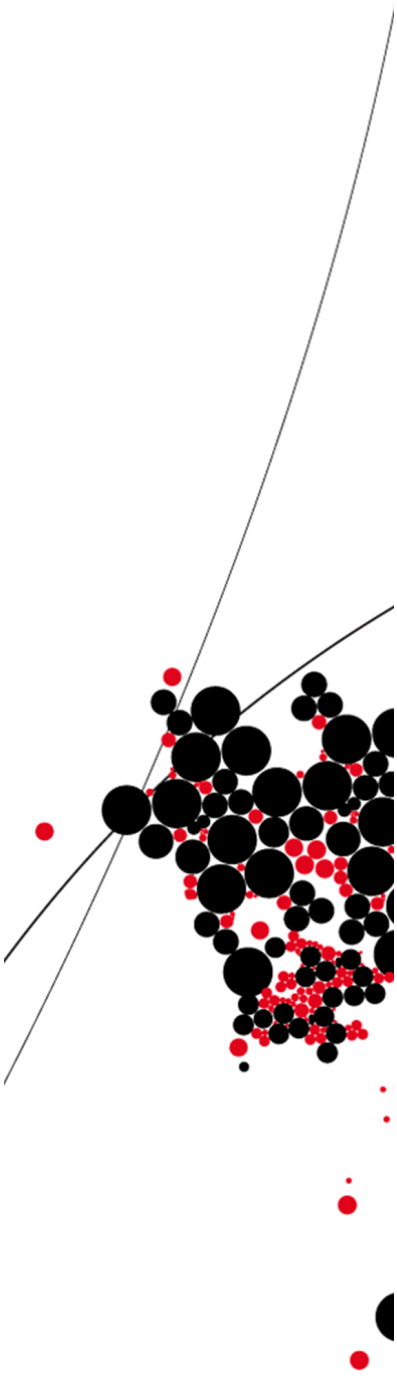
UNIVERSITY OF TWENTE.

Faculty of Science & Technology

The Effect of an Oxygen Gradient on Activation and Cellular Survival of Neuronal Networks

Luna Hagers
Biomedical Engineering
BSc. Thesis
July 2024



Supervisors:
dr. ir. J. le Feber
dr. G. C. Hassink
C. Cofino Fabres, MSc

Dept. of Clinical Neurophysiology
Faculty of Science and Technology
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

ACKNOWLEDGEMENTS

First of all, I would like to thank Oskari Kulta for allowing me to use his data, acquired knowledge on the subject and for answering all of my questions. I also wanted to thank my supervisors Dr. Ir. Joost le Feber, Dr. Gerco C. Hassink and Carla Cofino Fabres, MSc for all the guidance and support during these few weeks.

Luna Hagers

Enschede, 04-07-2024

Abstract

During an ischemic stroke, oxygen and glucose deficiencies are present, due to a decrease in blood perfusion. In the core, cells are irreversibly damaged and die via necrosis within minutes as apposed to cells surrounding the core, also known as the penumbra. In the penumbra, cells remain structurally intact, but synaptic failure reduces the amount of activity. If these cells cannot be restored, due to lack of reperfusion, they die via apoptosis hours or days later. This characteristic makes the penumbra a highly targeted region for potential treatments, however the exact mechanisms behind this balance between recovery and cell death are incompletely understood. Therefore we tried to mimic a more *in vivo* penumbra by modifying a previously published oxygen gradient generator and combining it with electrophysiological recordings and live fluorescent imaging of apoptotic and dead cells on neuronal cultures. During a 36 hour measurement, cells were exposed to 6 hours of normoxia, followed by 24 hours of gradient and finally another 6 hours of normoxia. The electrophysiological results showed that when the severity of hypoxia increased, the activity of the neurons decreased more rapidly. After 8 hours the activity increased back to baseline, due to disinhibition, followed by a large overshoot caused by medium evaporation. The fluorescent imaging showed that cells died from apoptosis and that this increased as the oxygen concentration decreased. If these findings are compared to previous studies where neuronal cultures were exposed to constant depths of hypoxia rather than a gradient, electrophysiological as well as the fluorescence results show similar patterns. This rejects the hypothesis that different areas in the gradient influence the behaviour of neurons in activity and survival.

