10th International Workshop on Innovative Simulation for Healthcare
18th International Multidisciplinary Modeling & Simulation Multiconference

ISSN 2724-0371 ISBN 978-88-85741-65-2 © 2021 The Authors.
doi: 10.46354/i3m.2021.iwish.005

Automated Evaluation of Cell Viability in Microfluidic Spheroid Arrays

Jonas Schurr[1,3,*], Christoph Eilenberger[2], Peter Ertl[2], Josef Scharinger[3] and Stephan Winkler[1,3]

[1]University of Applied Sciences Upper Austria, Bioinformatics, Softwarepark 11-13, 4232 Hagenberg, Austria
[2]Faculty of Technical Chemistry, Institute of Applied Synthetic Chemistry and Institute of Chemical Technologies and Analytics, Vienna University of Technology, Getreidemarkt 9, 1060 Vienna, Austria
[3]Johannes Kepler University, Department of Computer Science, Altenberger Str. 69, 4040 Linz, Austria

[*]Corresponding author. Email address: jonas.schurr@fh-hagenberg.at

Abstract

Three-dimensional (3D) spheroid arrays promise improved predictability due to their higher physiological relevance. They have the potential to improve drug screening outcomes in preclinical studies. Despite the advances, they can often lead to non-reproducible and unpredictable results. To support the development and subsequent analyses of spheroid arrays, we present a method for analyzing and evaluating cell viability in these. We provide a fast and easy-to-use fully automated workflow for the viability analysis in fluorescence images of cell aggregates within these arrays. The algorithm consists of multiple image processing algorithms for the segmentation and mapping of a priori knowledge about the array layout. The segmentation step is based on Otsu's thresholding followed by morphological filtering to obliterate the necessity of input parameters. No preprocessing is required. Besides, the algorithm offers the possibility of applying an additional flood fill algorithm. Subsequently, a k-means algorithm allows a fast image independent mapping of the grid to identify the cell aggregates. The complete workflow allows the extraction of essential metrics describing the viability of each cell aggregate. With our automated approach, we can show an increase in accuracy compared to simple manual segmentation. Additionally, the objectivity is increased by reducing human intervention. Furthermore, the needed analysis time is shortened and the information extraction and evaluation process is simplified.

Keywords: Image Processing Methods; Fluorescence Microscopy; Image Analysis; Bioinformatics

1. Introduction and state of the art

Multicellular spheroids are three-dimensional (3D) cell aggregates representing a physiologically relevant micro environment in vitro. They are used in preclinical in vitro models and can extensively improve the predictability of such studies. Multicellular 3D spheroids can simulate a more realistic cell contact than their two-dimensional counterparts due to a similar structure to tissues within the body. This allows more similar conditions within a body than standard two-dimensional approaches within a petri dish (Cui et al., 2017). With this approach, more and improved interactions with substances can be realized. This can result in better drug screening outcomes within preclinical studies, also by reducing the costs and reliability (Begley and Ioannidis, 2015). Despite their advantages, high complexity and lack of automatization and standardization of this method reduce current practicability (Eilenberger

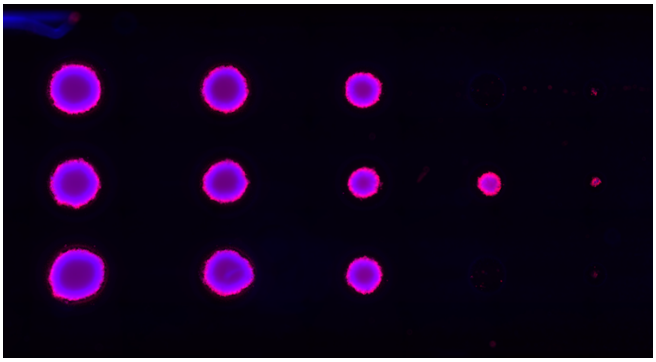

Figure 1. Example image of spheroids to be analyzed

et al., 2021). Within this paper, we want to present an easy-to-use algorithm for automating the analysis and evaluation of multicellular 3D spheroids. The workflow supports and further improves the evaluation process and the practicability of multi-sized spheroid arrays.

To use these arrays, cell nuclei (colored with Hoechst 3342, blue) and dead cells (colored with Ethidium-homodimer-1, red) of spheroids were stained and imaged using DAPI (excitation: 390nm, emission: 460 nm) and TRITC filters (excitation: 530 nm, emission: 645 nm). The images represent a two-dimensional view from above the spheroid as shown in Figure 1. The color emissions at the ends of the visible color spectrum are used to characterize the cell aggregates. It allows the differentiation of living and dead cells. This differentiation allows the determination of the viability and quality of the cell aggregates.

Advancements in multi-sized spheroid arrays can lead to improved drug screening outcomes in preclinical studies (Begley and Ioannidis, 2015). They open the possibility of experiments closer to human physiology. These experiments allow faster and improved drug screening, which can lead to cheaper and faster drug development. To allow the comparison and evaluation of the development of multiple variants of multi-sized spheroid arrays and to simplify and automate the analysis process of developed arrays, automatization is necessary, and human interaction must be reduced to a minimum. This not only saves time but also leads to more objective analysis and better reproducibility.

