Cellects, a software to quantify cell expansion and motion

Aurèle Boussard*[1], Patrick Arrufat, Audrey Dussutour and Alfonso Pérez-Escudero

Rue Marianne Grunberg-Manago, 31400 Toulouse, France

Research Centre on Animal Cognition (CRCA), Centre for Integrative Biology (CBI), Toulouse University, CNRS

*Correspondence: aurele.boussard@gmail.com

[1] Lead Contact

Highlights

- We present Cellects, a tool to quantify growth and motion of organisms automatically.
- Cellects is easy to install and use thanks to its user-friendly interface.
- It works with a wide range of organisms, providing many geometrical descriptors.
- Our validation showed high accuracy even in challenging experimental conditions.

Motivation

Growth and motion quantification is a crucial step in studying the evolution, growth and behavior of many species. However, there is no free and easy to use software to automatically quantify the growth of an organism, and that works across a wide range of species. To fill this gap, we developed Cellects, an open-source software that quantifies growth and motion under any conditions and for any species.

Summary

Automated quantification offers unique opportunities to study biological phenomena, increasing reproducibility, replicability, accuracy, and throughput, while reducing observer biases. We present Cellects, a tool to quantify growth and motion in 2D. This software operates with image sequences containing specimens growing and moving on an immobile flat surface. Its user-friendly interface makes it easy to adjust the quantification parameters to cover a wide range of species and conditions, and includes tools to validate the results and correct mistakes if necessary. The software provides the region covered by the specimens at each point of time, as well as many geometrical descriptors that characterize it. We validated Cellects with

Physarum polycephalum, which is particularly difficult to detect because of its complex shape and internal heterogeneity. This validation covered five different conditions with different background and lighting, and found Cellects to be highly accurate in all cases. Cellects' main strengths are its broad scope of action, automated computation of a variety of geometrical descriptors, easy installation and user-friendly interface.

github link: https://github.com/Aurele-B/Cellects

Keywords

growth and motion quantification, 2D biological growth, automated tracking, cell expansion, biotechnology, video analysis, time-lapse analysis, software, high throughput, microbiology, mycology, pharmacology, toxicology, ecology, behavioural ecology, ethology.

Introduction

All species exhibit growth, be it at the individual level (through development) or at the population level (through reproduction). Recent advances in imaging techniques have made it relatively easy and inexpensive to capture biological growth at almost all scales, including molecular aggregation[1], bacterial growth,[2] human human brain development,[3] forest recovery[4] and many more. These techniques have produced an explosion in the availability of datasets in the form of videos or time-lapses, and analysis of these datasets has become an important bottleneck. Automating this analysis presents challenges, especially when different parts of the organism have different appearance (for example, newly formed parts usually have different coloration or density than older parts), when the contrast between the organism and the

background is poor, and when the growth pattern is complex and unpredictable. Also, different organisms exhibit very different characteristics, requiring different segmentation and analysis approaches, making it difficult to develop a general-purpose software that can quantify growth for a wide variety of organisms in challenging conditions.

The main limitations that prevent current systems from being widely used are the following. First, most growth quantification algorithms focus on a single type of organism, such as bacteria,[5–10] yeasts ,[11,12] slime molds,[13–16] plants,[17,18] tumors[19] or stem cells[20].

Second, many of them lack a graphical user-friendly interface.[5,8,10,11,13–16,21]

Third, when equipped with a user interface, these softwares are commercial or rely on a commercial programming language.[6,7,12,20] Since they are not free and open source, commercial programs cannot be improved by their users or other developers.

Fourth, apart from a few commercial programs,[7,20] automated growth quantification programs suffer from a narrow range of output variables. To the best of our knowledge, the only free and open source programs having a wide inventory of output variables required to perform the analysis manually or using macros (e.g. ImageJ and Fiji[22]).

Fifth, no published software can be adjusted to a wide variety of lighting and coloring to adapt to different types of organisms and experimental conditions.

We developed Cellects to fill these gaps in the inventory of growth quantification programs. It is easy to install, and has a user-friendly interface that makes it easy to find the right settings in a broad range of color and lighting conditions, so that it can be easily applied to diverse conditions and taxa. It outputs a wide variety of geometry descriptors in addition to its raw results, and it also includes tools to review and validate the results.

Cellects has been tested on *Physarum polycephalum* plasmodia and a developing embryo of *Coturnix japonica*, but it has been designed to be useful for many other taxa, including Fungi, Bacteria, Myxomycota, and Plants.

This paper is organized as follows. Section I presents the Cellects' capabilities and the conditions in which it can be useful. Section II briefly describes the quantification methodology. Section III presents a validation in various biological and lighting conditions. The images used for validation and the codes are available in public repositories (see data availability). A complete algorithmic description of the software can be found in the supplements (SI - Developer-oriented description of the software) and the github repository includes a user manual.

Software capabilities

Cellects quantifies the growth and motion of one or several individuals, finding the region(s) occupied by them in every frame of a video (SI - Video 1). These regions are the raw output of the software, which also computes derived variables such as area (Figure 1c), perimeter, eccentricity (Figure 1d), convexity (Figure 1e), solidity (Figure 1e), and others (SI - Table 1). Cellects provides a time series of all the variables listed in the SI - Table 1, and saves them in .csv files.

