Cheetah: a computational toolkit for cybergenetic control

Elisa Pedone[1,2,#], Irene de Cesare[1,#], Criseida Zamora[1,3,#], David Haener[1,#], Lorena Postiglione[1], Barbara Shannon[3,4], Nigel Savery[3,4], Claire S. Grierson[3,5], Mario di Bernardo[1,3,6], Thomas E. Gorochowski[3,5,+,*], Lucia Marucci[1,2,3,+*]

[1] Department of Engineering Mathematics, University of Bristol, Woodland Road, Bristol, UK
[2] School of Cellular and Molecular Medicine, University of Bristol, University Walk, Bristol, UK
[3] BrisSynBio, Life Sciences Building, Tyndall Avenue, Bristol, UK
[4] School of Biochemistry, University of Bristol, University Walk, Bristol, UK
[5] School of Biological Sciences, University of Bristol, Tyndall Avenue, Bristol, UK
[6] Department of EE and ICT, University of Naples Federico II, Via Claudio 21, Naples, Italy

[#] These authors contributed equally to this work and should be considered co-first authors
[+] These authors contributed equally to this work and should be considered co-senior authors
[*] Correspondence should be addressed to T.E.G. (thomas.gorochowski@bristol.ac.uk) and L.M. (lucia.marucci@bristol.ac.uk)

Keywords: image analysis; microscopy; feedback control; deep learning; convolutional neural network; U-Net; cybergenetics; synthetic biology; biological control

Abstract

Advances in microscopy, microfluidics and optogenetics enable single-cell monitoring and environmental regulation and offer the means to control cellular phenotypes. The development of such systems is challenging and often results in bespoke setups that hinder reproducibility. To address this, we introduce Cheetah – a flexible computational toolkit that simplifies the integration of real-time microscopy analysis with algorithms for cellular control. Central to the platform is an image segmentation system based on the versatile U-Net convolutional neural network. This is supplemented with functionality to robustly count, characterise and control cells over time. We demonstrate Cheetah's core capabilities by analysing long-term bacterial and mammalian cell growth and by dynamically controlling protein expression in mammalian cells. In all cases, Cheetah's segmentation accuracy exceeds that of a commonly used thresholding-based method, allowing for more accurate control signals to be generated. Availability of this easy-to-use platform will make control engineering techniques more accessible and offer new ways to probe and manipulate living cells.

Introduction

Modern automated microscopy techniques enable researchers to collect vast amounts of single-cell imaging data at high temporal resolutions. This has resulted in time-lapse microscopy becoming the go to method for studying cellular dynamics, enabling the quantification of processes such as stochastic fluctuations during gene expression [1–3], emerging oscillatory patterns in protein concentrations [4], lineage selection [5,6], and many more [7].

To make sense of microscopy images, segmentation is performed whereby an image is broken up into regions corresponding to specific features of interest (e.g. cells and the background). Image segmentation allows for the accurate quantification of cellular phenotypes encoded by visual cues (e.g. fluorescence) by ensuring only those pixels corresponding to a cell are considered. A range of segmentation algorithms have been proposed to automatically analyse images of various organisms and tissues [3,8–11]. The most common of these are thresholding [12] and seeded watershed [13] methods, which are available in many scientific image processing toolkits. Commercial software packages also implement this type of functionality, enabling both automated image acquisition and analysis (e.g. NIS-Elements, Nikon). While these proprietary systems are user-friendly requiring no programming skills to be used, they are often difficult to tailor for specific needs and cannot be easily extended to new forms of analysis.

More recently, deep learning-based approaches to image segmentation have emerged [7,14–17]. Compared to the more common thresholding-based approaches [12], deep learning methods tend to require more significant computational resources when running on traditional computer architectures, and often require the time-consuming manual step of generating large numbers of classified images for training. However, once trained deep learning methods are generally more robust to varying image quality and provide comparable [18] or superior segmentation accuracy [17] to thresholding-based methods.

The accuracy and robustness of a segmentation method are particularly important for online applications. For example, where an environment is dynamically controlled during an experiment in response to changes in cell state. Real-time image analysis and segmentation allows for the implementation of external feedback control [19,20]. Typically, in such an experiment a combined microfluidic and microscopy platform is used to allow for images of single cells to be continually captured and analysed, with changes immediately processed. The state of the cells is generally signalled by the expression of a fluorescence protein that can be dynamically monitored and used as input to a control algorithm. The comparison of this cellular signal to the desired reference *in silico* allows a control signal to be generated by computer software that can be used to alter the cellular environment and perturb the cellular state in the required way (closing the loop). Generally, these experiments require the cells to be genetically engineered to transmit their state using fluorescence and respond to specific environmental stimuli in a prescribed way. This combination of computational, physical, and genetic aspects has resulted in this type of

approach being termed external cybergenetic control and has been successfully applied for gene expression regulation in yeast [21–24], bacteria [25] and mammalian cells [26]. Such external feedback control can also be implemented using optogenetics [2,27] and in combination with flow cytometry for online measurement of the control output [28]. When compared to embedded cellular controllers (where both the controlled process and the controller are implemented within the cell using synthetic regulatory networks), external controllers benefit from requiring only minimal cellular modification, placing little burden on a cell; also, a single control platform can be used for the automatic regulation of different cellular processes across cellular species (e.g. gene expression [21,22], cell growth [28], cytosol-nuclear protein translocation [29]).

In terms of software, while control algorithms such as proportional integral, model predictive control and zero average dynamics are versatile enough to be used in many contexts[30], an online segmentation algorithm usually needs to be tailored given the cell type and the image acquisition settings. For example, if using a thresholding-based approach, various parameters in the segmentation code must be adjusted by trial-and-error before running a closed-loop control experiment. Furthermore, these settings must not significantly change during an experiment (e.g. due to a loss of focus), otherwise accuracy will be compromised. If the online measurements deviate from the real state of the cells, the overall control experiments will fail as inputs become calibrated to a miscalculated control error.

In this work, we aim to address these difficulties by developing a computational toolkit called Cheetah to help simplify external cybergenetic control applications. We demonstrate its core functionality and flexibility by both post-processing time-lapse data for bacterial and mammalian cell growth in a microfluidic chip and external feedback control of gene expression in mammalian cells. We demonstrate Cheetah's increased robustness compared to the widely used Otsu thresholding-based method [12,31] and show how poor segmentation can lead to miscomputed control error and the possible failure of an experiment. Cheetah has a broad range of potential applications from post-experiment image analysis to robust real-time feedback control. Access to these capabilities in an easy-to-use package will help simplifying the integration of control engineering techniques into cell imaging platforms and offer new ways of robustly regulating the behaviour of living cells.