The algorithm is based on image processing algorithms for the automated analysis of the viability and the size of the cell aggregates. Segmentation and artifact handling is needed. Additionally, mapping a known grid structure is necessary to identify existing and estimate missing cell aggregates. Thresholding and region growing are used for segmentation. The information about the dimensions of the arrays is used to estimate the position of the aggregates and to predict the positions of missing cell aggregates within the image. This allows a robust execution and automated analysis of these images while reducing human interaction.

No fully automated analysis approach regarding multi-sized spheroid arrays has been found to this date. There exist multiple projects where multicellular 3D spheroids arrays were developed. However, in most projects, subsequent analyses are executed manually. Important features like viability and size are extracted manually using general image processing tools like ImageJ or dedicated software of microscope manufacturers (Sirenko et al., 2015; Eilenberger et al., 2021). While these tools often offer a broad range of algorithms, multiple steps must be executed manually or presume expert knowledge in image processing. This clearly states the lack of easy-to-use software workflows for fully automated analyses of such images.

Multiple approaches for segmentation and grid mapping do exist and can be used for biological or medical images (Shao et al., 2019). For segmentation, Otsu's thresholding method is suitable due to an automated separation of foreground and background with no need for user input (Sezgin and Sankur, 2004). Reducing the number of input parameters can further simplify the usage for biologists. More advanced or supervised segmentation approaches can further increase the results but need a lot of labeled data that is often not available.

2. Methods and Implementation

The following algorithm consists of multiple steps, as shown in Figure 2. The proposed algorithm and functionalities are implemented in python 3.8. The used image processing methods are implemented in OpenCV or scikit-image (van der Walt et al., 2014). While user interaction is not required, essential parameters can be changed. Especially the cropping percentage for removing large border areas and the possibility of a flood fill algorithm for the segmentation is of particular interest since it can improve the outcome depending on the given image. Additionally, for estimating the area of the cell aggregates, it is necessary to provide the resolution of the image in micrometers per pixel.

2.1. Preprocessing

The image orientation must be adjusted or considered accordingly since the images can be horizontally or vertically flipped and rotated by 90 degrees. Therefore, the algorithm must be robust against all mentioned alterations. This is achieved by making no assumptions based on the rotation or pattern besides a specific number of rows, which was three in our analyzed experiments case. The algorithm also can handle more or also fewer rows if needed. The rotation is detected and adjusted automatically on the basis of the images shape.

Additionally, to remove artifacts on the image's outer areas and to reduce processing time, the image is cropped at each size at a given percentage. In our

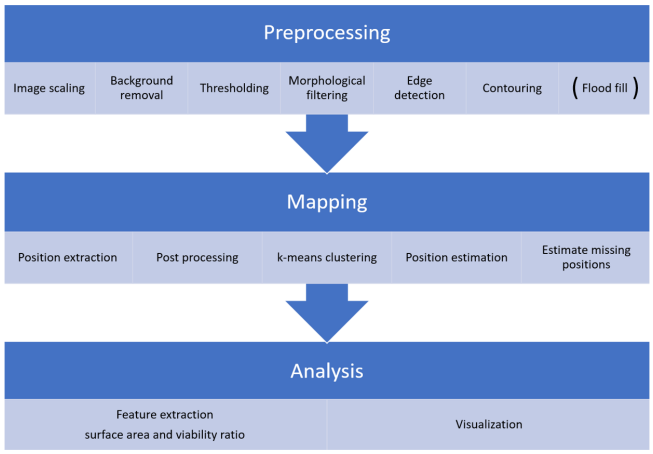

Figure 2. Workflow overview

case, the cell aggregates appeared between 10 and 20 percent of the images size within the image.

Background removal is needed because only the difference to the background is needed to identify the color ratio correctly. The colored cell aggregates can influence the background with a glow in a specific color. Furthermore, the background must be removed. This is done based on the RGB value of a specific percentile of all pixels. In our case, 80 percent leads to the best results since with 80 percent, the glowing effect closer to the aggregates could be considered. The extracted background is subtracted or, if higher than the actual value of a pixel, is set to black. Finally, the background is removed on the RGB image since we want to remove the background for each color channel to get clean results when identifying the red-blue ratio. A comparison between raw image and an image with removed background is shown in Figure 3.

To segment and identify the cell aggregates, a binarization by Otsu's thresholding followed by morphological filtering to reduce the amount of noise and artifacts is suitable. To allow Otsu's thresholding, the image must be converted to grayscale. Otsu's thresholding was used because of alternating grayscale differences within but also between multiple images. In comparison, simple thresholding was used initially but did not produce robust results and requires user input. An output of the thresholding can be seen in the top left in Figure 4.