Cellects is easy to install and use. A stand-alone executable installs Cellects on Windows with a single click and no additional dependencies. Alternatively, it can be installed on any operating system as a Python package (See installation tutorial at S1). It has a user-friendly interface that helps the user select the optimal parameters for any condition (Figure 2), and also includes a tool to validate and correct mistakes (see the user manual).

Cellects has several features to increase its flexibility. First, it works on images or videos containing multiple arenas, which are automatically detected and can have a variety of shapes. Second, within each arena, Cellects can track one or several individuals, which may appear, disappear, grow, connect and disconnect with each other. Third, it can sequentially analyze several folders. Fourth, while it requires an immobile background, it is prepared to work with videos where a drift-correction has been applied.

Finally, Cellects' main strength is to be able to detect the growth of the cells even when the contrast is relatively poor. It does this by combining two ingredients: a flexible interface where the user can find the most suitable parameters, and a powerful segmentation routine that uses the growth itself to help the segmentation of each frame, using dynamical information, and not only the information contained in each image. In the next section, we will see how this algorithm works.

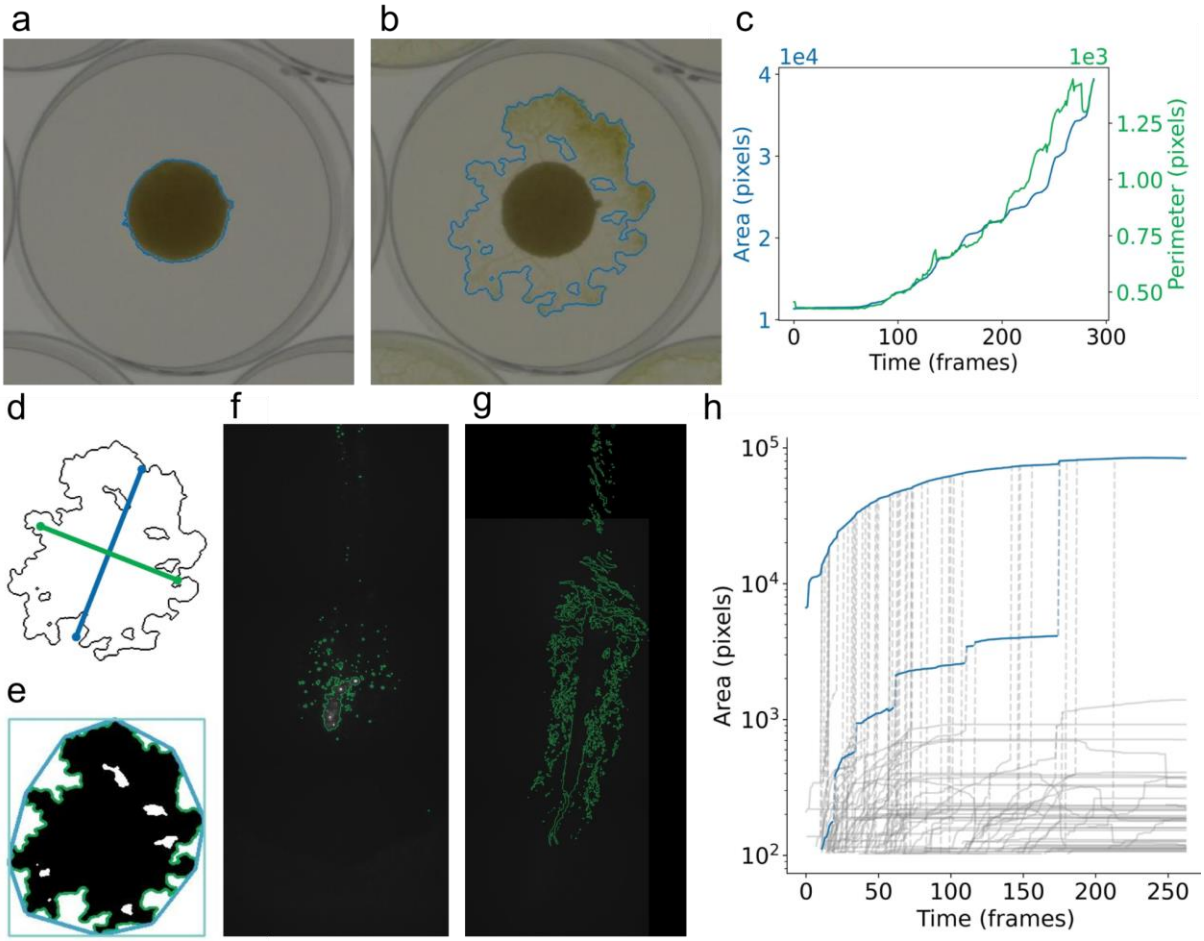

Figure 1: Cellects detects and analyzes growing or moving cells or colonies. a) Image of a giant unicellular organism (*P. polycephalum*), with cellect's initial segmentation (blue line). For a complete example see Supplemental - Video 1. **b)** Same as (a), but after 24h. **c)** Area and perimeter of the organism shown in boxes (a) and (b), over time. d) Directions for the major and minor axes of the shape detected in (b). Eccentricity measures the length difference between these two axes. e) Minimal convex hull (blue) and minimal bounding rectangle (green). Convexity compares the perimeter of the cell with that of the minimal convex hull, while solidity compares the areas. Rectangularity and circularity quantify the similarity of the cell with a rectangle and a circle, respectively. g) Image of cellular progenitors marked with GFP during the development of a quail embryo (*C. japonica*, caudal part of stage HH11, see [23]). Green lines show the segmentation. h) Same as (g), but for a later time step. **g)** and **h)** images of the segmentation of the growth and motion of cellular progenitors. **f)** Area against time of connected groups of the cells shown in (g), (h) Each vertical line is a fusion event between two groups of cellular progenitors. One of these fusions has been highlighted in blue for clarity.

