, P. *et al.* Reconstructing cell cycle and disease progression using deep learning. *Nature Communications* **8**, 463 (2017).

[12] Shetab Boushehri, S. *et al.* Explainable machine learning for profiling the immunological synapse and functional characterization of therapeutic antibodies. *Nature Communications* **14**, 7888 (2023).

[13] German, Y. *et al.* Morphological profiling of human T and NK lymphocytes by high-content cell imaging. *Cell Reports* **36**, 109318 (2021).

[14] Chandrasekaran, S. N., Ceulemans, H., Boyd, J. D. & Carpenter, A. E. Image-based profiling for drug discovery: Due for a machine-learning upgrade? *Nature Reviews Drug Discovery* **20**, 145–159 (2021).

[15] Olofsson, P. E. *et al.* Distinct Migration and Contact Dynamics of Resting and IL-2-Activated Human Natural Killer Cells. *Frontiers in Immunology* **5** (2014).

[16] Lu, H. *et al.* TIMING 2.0: High-throughput single-cell profiling of dynamic cell–cell interactions by time-lapse imaging microscopy in nanowell grids. *Bioinformatics* **35**, 706–708 (2019).

[17] Alieva, M., Wezenaar, A. K. L., Wehrens, E. J. & Rios, A. C. Bridging live-cell imaging and next-generation cancer treatment. *Nature Reviews Cancer* (2023).

[18] Vanherberghen, B. *et al.* Classification of human natural killer cells based on migration behavior and cytotoxic response. *Blood* **121**, 1326–1334 (2013).

[19] Wiggins, L. *et al.* The CellPhe toolkit for cell phenotyping using time-lapse imaging and pattern recognition. *Nature Communications* **14**, 1854 (2023).

[20] Salles, A. *et al.* Barcoding T Cell Calcium Response Diversity with Methods for Automated and Accurate Analysis of Cell Signals (MAAACS). *PLoS Computational Biology* **9**, e1003245 (2013).

[21] This, S., Costantino, S. & Melichar, H. J. Machine learning predictions of T cell antigen specificity from intracellular calcium dynamics. *Science Advances* **10**, eadk2298 (2024).

[22] Sità, L. *et al.* A deep-learning approach for online cell identification and trace extraction in functional two-photon calcium imaging. *Nature Communications* **13**, 1529 (2022).

[23] Lee, M. *et al.* Deep-learning-based three-dimensional label-free tracking and analysis of immunological synapses of CAR-T cells. *eLife* **9**, e49023 (2020).

[24] Ichise, H. *et al.* Functional visualization of NK cell-mediated killing of metastatic single tumor cells. *eLife* **11**, e76269 (2022).

[25] Liu, B. *et al.* Deep Learning–Based 3D Single-Cell Imaging Analysis Pipeline Enables Quantification of Cell–Cell Interaction Dynamics in the Tumor Microenvironment. *Cancer Research* **84**, 517–526 (2024).

[26] Dekkers, J. F. *et al.* Uncovering the mode of action of engineered T cells in patient cancer organoids. *Nature Biotechnology* **41**, 60–69 (2023).

[27] Roudot, P. *et al.* U-track3D: Measuring, navigating, and validating dense particle trajectories in three dimensions. *Cell Reports Methods* **3**, 100655 (2023).

[28] Berg, S. *et al.* Ilastik: Interactive machine learning for (bio)image analysis. *Nature Methods* **16**, 1226–1232 (2019).

[29] Moen, E. *et al.* Deep learning for cellular image analysis. *Nature Methods* **16**, 1233–1246 (2019).

[30] Arzt, M. *et al.* LABKIT: Labeling and Segmentation Toolkit for Big Image Data. *Frontiers in Computer Science* **4**, 777728 (2022).

[31] Jan, M., Spangaro, A., Lenartowicz, M. & Mattiazzi Usaj, M. From pixels to insights: Machine learning and deep learning for bioimage analysis. *BioEssays* **46**, 2300114 (2024).

[32] Shroff, H., Testa, I., Jug, F. & Manley, S. Live-cell imaging powered by computation. *Nature Reviews Molecular Cell Biology* (2024).

[33] Vicar, T. *et al.* Cell segmentation methods for label-free contrast microscopy: Review and comprehensive comparison. *BMC Bioinformatics* **20**, 360 (2019).

[34] Falk, T. *et al.* U-Net: Deep learning for cell counting, detection, and morphometry. *Nature Methods* **16**, 67–70 (2019).

[35] Weigert, M., Schmidt, U., Haase, R., Sugawara, K. & Myers, G. *Star-convex Polyhedra for 3D Object Detection and Segmentation in Microscopy*, 3655–3662 (IEEE, Snowmass Village, CO, USA, 2020).

[36] Stringer, C., Wang, T., Michaelos, M. & Pachitariu, M. Cellpose: A generalist algorithm for cellular segmentation. *Nature Methods* **18**, 100–106 (2021).