Multiple morphological filtering steps are used with a rectangular shaped structure element on the inverted image to reduce salt and pepper noise and to merge cell aggregates with surrounding loose fragments. First, an opening operator with a structure element of 21x21 was performed to remove white noise. With this filter, the artifacts and noise within cell aggregates can also be removed. Additionally, trenches and cuts into cell aggregates can be reduced as seen in the top right in Figure 4. It also connects cell aggregates with their surrounding fragments due to the first performed erosion. Following these steps, a closing operation with a smaller structure element size of 15x15 was performed. These steps are consisting of a dilation step followed by an erosion step. Which reduces or even removes pepper noise outside of the cell aggregates as seen in the bottom left in Figure 4. The order of these two filters assures the connection of smaller fragments to prevent their removal while removing noise. This is necessary to keep tiny cell aggregates. As a final step, an additional erosion step is performed with multiple iterations with a structure element size of 5x5 as seen in the bottom right in Figure 4.

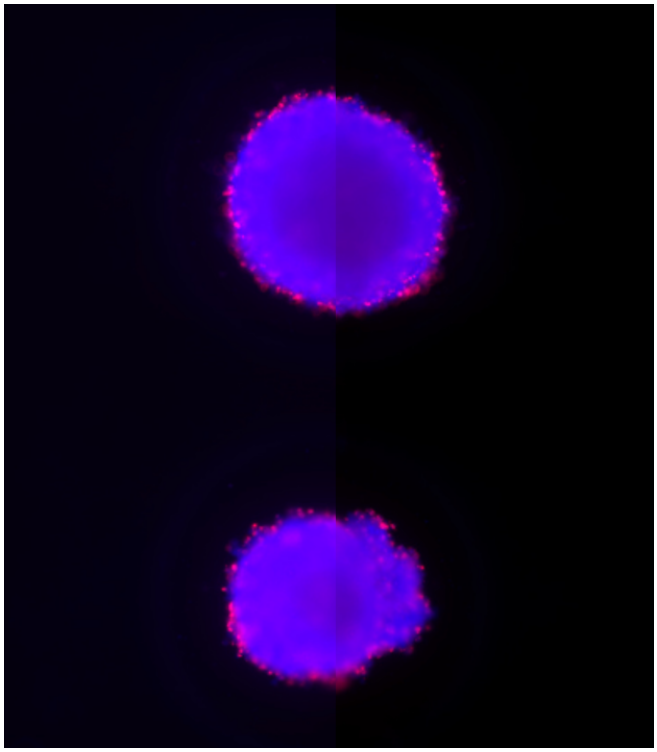

Figure 3. Left side: raw image, right side: after removed background

The erosion assures the connection of remaining fragments and removes existing trenches while also making smaller cell aggregates larger for following contour detection.

Additionally, due to Otsu's thresholding, not all pixels belonging to a cell aggregate are correctly assigned. The segmentation based on Otsu's thresholding does not include directly surrounding pixels, which are still part of the cell aggregates, especially for small aggregates since they have lower brightness. A final erosion counteracts this disadvantage of Otsu's thresholding. Edge detection was performed on the binary image before contouring the segmented cell aggregates to extract each aggregate.

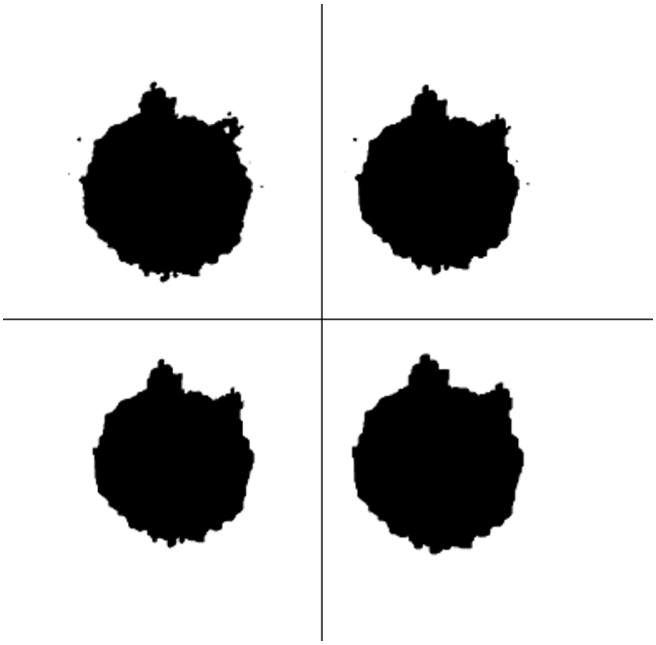

Figure 4. Intermediate results after each segmentation step: top left: Otsu's thresholding, top right: opening, bottom left: closing, bottom right: erosion

2.2. Flood fill

Additionally, a flood fill algorithm can be used to improve the segmentation based on Otsu's thresholding. In some cases, especially if the brightness of cell aggregates is slowly fading out, Otsu's thresholding followed by morphological filtering is not sufficient. Too many pixels will be missed and will not be recognized as part of a cell aggregate. To surpass this problem, a flood fill algorithm can be used to cover the whole area of the cell aggregate. For each potential contour representing a potential cell aggregate, the gray value is extracted at the center point. An overview of the full subworkflow can be seen in Figure 5

