



Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

SpheroidJ: An Open-Source Set of Tools for Spheroid Segmentation

David Lacalle^a, Héctor Alfonso Castro-Abril^{b,c,d}, Teodora Randelovic^{b,c}, César Domínguez^a,
Jónathan Heras^{a,*}, Eloy Mata^a, Gadea Mata^e, Yolanda Méndez^a, Vico Pascual^a,
Ignacio Ochoa^{b,c,f}

^a Department of Mathematics and Computer Science, University of La Rioja, Spain

^b Tissue MicroEnvironment (TME) lab, Institute for Health Research Aragón (IIS Aragón), Zaragoza, Spain

^c Aragon Institute of Engineering Research (I3A), University of Zaragoza, Zaragoza, Spain

^d Grupo de modelado y métodos numéricos en Ingeniería, Universidad Nacional de Colombia, Colombia

^e Confocal Microscopy Core Unit, Spanish National Cancer Research Centre, Madrid, Spain

^f Biomedical Research Networking Centre in Bioengineering, Biomaterials and Nanomedicine (CIBER-BBN), Instituto de Salud Carlos III, Madrid, Spain

ARTICLE INFO

Article history:

Received 9 September 2020

Accepted 8 November 2020

Available online xxx

Keywords:

Spheroids

Segmentation

Deep Learning

ImageJ

Java

Python

ABSTRACT

Background and objectives: Spheroids are the most widely used 3D models for studying the effects of different micro-environmental characteristics on tumour behaviour, and for testing different preclinical and clinical treatments. In order to speed up the study of spheroids, imaging methods that automatically segment and measure spheroids are instrumental; and, several approaches for automatic segmentation of spheroid images exist in the literature. However, those methods fail to generalise to a diversity of experimental conditions. The aim of this work is the development of a set of tools for spheroid segmentation that works in a diversity of settings.

Methods: In this work, we have tackled the spheroid segmentation task by first developing a generic segmentation algorithm that can be easily adapted to different scenarios. This generic algorithm has been employed to reduce the burden of annotating a dataset of images that, in turn, has been employed to train several deep learning architectures for semantic segmentation. Both our generic algorithm and the constructed deep learning models have been tested with several datasets of spheroid images where the spheroids were grown under several experimental conditions, and the images acquired using different equipment.

Results: The developed generic algorithm can be particularised to different scenarios; however, those particular algorithms fail to generalise to different conditions. By contrast, the best deep learning model, constructed using the HRNet-Seg architecture, generalises properly to a diversity of scenarios. In order to facilitate the dissemination and use of our algorithms and models, we present SpheroidJ, a set of open-source tools for spheroid segmentation.

Conclusions: In this work, we have developed an algorithm and trained several models for spheroid segmentation that can be employed with images acquired under different conditions. Thanks to this work, the analysis of spheroids acquired under different conditions will be more reliable and comparable; and, the developed tools will help to advance our understanding of tumour behaviour.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Cancer is the collective denomination for a group of diseases characterised by abnormal cell growth that can potentially disseminate, invade, and colonise different parts of the body. It is the second leading cause of death in the world, with approximately

9.6 million deaths per year [1]. A lot of important mechanisms of tumour progression have been described, which enabled development of treatments for various tumour types. However, the process of implementing one treatment in clinics is a long, expensive and complex process, as the treatment has to pass different proof stages. Namely, from tens of thousands of drugs tested, only one gets the approval for use [2]. This happens because most investigations have been done in two-dimensional (2D) cell culture, and on animal models. However, none of them represents properly the human organism and its response to treatments.

* Corresponding author.

E-mail address: jonathan.heras@unirioja.es (J. Heras).

For example, 2D cultures fail to reproduce the complex structure of tumours and their interactions with the surrounding tissue, whereas animal models fail to fully mimic the *in vivo* situation of a human cancer patient. Hence, it is important to have a biomimetic preclinical model since such models may shorten preclinical trials and give more reliable results [3,4]. Lately, three-dimensional (3D) cell cultures are being developed to include cell-cell and cell-extracellular matrix interactions and all physico-chemical characteristics of microenvironment, as they have been described to play an important role in tumour progression and response to treatment [5].

Spheroids are the most widely used 3D models since they can be used for studying the effects of different micro-environmental characteristics on tumour behaviour and for testing different pre-clinical and clinical treatments. They are cellular aggregates that represent correctly cell-cell interactions, and formation of oxygen and nutrient gradients [6]. These induce the formation of necrotic core inside the spheroid, a common feature of solid tumours that is impossible to reproduce in 2D systems. Spheroids can be grown in suspension, replicating isolated solid tumour, or embedded in extracellular matrix proteins, simulating the invasive capacity of tumour cells. Both necrosis and invasion are indicators of tumour progression and prognosis and their inclusion in a 3D model is essential for obtaining a more accurate representation of an *in vivo* cancer.

However, since the interactions in 3D models are radically different from traditional 2D cultures, changes in imaging systems and analysing programs must be made to capture the new complexities. In particular, imaging methods that automatically segment and measure batches of spheroid images are instrumental for further analysis. Several software tools for spheroid segmentation are available in the literature in the form of ImageJ plugins [7,8], Matlab packages [9,10] or standalone programs [11,12]. In addition, several commercial systems, like Celigo [13] or Phaedra [14]; or tools designed to work with concrete microscopes, such as Re-ViMS [15] and qVista [16], have been released. Due to the variance in sizes, shapes and textures of spheroids, all these tools are specialised in images acquired under certain conditions, and fail to generalise properly. An approach to deal with the generalisation problem is the application of data-based methods like deep learning [17,18]; however, deep learning models for spheroid segmentation [19] are not freely available, and have not been tested in a diversity of experimental conditions.