Results

The Cheetah computational toolkit

Cheetah is a Python package designed to support closed-loop control in cybergenetic applications (i.e. systems that combine computational and genetic elements). It combines real-time image segmentation using the U-Net convolutional neural network (CNN) [15,32] with image analysis and cellular control algorithms. U-Net was chosen for segmentation because it has been proven reliable for a wide range of applications in systems and synthetic biology [15,33,34]. Cheetah

implements U-Net using Keras and to avoid overfitting, regularization can be customised to use either batch normalisation or a dropout rate (all examples in this work use batch normalisation).

Cheetah is composed of four modules (**Figure 1**). The first module supports the generation of training data for the U-Net model. Creation of training data can be laborious, therefore a 'DataAugmentor' class is provided to allow for a few labelled training images to be resampled and manipulated, generating a large augmented set of training images. This works by sampling subregions of manually labelled images and then randomly applies image rotations, vertical and horizontal flips, scaling and shearing operations, and adjustments to the image histogram to simulate varying illumination levels. The use of these augmented training sets allows an accurate segmentation model to be trained using a far smaller number of manually labelled images [15].

The second module is focused on the segmentation of images into various classes (e.g., class 1 = background, class 2 = cell). This functionality is defined within the 'Segmenter' class, which also includes functions to train the built-in U-Net model, to save and load the parameters for previously trained models, and to use a model for predicting the class of each pixel in a new image or image stack.

The third module takes segmented images as an input and can apply a range of common analyses. These include the extraction of pixel intensity histograms for a particular segmentation class (e.g. the intensity of all pixels within cells), the ability to classify and label separate cells, and to track cells across a time-series of images (provided movement is limited between frames).

Finally, the fourth module allows for the implementation of user-defined feedback control algorithms. These are implemented by extending the 'ControlAlgorithm' class, which includes placeholder functions for initialising the control setup and an execution loop that continually processes images and generates a control output that will be used to actuate the experimental setup. Built-in functions for Relay, Proportional–Integral (PI) and Proportional–Integral–Derivative (PID) control are provided as examples.

Robust image segmentation and analysis of bacteria and mammalian cells

To demonstrate the core functionality of Cheetah, we made use of an integrated microfluidics and imaging platform that we have previously used for external feedback control of engineered bacterial and mammalian cells [29] (**Methods**). Previous time-lapse videos were collated and analysed using Cheetah and comparisons made to the same analyses performed using the common Otsu thresholding-based segmentation method.

We began by post-processing an open-loop time-lapse experiment of *Escherichia coli* cells containing a genetic construct which uses an orthogonal σ/anti-σ pair to regulate expression of a green fluorescent protein (*gfp*) gene [35] (**Methods**). The experiment consisted of cells being grown in a microfluidic device designed for long-term bacterial culture [36] (**Figure 2A**) and images

(including fluorescence) were acquired every 5 minutes over a 24-hour period (**Methods**). Before Cheetah could be used for analysis, it was necessary to train the system to be able to detect the bacteria in our experiment. This was done by manually annotating only 2 large images (512 × 512 pixels) containing 329 cells in total, with each pixel labelled as either 'background', 'cell border', or 'cell interior'. These training images were augmented using Cheetah's DataAugmentor class to create a final set of 60 smaller annotated images (256 × 256 pixels). Using this set of images allowed for a 99.5% segmentation accuracy to be reached after training (**Methods**). Once trained, Cheetah segmentation masks were generated and used to calculate the number of cells and average GFP fluorescence per cell (**Figures 2B**, **2C**). These results were compared to similar analyses using segmentation masks generated using an Otsu-thresholding based approach that we [29] and others [21,22] have previously implemented in a similar experimental setup (**Supplementary Movie 1**; **Methods**).

There were several clear differences between the two segmentation methods. First, Cheetah gave more robust segmentation results, being able to accurately isolate the bacterial cells from their environment (**Figure 2B**). This differed from the Otsu segmentation method, which struggled due to the edges of the microfluidic chamber and noise within the empty chamber that generated high-contrast features. This resulted in the walls and empty regions of the chamber being classified as cells, and caused a large reduction in GFP fluorescence per cell for the Otsu method at the start of the experiment, when only a few cells were present (**Figure 2B**). As the experiment progressed, the impact of these misclassified regions was reduced as the majority of the image was covered in cells and so their impact was negligible. Furthermore, the Otsu method struggled to precisely distinguish individual cells, showing a visibly lower cell count once the chamber was filled with bacteria (**Figure 2C**). In contrast, Cheetah was not affected by any of these aspects and provided robust and reliable estimates of cell number and fluorescence per cell (**Figure 2C**) for the entire duration of the experiment. It should be noted that the significant difference of ~2600 arbitrary units (a.u.) in GFP fluorescence per cell at the beginning of the experiment between the methods would be a major problem for estimating a control signal, potentially causing large unwanted perturbations to the cells if used in an external feedback control system.

Bacterial cells generally have a simple and fairly consistent morphology across a population, which simplifies their classification. A more challenging problem is the analysis of mammalian cells whose shape can significantly vary over time. To assess Cheetah's ability to handle these more complex cell types, we tested its ability to accurately isolate and characterise mouse embryonic stem cells (mESCs). Unlike in the bacterial example, mammalian cells can often die during an experiment, causing quantification of fluorescence to be influenced by these inactive cells. Ideally, dead cells should be excluded when calculating average fluorescence values, but often are not due to difficulties distinguishing each type with standard methods.

Fortunately, this capability can be easily enabled in Cheetah due to the underlying U-Net segmentation model allowing for additional label types. Therefore, to analyse mammalian cells using Cheetah, we manually annotated 34 large images (1280 × 1056 pixels) containing 314 clones in total, with each pixel labelled as either 'background', 'cell border', 'alive cell interior', or 'dead cell interior' based on human knowledge regarding the generally smaller, disconnected and spherical shape of dead cells within a microfluidic chamber. Again, the DataAugmentor class was used to generate a final set of 536 smaller annotated images (512 × 512 pixels) which enabled Cheetah to reach a segmentation accuracy of 98% after training (**Methods**).