[37] Pachitariu, M. & Stringer, C. Cellpose 2.0: How to train your own model. *Nature Methods* **19**, 1634–1641 (2022).

[38] Greenwald, N. F. *et al.* Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. *Nature Biotechnology* **40**, 555–565 (2022).

[39] Edlund, C. *et al.* LIVECell—A large-scale dataset for label-free live cell segmentation. *Nature Methods* **18**, 1038–1045 (2021).

[40] Wen, C. *et al.* 3DeeCellTracker, a deep learning-based pipeline for segmenting and tracking cells in 3D time lapse images. *eLife* **10**, e59187 (2021).

[41] Ulicna, K., Vallardi, G., Charras, G. & Lowe, A. R. Automated deep lineage tree analysis using a Bayesian single cell tracking approach. *Frontiers in Computer Science* **3**, 734559 (2021).

[42] Ershov, D. *et al.* TrackMate 7: Integrating state-of-the-art segmentation algorithms into tracking pipelines. *Nature Methods* (2022).

[43] Cuny, A. P., Ponti, A., Kündig, T., Rudolf, F. & Stelling, J. Cell region fingerprints enable highly precise single-cell tracking and lineage reconstruction. *Nature Methods* **19**, 1276–1285 (2022).

[44] Soelistyo, C. J., Ulicna, K. & Lowe, A. R. Machine learning enhanced cell tracking. *Frontiers in Bioinformatics* **3**, 1228989 (2023).

[45] Ollion, J., Maliet, M., Giuglaris, C., Vacher, É. & Deforet, M. DiSTNet2D: Leveraging Long-Range Temporal Information for Efficient Segmentation and Tracking. *PRX Life* **2**, 023004 (2024).

[46] Lugagne, J.-B., Lin, H. & Dunlop, M. J. DeLTA: Automated cell segmentation, tracking, and lineage reconstruction using deep learning. *PLOS Computational Biology* **16**, e1007673 (2020).

[47] Padovani, F., Mairhörmann, B., Falter-Braun, P., Lengefeld, J. & Schmoller, K. M. Segmentation, tracking and cell cycle analysis of live-cell imaging data with Cell-ACDC. *BMC Biology* **20**, 174 (2022).

[48] Schwartz, M. S. *et al.* Caliban: Accurate cell tracking and lineage construction in live-cell imaging experiments with deep learning (2023).

[49] Schindelin, J. *et al.* Fiji: An open-source platform for biological-image analysis. *Nature Methods* **9**, 676–682 (2012).

[50] De Chaumont, F. *et al.* Icy: An open bioimage informatics platform for extended reproducible research. *Nature Methods* **9**, 690–696 (2012).

[51] McQuin, C. *et al.* CellProfiler 3.0: Next-generation image processing for biology. *PLOS Biology* **16**, e2005970 (2018).

[52] Gómez-de-Mariscal, E. *et al.* DeepImageJ: A user-friendly environment to run deep learning models in ImageJ. *Nature Methods* **18**, 1192–1195 (2021).

[53] Stirling, D. R. *et al.* CellProfiler 4: Improvements in speed, utility and usability. *BMC Bioinformatics* **22**, 433 (2021).

[54] Körber, N. MIA is an open-source standalone deep learning application for microscopic image analysis. *Cell Reports Methods* **3**, 100517 (2023).

[55] Windhager, J. *et al.* An end-to-end workflow for multiplexed image processing and analysis. *Nature Protocols* (2023).

[56] von Chamier, L. *et al.* Democratising deep learning for microscopy with ZeroCostDL4Mic. *Nature Communications* **12**, 2276 (2021).

[57] Edelstein, A. D. *et al.* Advanced methods of microscope control using μManager software. *Journal of Biological Methods* **1**, 10 (2014).

[58] Gonzalez, C. *et al.* Nanobody-CD16 Catch Bond Reveals NK Cell Mechanosensitivity. *Biophysical Journal* **116**, 1516–1526 (2019).

[59] Robert, P., Limozin, L., Van Der Merwe, P. A. & Bongrand, P. CD8 Co-Receptor Enhances T-Cell Activation without Any Effect on Initial Attachment. *Cells* **10**, 429 (2021).

[60] Turini, M., Chames, P., Bruhns, P., Baty, D. & Kerfelec, B. A FcgammaRIII-engaging bispecific antibody expands the range of HER2-expressing breast tumors eligible to antibody therapy. *Oncotarget* **5**, 5304–5319 (2014).

[61] Limozin, L. & Sengupta, K. Quantitative Reflection Interference Contrast Microscopy (RICM) in Soft Matter and Cell Adhesion. *ChemPhysChem* **10**, 2752–2768 (2009).

[62] Sengupta, K., Aranda-Espinoza, H., Smith, L., Janmey, P. & Hammer, D. Spreading of Neutrophils: From Activation to Migration. *Biophysical Journal* **91**, 4638–4648 (2006).