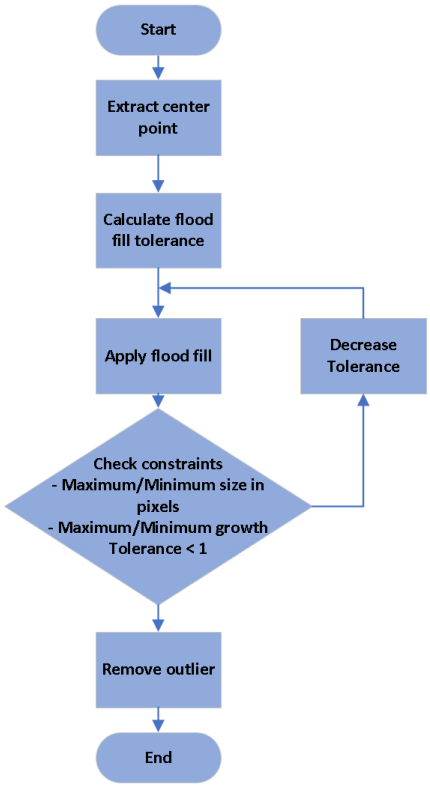

Figure 5. Flood fill subworkflow overview

The used tolerance for the flood fill algorithm must fulfill two requirements. First, it must be large enough to cover the whole cell fragment, which can be a large value due to the low gradient in color in some cell aggregates. Nevertheless, at the same time, it must not be too large to prevent the segmentation of the background of the surrounding area of a cell aggregate which is slightly brighter than the background due to the glow of the cell aggregates. To achieve this, the tolerance is first set to a large value (25 was identified as suitable). To adapt the value for much brighter images, the tolerance is set to the intensity of the center position if the color in the center is larger. This is the highest used tolerance, and based on additional constraints, the tolerance is lowered until it the covered area reaches a good coverage of the cell. Since this is a computationally intensive method before this stepwise approximation, a guess for the tolerance based on 90 percent of the intensity of the center point is made. This value represents a value that does include all values except the lowest 10 percent. This 10 percent estimates and represents the unwanted glow of the cell aggregates.

To check if the first guess is feasible, the mentioned constraints are checked. Four constraints must be met. On one side, the maximum size of the area is capped at two percent of the image size since this value represents the maximum size of a cell aggregate. Additionally, the minimum size of the area in pixels is also delimiting the tolerance form getting too low. As an additional constraint, the factor between the original size of the cell aggregate and the new size of the area based on the flood fill algorithm is checked. This prevents the area by the given tolerance from getting too large. Since the size of two percent of the whole image makes sense for a large cell aggregate but not for a small cell aggregate. Therefore, additionally, a maximum increase in size is used as constraint. Also, a reduction of the detected area with flood fill is prevented. Based on these constraints and if the first guess did not provide relevant results, the tolerance is lowered starting from the intensity of the center point until it is one or triggers one of the mentioned constraints. With this approach, it can be assured that the tolerance is found automatically, and the flood fill algorithm does provide accurate results. Again, to retrieve the segmented areas, the

external contours are extracted.

After the flood-fill algorithm with its optimization step, filtering as post-processing is applied. Based on the size increase of the area, outliers are removed. This can be necessary because the mentioned constraints are not too strict to prevent the removal of wanted true cell aggregates. However, this allows the flood fill algorithm also to segment artifacts close to cell aggregates. To get an overview of the general growth of the surface area within the whole image, for each potential cell aggregate, the factor to its previous size is calculated. If a detected cell aggregate had a much larger growth than others, it will be removed, especially if it is identified as an outlier. The outlier removal is based on the z score with a threshold of three.

To finalize, if no suitable tolerance for the flood fill can be found, the original contour will be kept. All newly detected contours, again, are extracted by the minimum enclosing circle to get the center of the cell aggregate.

2.3. Mapping

The next step is contouring after the preprocessing is done. To extract the masked segments individually, all contours are extracted. To prevent too small segments, the contours are filtered based on a minimum size which the user can adapt. To be independent of the size of the image, the used value is given as a ratio. The single contours allow the distinction into unique cell aggregates to identify and compare their features. Their properties are necessary for the final analysis and the mapping onto the grid based on their position in the image. Most of the wrongly masked cell aggregates are removed at this stage, and the cell aggregates can be mapped and analyzed.

For the mapping, the maximum dimension of the grid must be given. In the images of the analyzed experiments, the maximum number of cell aggregates was fifteen with three rows and five columns. The algorithm must deal with missing cell aggregates since these are not always available. Since these are multi size experiments the size of the aggregates is reduced in each column. This is not always the case so this is not a constraint for the algorithm, but this leads to small positions or even positions without cell aggregates. A grid with the given dimension is generated as a template, first without specific positions.

To extract the specific position of a single cell, aggregate the midpoint of a contour is calculated. Based on the midpoint and the radius of a contour which is defined by the minimum enclosing circle, a position for each cell aggregation within the image is defined. Finally, based on these positions, the position within the grid is defined.