Next, we tested Cheetah using images from a 29-hour open-loop time-lapse experiment where engineered mESCs were grown in a microfluidic chamber that enabled long-term imaging (**Figure 2D**). mESCs were modified to carry an inducible genetic construct that expressed an mCherry fluorescent protein (**Methods**). As before, we compared the performance of cell segmentation and average mCherry fluorescence of Cheetah versus an Otsu segmentation approach (**Supplementary Movie 2**). Similar to the bacterial results, the Otsu method misclassified the walls of the microfluidic chamber as cells and struggled to precisely isolate cell bodies within the chamber (**Figure 2E**). The Otsu method was also not able to distinguish between alive and dead cells, resulting in measurements that combined both categories. When compared to the more accurate results generated by Cheetah, the Otsu method led to a slightly lower estimation of average mCherry fluorescence (**Figure 2F**). Cheetah was able to classify alive and dead cells and although not perfect, its ability to remove even some dead cells helped to improve its estimate of alive cell mCherry fluorescence, which was found to be marginally higher than for dead cells (**Figure 2F**).

External feedback control of protein expression in mammalian cells

Having demonstrated the ability for Cheetah to robustly perform image analysis, we next attempted to validate its use for real-time external control of mammalian cells. Using the same engineered mESCs from the previous experiment, we employed an automated microscopy and fluidic control platform that allows for real-time live-cell imaging within microfluidic chips and the precise control of media and chemical inducers fed to the cells by the movement of motorised syringes (**Figure 3A**) [26]. To allow for cells to be controlled by this system, mESCs carried a dual-input genetic construct where an mCherry fluorescent protein fused to a destabilising-domain (DD) was under the control of a 'Tet-On' promoter (**Figure 3B**, **Methods**) [26]. This allowed the mCherry reporter to be switched 'on' by the combined presence of doxycycline (Doxy) and trimethoprim (TMP). By varying the concentration of these chemicals using the experimental platform in response to the deviation between the current mCherry fluorescence of the cells and the desired reference value, closed-loop real-time control of the cells could be achieved.

To test the effectiveness of Cheetah for external *in silico* feedback control, mESCs carrying the dual-input genetic construct were exposed overnight to high concentrations of Doxy (1 µg/mL) and TMP (100 mM) to cause strong mCherry expression. These cells were then seeded into a microfluidic chip placed on our control platform (**Figure 3A**) and a Relay control algorithm [26] was used to allow for set-point regulation of mCherry expression over a period of 24 hours (**Methods**). In this case, we selected a desired reference average mCherry fluorescence of 10 arbitrary units (a.u.), which was half of the saturating mCherry fluorescence reached overnight. For closed-loop feedback control, images were streamed to Cheetah every 60 minutes; each image was immediately segmented, and the mask generated for alive cells was used to estimate average mCherry fluorescence. This data was then fed to an external system to actuate the necessary control action (i.e. movement of the syringes and thus change in Dox and TMP concentration experienced by the cells) on the experimental platform.

Results from this experiment showed the ability for the platform to accurately control average mCherry fluorescence from the cells throughout the experiment (**Figures 3C**, **3D**; **Supplementary Movie 3)**. We manually annotated 4 frames of the time-lapse data at 0, 8, 16, and 24 hours and compared the average mCherry fluorescence calculated using these masks and those automatically generated by Cheetah. Close agreement was found for the alive and dead cells for most time points, with the only major deviation being for dead cells at 0 hours. Dead cells are often difficult to distinguish from living cells, so some differences, especially during seeding where cells are becoming accustomed to their new environment, would be expected (**Figure 3E**).

For comparison, we ran the identical time-lapse imaging data offline through the Otsu segmentation method used in the previous section (**Figure 3D**; **Supplementary Movie 3**). Estimates of average mCherry fluorescence saw much lower levels due to misclassification of the chamber walls. Such incorrect estimation of fluorescence would have resulted in the mistaken triggering of the control input throughout the experiment.

Discussion

As our ability to create cybergenetic systems that combine computational, physical, and biological elements advances, the need for supporting software to coordinate and control these systems will grow. Cheetah is an attempt to simplify this process by providing an easy-to-use computational toolkit that while containing core functionality to speed up most projects, is also highly adaptable to new needs. Here, we have demonstrated Cheetah's abilities to rapidly classify and segment two morphologically different cell types in two different microfluidic settings. We show that Cheetah can rapidly compute highly accurate image segmentation (99.5% and 98% for *E. coli* and mESCs, respectively) even when trained using only a small number of manually annotated images (2 and 34 images for *E. coli* and mESCs, respectively). Furthermore,

we demonstrate how these capabilities allow for accurate control signals to be generated for external feedback control applications. In particular, the ability for Cheetah to not only segment, but also classify cells as potentially 'dead' or 'alive' enables it to filter out non-viable cells and leads to improved accuracy, as compared to a commonly used Otsu thresholding-based method. In addition to segmentation and control algorithms, Cheetah also includes a wide range of built in analysis for labelling cells, tracking their position across frames and using this information to enable analysis of single-cell properties like fluorescence (see **Supplementary Movie 4** for an example). Being able to automate the creation of analysis dashboards using these capabilities, will also help speed up the discovery of subtle behaviours in populations of cells and offer the means to reanalyse existing time lapse microscopy data in more depth.

While the focus here has been on demonstrating the major functionality of Cheetah, we anticipate that to can be applied much more broadly for applications across the field of synthetic biology. For example, using it within custom-built platforms able to perform imaging and dynamic light patterning [27,28] to control single-cell and guide collective behaviours [37]. Furthermore, the code provided in the toolkit can easily be refined, customised and extended to allow for new features to be implemented. As such, Cheetah is a public, open-source project hosted on GitHub and welcomes contributions from the wider community.

Finally, we expect the deep learning methods that are central to Cheetah's capabilities to play an increasingly important role in synthetic biology. In the context of external feedback control, the combination of deep learning-based label-free cell classification [38,39], online training approaches, model-free control strategies (e.g. reinforcement learning-based feedback control), and the availability of tunable genetic parts [26,35,40,41] could be instrumental in unlocking the potential for control engineering techniques in biology. This will open up new avenues to create reliable and robust synthetic biological systems, much like how control engineering has revolutionised other fields.

Methods

Cheetah training process

Training of the U-Net convolutional neural network within Cheetah was performed using a Dell Precision 5530 laptop (Intel Core i7-8850H CPU, 16 GB RAM, and 512 GB NVMe SSD) running Windows 10, connected to a Sonnet eGFX Breakaway Box 550 hosting an NVIDIA Titan Xp GPU with 12 GB GDDR5X RAM. For all organisms, the full set of annotated images were randomly split with 70% used for training and the remaining 30% used for validation.