[63] Dillard, P., Varma, R., Sengupta, K. & Limozin, L. Ligand-Mediated Friction Determines Morphodynamics of Spreading T Cells. *Biophysical Journal* **107**, 2629–2638 (2014).

[64] Pang, M., Roy, T. K., Wu, X. & Tan, K. CelloType: A Unified Model for Segmentation and Classification of Tissue Images (2024).

[65] Juppet, Q. *et al.* Deep Learning Enables Individual Xenograft Cell Classification in Histological Images by Analysis of Contextual Features. *Journal of Mammary Gland Biology and Neoplasia* **26**, 101–112 (2021).

[66] Gonzalez Gutierrez, C. *et al.* Decoupling individual host response and immune cell engager cytotoxic potency (2024).

[67] Versaevel, M., Grevesse, T. & Gabriele, S. Spatial coordination between cell and nuclear shape within micropatterned endothelial cells. *Nature Communications* **3** (2012).

[68] Brodovitch, A. *et al.* T lymphocytes need less than 3 min to discriminate between peptide MHCs with similar TCR-binding parameters: Cellular immune response. *European Journal of Immunology* **45**, 1635–1642 (2015).

[69] Israel, U. *et al.* A Foundation Model for Cell Segmentation. Preprint, Bioinformatics (2023).

[70] Ma, J. *et al.* The multimodality cell segmentation challenge: Toward universal solutions. *Nature Methods* **21**, 1103–1113 (2024).

[71] Ravi, N. *et al.* SAM 2: Segment Anything in Images and Videos (2024). 2408.00714.

[72] Even-Desrumeaux, K. *et al.* Masked Selection: A Straightforward and Flexible Approach for the Selection of Binders Against Specific Epitopes and Differentially Expressed Proteins by Phage Display. *Molecular & Cellular Proteomics* **13**, 653–665 (2014).

[73] Rozan, C. *et al.* Single-Domain Antibody-Based and Linker-Free Bispecific Antibodies Targeting Fc RIII Induce Potent Antitumor Activity without Recruiting Regulatory T Cells. *Molecular Cancer Therapeutics* **12**, 1481–1491 (2013).

[74] Raynaud, A. *et al.* Anti-NKG2D single domain-based antibodies for the modulation of anti-tumor immune response. *OncoImmunology* **10**, 1854529 (2021).

[75] Barshir, R. *et al.* The TissueNet database of human tissue protein–protein interactions. *Nucleic Acids Research* **41**, D841–D844 (2013).

[76] Caicedo, J. C. *et al.* Nucleus segmentation across imaging experiments: The 2018 Data Science Bowl. *Nature Methods* **16**, 1247–1253 (2019).

[77] Kumar, N. *et al.* A Multi-Organ Nucleus Segmentation Challenge. *IEEE Transactions on Medical Imaging* **39**, 1380–1391 (2020).

[78] Naylor, P., Lae, M., Reyal, F. & Walter, T. Segmentation of Nuclei in Histopathology Images by Deep Regression of the Distance Map. *IEEE Transactions on Medical Imaging* **38**, 448–459 (2019).

Supplementary Tables

Table S2 Generalist deep learning segmentation models. This table lists the different generalist models (Cellpose or StarDist) which can be called natively in Celldetective. The sample images are cropped to (200 × 200) px and rescaled homogeneously to fit in the table.

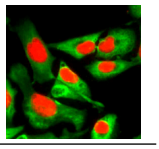
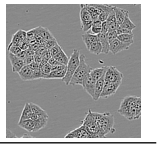
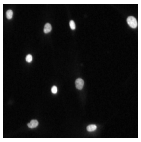
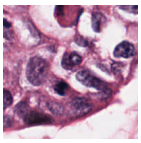
name	modalities	# channels	dataset	sample image
`CP_cyto3`	cytoplasm nucleus	2	Cellpose [37] & user-submitted images	
`CP_livecell`	cytoplasm (BF) black	2	LiveCell [39]	
`CP_tissuenet`	cytoplasm nucleus	2	TissueNet[75]	/
`CP_nuclei`	nucleus black	2	?	/
`SD_versatile_fluo`	nucleus	1	subset of DSB 2018 [76]	
`SD_versatile_he`	H&E RGB	1	MonoNuSeg 2018 [77] TNBC 2018 [78]	

Table S3 MCF7 nuclei segmentation models in the presence of primary NK cells. Each model was trained on the same dataset of ADCC images, picking only the relevant channels.

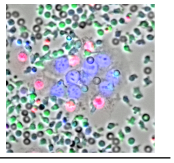
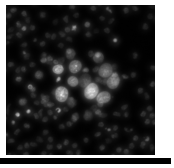
Name	Channels	Type	Pretrained	Spatial calib. (μm)	sample image
`mcf7_nuc_multimodal`	Hoechst Brightfield CFSE PI	StarDist	None	0.3112	
`mcf7_nuc_stardist_transfer`	Hoechst	StarDist	`SD_versatile_fluo`	0.3112	

Table S4 Primary NK segmentation models. The models have been trained on a dataset of annotated primary NKs in ADCC images (*primary_NKs_w_MCF7*).