In this paper, we approach the generalisation problem by combining traditional imaging processing methods and deep learning techniques. Namely, the contributions of this work are as follows:

- First of all, we present a generic spheroid segmentation algorithm that can be particularised to different conditions, see Section 3. Such a segmentation algorithm has been employed to create a dataset of annotated images to train several deep-learning based segmentation models, see Section 4. The dataset of images is freely available at the project webpage for further comparisons.
- We have conducted a thorough comparison of different variants of our generic algorithm and our deep learning models with images of spheroids with different sizes, shapes and textures, see Section 5. We also compare our approach with several open-source tools, and show how well it generalises to different experimental conditions.
- Finally, we have released our algorithms and models in the form of an open-source and freely available set of tools; namely, an ImageJ plugin and a user-friendly and standalone application called SpheroidJ, see Section 6.

2. Materials and methods

In our experiments, we have employed images from two different tumour spheroids under different experimental conditions. In addition, images were captured using different equipment (microscopes) and conditions (focus and magnification).

Human glioblastoma cell lines U87-MG and U251-MG and colorectal cancer cell line HCT-116 were purchased from Sigma Aldrich and American Type Culture Collection, respectively. All cell lines were cultured in high-glucose Dulbecco's modified Eagle's medium (DMEM) (Lonza, BE12-614F), supplemented with 10% foetal bovine serum (FBS) (Sigma, F7524), 1% L-glutamine (Lonza, 17-605C) and 1% penicillin/streptomycin (Lonza, 17-602E). In order to follow HCT-116 cells easier during a long period of time, they were transduced with a green fluorescent protein-expressing lentiviral vector, so while alive, cells produce fluorescent protein. All cell lines were grown in humidified incubator with 5% CO₂ and trypsinised twice a week.

Spheroids were formed using hanging drop method, which ensures the formation of one single spheroid per drop. Shortly, cells were harvested and resuspended at 40000 cells/mL in complete DMEM medium supplemented with 20% methocel. Drops of 25 μ L were placed on the top of a petri dish and left for 48h for spheroid formation. For suspension culture, spheroids were transferred to round bottom 96 well plate (Sarstedt, 83.3925.500) treated with Anti-adherence rinsing solution (Stemcell, 07010). To investigate the importance of nutrients or growth factors present in microenvironment, spheroids were grown in media with different chemical composition. Besides, suspension culture was used to evaluate the efficacy of tested drugs. For invasion assays, spheroids were embedded in rat tail type I collagen hydrogels. Different final concentrations of collagen enabled studying the effect of different matrix stiffness on spheroid behaviour.

Spheroid growth and invasion were followed for up to two months by brightfield and fluorescence imaging, using Nikon Eclipse Ti-E C1 and Leica DMi8 microscopes. Transduced cells were imaged using GFP filter set. Images were acquired using 2x and 10x magnification on Nikon microscope and 5x on Leica microscope. The images were organised in 6 datasets (3 brightfield datasets and 3 fluorescence datasets), and their features are summarised in Table 1 – the datasets are available at the project webpage. In addition to those datasets, we have also employed the dataset provided in [7]. A sample from each dataset is provided in Fig. 1. As can be seen in such a figure, our images contain a single spheroid per image since we employ the hanging drop method, which ensures the formation of one single spheroid per drop, and permits us to study the behaviour of the tumour and the effect of tested compounds, excluding interactions with other spheroids.

3. A generic segmentation algorithm

In this section, we present our generic algorithm for segmenting spheroids. Such an algorithm can be particularised in different ways to produce distinct segmentation procedures that are useful for several scenarios.

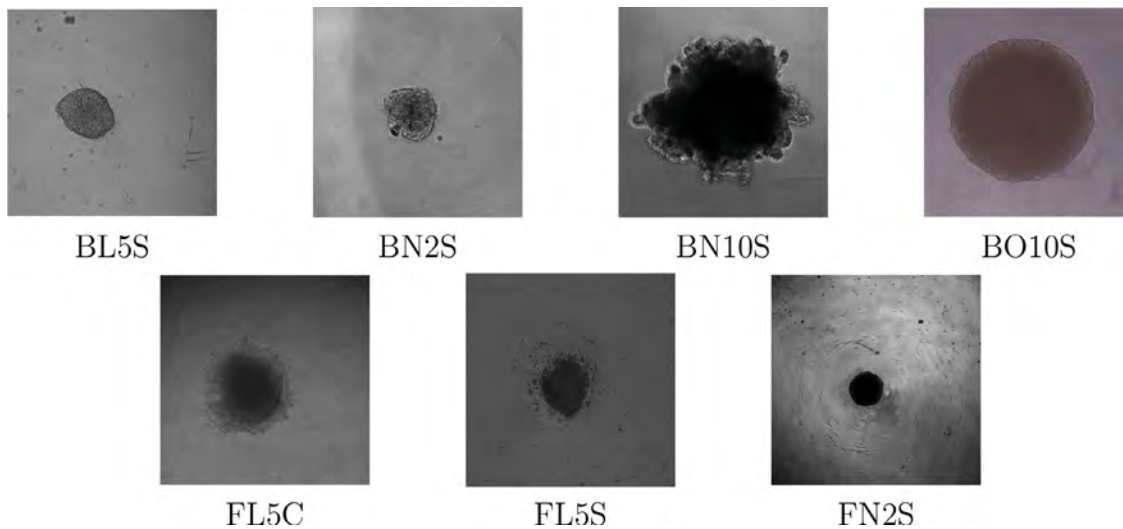
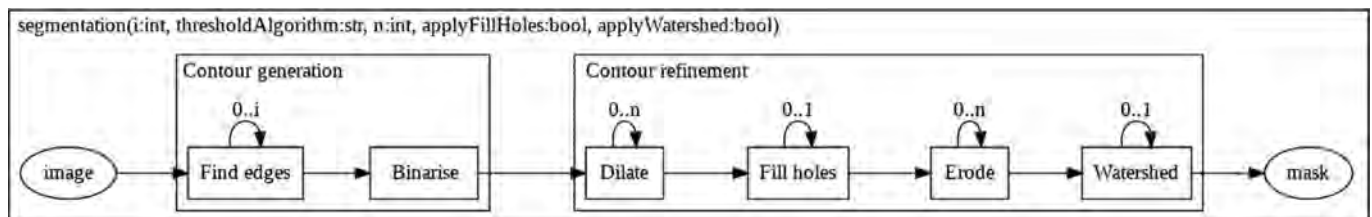
3.1. Generic segmentation algorithm