Otsu thresholding-based segmentation algorithm

The Otsu segmentation method is based on pixel intensity levels and relies on the definition of grey threshold values used to divide a grayscale image into its components creating a binary

mask [12]. The simplest version of this algorithm allows for the identification of two-pixel classes, background and foreground, by using a single threshold level that aims to minimise the intra-class variance. More sophisticated versions of the algorithm couple global thresholding, previously described, to local thresholding, which computes dividing grey-intensity levels on smaller patches of the same image in order to boost the algorithm accuracy. In this work, segmentation of bacteria was computed using the Otsu method with global and local thresholding. The algorithm distinguished the foreground (single bacterial cells) from the background in each image of the time-lapse experiment. The global thresholding calculated the global area where cells are located, and the local thresholding found the centres and edges to differentiate individual cells in a binary mask. The final mask contained the boundaries and interiors of every segmented cell. This mask was overlaid to the fluorescence image field to calculate the fluorescence as the sum of all pixels in the segmented area minus the background fluorescence value. The average fluorescence across the bacterial population was then calculated as the mean of the fluorescence exhibited by all the objects in the final mask. Mammalian cells fluorescence was computed as the average pixel intensity value of pre-masked fluorescent images to which an average background intensity was subtracted, to take into account possible oscillations of microscopy's light intensity. Masked images were obtained using the global thresholding strategy. For further details and access to the code, we refer the reader to de Cesare et al.[31]

Bacterial strains, media and cell culture

Experiments with bacteria used a previously generated *E. coli* strain [35]. Luria−Bertani (LB) medium (113002065, MP Biomedicals) supplemented with 50 µg/mL kanamycin (K4000, Sigma-Aldrich), 100 µg/mL ampicillin (A9518, Sigma-Aldrich) and 25 µg/mL chloramphenicol (C0378, Sigma-Aldrich) was used for all bacterial cell culture and microfluidics experiments. For microfluidic experiments, a single colony was used to seed 5 mL of LB media with antibiotics and grown overnight (approximately 16 hours) at 37°C with shaking at 200 rpm. 300 µL of the overnight culture was used to seed 300 mL of fresh LB medium with antibiotics. This culture was grown to an optical density at 600 nm of 0.3. The culture was then centrifuged at 2200 × *g* for 15 min and resuspended in 1.5 mL of fresh LB medium supplemented with 0.075% Tween-20 (P1379, Sigma-Aldrich) and antibiotics before loading into the microfluidic device.

Mammalian cell lines, media and culture

Experiments with mammalian cells used a previously generated mouse Embryonic Stem Cell (mESC) line [26]. Briefly, mESCs were subjected to two rounds of infection and drug-selection to stably express the transactivator (EF1a-rtTA, Neomycin) and the doxycycline-inducible vector (pLVX_TRE3GDDmCherry, Puromycin; Addgene plasmid #108679). Selected cells were

expanded and grown on gelatin-coated dishes in knockout Dulbecco's modified Eagle's medium (DMEM D5796, Sigma) supplemented with 15% fetal bovine serum (F7524, Sigma), 1 x nonessential amino acids (11140035, Thermo Fisher), 2 mM L-Glutamine (25030024, Thermo Fisher), 100 µM 2-mercaptoethanol (31350010, Thermo Fisher), 1 mM Sodium Pyruvate (11360039, Thermo Fisher), 1 X Penicillin/Streptomycin (P4458, Sigma) and 1000 U/mL LIF (250-02, Peprotech).

Microfluidic devices and loading

For *E. coli*, the microfluidic device used was developed by Mondragón-Palomino and colleagues at the University of California, San Diego [36]. A replica of the silicon mould was donated to our group. Soft lithography was used to form the microfluidic device which contains 48 trapping chambers and 6 inlet/outlet ports. Before each experiment, a wetting protocol was used to remove any air bubbles and debris from inside the device. The device was then mounted onto the stage of an inverted widefield fluorescence microscope, enclosed inside an incubation chamber set to 37°C (Pecon) and connected to fluidic lines. A cell loading protocol, trapping individual cells in the chambers of the device was performed via the C port. Ports W1 and W2 were used as waste ports, the C port became a waste port once the experiment had begun. Ports B and I were connected to an actuation system for motorised control of syringes to deliver fresh media and inputs to the cells growing inside the device. The R port was used as a mixing port. The microscope (see below for details) was programmed to take phase contrast (PhC), green fluorescence and red fluorescence images of the cells growing inside three different trapping chambers every 5 minutes. Green fluorescence images were used for the detection of sfGFP and red fluorescence images were used for the detection of the sulforhodamine B dye (230162, Sigma-Aldrich), used to detect the correct flow of inputs.

For mESCs, microfluidic chip loading and imaging were performed as reported previously [26]. The microfluidic device we used was designed in the laboratory of Prof Jeff Hasty at the University California in San Diego. It consists of 5 ports for cell lading and media input/output, 33 individual chambers for cell growth and imaging, and a channel for controlled flow perfusion [42]. The chip was fulfilled with complete mESC media supplemented with 1 µg/mL Doxy (D9891, Sigma) and 100 nM TMP (T7883, Sigma) flowing from port 5 followed by port 1 before the cell loading. Cells from a sub-confluent petri dish (60 cm in diameter) were washed with sterile Phosphate Buffered Saline (PBS D8537, Sigma), trypsinised for 2-3 min at room temperature and centrifuged at 1000 rpm for 5 min. Pelleted cells were resuspended in 200 µL of complete mESC medium+Doxy/TMP and gently loaded from port 1 using a 2 mL syringe, while applying constant vacuum suction to ports 3 and 4. The vacuum enables cell trapping by facilitating air release from the chambers. The chip was kept for 24 hours in a tissue culture incubator (5% CO_2, 37°C) under constant Doxy/TMP perfusion to induce mCherry expression before the time-lapse.

The day after, the device was transferred on the widefield microscope and connected to the actuation system that consists of two motor-controlled syringes (http://biodynamics.ucsd.edu/dialawave/) connected to port 6 and 7. One syringe contains Doxy, TMP and 1 μM of Atto488 green fluorescent dye (41051-1MG-F, ThermoFisher), whereas the other only contains plain mESC media. Ports 1, 2 and 5 were connected to stating syringes to balance the flow of media from ports 6 and 7, ensuring constant perfusion and avoiding backflow. During the open-loop experiment (**Figures 2E**, **2F**) mESCs were exposed to plain media for the entire duration of the time-lapse, whereas dynamic switching between plain and Doxy/TMP media was automatically controlled during the closed-loop experiment (**Figures 3C**, **3D**) to reach and maintain a desired reference red fluorescence level.