To prevent errors by using wrongly detected segments or noise for mapping, additional filtering is performed. To prevent the mapping of artifacts or remaining fragments close to cell aggregates are filtered based on their size and neighborhood. Within a given range, represented as a percentage of the image size to allow the analysis of different image sizes, the neighborhood is identified. In this neighborhood, the minor contours are removed, and only the biggest one is kept. This approach is valid because the aggregates are always the biggest structure within the area of correct positions in the mapped grid. The size of the contour is defined by there are measured in pixels. The used neighborhood is given by 10 percent in height and 15 percent in width. There is a difference in height and width because the horizontal distance between multiple cell aggregates is larger than the vertical distance. Therefore, the distance should be at least as large as half of the distance between multiple cell aggregates to remove artifacts in between. But at the same time, it must not be too large to prevent the removal of smaller correct contoured cell aggregates. In Figure 6 the remaining contours can be seen. It can also be seen that far off artifacts cannot be removed in this step since the filtering is based on the distance.

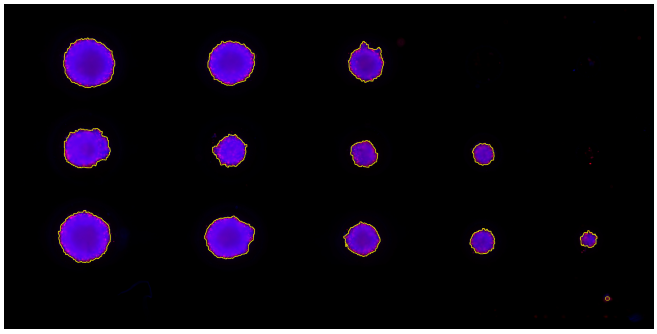

Figure 6. Contours after neighborhood filtering

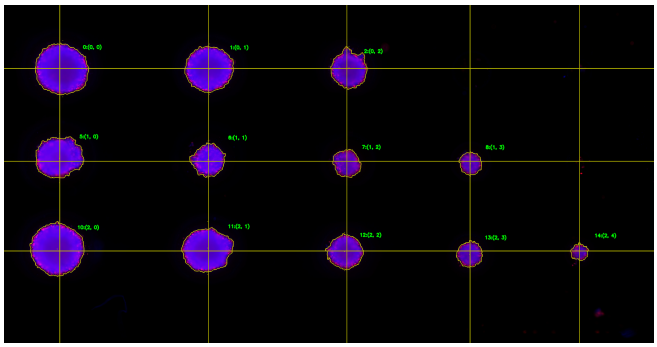

Figure 7. Clustered cell aggregates representing rows and columns

After performing the mentioned preprocessing steps like the contouring of the segments, the definition of the maximum dimension, the extraction of the center point, and the additional filtering to remove noise and artifacts, the mapping of the potential cell aggregate candidates onto the actual grid can be performed.

To map the contoured segments representing cell aggregates to the grid which is defined by the given number of rows and maximum number of columns, a clustering algorithm is used. A k-means algorithm is suitable since its usage, is not dependent on the size of the image because the processed images can have high resolutions (more than 20,000x10,000 pixels) (MacQueen et al., 1967). This approach assures low execution time, also for big images. To assign the detected contours according to their position in the grid, a clustering based on the y-coordinates of the contour centers is executed. This separates the found contours into three clusters and assigns them to their best-fitting row in the grid. The cluster count is defined by the known number of rows of the grid. Similarly to the clustering of the y positions for the assignment of the contours to the row, clustering based on the x-coordinates can be done. The problem here is that it is not known how many columns there will be. This is because of possibly missing cell aggregates at certain positions. To avoid this problem and estimate a suitable number of clusters, the clustering is done multiple times for different cluster numbers. The number of clusters is starting from 2 to the given maximum number of columns in the grid. The best number for k (number of clusters) is evaluated based on the silhouette score. With this best number of clusters, the clustering is applied to the x-coordinates of the contours to assign their position within the grid. Each potential cell aggregate is now assigned to its best-fitting cluster, which also defines the position in the grid. The cluster with the lowest y values is the most upper one and vice versa. Same counts for the x values from left to right. If multiple potential candidates would map onto the same position, the biggest one is taken because smaller ones are often artifacts. The result of this step can be seen in Figure 7.

Probably missing positions also must be evaluated since they can contain tiny fragments or a low signal. Also, if there is no signal, the not available cell aggregate position must be assigned to the grid. The position in the grid also must be marked as empty for further analyses. To achieve this, the grid is iterated row-wise and positions in the grid. Mostly the last column is without an assigned contour. If there is at least one cell aggregate in a column, the others can be estimated. If there is no cell aggregate in a column, the aggregates in the corresponding row cannot be guessed. This is because there is no uniform distance between multiple cell aggregates. If there is one, the center point is guessed by the intersection of the average of the other detected aggregates within the column and row for the corresponding aggregate. The size is guessed based on the average radius of this column. The column is chosen for the size estimation since the size of the wells within one column is the same but not within a row. With this last step, all fifteen positions given by a grid defined by three rows and five columns are assigned. For positions where cell aggregates can be found and assigned, the segmented cell aggregate is defined. If no aggregates for specific positions are available, an estimation is made based on the average size of similar aggregates. For the estimated positions, no area is calculated since no aggregates are available.