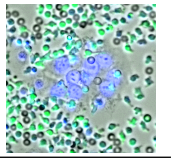
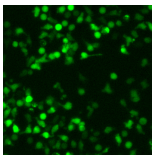
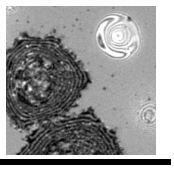
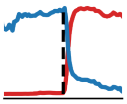
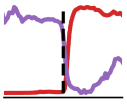
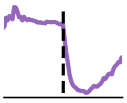
Name	Channels	Type	Pretrained	Spatial calib. (μm)	sample image
`primNK_multimodal`	Brightfield CFSE Hoechst	Cellpose	None	0.2178	
`primNK_cfse`	CFSE None	Cellpose	`CP_cyto2`	0.2178	
`lymphocytes_ricm`	RICM	Cellpose	None	0.2	

Table S5 Event detection models. We trained the following 1D DL models to classify and regress events of interest. The mean event response, centered at the event time is shown for each channel in the pattern column.

Name	Signals	Task	Pattern
lysis_H_PI	Hoechst PI	Strong PI intake	
lysis_PI_area	PI area	Strong PI intake	
NucCond	area	Nucleus shrinking	

Supplementary Figures

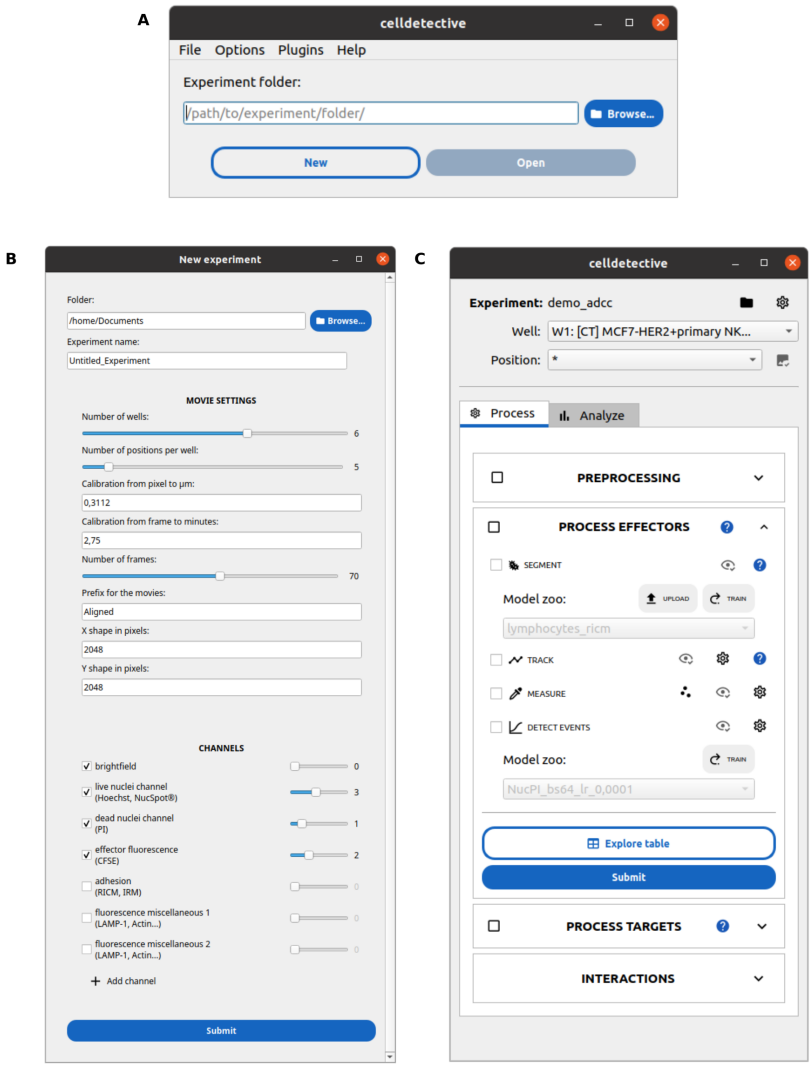


Fig. S6 Main GUI windows: A) Experiment project selection window. B) GUI to generate a new experiment, where the user provides the metadata. C) Main interface after loading a project. The process block for the effector population is unravelled, showing the 4 main steps detailed in Fig. 1.