Live-cell imaging

Time-lapse microscopy for both *E. coli* and mESCs were performed using a Leica DMi8 inverted microscope equipped with an environmental control chamber (PeCon) for long-term temperature control and CO2 enrichment where necessary. The Adaptive Focus Control (AFC) ensures focus is maintained during the entire time-course experiment. Imaging of *E. coli* cells was performed using a 100X objective every 5 min using an AndoriXON 897 ultra back-illuminated EMCCD (512 × 512 pixel 16 μm pixels, 16-bit, 56 fps at full frame) in a temperature-controlled environment. Imaging of mESCs was performed using a 20X objective every 60 minutes in a temperature and CO_2 controlled environment. The experimental set-up includes consecutive acquisition in three channels (phase contrast, green fluorescence and red fluorescence).

Relay control algorithm

The Relay Control algorithm provides at each timepoint a control action that aims to minimise the error signal (*e*, defined as the difference between a reference signal and the process output). Formally, the controller generates the following control input

$$u(t) = \begin{cases} u_1 \ \ if \ e(t) > 0 \\ u_2 \ \ if \ e(t) \leq 0 \end{cases} \quad (1)$$

to decrease the error. In our experiments, the control input u_1 corresponds to providing cells culture media supplemented with Doxy/TMP, while u_2 corresponds to providing cells with plain media. The algorithm also implements a 5% hysteresis interval around the set-point to avoid chattering in the control signal.

General computational analysis and tools

Computational analysis was performed by custom scripts run using Python 3.6.8 and the following packages: tensorflow 1.14.0, keras 2.2.4, scikit-learn 0.21.2, scikit-image 0.15.0, numpy 1.16.4 and matplotlib 3.1.1. Genetic designs are visualised using DNAplotlib 1.0 [43,44] and

Synthetic Biology Open Language Visual (SBOL Visual) symbols [45]. Figures were composed using Omnigraffle 7.16 and Affinity Designer 1.8.3.

Data availability

The Cheetah Python package, analysis code and data presented in this work are available from the project GitHub repository at: https://www.github.com/BiocomputeLab/cheetah.

Acknowledgements

T.E.G. gratefully acknowledge the support of NVIDIA Corporation for the donation of a Titan Xp GPU used in this research. This work was supported by BrisSynBio, a BBSRC/EPSRC Synthetic Biology Research Centre grant BB/L01386X/1 (T.E.G., L.M.), a Royal Society University Research Fellowship grant UF160357 (T.E.G.), the EU Horizon 2020 research project COSY-BIO grant 766840 (L.M.), EPSRC grants EP/R041695/1 and EP/S01876X/1 (L.M.), and MRC grant MR/N021444/1 (L.M.)

Author contributions

L.M. and T.E.G. conceived of the project and supervised the work. D.H. developed the initial Keras implementation of the U-Net convolutional neural network. T.E.G. extended the U-Net implementation and developed the integrated Python package and cell analysis functions. E.P., C.Z., W.W. and L.P. performed experiments and generated training data. L.M. and T.E.G. wrote the paper with input from the other authors.

Conflicts of interest

The authors declare no competing financial interests.

References

1. Marucci, L. *et al.* β-Catenin Fluctuates in Mouse ESCs and Is Essential for Nanog-Mediated Reprogramming of Somatic Cells to Pluripotency. *Cell Rep.* **8**, 1686–1696 (2014).
2. Rullan, M., Benzinger, D., Schmidt, G. W., Milias-Argeitis, A. & Khammash, M. An Optogenetic Platform for Real-Time, Single-Cell Interrogation of Stochastic Transcriptional Regulation. *Mol. Cell* **70**, 745-756.e6 (2018).
3. Young, J. W. *et al.* Measuring single-cell gene expression dynamics in bacteria using fluorescence time-lapse microscopy. *Nat. Protoc.* **7**, 80–88 (2012).
4. Gritti, N., Kienle, S., Filina, O. & van Zon, J. S. Long-term time-lapse microscopy of C. elegans post-embryonic development. *Nat. Commun.* **7**, 12500 (2016).
5. Piltti, K. M. *et al.* Live-cell time-lapse imaging and single-cell tracking of in vitro cultured neural stem cells – Tools for analyzing dynamics of cell cycle, migration, and lineage selection. *Neural Stem Cells Health Dis.* **133**, 81–90 (2018).
6. Brockway, N. L. *et al.* Multicolor lineage tracing using in vivo time-lapse imaging reveals coordinated death of clonally related cells in the developing vertebrate brain. *Dev. Biol.* **453**, 130–140 (2019).
7. Bougen-Zhukov, N., Loh, S. Y., Lee, H. K. & Loo, L.-H. Large-scale image-based screening and profiling of cellular phenotypes. *Cytometry A* **91**, 115–125 (2017).
8. E. Meijering. Cell Segmentation: 50 Years Down the Road. *IEEE Signal Process. Mag.* **29**, 140–145 (2012).
9. Stylianidou, S., Brennan, C., Nissen, S. B., Kuwada, N. J. & Wiggins, P. A. SuperSegger: robust image segmentation, analysis and lineage tracking of bacterial cells. *Mol. Microbiol.* **102**, 690–700 (2016).
10. Kamentsky, L. *et al.* Improved structure, function and compatibility for CellProfiler: modular high-throughput image analysis software. *Bioinformatics* **27**, 1179–1180 (2011).
11. Eliceiri, K. W. *et al.* Biological imaging software tools. *Nat. Methods* **9**, 697–710 (2012).
12. Otsu, N. A Threshold Selection Method from Gray-Level Histograns. *IEEE Trans. Syst. Man Cybern.* **9**, 62–66 (1979).