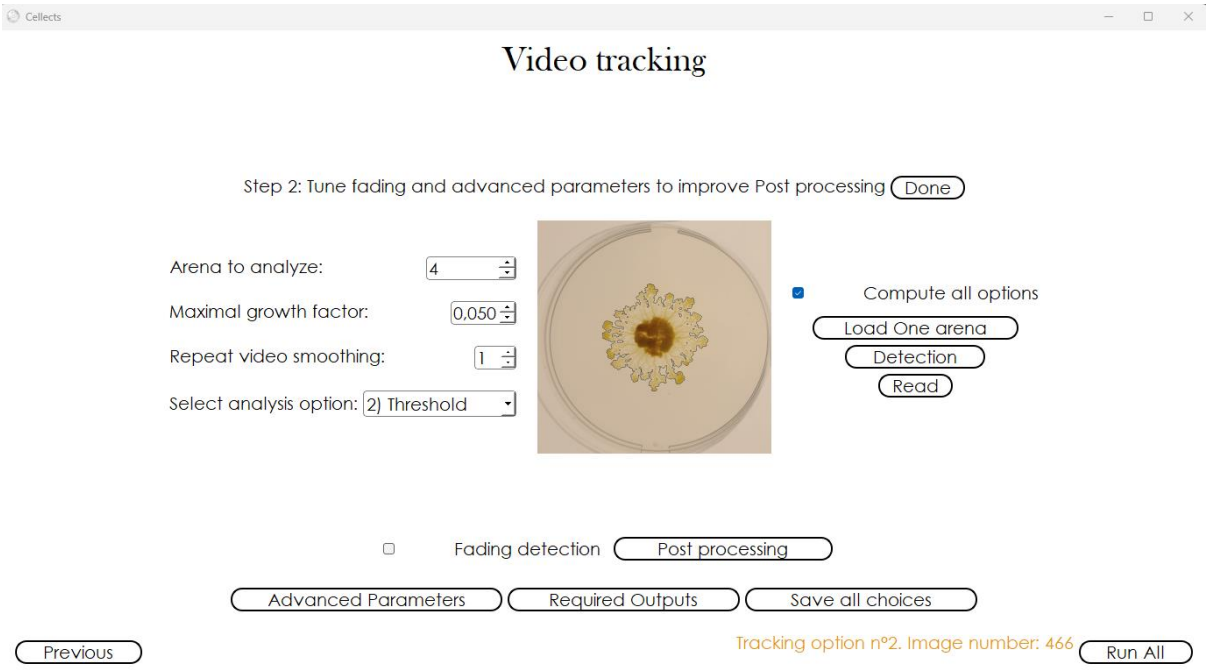

Figure 2: Snapshot of the user interface during the analysis of one arena. The [github repository]() includes an updated user manual.

Image analysis methodology

Cellects uses traditional computer vision algorithms, combining different techniques to enhance flexibility. An automatic algorithm selects the most promising method for each dataset, so that in most cases the user does not need to fine-tune any analysis parameters. The analysis proceeds in four steps: perform segmentation of the first image, delineate arenas, perform segmentation of the whole video, and improve that video segmentation.

The first step of the process converts the first image (Figure 3a) into a grayscale image that maximizes contrast between the specimens and the background (Figure 3b). To do this, Cellects uses linear combinations of several color channels, from different color spaces, and displays the resulting segmentation in the interface (Figure 3c in blue). The user can easily choose among several automatically generated combinations, selecting the one that provides the best

segmentation (See S3.1 for more information). In the few cases in which none of them works well enough, the user can fine tune them in three ways: With a drawing tool that tells the software where to find cell/background to improve its automatic training, by manually changing the parameters that determine the linear combination of the color spaces, or choosing a different segmentation algorithm (between otsu thresholding and k-means). When selected, background subtraction occurs just before segmentation. This manual fine-tuning of the parameters typically takes a few minutes and greatly enhances the ability of Cellects to work with diverse specimens and experimental conditions.

Once the segmentation of the first frame is done, Cellects uses the initial position of the specimens to automatically detect all the arenas (whenever there is more than one; see details in S3.2). Arena detection can also be corrected manually if needed. Then, the images are cropped to analyze each arena separately, and from now on we will describe the process for a single arena.