Given an image containing a spheroid, our generic algorithm aims to produce a mask for the region that contains it. Our algorithm, that is diagrammatically described in Fig. 2, is based on the sequential application of several image processing techniques, such as edge detection or thresholding, and morphological operations like dilation or erosion. Namely, the procedure can be split into two steps: contour generation and contour refinement.

Table 1

Features of the 7 datasets employed in this work. The datasets are named with the following convention: the first character of the name indicates whether is a brightfield (B) or a fluorescence dataset (F); the second, the microscope; the third, the magnification; and, the fourth, the culture media.

Dataset	Method	# Images	Image size	Microscope	Magnification	Format	Type	Culture
BL5S	Brightfield	50	1296 × 966	Leica	5x	TIFF	RGB	Suspension
BN2S	Brightfield	154	1002 × 1004	Nikon	2x	ND2	Gray 16bits	Suspension
BN10S	Brightfield	105	1002 × 1004	Nikon	10x	ND2	Gray 16bits	Suspension
FL5C	Fluorescence	19	1296 × 966	Leica	5x	TIFF	RGB	Collagen
FL5S	Fluorescence	50	1296 × 966	Leica	5x	TIFF	RGB	Suspension
FN2S	Fluorescence	34	1002 × 1004	Nikon	2x	ND2	Gray 16bits	Suspension
BO10S [7]	Brightfield	64	3136 × 2152	Olympus	10x	JPG	RGB	Suspension

**Fig. 1.** Samples from the 7 datasets employed in this work.**Fig. 2.** Generic algorithm for spheroid segmentation.

In the first step, the algorithm tries to find the contour of the spheroid, either by binarising the image, or by finding the edges of the image and later binarising it. This step can be particularised in two different ways. First, the Sobel edge detector [20] can be iterated several times on the image to detect a closed contour, this iterative procedure can be employed when the edges of the spheroid are not clear, and, it terminates when a region that satisfies some conditions, related to size and solidity, is found or a number of iterations fixed by the user is reached. And, second, the threshold values to binarise the image can be fixed manually, or automatically selected by using algorithms like IsoData [21] or Otsu [22].

Once the contour of the image is generated, the second step of our algorithm refines such a contour to find the region where the spheroid is located. First of all, the algorithm tries to close the contour region by applying several times the dilation operation, and subsequently filling the holes produced in the image. The dilation operation has the undesirable effect of producing a segmentation that is bigger than the actual region of the spheroid; therefore, an erosion operation is applied, as many times as the dilation operation was applied, to adjust that region. Finally, the watershed operation [23] is applied to remove artefacts that do not belong to the

spheroid. An example showing the application of our procedure is depicted in Fig. 3.

As it can be noted from the above description, our generic algorithm can be customised by fixing 5 parameters: (1) the number of iterations that the Sobel edge detector is applied; (2) the thresholding method; (3) the number of times that the dilation and erosion operations are employed; (4) whether the fill holes operation is applied; and (5) whether the watershed operation is employed.

3.2. Particular algorithms

Due to the different nature of spheroid images, we have particularised our generic algorithm using 5 strategies; that is, using different values for the 5 parameters of our segmentation algorithm. In addition, several variants of our algorithm are combined to deal with those cases where a proper spheroid mask is not generated. We consider a mask as valid when it has a minimum size and satisfies some solidity conditions – note that these conditions depend on the particular characteristics of the spheroid image. The rest of this section is devoted to present the 5 versions of our algorithm.

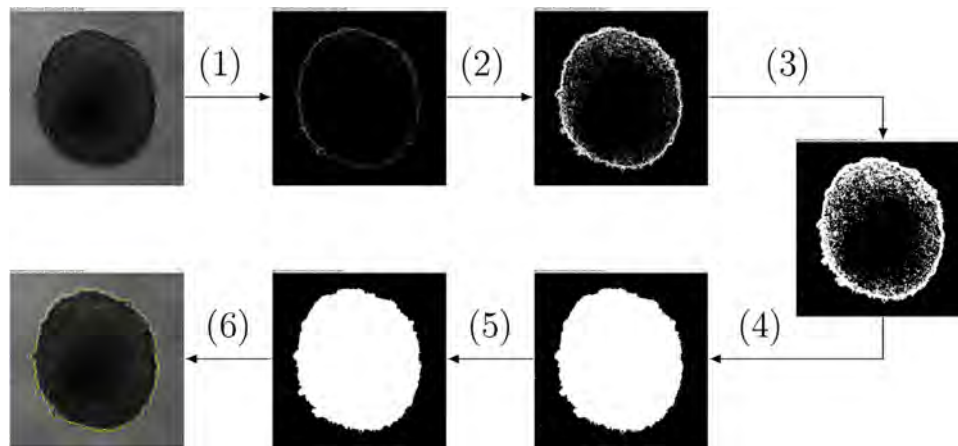


Fig. 3. Example of the application of our generic algorithm for a sample from the BN10S dataset. (1) Find edges; (2) Binarising; (3) Dilation; (4) Fill holes; (5) Erosion; (6) Final segmentation (in yellow).