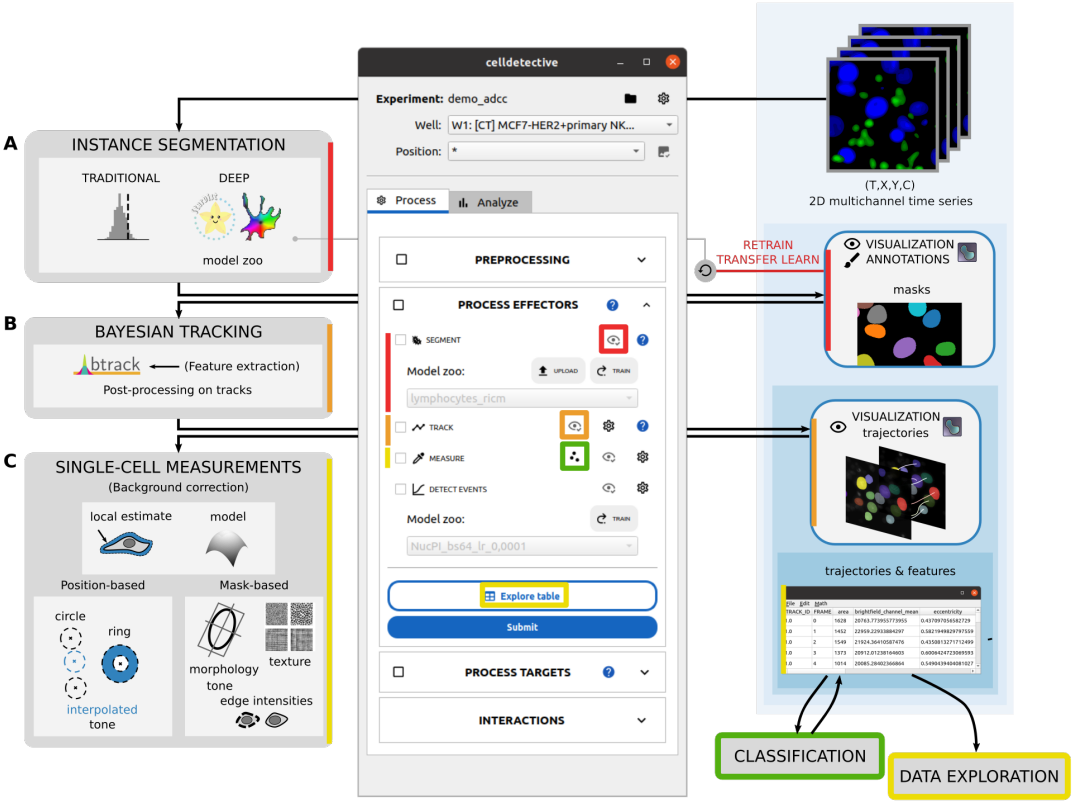


Fig. S7 Processing modules to extract single cells: Processing modules are shown on the left, with brief illustrations of the main options, while the output and visualization modules are represented on the right side, following Celldetective's graphical structure. A typical pipeline features successively (from top to bottom): A) an instance segmentation (either with traditional thresholding method or a deep learning model, stored in a model zoo, using StarDist [35] or Cellpose [37]). The output data are masks, which can be visualized and annotated using napari [56], for correction or model retraining. B) Bayesian tracking associated with feature extraction (btrack [41]), the output data being trajectories, which can also be visualized with napari. C) Single cell measurements of intensity, morphology, texture (based on cell position or on mask), the output being tables of trajectories enriched with features.