The third step is the segmentation of all frames of each single-arena video. This segmentation will consist in transforming the grayscale video resulting from the color space combination conversion into a binary video of presence/absence. To do this, Cellects provides several different options to detect specimen motion and growth throughout the video (see a detailed description in S3.4): In simple datasets with strong contrast between specimens and background, Cellects can simply segment each image by thresholding. In more challenging conditions, the algorithm tracks the intensity of each pixel over time, using this dynamical information to determine when a pixel has been covered. This is done through an automatically determined threshold on the intensity (Figure 3d, blue) or on its derivative (Figure 3d, green). Additionally, Cellects can apply the logical operators AND or OR to these intensity and

derivative thresholds. The default option is the dynamical intensity threshold and it works in many cases, but the user interface lets the user quickly check the results of different options and choose the best one by visual inspection of the segmentation result in different frames (Figure 3e).

The final step consists in improving the segmentation (see S3.5 for more information). Cellects will first apply several filters that consistently improve the results, such as checking that each detected pixel was also detected at least twice in the three previous frames, omitting images containing too many detected pixels, and performing morphological opening and closing. Optionally, the user can activate the detection of areas left by the cell (See S3.5.B for details).

Additionally, two optional algorithms correct particular types of errors. The first algorithm is useful when the substrate on which the cells are at the first image is of a different color than the substrate on which they will grow, expand or move. This color difference may produce holes in the segmentation and we developed an optional algorithm to correct this kind of error around the initial shape. The second algorithm should be used when each arena contains a single specimen, which should generate a single connected component. We can use this information to correct mistakes in models such as *P. polycephalum*, whose strong heterogeneity produces large variations of opacity. In these cases, segmentation may fail in the most transparent parts of the specimens and identify two disconnected components (Figure 3f,g). The correction algorithm merges these disconnecting components by finding the most likely pixels connecting them and the most likely times at which those pixels were covered during growth (Figure 3h).

Once the image analysis is finished, the software determines the value of each morphological descriptor at each time frame (SI - Table 1). Finally, Cellects saves a new video for each arena with the original video next to the converted video displaying the segmentation result, so that the user can easily validate the result. If an arena shows a poor segmentation result, the user can re-analyze it, tuning all parameters for that specific arena.

To create Cellects and this paper, we used python 3.10 and the python packages Numpy,[24] OpenCV,[25] Pandas,[26] SciPy,[27] Matplotlib.[28] (See more in SI - Table 2)