A1. Threshold The first strategy is based on just binarising the spheroid images by using the IsoData method [21]. In those cases where such a direct approach does not produce a valid mask, we sequentially binarise the image, dilate it, fill the holes, erode the image, and, finally, apply the watershed operation. This straightforward approach is useful when the spheroid image can be clearly distinguished from the background of the image.

A2. Edges The second strategy does not directly binarise the image but it firstly finds the edges of the image, and subsequently binarise the image using the IsoData method. In case that the method does not work, the number of iterations that the find edges operation is applied is increased. The process stops after a valid mask is found or when a number of iterations is reached.

A3. Threshold+Edges This approach is a sequential application of Algorithms A1 and A2. Namely, it starts applying the threshold approach, and if it fails to find a valid mask, it applies the edges approach.

A4. Threshold & edges This strategy applies both Algorithms A1 and A2 to the input image, adds the two resulting masks, and fills the holes of the resulting mask to produce the final output.

A5. Fluorescence Finally, we have designed an algorithm that takes advantage of images acquired with fluorescence. To this aim, the normal image is processed by sequentially finding its edges and binarising it; and, the fluorescence image is binarised using the IsoData thresholding method. The two images produced in this way are combined using the AND binary operation to output the mask.

4. Deep learning segmentation algorithms

In this section, we present the deep learning approach that we have followed for segmenting spheroids¹. Nowadays, deep learning techniques are the state of the art approach to deal with computer vision tasks particularly in medicine [24–27]. The main drawback of deep learning methods is that they are data demanding, and require a considerable number of annotated images to train models from scratch [28]. In this work, we face this problem by using two widely employed techniques to train a model with a small dataset: *data augmentation* [29] (a technique that consists in generating new training samples from the original training dataset by applying transformations that do not alter the class of the data) and *transfer learning* [30] (a technique that reuses the knowledge

learned in a different task to train a new model). In addition, we partially automatise the annotation task by using the generic algorithm presented in the previous section. Namely, we constructed 4 datasets for training several deep-learning based algorithms that produce as a result different segmentation models.

The datasets employed for training the deep learning algorithms were constructed as follows. First, a total of 1645 images were acquired using the settings from datasets BL5S, BN2S and BN10S presented in Table 1 – these images are independent from the datasets employed for evaluation and presented in Table 1. From those images, 838 images were manually annotated by experts using the free hand tool of ImageJ [31]. Those 838 images form the manually annotated dataset; and such a dataset was split into a training set of 621 images, and a validation set of 217 images. The training dataset was augmented in three different ways: (1) using data augmentation (by applying on-the-fly dihedral transformations and rotations with the Albumentations library [32]); (2) automatically annotating the other 807 acquired images with the versions A1–A4 of the generic algorithm presented in Section 3.2 (the annotations were manually validated, and for each image the best result produced by algorithms A1–A4 was selected; a total of 359 automatically annotated images were dismissed since none of the algorithms produced a valid result); and (3) applying data augmentation to (2).

From the training datasets, we fine-tuned several deep-learning segmentation algorithms [30]; namely, we have trained 5 architectures: U-Net [17] (with a Resnet 34 backbone), DeepLabV3+ [33] (with a Resnet 50 backbone), Mask RCNN [34] (with a Resnet 50 backbone), HRNet-Seg [35] (with an HRNet W30 backbone) and U²-net [36] (with its underlying backbone). The definition of those 5 architectures are available in the SemTorch package². All the architectures were trained with the libraries PyTorch [37] and FastAI [38] and using a GPU Nvidia RTX 2080 Ti. In order to set the learning rate for the different architectures, we employed the procedure presented in [38]; and, we applied early stopping in all the architectures to avoid overfitting. As a result of the training process, 5 models were produced that can be used for inference by providing them a spheroid image as input, and they will output the mask associated with the segmentation.

The results achieved by the constructed models in the validation set are presented in Table 2. The metric employed to measure

¹ The notebooks, models, and datasets employed in this section are available at <https://github.com/WaterKnight1998/Deep-Tumour-Spheroid>.

² The SemTorch package is available at <https://github.com/WaterKnight1998/SemTorch>.

Table 2

Jaccard index for the validation set obtained by each model using different training datasets and augmentation regimes. The best result is highlighted in bold face.

	Manual	Manual (Augmented)	Manual & Automatic	Manual & Automatic (Augmented)
DeepLabV3+	0.9267	0.9255	0.9502	0.9419
HRNet-seg	0.9141	0.9333	0.9512	0.9478
Mask-RCNN	0.8781	0.8796	0.8942	0.8926
U-Net	0.8819	0.9209	0.9431	0.9466
U ² -Net	0.8857	0.8657	0.9462	0.9408

Table 3

Mean (and standard deviation) for the brightfield datasets. The best result for each dataset is highlighted in bold face, *** $\rho < 0.001$, > significant difference between methods. In, Iv, K, C, and D stand for Insidia, Ivanov, K-means, Canny, and Deep, respectively.