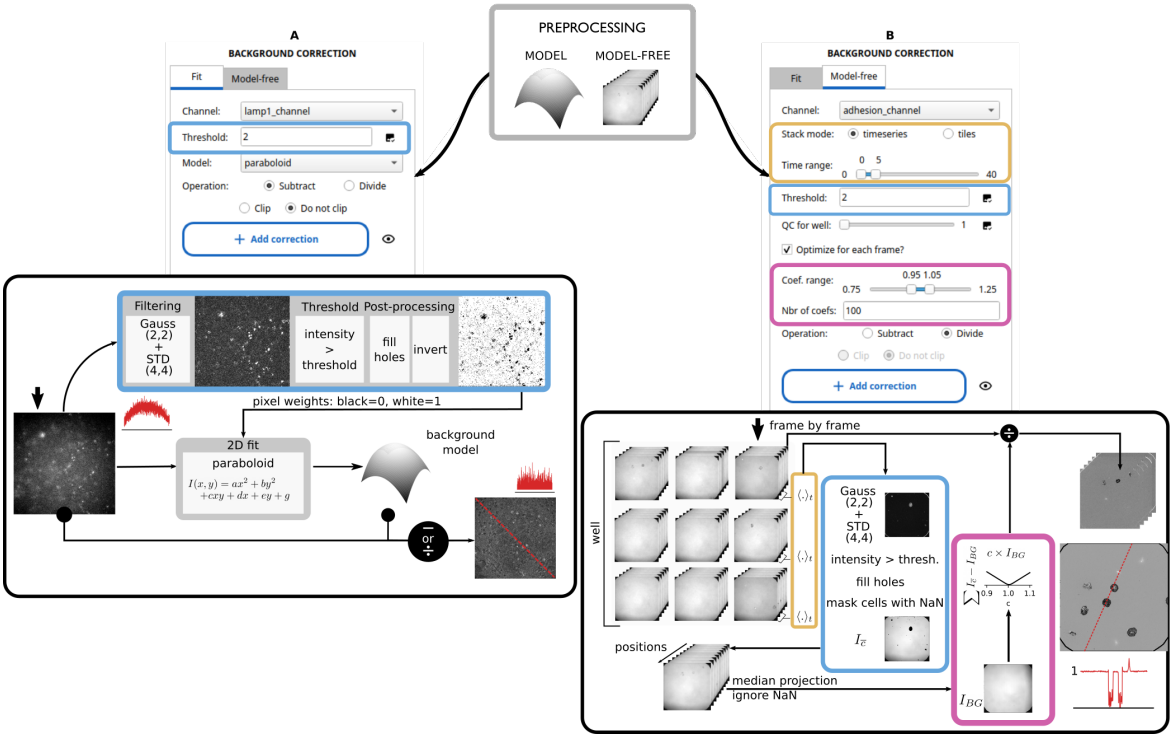


Fig. S8 Background correction methods. Screenshots of the background correction tabs in the software and illustration of the associated pipeline. Colored bounding boxes match graphical parameters to their role in the pipeline. The thick black arrows indicate the starting point of the pipelines. A) The steps for a fit-based correction involve selecting a channel of interest, setting a threshold on the standard-deviation filtered image to exclude the cells (done graphically with a viewer). Then a model (either plane or paraboloid) is fit on the background pixels for each image. The fitted background is either subtracted or divided, with or without clipping. B) With the model-free approach, a single background is reconstructed per well, following the same threshold approach but leveraging the multi-positional information to construct a median background. Instead of applying the background directly, it is amplified to minimize the difference with the current image's background. As with A), the background is then either subtracted or divided, with or without clipping.