3. Results and Discussion

For a final human evaluation, the result is visualized. The results contain the contours of the found aggregates. Additionally, the grid with three rows and five columns is represented by lines. Positions with an estimated cell aggregate are represented by a circle estimated by the average size of other aggregates within a column. All positions contain the center point of the minimum enclosing circle. For reference to the final analyses and extracted measures, the positions are consecutively numbered in reading direction and with their corresponding x and y positions in the grid. After the last step of the image processing and mapping to the grid, the needed measures can be extracted, and further analyses can be executed. To allow immediate and automated additional analyses, the results are exported as csv files. The extracted features and their calculation can be seen in Table 1. The most important values are the number of red cells, especially the red ratio to the blue channel. This value gives the number of dead cells and, therefore also the viability of a cell aggregate. This is an essential measurement when screening substances or drugs. Since it gives insights on the effect of specific substances but also the reliability of the used well. This value can also be used for comparison of other screening methods. Also, the size is an important measurement for evaluation. It is used to assure similar sizes over multiple experiments. Additionally, it allows comparing the effect of different substances onto different sizes of cell aggregates.

The workflow allows an automated analysis of multiple features in especially the viability and area of cell aggregates within multicellular 3D spheroid assays. The usability is shown in Figure 8. It provides a good accuracy especially compared to manual analyses, which are often performed by approximated polygons or circles. The algorithm has been successfully used on multiple assay images including small artifacts and different levels of growth of cell aggregates. In general, no input parameters from the user are needed for the whole workflow. Only the cropping percentage (if not already cropped manually in advance) can improve the alignment of the cells along the grid by cutting out disturbing far-off artifacts. Since the threshold for the flood fill algorithm is calculated automatically only the parameters of the morphological filters could additionally be of interest. In none of our experiments this adjustment was necessary. The final features as

Table 1. Extracted features of cell aggregates

Feature	Description	Formula
ratio	intensity ratio of given channel to both channels	$(sum(CHANNEL\_A)/(sum(CHANNEL\_A) + sum(CHANNEL\_B)) * 100$
area	estimation of area of cell aggregates in square micrometer	$PIXEL\_COUNT * PIXEL\_SIZE$
channel	Name of related color channel for calculations	–
average	arithmetic mean of absolute intensity of given channel	$mean(CHANNEL)$
channel ratio	ratio of maximum intensity of given channel	$sum(CHANNEL)/(PIXEL\_COUNT * 255)$
minimum	smallest intensity of given channel	$min(CHANNEL)$
maximum	highest intensity of given channel	$max(CHANNEL)$
intensity sum	total sum of intensity of given channel	$sum(CHANNEL)$
binary sum	total amount of pixels with higher intensity compared to other channel	$count(CHANNEL\_A > CHANNEL\_B)$
binary ratio	ratio of pixels with higher intensity compared to total amount of pixels	$count(CHANNEL\_A > CHANNEL\_B)/PIXEL\_COUNT$

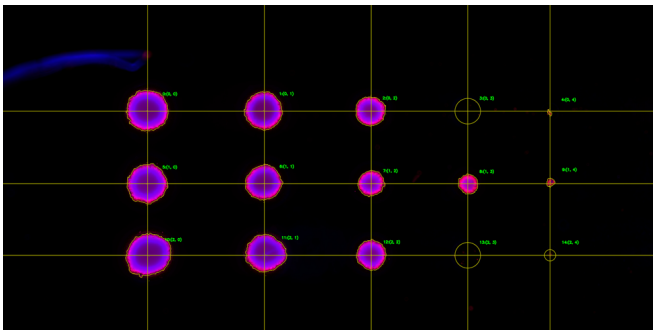

Figure 8. Example of final result showing detected cell aggregates

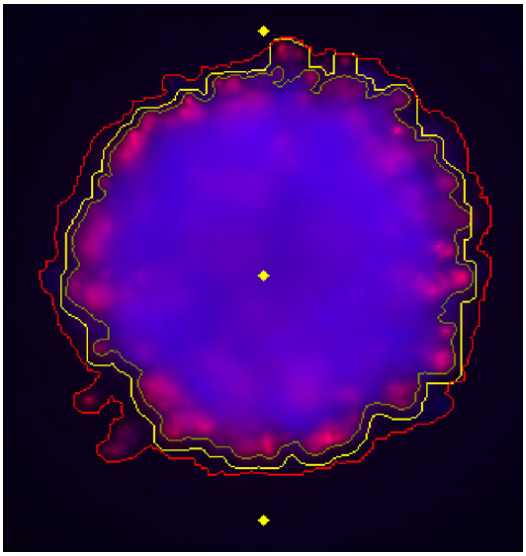

Figure 9. Comparison of segmentation steps

shown in 1 are calculated automatically for each cell aggregate. In Figure 8 you can also see the estimated missing cell aggregates with a rough estimate represented by a circle. Area estimations are not calculated for these estimated positions due to a missing or too weak signal.