	Insidia	Ivanov	K-means	Canny	A1	A2	A3	A4	Deep	Friedman Test	Dunn test
BL5S	0(0)	0(0)	0(0)	0.64 (0.30)	0.55(0.33)	0.31(0.42)	0.63(0.39)	0(0)	0.75(0.25)	293.231***	D,C,A3,A1>A2,A4,In,Iv,K
BN2S	0.65(0.35)	0.2(0.36)	0.14 (0.30)	0.85 (0.09)	0.93(0.04)	0.94(0.02)	0.72(0.35)	0.73(0.35)	0.96 (0.01)	801.601***	D>A2,A1>C,A4,A3,In>Iv, K
BN10S	0.84(0.07)	0.03(0.18)	0.26 (0.36)	0.38 (0.10)	0.65(0.38)	0.69(0.42)	0.6(0.42)	0.95(0.01)	0.97 (0.01)	380.116***	D>A4>In,A2,A1,A3,C,K>Iv
BO10S	0.91(0.09)	0.94(0.17)	0.77(0.35)	0.88 (0.05)	0.94(0.03)	0.42(0.42)	0.79(0.36)	0.88(0.10)	0.92 (0.03)	254.627***	A1,Iv>D,In,A4,C,A3,K>A2
Combined	0.64(0.37)	0.28(0.43)	0.27(0.40)	0.72(0.24)	0.81(0.27)	0.68(0.41)	0.7(0.39)	0.74(0.35)	0.92 (0.12)	298.238***	D>A1>A4,A3,A2>C,In>Iv,K

the accuracy of the different methods is the IoU, also known as Jaccard index – this metric measures the area of intersection between the ground truth and the predicted region over the area of union between the ground truth and the predicted region. As can be seen in Table 2, data augmentation does not produce considerable benefits in this context; by contrast, better results are obtained for all the architectures when the automatically annotated dataset is employed; namely, the model trained using the HRNet-seg architecture achieves the best results. In the next section, we show that this model generalises to images that were not employed during the training process, even if they were acquired with a different setting.

5. Results and discussion

In this section, we compare the different versions of our generic algorithm and the best model constructed in the previous section. To this aim, we have employed the datasets presented in Section 2. In addition, we include in the comparison two classical segmentation algorithms that are K-means clustering [39] and Canny segmentation [40], and two open-source Fiji macros: Insidia [8], and the macro presented in [7] (from now on, this macro will be called “Ivanov”) – other tools exist in the literature, but they are not freely available or only work with concrete microscopes. As in the previous section, the Jaccard index is employed to evaluate the different algorithms. We first analyse the 4 brightfield datasets; and, after that, the fluorescence datasets.

The results for the brightfield datasets are presented in Table 3. A statistical analysis of such results is also included. In particular, Friedman tests were carried out to compare the total scores for the seven methods. When significant differences among the methods were found, a Dunn-Bonferroni pairwise post hoc test was also included. We can notice that the deep model achieves the best results for the datasets acquired with the same settings that those used for training the model; moreover, it generalises properly to a dataset of images acquired with a completely different setting (the BO10S dataset) and the results are close to the best models. Hence, even if the particular versions of the generic algorithm can obtain good results for different datasets, they must be adjusted; by contrast, the deep model can be employed across all datasets without modifying it. In addition, our generic method and the deep model outperform both Insidia and Ivanov tools, two macros that are based on image processing techniques and suffer from the same generalisation problem as the particular version of our generic algorithm: they work properly for a particular kind of images, but

they fail when the settings are changed. Finally, the deep model also obtains better results than those achieved by the general purpose algorithms (K-means clustering and Canny).

On the contrary to the brightfield datasets, the deep model only achieves the best results in one of the datasets; and, Algorithm A5 produces better results than the other algorithms for the other fluorescence datasets, see Table 4. This is due to the fact that, Algorithm A5, as well as humans, not only uses the brightfield image for segmentation, but it also takes advantage of the fluorescence image where the location of the spheroid region is clearly defined – the accuracy of all the other studied methods and macros is considerably lower than the accuracy obtained by Algorithm A5 since they only consider the brightfield image. However, since the spheroid region of a fluorescence image does not perfectly adjust to the spheroid, this produces a lower accuracy than the methods for the brightfield datasets.

As we have seen throughout this section, our deep learning model can be successfully applied to a wide variety of settings; and, additionally, it might be worth trying different approaches based on our generic algorithm, specially when working with fluorescence images. To facilitate the dissemination and use of our methods, we have developed a tool called SpheroidJ.

6. SpheroidJ

We have released our methods in an open-source and freely available program, called SpheroidJ. This tool can be employed as