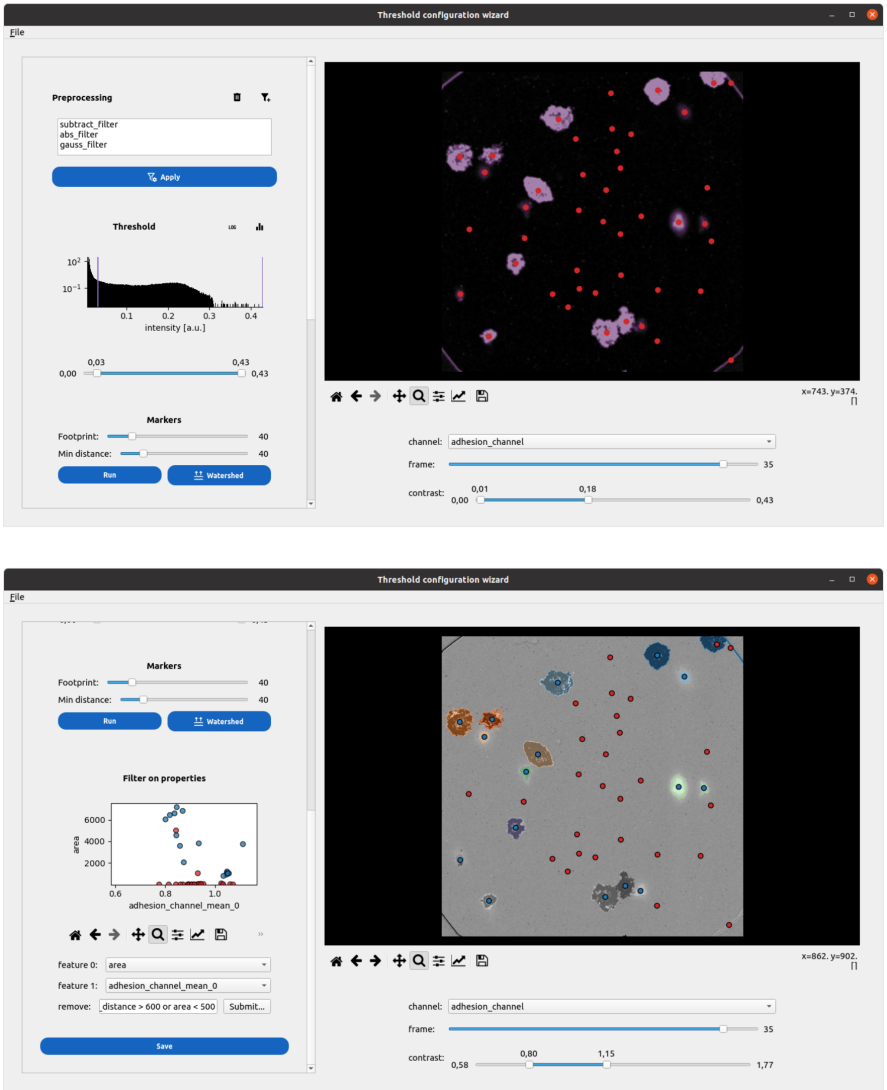


Fig. S9 Designing a traditional segmentation pipeline. Screenshots of the traditional segmentation pipeline configuration interface applied to a normalized RICM image of spreading primary NK cells. Top: User-selected mathematical operations (subtraction, absolute value) and preprocessing filters (Gaussian blur) are applied to the image, with results displayed in real-time on the right. A binary mask, set using an interactive thresholding slider, is overlaid on the transformed image in purple. Spot detection is then performed on the Euclidean Distance Transform of the binary mask, using parameters for footprint size and minimum object distance; detected spots are shown as red scatter points. Bottom: After applying the watershed algorithm, the instance segmentation of cells is overlaid on the original normalized RICM image. The user can set filters on object features to exclude false positives; rejected cells are marked in red in both the feature scatter plot and the image.

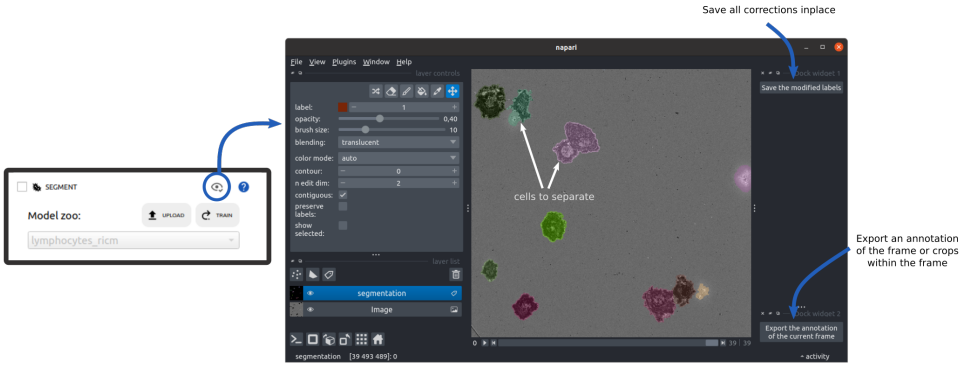


Fig. S10 Segmentation correction and annotation. To view segmentation results, the user can click the "eye" button in the segmentation module, which opens images and labels in napari. Shown is a screenshot of a segmentation outcome for a normalized RICM image of spreading primary NK cells, with labels overlaid on the image. The user can correct the masks directly on the image, for example, to separate cells indicated by white arrows. New labels can be assigned and saved, and the full set of corrected masks can either be saved in place (top-right plugin) or exported as a training sample for a segmentation model (bottom-right plugin).

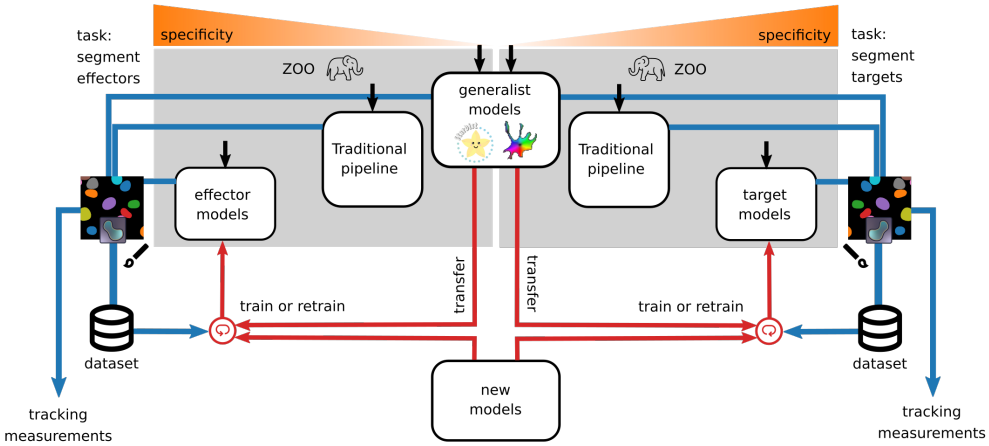


Fig. S11 Overview of segmentation strategies. The principle is illustrated for a mixture of two cell populations, effector and targets. Celldetective provides several entry points (black arrows) to perform segmentation, with the intent of segmenting specifically a cell population (left: effectors, right: targets). The generalist DL models are listed in Tab. S2. Specific DL models are listed in Tab. S3 for targets and Tab. S4 for effectors. The traditional pipeline refers to a thresholding method accessible through the GUI (see Fig. S9) and which can serve as a starting point to prepare a dataset for training a new DL model. The masks output from each segmentation technique can be visualized and manually corrected in napari. Exporting those corrections into a dataset of paired image/masks can be used either to fit a generalist model (transfer learning) or train a new model from scratch.