∗∗ ***Dutch version*** ∗∗

Tijdens een ischemische beroerte ontstaan er zuurstof en glucose tekorten door een verlaging in bloed perfusie. In de kern worden neuronen onomkeerbaar beschadigd en gaan ze dood na een aantal minuten via necrose in tegen stelling tot het omliggende weefsel, ook wel de penumbra genoemd. In dit gebied zijn cellen nog intact, maar door veel synaptische afbraak zijn de neuronen minder actief. Als er snel genoeg reperfusie van bloed is zullen de synapsen zich weer herstellen, echter als dat niet het geval is zullen ook deze cellen dood gaan, maar dan via apoptose, uren of dagen na het incident. De exacte mechanismen die de balans in stand houden van cel dood of herstel in de penumbra zijn nog niet duidelijk. Om dit beter in kaart te brengen is er een *in vitro* model gemaakt die de penumbra nabootst door middel van een zuurstof gradiënt dat blootgesteld word over neuronale netwerken. Tijdens een meting van 36 uur, worden er zowel elektrofysiologische opnames als live fluorescente afbeelding van dode en apoptotische cellen gemaakt. In de eerste 6 uur worden cellen blootgesteld aan normoxie, gevolgd door een zuurstof gradiënt van 24 uur en hierna nog een keer 6 uur normoxie. De resultaten van de elektrofysiologie lieten zien dat hoe minder zuurstof er aanwezig was hoe minder activiteit de neuronen hadden. Na 8 uur ging de activiteit weer omhoog, door disinhibitie, gevolgd door een grote piek door verdamping van medium. De fluorescente afbeeldingen lieten zien dat er meer apoptose plaats vond als er minder zuurstof aanwezig was. Als deze bevindingen worden vergeleken met voorgaande experimenten, waarbij neuronale netwerken werden blootgesteld aan één constante zuurstof concentratie zagen we zowel voor de elektrofysiologische als de fluorescente afbeeldingen dezelfde patronen. De hypothese dat verschillende gebieden in het gradiënt invloed hebben op elkaar qua gedrag in activiteit en overleving, kan worden verworpen.

I. INTRODUCTION

Strokes are the second leading cause of death worldwide [1], where over 62% of all strokes are ischemic [2]. During an ischemic stroke there is a decrease in blood perfusion to certain parts of the brain due to an occlusion caused by a thrombus or an emboli [3], resulting in oxygen and glucose deficiencies [4].

In the core of an ischemic stroke, cells are irreversibly damaged due to perfusion levels of 10 mL/100 g/min [4], which normally lie around 45-60 mL/100 g/min [5], resulting in passive cell death, also known as necrosis, within minutes [6][7]. This, in turn, leads to a loss of neuronal function, leakage of cellular components and inflammatory responses [3]. The area surrounding the core, the penumbra, with perfusion levels between 14 and 35 mL/100 g/min [4], has the capacity to salvage functional impaired cells if there is quick enough reperfusion of oxygen and glucose [8]. These impaired neurons are initially intact, but due to synaptic failure their activity reduces [9]. If they cannot be restored because of a lack of reperfusion and activity, the neurons will likely undergo apoptosis, programmed cell death several hours or days later [6] [7]. The exact neurophysiological mechanisms that regulate this balance between recovery or cell death are still unknown, which makes the penumbra a highly targeted region for research that could ultimately lead to the treatment of ischemic stroke patients.

Over the last two decades there have been several studies enlightening the mechanisms involved in ischemic strokes, specifically the penumbra [10][11][12][13][14][15]. A common method they implemented in their research was an *in vitro* model, where neuronal network cultures were exposed to a constant depth of hypoxia. In such an isolated model of the penumbra, cell survival and recovery of activity increased upon mild stimulation via optogenetic and electrical activation [15]. A lack of interactions between the penumbra and healthy tissues or the ischemic core, prevents this model from optimizing stimulation parameters to improve cell survival or affect therapeutic activation. When you look at the complete environment of an ischemic stroke, these different areas could be stimulating the penumbra on their own. These stimulating factors are for example glutamate and potassium ions being released from necrotic cells in the core and persisting activity in healthy tissues [4]. Because of this reason a new experimental setup was designed to impose an oxygen gradient on neuronal cells to better mimic conditions in the *in vivo* penumbra.