13. Beucher, S. & Lantuejoul, C. Use of Watersheds in Contour Detection. in *International Workshop on Image Processing: Real-Time Edge and Motion Detection/Estimation* (1979).
14. C. Sommer, C. Straehle, U. Köthe & F. A. Hamprecht. Ilastik: Interactive learning and segmentation toolkit. in *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro* 230–233 (2011). doi:10.1109/ISBI.2011.5872394.
15. Falk, T. *et al.* U-Net: deep learning for cell counting, detection, and morphometry. *Nat. Methods* **16**, 67–70 (2019).
16. Hilsenbeck, O. *et al.* fastER: a user-friendly tool for ultrafast and robust cell segmentation in large-scale microscopy. *Bioinformatics* **33**, 2020–2028 (2017).
17. Caicedo, J. C. *et al.* Evaluation of Deep Learning Strategies for Nucleus Segmentation in Fluorescence Images. *Cytometry A* **95**, 952–965 (2019).
18. Van Valen, D. A. *et al.* Deep Learning Automates the Quantitative Analysis of Individual Cells in Live-Cell Imaging Experiments. *PLOS Comput. Biol.* **12**, e1005177 (2016).
19. M. Khammash, M. Di Bernardo & D. Di Bernardo. Cybergenetics: Theory and Methods for Genetic Control System. in *2019 IEEE 58th Conference on Decision and Control (CDC)* 916–926 (2019). doi:10.1109/CDC40024.2019.9030209.
20. Del Vecchio, D., Dy, A. J. & Qian, Y. Control theory meets synthetic biology. *J. R. Soc. Interface* **13**, 20160380 (2016).
21. Menolascina, F. *et al.* In-Vivo Real-Time Control of Protein Expression from Endogenous and Synthetic Gene Networks. *PLOS Comput. Biol.* **10**, e1003625 (2014).
22. F. Menolascina, M. di Bernardo & D. di Bernardo. Design and implementation of a feedback control strategy for IRMA, a novel synthetic gene regulatory network. in *49th IEEE Conference on Decision and Control (CDC)* 2535–2540 (2010).
23. Uhlendorf, J. *et al.* Long-term model predictive control of gene expression at the population and single-cell levels. *Proc. Natl. Acad. Sci.* **109**, 14271 (2012).
24. Perrino, G., Wilson, C., Santorelli, M. & di Bernardo, D. Quantitative Characterization of α-Synuclein Aggregation in Living Cells through Automated Microfluidics Feedback Control. *Cell Rep.* **27**, 916-927.e5 (2019).

25. Lugagne, J.-B. *et al.* Balancing a genetic toggle switch by real-time feedback control and periodic forcing. *Nat. Commun.* **8**, 1671 (2017).

26. Pedone, E. *et al.* A tunable dual-input system for on-demand dynamic gene expression regulation. *Nat. Commun.* **10**, 4481 (2019).

27. A. M. R. Denniss, T. E. Gorochowski & S. Hauert. Augmented reality for the engineering of collective behaviours in microsystems. in *2019 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS)* 1–6 (2019). doi:10.1109/MARSS.2019.8860907.

28. Milias-Argeitis, A., Rullan, M., Aoki, S. K., Buchmann, P. & Khammash, M. Automated optogenetic feedback control for precise and robust regulation of gene expression and cell growth. *Nat. Commun.* **7**, 12546 (2016).

29. Postiglione, L. *et al.* Regulation of Gene Expression and Signaling Pathway Activity in Mammalian Cells by Automated Microfluidics Feedback Control. *ACS Synth. Biol.* **7**, 2558–2565 (2018).

30. Fiore, G., Perrino, G., di Bernardo, M. & di Bernardo, D. In Vivo Real-Time Control of Gene Expression: A Comparative Analysis of Feedback Control Strategies in Yeast. *ACS Synth. Biol.* **5**, 154–162 (2016).

31. de Cesare, I. *et al.* ChipSeg: an automatic tool to segment bacteria and mammalian cells cultured in microfluidic devices. *bioRxiv* 2020.08.03.225045 (2020) doi:10.1101/2020.08.03.225045.

32. Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (eds. Navab, N., Hornegger, J., Wells, W. M. & Frangi, A. F.) 234–241 (Springer International Publishing, 2015).

33. Lugagne, J.-B., Lin, H. & Dunlop, M. J. DeLTA: Automated cell segmentation, tracking, and lineage reconstruction using deep learning. *PLOS Comput. Biol.* **16**, e1007673 (2020).

34. Weigert, M. *et al.* Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nat. Methods* **15**, 1090–1097 (2018).

35. Annunziata, F. *et al.* An Orthogonal Multi-input Integration System to Control Gene Expression in Escherichia coli. *ACS Synth. Biol.* **6**, 1816–1824 (2017).
36. Mondragón-Palomino, O., Danino, T., Selimkhanov, J., Tsimring, L. & Hasty, J. Entrainment of a Population of Synthetic Genetic Oscillators. *Science* **333**, 1315 (2011).
37. Gorochowski, T. E. *et al.* Toward engineering biosystems with emergent collective functions. *Front. Bioeng. Biotechnol.* **8**, 705 (2020).
38. Chen, C. L. *et al.* Deep Learning in Label-free Cell Classification. *Sci. Rep.* **6**, 21471 (2016).
39. Matek, C., Schwarz, S., Spiekermann, K. & Marr, C. Human-level recognition of blast cells in acute myeloid leukaemia with convolutional neural networks. *Nat. Mach. Intell.* **1**, 538–544 (2019).
40. Bartoli, V., Meaker, G. A., di Bernardo, M. & Gorochowski, T. E. Tunable genetic devices through simultaneous control of transcription and translation. *Nat. Commun.* **11**, 2095 (2020).
41. Baumschlager, A., Aoki, S. K. & Khammash, M. Dynamic Blue Light-Inducible T7 RNA Polymerases (Opto-T7RNAPs) for Precise Spatiotemporal Gene Expression Control. *ACS Synth. Biol.* **6**, 2157–2167 (2017).
42. Kolnik, M., Tsimring, L. S. & Hasty, J. Vacuum-assisted cell loading enables shear-free mammalian microfluidic culture. *Lab. Chip* **12**, 4732–4737 (2012).
43. Der, B. S. *et al.* DNAplotlib: Programmable Visualization of Genetic Designs and Associated Data. *ACS Synth. Biol.* **6**, 1115–1119 (2017).
44. Bartoli, V., Dixon, D. O. R. & Gorochowski, T. E. Automated Visualization of Genetic Designs Using DNAplotlib. in *Synthetic Biology: Methods and Protocols* (ed. Braman, J. C.) 399–409 (Springer New York, 2018). doi:10.1007/978-1-4939-7795-6_22.
45. Beal, J. *et al.* Communicating Structure and Function in Synthetic Biology Diagrams. *ACS Synth. Biol.* **8**, 1818–1825 (2019).