Multiple intermediate steps and methods are shown within Figure 9. The inner dark yellow line shows the intermediate state before morphological filtering. After morphological filtering, the contour is slightly widened and also includes more parts of the aggregate on the surrounding area which was not included by thresholding. The outer red line shows the result of the contouring after the segmentation by the flood fill algorithm. You can see that all outlying parts of the cell aggregate are also included into the contour. With this optional approach all parts of the cell aggregate will be included into the contour.

The accuracy of the segmentation can be seen in Figure 10. As measure, the dice coefficient was used to show and compare the overlap of aggregates compared to the ground truth (Zou et al., 2004). All three methods are evaluated based on manually accurate labeled images which are used as ground truth. The given accuracy values are based on the average of all appearing aggregates in one of the three images (A, B and C). The dice coefficient is calculated and compared between the three different methods of segmentation. The first bar (gray) shows the score for the simple manual labeling which is a circle around the aggregate. This kind of labeling was mostly used as the simplified standard procedure in the manual evaluation process. The second bar (blue) shows the dice coefficient based on the segmentation without a following flood fill, only based on the morphological filtering. The third bar (orange) shows the results of the segmentation with the use of the additional flood fill algorithm. In all three images at least one of the results of the automated methods (morphological, flood fill) shows a good dice coefficient, which means there is a high overlapping area between the ground truth and the estimated segmentation. Also, in all three images the automated methods provide increased accuracy compared to the manual segmentation. Since very small cell aggregates can be missed in the mapping process the average dice coefficient is strongly influenced by them. If only detected aggregatesare used for the dice coefficient calculation, the average dice coefficient is even higher (01 (morph): 0.957, 001 (morph): , 0.972). As mentioned before cell aggregates can have different transitions to the background. A low gradient can be found in the third image (C). In this image also the importance of the additional flood fill algorithm can be seen. Otsu's thresholding

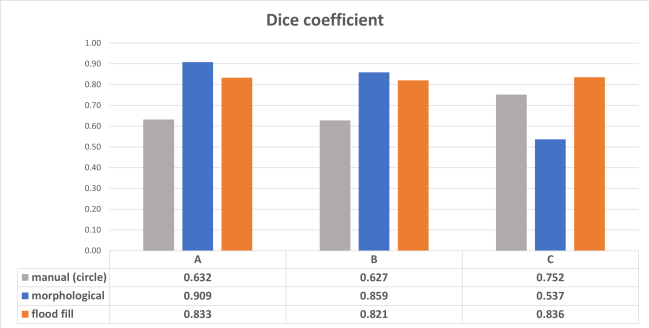

Figure 10. Comparison of results by dice coefficient between used segmentation methods. Gray: manual segmentation, blue: segmentation with morphological filtering, orange: segmentation with additional flood fill

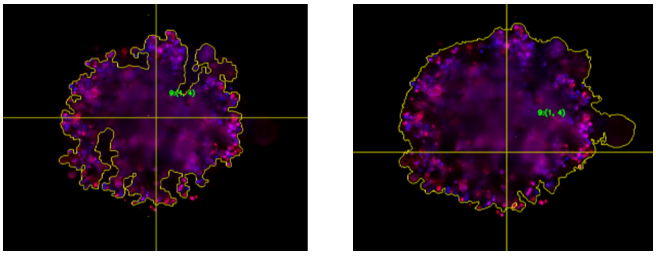

Figure 11. Examples of fragmented cell aggregate

with morphological filtering alone provides too many false negatives and misses too many pixels of the aggregate resulting in a low dice, whereas the flood fill algorithm provides good results similar to the other two images with the morphological filtering. Therefore the results strongly depend on the used automated method. Whereas the morphological filtering alone provide good results in the first two images the accuracy drops in the third image. But in this case the additional flood fill algorithm is necessary and therefore provides good results.

With the usage of the flood fill algorithm for region growing also large trenches and gaps within a cell aggregate can be contoured correctly. This is especially important for large gaps, as shown in Figure 11 where morphological filtering is insufficient due to a small structure element for these cases.

As shown in Figure 12 through Otsu's thresholding but also due to the post processing and filtering of the neighborhood artifacts close to the cell aggregates are correctly ignored. Also darker artifacts or even brighter artifacts within a certain distance are dropped as possible cell candidates.