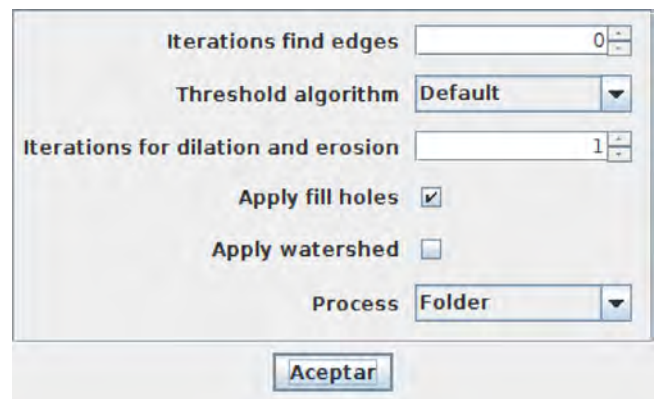
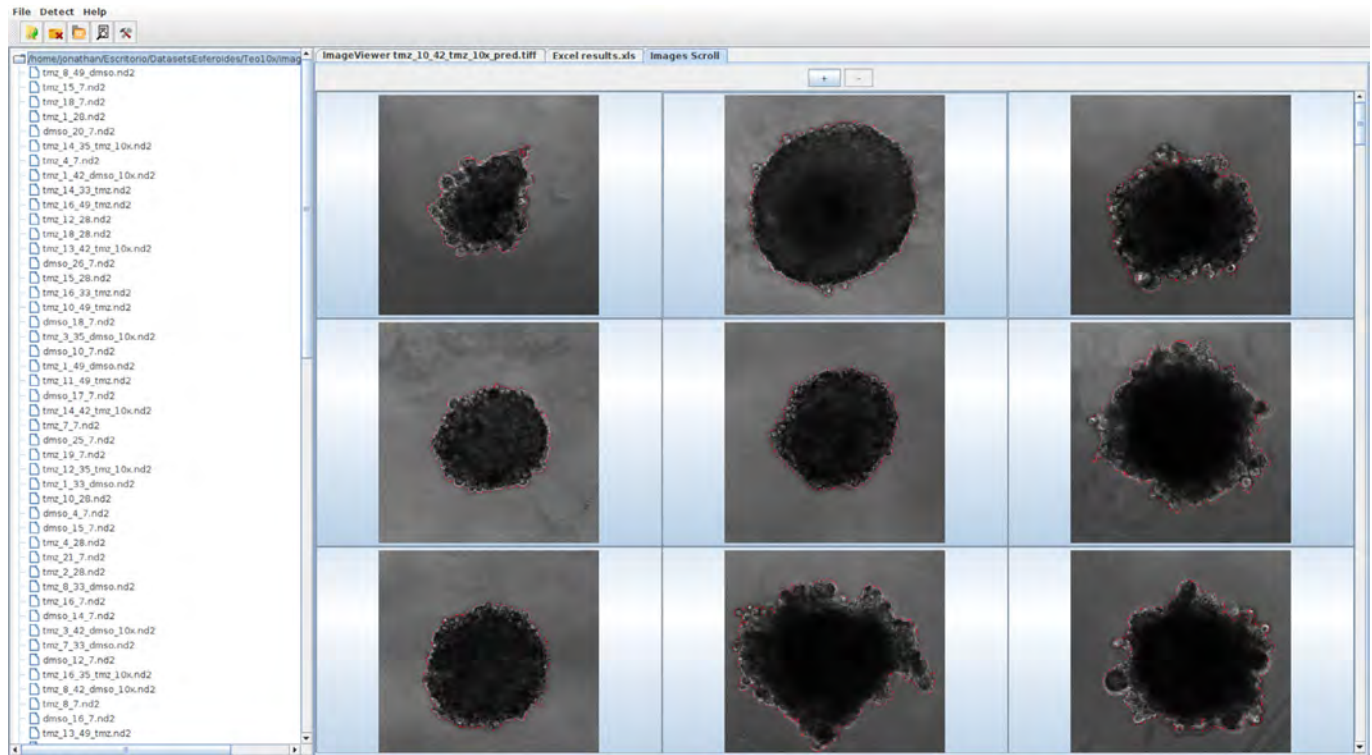


Fig. 4. SpheroidJ plugin window to configure the segmentation algorithm.

Table 4

Mean (and standard deviation) for the fluorescence datasets. The best result for each dataset is highlighted in bold face, $***\rho < 0.001$, > significant difference between methods. In, Iv, K, C, and D stand for Insidia, Ivanov, K-means, Canny, and Deep, respectively.

	Insidia	Ivanov	K-means	Canny	A1	A2	A3	A4	A5	Deep	Friedman Test	Dunn test
FL5C	0.12(0.24)	0.09(0.28)	0.21(0.29)	0.32(0.25)	0.53(0.37)	0(0)	0.4(0.37)	0(0)	0.67(0.17)	0.71 (0.30)	112.844***	D,A5,A1>A3,C,K,In,Iv,A2,A4
FL5S	0.51(0.24)	0.04(0.1)	0.08(0.11)	0.07(0.13)	0.31(0.21)	0.04(0.14)	0.42(0.27)	0(0)	0.89(0.07)	0.70 (0.26)	268.033***	A5,D>In,A3,A1,A2,Iv,K,C,A4
FN2S	0.03(0.02)	0(0)	0.02(0.01)	0.29(0.40)	0.65(0.3)	0.47(0.36)	0.02(0.16)	0.05(0.04)	0.82(0.17)	0.78 (0.20)	211.293***	A5,D,A1>A2,C,A4,In,K,A3,Iv
Combined	0.25(0.29)	0.03(0.15)	0.08(0.16)	0.20(0.30)	0.48(0.32)	0.19(0.32)	0.27(0.32)	0.03(0.10)	0.82(0.16)	0.74 (0.25)	442.198***	A5,D>A1>A3,In,A2,C,K,A4,Iv

**Fig. 5.** SpheroidJ standalone application.

a Python library, an ImageJ plugin, and also as a standalone application. The Python library is an API that provides access to the deep learning model; the ImageJ plugin and the standalone application include all the methods presented in this work, and deserve a more detailed explanation.

ImageJ [31] is an image-analysis tool that has been widely employed to deal with many problems in life sciences, and that can be easily extended by means of plugins. The SpheroidJ plugin can be called from the ImageJ interface and provides two executions modes: the batch mode and the experimental mode. The batch mode allows the users to segment the spheroid images of a folder by employing either any of the 5 algorithms presented in Section 3.2 or the deep model presented in Section 4 – this mode has been designed to include other methods and models in the future. The experimental mode allows the users to configure the generic algorithm presented in Section 3.1 to deal with their own images. The parameters of the algorithm can be configured from the window presented in Fig. 4 and applied either to a single image or a folder of images. The result outputted by both modes are the spheroid segmentations, and an Excel file with a summary of measures (such as the area, perimeter, circularity or Feret's diameter) extracted from the segmented images.