In this research the oxygen gradient generator [4][16] is used to correlate local oxygen concentrations with neuronal activity and cell survival, it enables electrophysiological experiments alongside live fluorescent imaging of apoptosis and cell death.

II. MATERIALS & METHODS

A. Cell Culture

On the day of birth, neuronal cells of newborn Wistar rats were acquired by dissecting the head and isolating the cortices in Dulbecco's Modified Eagle Medium (DMEM, ThermoFisher, 21885-017). All surgical and experimental procedures complied with Dutch and European laws and guidelines (AVD110002016802). The cortices were treated with trypsin (Invitrogen, 25200-056) and then dissociated by trituration before seeding them on MEAs (Multichannel Systems, Reutlingen, Germany), precoated with a 50 g/mL solution of polyethylene-imine (PEI, Sigma-Aldrich, 181978-5g), with a concentration of $1 * 10^6$ cells/mL. On top of the MEA a silicone culture chamber was attached, to create a well that can contain 1.4 mL of culture medium, see Figure 1A. The culture medium contained Neurolbasal A medium (ThermoFisher, A2477501), B27 (ThermoFisher, 17504044), D-glucose (1.24 g/10mL stock), Pen/strep (ThermoFisher, 10378016), nerve growth factor (NGF, Sigma-Aldrich, n6009) and vitamin C (100 mg Ascorbic-acid in 10 mL H_2O). Cultures were incubated under standard conditions of 37°C, high humidity and 5% CO_2 for at least 3 weeks, to let the neurons mature. Half of the culture medium was refreshed twice a week and 12-24 hours before an experiment. All experiments were done during DIV21-DIV28. The day of the experiment additional medium was added to acquire a total volume of 1.4 mL at the start of experiments.

B. Gradient Generator Setup

The oxygen gradient generator [16] has two inserts through which a gas mixture flows, see Figure 1C. Depending on the settings one can create an oxygen gradient across the electrode area of the MEA, where the neuronal network is cultured. The gradient generator is made up of three parts, namely a PDMS culture chamber on top of a 60MEA200/30iR-Ti-w/o layout, an oxygen gradient lid and a lid lock to hold the system in place (Figure 1A). The MEA used, is made up of an 8 by 8 grid of 59 TiN electrodes (Ø: 30 μm) spaced 200 μm from each other, resulting in a recording area of 1.4 mm, and one reference electrode, see Figure 1B. The distance between the inserts is 1.5 mm to ensure a wide enough gradient over the MEA.

C. Gradient Generator Settings

The inserts of the gradient generator were attached to two sets of three flow controllers containing compressed air, nitrogen and carbon dioxide. Between the flow controllers and the inserts a bypass was used to reduce the flow to 5 mL/min to ensure that the membrane will stay intact during the measurement. This constant flow was verified beforehand by filling one large beaker halfway with water and one 10 mL cylindrical measuring flask almost to the top. The measuring flask was then placed upside down in the beaker without spilling too much of the water in the flask. This creates an air bubble at the top, which will be used to measure the flow. One of the inserts is placed under the cylindrical flask and when air flows through, the bubble becomes larger. During one minute the volume of the bubble is measured according to the marks on the flask, revealing what flow passes through the inserts.