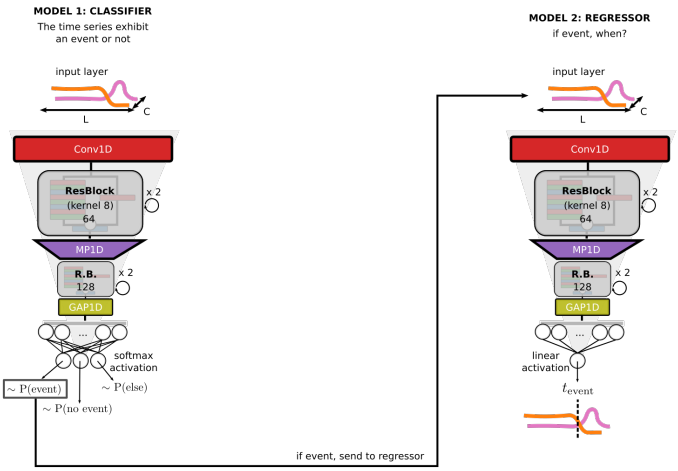


Fig. S12 Event detection model architectures. The event detection models consist of two CNN-based models called back to back during inference to 1) classify single-cell time series and 2) detect the event times from the same time series. The models share a similar architecture of a first 1D convolution, followed by ResBlocks encoding the time series into either three neurons with a softmax activation for the classification problem (three classes) or one neuron with a linear activation for the regression problem. As illustrated here, only the time series that were classified as "event" are sent to the regressor model.

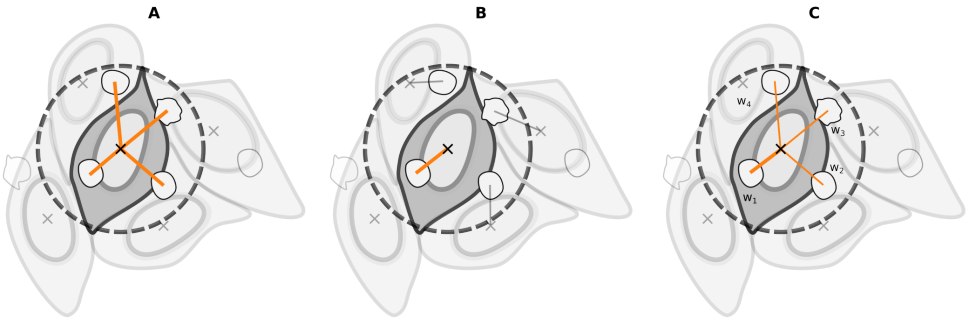


Fig. S13 Neighbor counting methods: A) All neighboring cells in contact or within the isotropic neighborhood are linked to the reference cell (inclusive method). B) Each neighboring cell is linked exclusively to the closest reference cell (exclusive method). C) Neighboring cells in contact or within the isotropic neighborhood are linked to the reference cell, with weights inversely proportional to their number of neighbors.

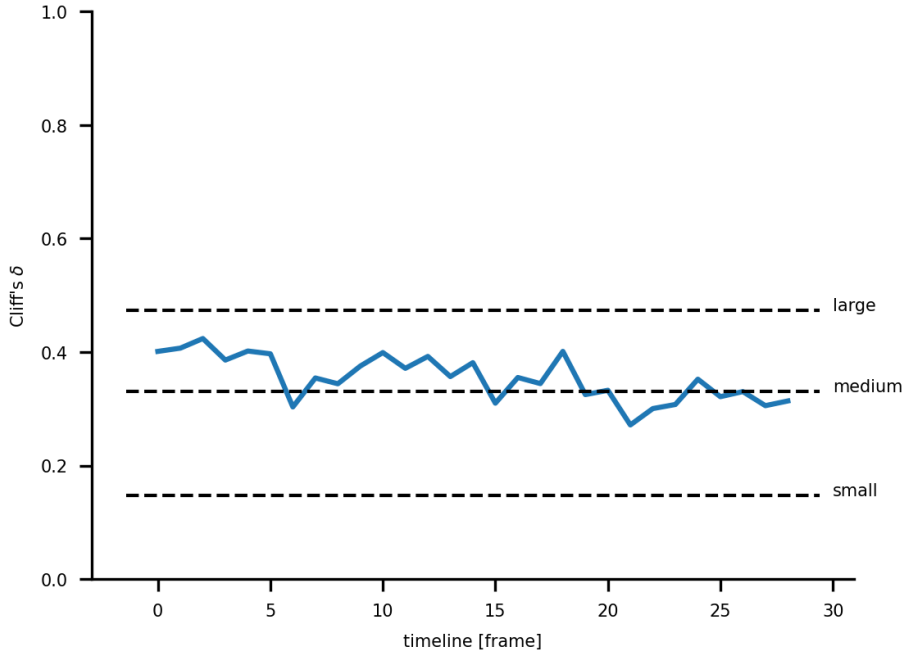


Fig. S14 Effect size at each time point: Cliff's Delta between the simulated sum of two distributions (CE4-21/off-contact/HER2+ + CE4-X/in-contact/HER2+) and CE4-21/in-contact/HER2+ computed at each timepoint, independently.