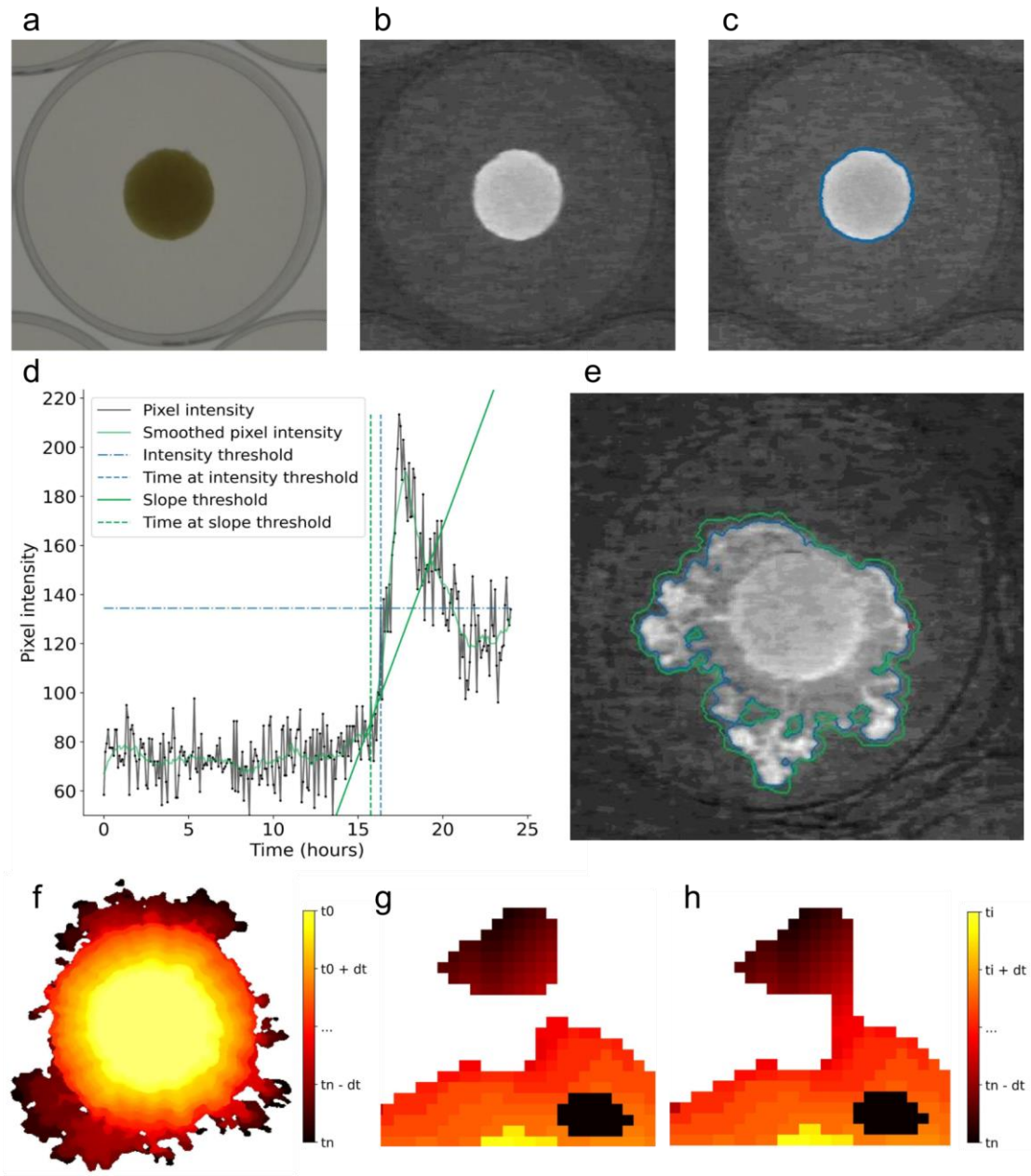

Figure 3: Specimens segmentation algorithms. a) First original image of one arena. **b)** Greyscale image of the same arena after conversion. **c)** Same as (b), showing the result of segmentation (blue line). **d)** Intensity curve for one pixel (marked in red in box e). Blue horizontal dashed-dotted line: Intensity threshold. Blue vertical dashed line: Time at which the intensity threshold is reached. Green vertical dashed line: Time at which the derivative threshold (illustrated by the solid green line) is reached. In this example, the slope segmentation method detects the covering of the pixel before the threshold segmentation method. Because of that, the area within the green contour is larger than the area within the blue contour in e). **e)** Grayscale image at time t=16:40, showing two possible segmentations (intensity in blue, derivative in green). Red dot: Pixel whose time series is shown in (d).**f)** Segmentation of a cell, showing a small disconnected component in the bottom. **g)** Zoom on the upper part of (g), showing a disconnection. h) Dynamic fixing of the disconnection. Colors indicate the estimated time at which each pixel was covered. Ranges from t0=0:0 to the current time tn=21:20 with an arbitrary time step dt.

Validation

We have validated Cellects on time-lapses of circular or square samples of the giant unicellular *P. polycephalum* (See the experimental procedures in S4). The structural heterogeneity of this cell gives it variations in transparency,[29] making it hard to detect, both with the naked eye and with a computer. Therefore, a program that can detect this cell should be able to detect more homogeneous specimens with less difficulty. We performed the validation in a wide range of conditions, in order to cover as many types of experiments as possible. In particular, our validation covered the following conditions:

a) High contrast, single color: LED's provided a bright, uniform and high-contrast background. The substrate underwent no significant desiccation, ensuring uniform imaging quality throughout the experiment, and image resolution was high (Fig 4a). This is our reference for optimal imaging conditions.
b) High contrast, multiple colors: good contrast but each arena contained a food patch in addition to the cells, adding another color to the immediate background (Fig 4b). Also, the absence of a protecting petri dish increased agar desiccation, which induced changes in the appearance of the cells and substrate as the experiment progressed.
c) Low contrast, dark and heterogeneous background, and significant desiccation (Fig 4c).
d) Lower contrast, bright homogeneous background, and no significant desiccation (Fig 4d).
e) Low contrast and low resolution: These experiments were performed on a non-transparent agar bridge, which decreases contrast with the cell, and the lowest resolution (Fig 4e).

We randomly selected 60 arenas per experiment to assess Cellects' accuracy. An ideal validation would compare Cellects' output with an independent annotation made by a human expert operator. However, this was not possible in our case: in our conditions contrast is often so low that, while it is easy to distinguish when a segmentation is clearly incorrect, in many cases there are several possible segmentations that seem correct (i.e. different expert operators, or even the same operator at different times, would choose different options). Because of this limitation, we first ran Cellects on our validation samples, and then reviewed all of them manually, tagging all cases in which the human operator was certain that there was a segmentation mistake. This method introduces a small bias in favor of Cellects, because the human operator might miss some mistakes or be too permissive in doubtful cases. However, the area covered by these doubtful regions is negligible in comparison with the areas affected by the clear mistakes that were unambiguously detected by the human operator, so we expect that a different validation with an independent ground truth would produce a virtually identical result.

Therefore, our validation procedure was as follows. First, we run Cellects on the videos of each of the experiments, following steps 1-4 described in section II: For each experiment, we selected the optimal option to create maximum-contrast grayscale images, made sure that all arenas were properly detected, selected the optimal option for the dynamical segmentation, and selected the optimal post-processing options. This tuning took approximately 20 minutes for each experiment (see all parameters in SI - Table 3). Then, we run the program to perform automatic segmentation. It is important to note that we did not perform any correction after the automatic segmentation. As described in section II Cellects includes a tool to re-analyze defective arenas with different parameters, but we did not use it to validate the software.

For each experiment, we loaded the result of the segmentation of each arena (blue lines in Figure 4a-e, bottom) in image J[22] and we manually corrected all detectable errors (red lines in Figure 4a-e, bottom). We then compared Cellects segmentations with the corrected images to get the proportion of the cell that has been accurately detected (cell accuracy = true positives / (true positives + false negatives)) and the proportion of the background that has been accurately detected (Background accuracy = true negatives / (true negatives + false positives)). For each experiment, we averaged these two accuracies to estimate the overall accuracy (Figure 4f).