This ImageJ plugin has a main drawback: it does not provide a simple way of visualising and editing the segmentation results when dealing with a folder of images. This issue has been tackled with the development of a user-friendly and standalone applica-

tion. This tool provides the same functionality explained for the ImageJ plugin, but after the segmentation process, it shows the results using the interface presented in Fig. 5. In this way, the users can easily inspect the segmentation result, try different algorithms for a single image, and manually edit the segmented region if it was not properly detected.

7. Conclusions

Due to the variance in sizes, shapes and textures of spheroids, it is challenging to define a set of rules to segment them. In this paper, we have tackled this challenge by designing a customisable algorithm that can be successfully adapted to different kinds of spheroid images. In addition, this generic algorithm has been the basis to reduce the burden of annotating a dataset of images to train deep learning models. Both our generic algorithm and deep learning model can be applied in a wide variety of images; however, it is recommended to use the deep learning model since it does not require any configuration. SpheroidJ is specialised in images containing a single spheroid, but it can also be employed to segment images with multiple or combined spheroids.

In order to facilitate the dissemination of our methods, we have released SpheroidJ, an open-source that can be used both as an ImageJ plugin and a standalone application. Thanks to this work, the analysis of spheroids acquired under different conditions will be

more reliable and comparable; and, the developed tools will help to advance our understanding of tumour behaviour.

Availability and requirements

- Project name: SpheroidJ.
- Project home page: <https://github.com/joheras/SpheroidJ>.
- Operating system(s): Platform independent.
- Programming language: Java and Python.
- Other requirements: Java 8 and Python 3.6. The use of the deep learning segmentation model requires the installation of a Python package called [Deep-Tumour-Spheroid](#).
- License: GNU GPL v3.
- Any restrictions to use by non-academics: restrictions specified by GNU GPL v3.

The instructions to install and use SpheroidJ are provided in the project webpage. The code, datasets and ground truth employed in this work are also available at the project webpage.

Funding

This work was partially supported by the Ministerio de Economía y Competitividad [MTM2017-88804-P], Agencia de Desarrollo Económico de La Rioja [2017-I-IDD-00018].

Declaration of Competing Interest

The authors declare that they have no competing interests.