The flow controllers were operated by a custom made LabView program. In this program several parameters could be set accordingly, such as the maintenance flow, flush duration, type of gas mixture and the duration of this mixture over a period of time. For this experiment both inserts were set to undergo a maintenance flow of 0.3 L/min and a flush duration of 0 s. To establish a gradient one of the flow controllers is programmed to let out a mixture of 95% compressed air with 5% CO_2 while the other one lets out a mixture of 95% nitrogen with 5% CO_2. A normoxic environment is created by having both flow controllers let out a gas mixture of 95% compressed air with 5% CO_2. In the penumbra there is both an oxygen and glucose deficiency. However in this experiment we chose to only compromise the oxygen availability, due to less interruption during the recording and a more sterile environment. All four oxygen gradient generators have validated oxygen gradients based on the calibration curves done by Oskari Kulta in Figure 2 [16]. From Figure 2B the average oxygen concentration per column for every generator was estimated to be [14.9, 12.9, 10.9, 8.90, 6.90, 4.90, 2.90, 0.90] in percentiles.

D. Electrophysiological Recordings

A custom software program, driving a PCI-6023E data-acquisition (National Instruments, USA) at a sample frequency of 25 kHz/channel, was used to record activity at all 59 electrodes. Action potentials (APs) were detected with the use of a threshold crossing algorithm whilst the RMS noise level was continuously updated. The detection threshold was set to 5.5 times the RMS noise level. Timestamps, channel numbers, 6 ms of wave shape and estimated noise levels of an electrode were stored for each detected event. Artifact detection and removal of invalid APs was done in a separate script, in MATLAB, before the analysis of the recorded data. This was done by using an algorithm where the shapes of putative APs were used to preform off-line artifact detection, hereafter false positive APs were counted and removed [4].