In the best case scenario (experiment a, with high contrast, high resolution, uniform background, and weak desiccation), Cellects accurately detected both the cells and the background (99.76% average accuracy).

Experiments b and c still achieved maximum accuracy in the detection of the cells, but the more complex background and the desiccation of the agar made it more challenging to correctly identify the background. These challenges decreased the accuracy of the background detection only slightly, down to 97.74% in b and 97.50% in c. Experiment d had very poor contrast and significant desiccation, causing a slight decrease in accuracy both in the detection of the cells (down to 98.74%) and of the background (down to 96.86%). The poor resolution of experiment d produced similarly small drops in accuracy (97.66% for the cells and 97.56% for the background). In sum, Cellects showed excellent performance, with an accuracy higher than 97% even in the most challenging condition.

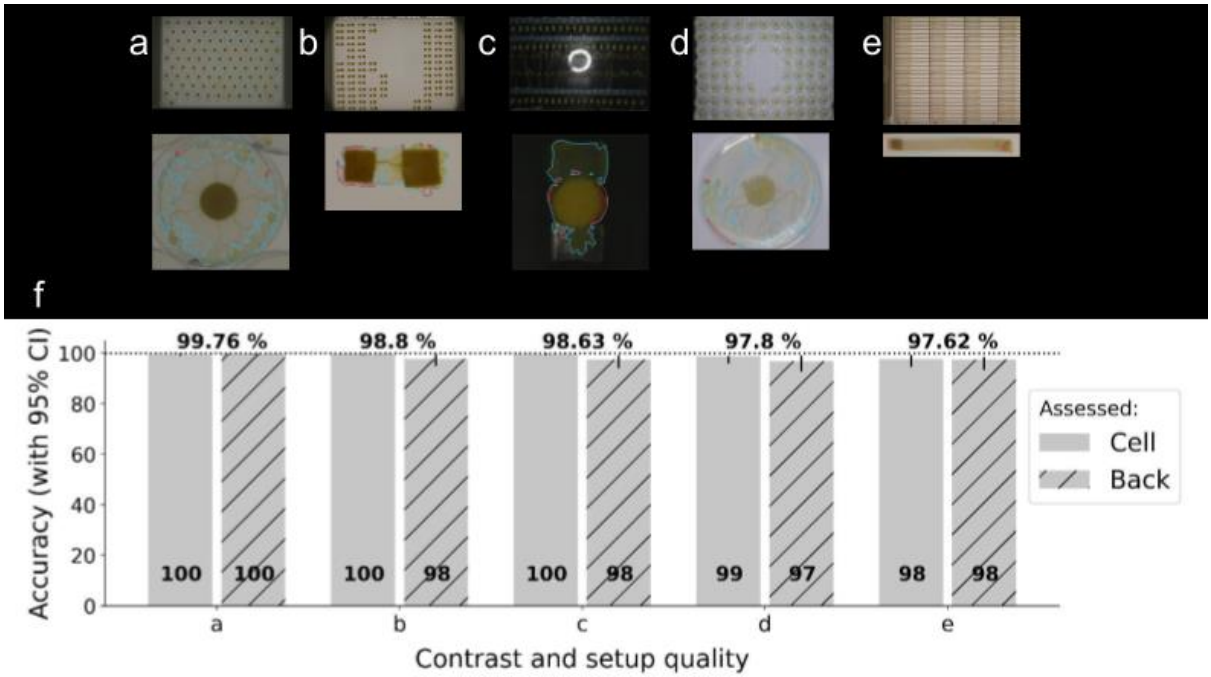

Figure 4: Accuracy of Cellects in 5 conditions with different types of backgrounds and arenas. a-e) Snapshots of the five conditions. The upper image contains the complete setup and the lower one is a zoom on one particular arena where we found segmentation mistakes. Cellects segmentation is in blue and manual correction is in red. **a)** High contrast and optimal setup. **b)** High contrast, heterogeneous colors. **c)** Low contrast and setup prone to desiccation. **d)** Low contrast and optimal setup. **e)** Low resolution. **f)** Barplot of the corresponding accuracies for each of the 5 conditions. The accuracy of Cellects to detect cellular material is in solid gray. The accuracy of Cellects to detect the background is in hatched gray. The average of these two accuracies (named overall accuracy) is written above each condition in black.

Discussion

Cellects is an image analysis software to study the growth or exploratory behavior of cells or cell colonies. It provides a wide range of dynamical and morphological descriptors to do so, and it is designed and validated to work in a wide range of conditions.

Cellects may be applied to many different studies. It may be used to test specific predictions by measuring one specific variable. For example, we know that the oscillatory dynamics of *P. polycephalum* play a role in the chemotactic behavior of this organism.[30] Cellects could be a

good tool to improve our understanding of this causal link by highlighting the effect of oscillatory variables on morphological or speed-related variables. Future versions of Cellects may have additional features to improve this analysis, such as methods to extract *P. polycephalum*'s tubular network, quantify its oscillations, and compute its fractal dimension. It may also be used to study correlations or causations between various factors. In this context, its applications can range from the effects of pollutants on living organisms to research for gene pleiotropic effects.