Figures and captions

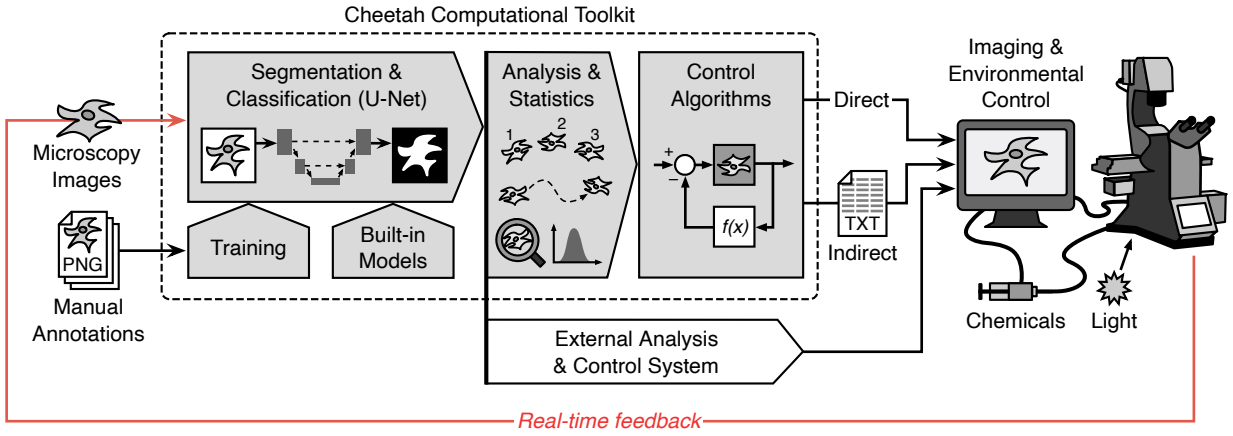

Figure 1: Overview of the Cheetah computational toolkit. Structure of Cheetah's core modules and their interactions (grey filled arrows and boxes). The modular nature of the toolkit allows elements to be used separately, e.g., enabling the use of the built-in segmentation functionality with external analysis and control systems (white pointed box). Control algorithms can either directly interface with the imaging and environmental control system or output their data to text files for use by the external system (i.e. an indirect interface).

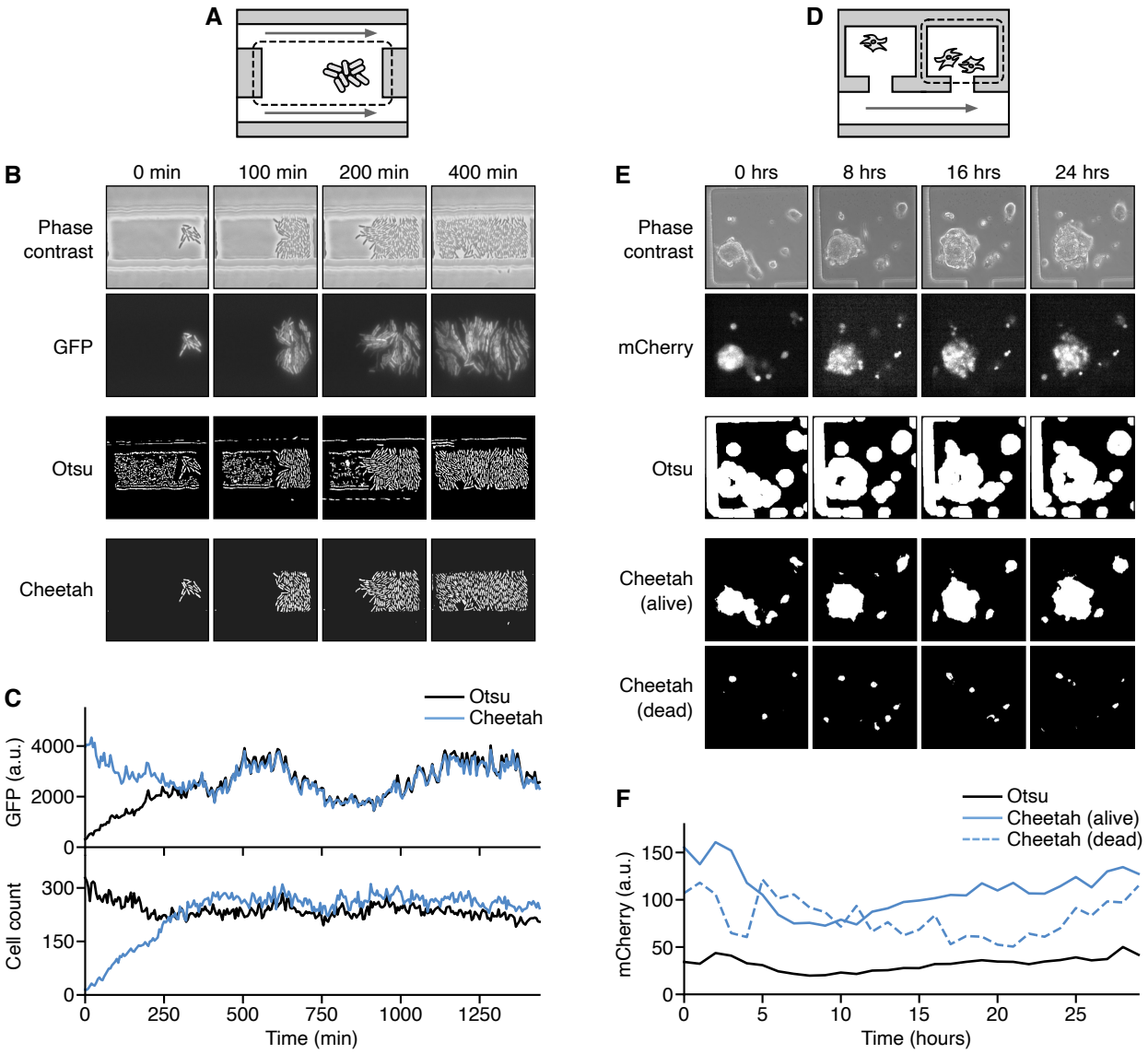

Figure 2: Monitoring and analysis of bacteria and mammalian cells in microfluidic chips. (**A**) Schematic of the microfluidic chamber used for bacterial growth and imaging. A typical imaging area is shown by the dashed box and flow of nutrients is shown by the grey arrows. (**B**) Time-lapse images of *Escherichia coli* cells growing in the microfluidic chamber for phase contrast and GFP fluorescence, as well as segmentation masks for cells generated using the Otsu method and Cheetah (white regions denote cells). (**C**) Average GFP fluorescence of the cell segmentation mask and cell count over time calculated using either the Otsu or Cheetah segmentation masks. (**D**) Schematic of the microfluidic chamber used for mouse embryonic stem cells (mESCs) growth and imaging. A typical imaging area is shown by the dashed box and flow of nutrients is shown by the grey arrows. (**E**) Time-lapse images of mESCs growing in the microfluidic chamber for phase contrast and mCherry fluorescence, as well as segmentation masks for cells generated using the Otsu method and Cheetah (white regions denote cells). For Cheetah, separate masks are shown for living and dead cells. (**F**) Average mCherry fluorescence of the cell segmentation mask over time calculated using either the Otsu method or Cheetah.