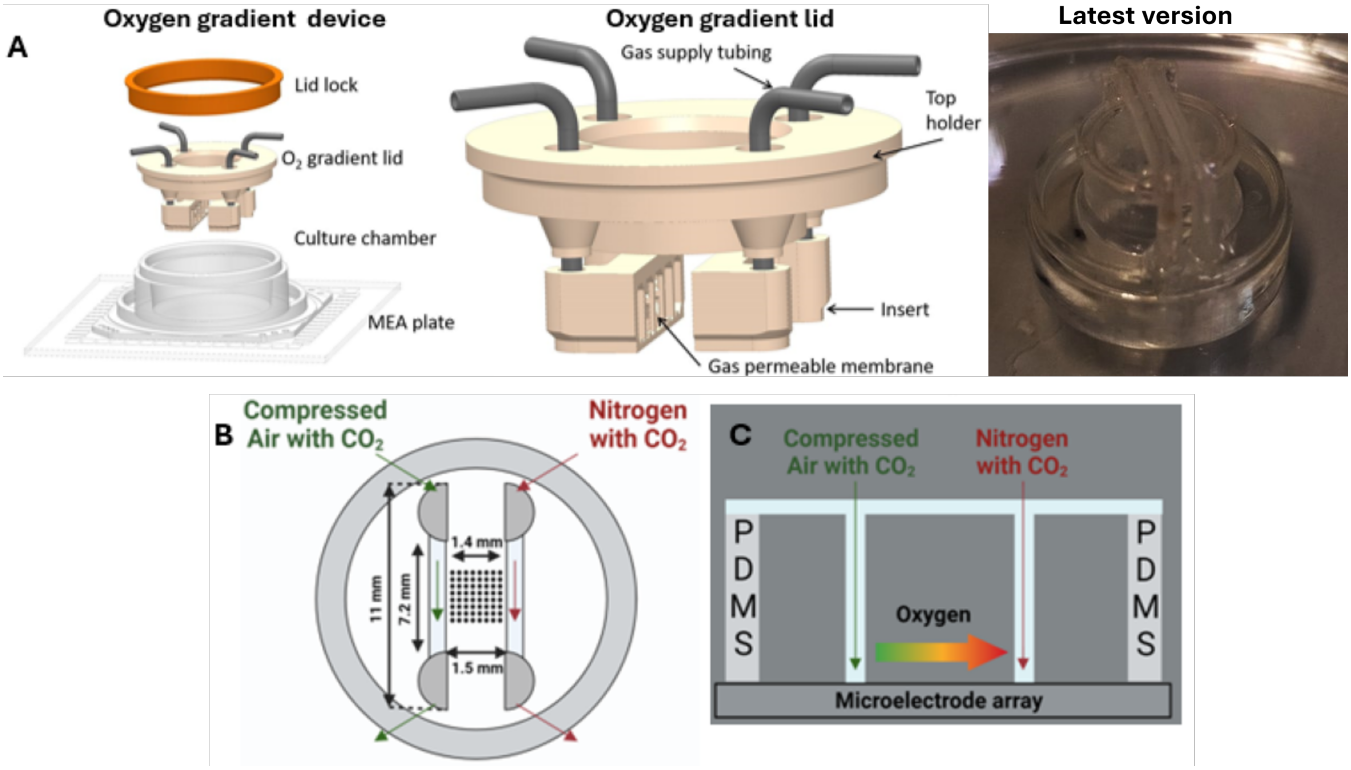


Fig. 1: A 3D model of the oxygen gradient generator (A) [4]. The lid lock seals the gradient lid on top the the PDMS culture chamber. In the middle this lid is depicted close up, consisting of a top holder and two inserts with gas supply tubing. Both inserts have a gas permeable silicone membrane allowing for gas exchange into the culture chamber. The right image shows the newest version that was used in the experiments. A top view (B) of the device sealed with the ring, indicating the dimensions of the device and the placement of the oxygen sensors grid. A schematic side view (C) shows this gas exchange more clearly. The compressed air is dragged towards nitrogen, creating a gradient of oxygen over the microelectrode array (MEA). These images were made by Oskari Kulta [16].

E. Cell Viability Recordings

Fluorescent imaging was done with a Nikon inverted microscope mounted with TMD-EF fluorescence equipment and a 4x objective. A live staining was added to the neuronal cultures to visualize apoptotic and dead cells. Based on the volume of the culture medium i.e. 1.4 mL, 1:500 diluted propidium iodide (PI, Sigma-Aldrich, p4170, 0.25 mM stock) was added to achieve a concentration of 0.5 μM and 1:1000 diluted caspase 3/7 (CAS3/7, Invitrogen, 2570382, 2 mM solution in DMSO) was added to achieve a concentration of 2.0 μM. A mercury lamp was used to excite the PI and CAS3/7 staining. To avoid photobleaching the cells were only exposed to light when an image was taken [17]. This was done with a custom made LabView program, where images were taken at a frequency of 4 images/hour and a Digicam Control program was used to acquire images on a computer in a remote controlled way with an exposure time of 0.5 s and an aperture of f/0.0.

F. Experiment Protocol

Before the start of an experiment, neuronal activity was measured to check the viability of the culture. Experiments were done only with cultures that showed synchronous bursting and at least 70% activity. An electrode was considered active if it recorded at least 1 spike/3 min. If these requirements were met, the live staining of PI and CAS3/7 could be added into the culture chamber 10 minutes before the experiment. Depending on the location of the active electrodes, the placement of the gradient generator on top of the culture chamber was either creating a row-wise or column-wise gradient (Figure 3). Accordingly, all analyses were done row or column-wise, but for clarity we will refer to column-wise gradients and analysis in the rest of this report. The generator was then secured with the lid lock and the mass was measured of the whole device.

At the start of an experiment the MEA was weighed and placed in a temperature-controlled recording setup [4] for a 36 hour recording during which the electrophysiological and fluorescence data were acquired. The first six hours were under normoxic conditions (baseline). After baseline, cultures were exposed to and oxygen gradient for 24 hours. Finally, cultures