Cellects uses traditional image analysis techniques, combining several of them to increase its flexibility. Its strength lies in a combination of automatic algorithms to select the best analysis technique for each dataset, coupled with a user-friendly interface that facilitates parameter tuning for difficult datasets.

Quantifying cells automatically allows transparent and reproducible analysis while reducing the observer bias. Since Cellects can be used in various conditions and to study various biological models, it can be a valid tool for studying context-dependent factors as well as interspecific comparisons.

Author contribution statement

Conceptualization, A.B., P.A., A.D., A.P.E.; methodology, A.B., P.A.; software, A.B.; investigation, A.B., A.D.; resources, A.B., P.A., A.D., A.P.E.; writing – original draft preparation, A.B.; writing – review and editing, A.B., A.P.E; visualization, A.B.; supervision, P.A., A.D., A.P.E.; project administration, A.B.; funding acquisition, A.D., A.P.E.

Funding statement

A.B. was supported by a grant from the 'Agence Nationale de la recherche' (ANR), reference no. ANR-17-CE02- 0019-01-SMARTCELL (PI A.D.). A.D. acknowledges support by the CNRS. APE acknowledges support from a CNRS Momentum grant and an ANR JCJC grant (ANR-22-CE02-0002, ForAnInstant).

The funders had no role in study design, data collection and interpretation, or the decision to submit the work for publication.

Data availability statement

Lead contact

Aurèle Boussard: aurele.boussard@gmail.com, ORCID: 0000-0002-6083-4272

Data and code availability

The data and the code to replicate the analyses can be found at:

https://datadryad.org/stash/share/nCvWIZoZ8-Wnxm0CjnPbbznUPw90RYdo1YVJEQkfLIY.

In addition, this repository contains an executable and a zip file to install Cellects on windows and a python wheel file to install it on macOs or Linux. The github repository of Cellects, available at https://github.com/Aurele-B/Cellects, contains its source code and a complete user manual (at https://github.com/Aurele-B/Cellects/blob/main/UserManual.md).

Boussard, Aurèle; Arrufat, Patrick; Dussutour, Audrey; Pérez-Escudero, Alfonso (2023). Cellects, a software to quantify cell expansion and motion [Software]. Zenodo. https://doi.org/10.5281/zenodo.8184830

Boussard, Aurèle; Arrufat, Patrick; Dussutour, Audrey; Pérez-Escudero, Alfonso (2023). Cellects, a software to quantify cell expansion and motion [Dataset]. Dryad. https://doi.org/10.5061/dryad.7wm37pvzp

Declaration of interests statement

The authors declare no competing interests.

Acknowledgements

We thank Audrey Bizet for her work on the experiment of Figure 4a, Charlotte Dupont and Paul-Antoine Badon for the experiment of Figure 4b, Nirosha Murugan for her help with the experiment of Figure 4d, Ana Lucía Morán Hernández for her help with the experiment of Figure 4e, and Florent Le Moël and Remi Giorno dit Journo for their help during software development.

Supplementary information

Supplementary information.docx

Supplemental - Video 1.mp4

Installation.md

UserManual.md

References

1. Jyothi, M., Nagarajan, V., and Chandiramouli, R. (2020). Investigation on adsorption properties of HCN and ClCN blood agents on teta–phosphorene nanosheets–A first–principles insight. Chem. Phys. *538*, 110896.
2. Moller, S., Kristensen, C.S., Poulsen, L.K., Carstensen, J.M., and Molin, S. (1995).