References

- [1] World Health Organization, WHO report on cancer: setting priorities, investing wisely and providing care for all, World Health Organization, 2020.
- [2] D. Rodenhizer, T. Dean, E. D'Arcangelo, et al., The Current Landscape of 3D In Vitro Tumor Models: What Cancer Hallmarks Are Accessible for Drug Discovery? *Advanced Healthcare Materials* 7 (8) (2018) 1701174, doi:10.1002/adhm.201701174.
- [3] E.L. Andrade, A. Benito, J. Cavalli, et al., Non-clinical studies required for new drug development Part I: Early in silico and in vitro studies, new target discovery and validation, proof of principles and robustness of animal studies, *Brazilian Journal of Medical and Biological Research* 19 (11) (2016) e5644, doi:10.1590/1414-431X20165644.
- [4] D. Lv, Z. Hu, L. Lu, et al., Three-dimensional cell culture: A powerful tool in tumor research and drug discovery (Review), *Oncology Letters* 14 (6) (2017) 6999–7010, doi:10.3892/ol.2017.7134.
- [5] J. Hoarau-Véhot, A. Rafii, C. Touboul, et al., Halfway between 2D and animal models: Are 3D cultures the ideal tool to study cancer-microenvironment interactions? *International Journal of Molecular Sciences* 19 (1) (2018) 181, doi:10.3390/ijms19010181.
- [6] S. Nath, G.R. Devi, Three-dimensional culture systems in cancer research: Focus on tumor spheroid model, *Pharmacology & Therapeutics* 163 (2016) 94–108, doi:10.1016/j.pharmthera.2016.03.013.
- [7] D.P. Ivanov, et al., Multiplexing spheroid volume, resazurin and acid phosphatase viability assays for high-throughput screening of tumour spheroids and stem cell neurosphere, *Plos One* 9 (8) (2014) e103817, doi:10.1371/journal.pone.0103817.
- [8] C. Moriconi, et al., Insidia: A fiji macro delivering high-throughput and high-content spheroid invasion analysis, *Biotechnology Journal* 12 (10) (2017) 1700140, doi:10.1002/biot.201700140.
- [9] Y. Hou, J. Konen, D.J. Brat, et al., Tasi: A software tool for spatial-temporal quantification of tumor spheroid dynamics, *Scientific Reports* 8 (2018) 7248, doi:10.1038/s41598-018-25337-4.
- [10] W. Chen, C. Wong, E. Vosburgh, et al., High-throughput image analysis of tumor spheroids: A user-friendly software application to measure the size of spheroids automatically and accurately, *Journal of Visualized Experiments* 89 (2014) e51639, doi:10.3791/51639.
- [11] F. Piccinini, Anasp: A software suite for automatic image analysis of multicellular spheroids, *Computer Methods and Programs in Biomedicine* 119 (1) (2015) 43–52, doi:10.1016/j.cmpb.2015.02.006.
- [12] L.R. Cisneros-Castillo, A.D. Oancea, C. Stüllen, et al., A novel computer-assisted approach to evaluate multicellular tumor spheroid invasion assay, *Scientific Reports* 6 (2016) 35099, doi:10.1038/srep35099.
- [13] M. Vinci, S. Gowan, F. Boxall, et al., Advances in establishment and analysis of three-dimensional tumor spheroid-based functional assays for target validation and drug evaluation, *BMC Biology* 10 (2012) 29, doi:10.1186/1741-7007-10-29.
- [14] F. Cornelissen, E. Gustin, M. Cik, Phaedra, a protocol-driven system for analysis and validation of high content imaging and flow cytometry, *Journal of Biomedical Screening* 17 (4) (2012) 496–506, doi:10.1177/1087057111432885.
- [15] F. Piccinini, A. Tese, M. Zanoni, Revims: Software tool for estimating the volumes of 3-d multicellular spheroids imaged using a light sheet fluorescence microscope, *BioTechniques* 63 (5) (2017) 227–229, doi:10.1177/1087057111432885.
- [16] L.R. Cisneros-Castillo, A.D. Oancea, C. Stüllen, et al., An imaging-based platform for high-content, quantitative evaluation of therapeutic response in 3d tumour models, *Scientific Reports* 4 (2015) 3751, doi:10.1038/srep03751.
- [17] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, in: *Lecture Notes in Computer Science*, 9351, 2015, pp. 234–241, doi:10.1007/978-3-319-24574-4_28.
- [18] L.C. Chen, G. Papandreou, F. Schroff, et al., Rethinking atrous convolution for semantic image segmentation, *ArXiv abs/1706.05587* (2017).
- [19] S.K. Sadanandan, J. Karlsson, C. Wühlby, Spheroid segmentation using multiscale deep adversarial networks, in: *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 36–41, doi:10.1109/ICCVW.2017.11.
- [20] N. Kanopoulos, N. Vasanthavada, R.L. Baker, Design of an image edge detection filter using the Sobel operator, *IEEE Journal of Solid-State Circuits* 23 (2) (1988) 358–367, doi:10.1109/4.996.
- [21] T.W. Ridler, S. Calvard, Picture thresholding using an iterative selection method, *IEEE Transaction on Systems, Man and Cybernetics* 8 (1978) 630–632, doi:10.1109/TSMC.1978.4310039.
- [22] N. Otsu, A threshold selection method from gray level histograms, *IEEE Transaction on Systems, Man and Cybernetics* 9 (1979) 62–66, doi:10.1109/TSMC.1979.4310076.
- [23] J. Roerdink, A. Meijster, The watershed transform: definitions, algorithms, and parallelization strategies, *Fundamenta Informaticae* 41 (2000) 187–228, doi:10.3233/FI-2000-411207.
- [24] S. Nema, A. Dudhane, S. Murala, et al., Rescuenet: An unpaired gan for brain tumor segmentation, *Biomedical Signal Processing and Control* 55 (2020) 101641, doi:10.1016/j.bspc.2019.101641.
- [25] H. Chen, Z. Qin, Y. Ding, et al., Brain tumor segmentation with deep convolutional symmetric neural network, *Neurocomputing* 392 (2020) 305–313, doi:10.1016/j.neucom.2019.01.111.
- [26] E. Goceri, Capsnet topology to classify tumours from brain images and comparative evaluation, *IET Image Processing* 14 (5) (2020) 882, doi:10.1049/iet-ipc.2019.0312.
- [27] E. Goceri, Diagnosis of alzheimer's disease with sobolev gradient-based optimization and 3d convolutional neural network, *International Journal for Numerical Methods in Biomedical Engineering* 35 (7) (2019) e3225, doi:10.1002/cnm.3225.
- [28] E. Goceri, Challenges and recent solutions for image segmentation in the era of deep learning, in: *2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 2019, pp. 1–6, doi:10.1109/IPTA.2019.8936087.
- [29] P. Simard, D. Steinkraus, J.C. Platt, Best practices for convolutional neural networks applied to visual document analysis, in: *International Conference on Document Analysis and Recognition*, in: *ICDAR'03*, 2, 2003, pp. 958–964, doi:10.1109/ICDAR.2003.1227801.
- [30] A.S. Razavian, H. Azizpour, J. Sullivan, et al., CNN features off-the-shelf: An astounding baseline for recognition, in: *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, in: *CVPRW'14*, 2014, pp. 512–519, doi:10.1109/CVPRW.2014.131.
- [31] C.T. Rueden, J. Schindelin, M.C. Hiner, et al., ImageJ2: ImageJ for the next generation of scientific image data, *BMC Bioinformatics* 18 (2017) 529, doi:10.1186/s12859-017-1934-z.
- [32] A. Buslaev, V.I. Iglovikov, E. Khvedchenya, et al., AlbuNet: Fast and flexible image augmentations, *Information* 11 (2) (2020) 125, doi:10.3390/info11020125.
- [33] L.C. Chen, Y. Zhu, G. Papandreou, et al., Encoder-decoder with atrous separable convolution for semantic image segmentation, in: *Computer Vision – ECCV 2018*, Springer International Publishing, 2018, pp. 833–851, doi:10.1007/978-3-030-01234-2_49.
- [34] K. He, G. Gkioxari, P. Dollár, et al., Mask r-cnn, *ArXiv abs/1703.06870* (2017).
- [35] J. Wang, K. Sun, T. Cheng, et al., Deep high-resolution representation learning for visual recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), doi:10.1109/tpami.2020.2983686.
- [36] X. Qin, Z. Zhang, C. Huang, et al., U²-net: Going deeper with nested u-structure for salient object detection, *Pattern Recognition* 106 (2020) 107404, doi:10.1016/j.patcog.2020.107404.
- [37] A. Paszke, S. Gross, F. Massa, et al., Pytorch: An imperative style, high-performance deep learning library, in: *Advances in Neural Information Processing Systems* 32, Curran Associates, Inc., 2019, pp. 8024–8035.
- [38] J. Howard, R. Thomas, Practical deep learning for coders, 2019, (<https://course.fast.ai/>).
- [39] D.L. Pham, C. Xu, J.L. Prince, Current methods in medical image segmentation, *Annual Review of Biomedical Engineering* 2 (2000) 315–337, doi:10.1146/annurev.bioeng.2.1.315.
- [40] J. Canny, A computational approach to edge detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8 (6) (1986) 679–698, doi:10.1109/TPAMI.1986.4767851.