Additionally, the whole execution of the workflow takes 10-60 seconds per image, resulting in an increased speedup when analyzing the cell aggregates. Because of the simple usage and fully automated approach, the analysis for each image decreased from around one hour per image to less than a minute, depending on the resolution of the image. Additionally,

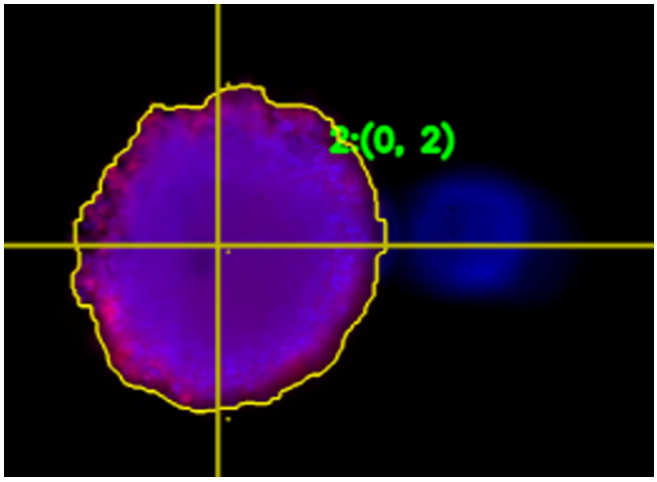

Figure 12. Example of a close ignored artifact

also the comparability of multiple results is increased, since the segmentation, and therefore the viability and area calculation, is deterministic, all images are analyzed with the same assumptions.

4. Conclusions

To summarize, the proposed method provides an easy-to-use algorithm for the automated analysis of the viability of 3D spheroidal cell aggregates in multi-sized spheroid arrays. Therefore, the fulfillment of the two main goals at this stage of the research, namely the viability analysis based on living and death cell ratio expressed by different color spectrums within fluorescent microscopy images and the automated size extraction, could be shown. Furthermore, it could be shown that the algorithm simplifies and generalizes the analysis process of multi-sized spheroid arrays. Additionally, by reducing human interaction also comparability between multiple experiments can be increased. Also, the segmentation accuracy could be increased compared to simple manual performed segmentation, which uses approximations like circles for the evaluation.

At the current state only a few data samples are available, since the algorithm was developed based on first biological experiments of the arrays. For the usage of this algorithm, it is assumed that artifacts are small and in low numbers. Within our experiments this was the case due to an antiadhesive coating in unwanted areas which results in a low number of artifacts within the image. Also, these artifacts have low intensity. Therefore, the used approach is suitable. But in a small number of cases, artifacts like merged cell aggregates or spilled fluorescent molecules appearing as unwanted artifacts within an image can prohibit good results. Additionally, in some cases, a column within a well-plate does not develop cell aggregates. In these cases, suggestions of their positions can be made, but they can only be within a specific range due to the

different distances between cell aggregates. Moreover, the estimation of missing columns is not used within the final analyses.

The workflow is currently used at the Vienna University of Technology in the Institute of Chemical Technologies and Analytics for mentioned analyses. For future usage automated segmentation algorithms like u-nets will be tested with more data and compared. This also allows a further reduction of internal parameters and more important user relevant inputs. In general dealing with a low amount of data and especially labeled data will be of high interest in the analysis of biological data and in bioimaging to support and allow ongoing biological experiments. To handle low amounts of data augmentation and transfer learning can be used. In addition to allow continuous and simplified labeling a feedback loop will be implemented. Therefore first traditional algorithms are necessary which do not require a high amount of data. These algorithms can be improved by more data dependent algorithms with the increased amount of labeled data. To further simplify the usage of this algorithm and to support the use of a feedback loop, additionally an interactive GUI for manual labeling will be implemented.

5. Acknowledgements

The Research described in this Paper was funded by the Christian-Doppler Forschungsgesellschaft (Josef Ressel Center for Phytogenic Drug Research).

References

Begley, C. G. and Ioannidis, J. P. (2015). Reproducibility in science: improving the standard for basic and preclinical research. *Circulation research*, 116(1):116–126.

Cui, X., Hartanto, Y., and Zhang, H. (2017). Advances in multicellular spheroids formation. *Journal of the Royal Society Interface*, 14(127):20160877.

Eilenberger, C., Rothbauer, M., Selinger, F., Gerhartl, A., Jordan, C., Harasek, M., Schädl, B., Grillari, J., Weghuber, J., Neuhaus, W., et al. (2021). A microfluidic multisize spheroid array for multiparametric screening of anticancer drugs and blood–brain barrier transport properties. *Advanced Science*, page 2004856.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Sezgin, M. and Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic imaging*, 13(1):146–165.

Shao, G., Li, D., Zhang, J., Yang, J., and Shangguan, Y. (2019). Automatic microarray image segmentation with clustering-based algorithms. *PloS one*, 14(1):e0210075.

Sirenko, O., Mitlo, T., Hesley, J., Luke, S., Owens, W., and Cromwell, E. F. (2015). High-content assays for characterizing the viability and morphology of 3d cancer spheroid cultures. *Assay and drug development technologies*, 13(7):402–414.

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., and the scikit-image contributors (2014). scikit-image: image processing in Python. *PeerJ*, 2:e453.

Zou, K. H., Warfield, S. K., Bharatha, A., Tempany, C. M., Kaus, M. R., Haker, S. J., Wells III, W. M., Jolesz, F. A., and Kikinis, R. (2004). Statistical validation of image segmentation quality based on a spatial overlap index1: scientific reports. *Academic radiology*, 11(2):178–189.