Bacterial growth on surfaces: automated image analysis for quantification of growth rate-related parameters. Appl. Environ. Microbiol. *61*, 741–748.
3. Rajagopalan, V., Scott, J., Habas, P.A., Kim, K., Corbett-Detig, J., Rousseau, F., Barkovich, A.J., Glenn, O.A., and Studholme, C. (2011). Local Tissue Growth Patterns Underlying Normal Fetal Human Brain Gyrification Quantified In Utero. J. Neurosci. *31*, 2878–2887. 10.1523/JNEUROSCI.5458-10.2011.
4. Hird, J.N., Kariyeva, J., and McDermid, G.J. (2021). Satellite Time Series and Google Earth Engine Democratize the Process of Forest-Recovery Monitoring over Large Areas. Remote Sens. *13*, 4745.
5. Ernebjerg, M., and Kishony, R. (2012). Distinct growth strategies of soil bacteria as revealed by large-scale colony tracking. Appl. Environ. Microbiol. *78*, 1345–1352.
6. Levin-Reisman, I., Fridman, O., and Balaban, N.Q. (2014). ScanLag: high-throughput quantification of colony growth and lag time. J. Vis. Exp. JoVE.
7. Bär, J., Boumasmoud, M., Kouyos, R.D., Zinkernagel, A.S., and Vulin, C. (2020). Efficient microbial colony growth dynamics quantification with ColTapp, an automated image analysis application. Sci. Rep. *10*, 16084.
8. Koch, R.A., Harmel, C., Alber, M., and Bahn, E. (2021). A framework for automated time-resolved analysis of cell colony growth after irradiation. Phys. Med. Biol. *66*, 035017.
9. Pandey, S., Park, Y., Ankita, A., and Phillips, G.J. (2021). Scan4CFU: Low-cost, open-source bacterial colony tracking over large areas and extended incubation times. HardwareX *10*, e00249.
10. Spedalieri, G., Piersimoni, L., Laurino, O., Braunstein, S.L., and Pirandola, S. (2020). Detecting and tracking bacteria with quantum light. Phys. Rev. Res. *2*, 043260.
11. Falconnet, D., Niemistö, A., Taylor, R., Ricicova, M., Galitski, T., Shmulevich, I., and Hansen, C.L. (2011). High-throughput tracking of single yeast cells in a microfluidic imaging matrix. Lab. Chip *11*, 466–473.
12. Doncic, A., Eser, U., Atay, O., and Skotheim, J.M. (2013). An algorithm to automate yeast segmentation and tracking. PLoS One *8*, e57970.
13. Halvorsrud, R., and Wagner, G. (1998). Growth patterns of the slime mold Physarum on a nonuniform substrate. Phys. Rev. E *57*, 941.
14. Vogel, D., Nicolis, S.C., Perez-Escudero, A., Nanjundiah, V., Sumpter, D.J., and Dussutour, A. (2015). Phenotypic variability in unicellular organisms: from calcium signalling to social behaviour. Proc. R. Soc. B Biol. Sci. *282*, 20152322.
15. Dirnberger, M., Mehlhorn, K., and Mehlhorn, T. (2017). Introducing the slime mold graph repository. J. Phys. Appl. Phys. *50*, 264001.
16. Patino-Ramirez, F., Arson, C., and Dussutour, A. (2021). Substrate and cell fusion influence on slime mold network dynamics. Sci. Rep. *11*, 1–20.
17. French, A., Ubeda-Tomás, S., Holman, T.J., Bennett, M.J., and Pridmore, T. (2009). High-throughput quantification of root growth using a novel image-analysis tool. Plant Physiol. *150*, 1784–1795.
18. Felekis, D., Muntwyler, S., Vogler, H., Beyeler, F., Grossniklaus, U., and Nelson, B.J. (2011). Quantifying growth mechanics of living, growing plant cells in situ using microrobotics. Micro Nano Lett. *6*, 311–316.
19. Haney, S.M., Thompson, P.M., Cloughesy, T.F., Alger, J.R., and Toga, A.W. (2001). Tracking tumor growth rates in patients with malignant gliomas: A test of two algorithms. Am. J. Neuroradiol. *22*, 73–82.
20. Yoshida, K., Okada, M., Nagasaka, R., Sasaki, H., Okada, M., Kanie, K., and Kato, R. (2019). Time-course colony tracking analysis for evaluating induced pluripotent stem cell culture processes. J. Biosci. Bioeng. *128*, 209–217.
21. Patino-Ramirez, F., Boussard, A., Arson, C., and Dussutour, A. (2019). Substrate

composition directs slime molds behavior. Sci. Rep. *9*, 1–14.
22. Schneider, C.A., Rasband, W.S., and Eliceiri, K.W. (2012). NIH Image to ImageJ: 25 years of image analysis. Nat. Methods *9*, 671–675.
23. Romanos, M., Allio, G., Roussigné, M., Combres, L., Escalas, N., Soula, C., Médevielle, F., Steventon, B., Trescases, A., and Bénazéraf, B. (2021). Cell-to-cell heterogeneity in Sox2 and Bra expression guides progenitor motility and destiny. ELife *10*, e66588.
24. Harris, C.R., Millman, K.J., Van Der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., et al. (2020). Array programming with NumPy. Nature *585*, 357–362.
25. Bradski, G. (2000). The openCV library. Dr Dobbs J. Softw. Tools Prof. Program. *25*, 120–123.
26. McKinney, W. and others (2010). Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference (Austin, TX), pp. 51–56.
27. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat. Methods *17*, 261–272.
28. Hunter, J.D. (2007). Matplotlib: A 2D graphics environment. Comput. Sci. Eng. *9*, 90–95.
29. Oettmeier, C., Brix, K., and Döbereiner, H.-G. (2017). Physarum polycephalum—a new take on a classic model system. J. Phys. Appl. Phys. *50*, 413001.
30. Durham, A., and Ridgway, E. (1976). Control of chemotaxis in Physarum polycephalum. J. Cell Biol. *69*, 218–223.
31. Rocha, L., Velho, L., and Carvalho, P.C.P. (2002). Image moments-based structuring and tracking of objects. In Proceedings. XV Brazilian Symposium on Computer Graphics and Image Processing (IEEE), pp. 99–105.
32. Vogel, D., Gautrais, J., Perna, A., Sumpter, D.J.T., Deneubourg, J.-L., and Dussutour, A. (2016). Transition from isotropic to digitated growth modulates network formation in Physarum polycephalum. J. Phys. Appl. Phys. *50*, 014002. 10.1088/1361-6463/50/1/014002.
33. Boussard, A., Fessel, A., Oettmeier, C., Briard, L., Döbereiner, H.-G., and Dussutour, A. (2021). Adaptive behaviour and learning in slime moulds: the role of oscillations. Philos. Trans. R. Soc. B Biol. Sci. *376*, 20190757. 10.1098/rstb.2019.0757.

Article info

Copyright

© 2024, Boussard et al.

This article is distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use and redistribution provided that the original author and source are credited.