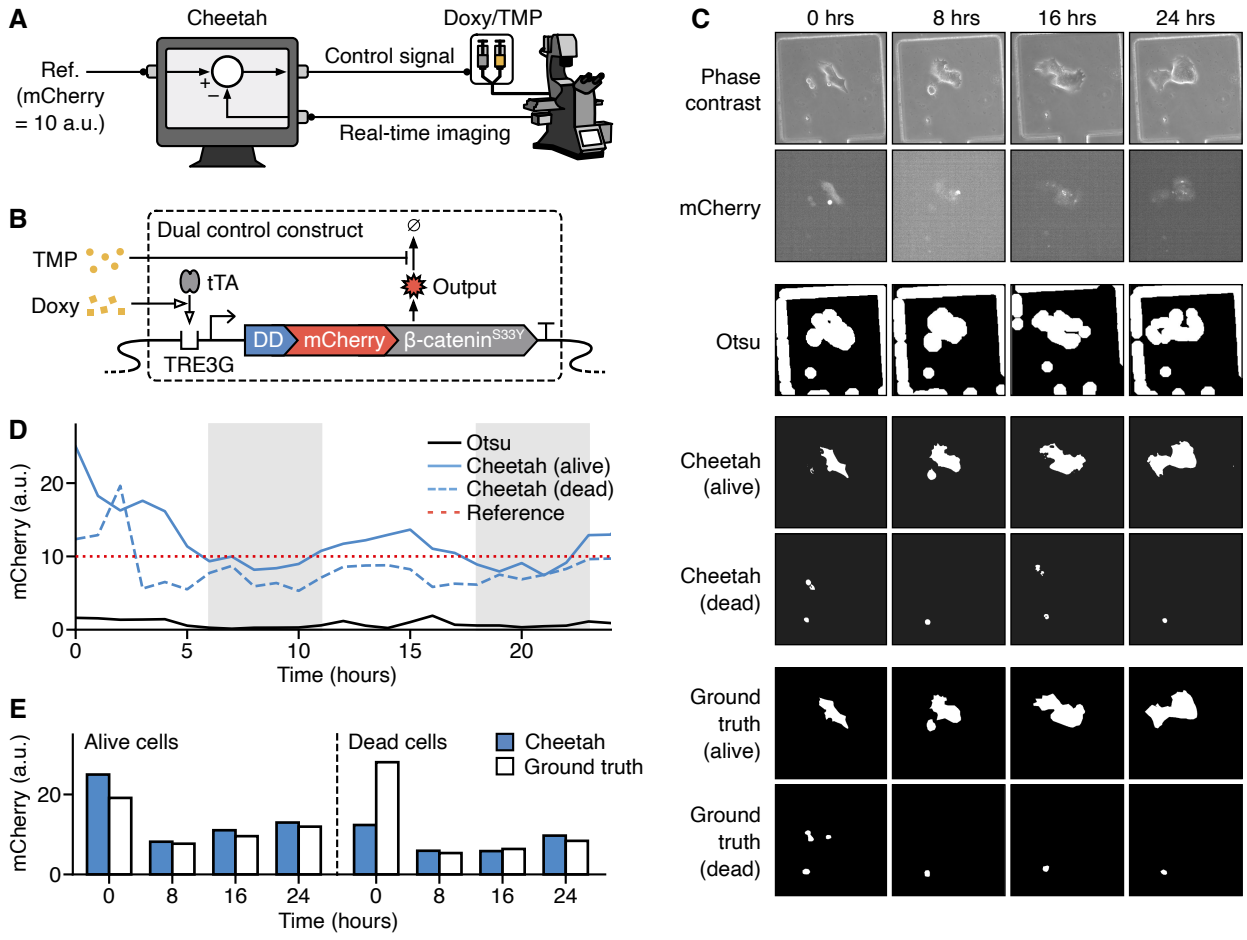

Figure 3: Closed-loop control of protein expression in mammalian cells. (**A**) Schematic of the microfluidic system used for external closed-loop control. A desired reference cellular mCherry fluorescence of 10 arbitrary units (a.u.) is shown. (**B**) Overview of genetic construct used to control mCherry expression [26]. Small molecules (TMP and Doxy) work in tandem to boost the expression level of mCherry. Regulation is due to a tetracycline transcriptional activator (tTA) and a destabilising domain (DD) which forms part of the mCherry reporter protein. (**C**) Time-lapse images of mouse embryonic stem cells (mESCs) growing in the system for phase contrast and mCherry fluorescence, as well as segmentation masks for cells generated using the Otsu method, Cheetah and manually annotated to give a ground truth (white regions denote cells). For Cheetah and the ground truth, separate masks are shown for living and dead cells. (**D**) Average mCherry fluorescence of the cell segmentation mask over time calculated using either the Otsu method or Cheetah. Red dotted line denotes the external reference that the controller aims to maintain (10 a.u.). Grey shaded regions show when the control signal triggered release of TMP and Doxy. Control signals were generated by using average mCherry fluorescence calculated using segmentation masks of alive cells from Cheetah. (**E**) Comparison of average mCherry fluorescence at specific time points during the experiment for segmentation masks generated by Cheetah and manually annotated (ground truth).

Supplementary Movie Captions

Supplementary Movie 1: Open-loop experiment of bacterial cells. Otsu- and Cheetah-based segmentation results are shown, comparing the computed masks, cell number and GFP fluorescence over time.

Supplementary Movie 2: Open-loop experiment of mouse embryonic stem cells. Otsu- and Cheetah-based segmentation results are shown, comparing the computed masks and mCherry fluorescence. Cheetah also classify cells as live and dead and provides fluorescent protein dynamics of each.

Supplementary Movie 3: External feedback control experiment of mouse embryonic stem cells performed using Cheetah-based segmentation. Offline Otsu- and online Cheetah-based segmentation results are shown, comparing the computed masks cell number and mCherry fluorescence. The control input provided during the experiment and the set-point control reference are also shown.

Supplementary Movie 4: Detailed analysis dashboard for bacteria growing in a microfluidic chip. Top two panels on the left show the phase contrast and GFP fluorescence images from the microscope. Top right panel shows detailed analysis of the phase contrast image with cells labelled by colour and a light grey bounding box and their centre of mass and major axis (i.e. orientation) denoted by a red circle and line, respectively. The bottom two panels show the time course of both cell count and single-cell GFP fluorescence (with the average shown as a solid line and ± the standard deviation depicted by the shaded area).