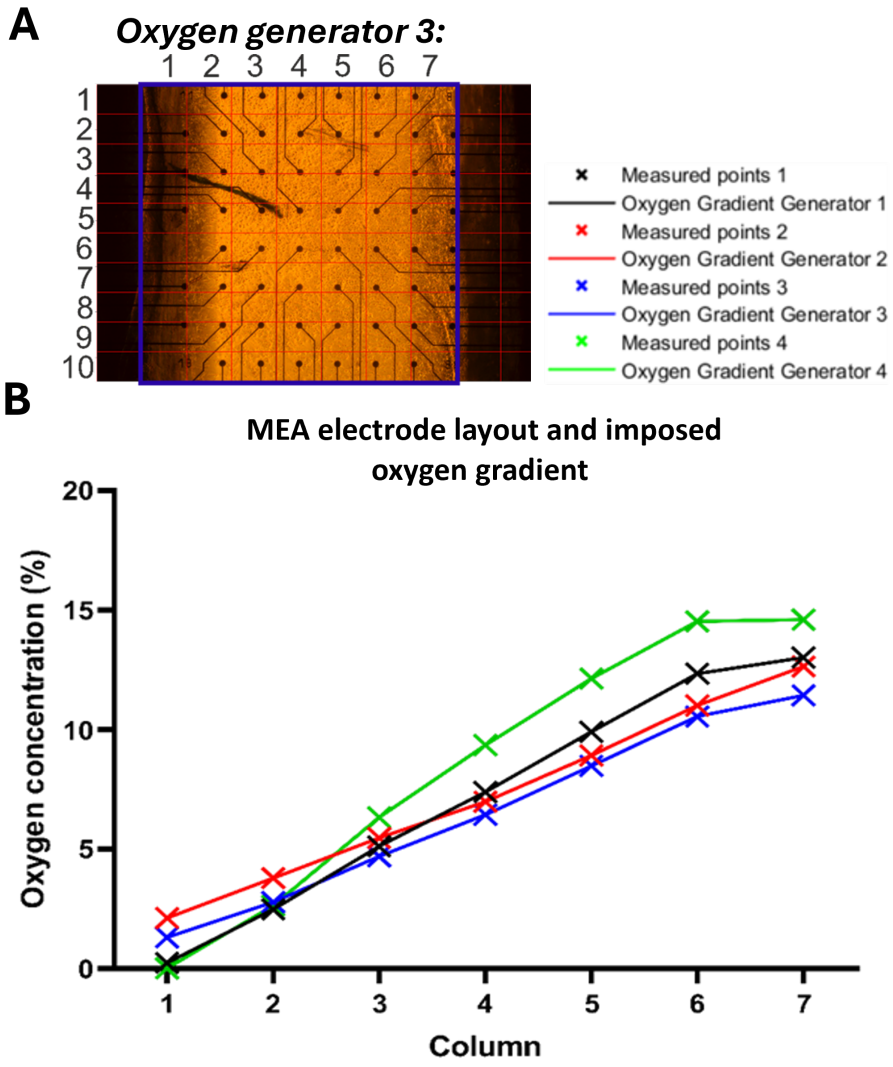


Fig. 2: (A) Indicates the area that was used to analyse the different oxygen concentrations per column. The graph in (B) shows the average oxygen concentration per column. A linear line of y = 2x-1.1 was fitted over the graph to indicate the average oxygen concentration per column for every generator. This resulted in the following concentrations from high to low in percentiles: [14.9, 12.9, 10.9, 8.90, 6.90, 4.90, 2.90, 0.90]. These images and the analysis of the calibration were done by Oskari Kulta [16].

were exposed to another six hours of normoxia. The MEA was taken out of the setup, weighed again and medium lost due to evaporation was replenished. Two days after the experiment an extra measurement of one hour was conducted under normoxic conditions. The difference with respect to the previous experiment was that on top of the culture chamber no gradient lid was needed, because we did not induce an oxygen gradient. Instead a cap was placed over the cells, to protect them from evaporation and bacterial infection. During these extra measurements, MEAs were placed under a Plexiglass hood that received a continuous gasflow (2 L/min) of 95% air and 5% CO_2. See Appendix V-B to find experimental notes on the last four experiments. For the experimental notes from Kulta go to the following thesis [16].

G. Analysis

Exclusion Criteria

Several criteria were established for both electrophysiological and fluorescence data. These criteria determined whether an experiment or parts of an experiment were excluded from the final analysis.

As stated before, an electrode is active when it can uphold a spike rate of 20 spikes/hour during the first six hours of normoxia. At least 50% of the electrodes should be active in a column to be used in the further analysis. In table I the

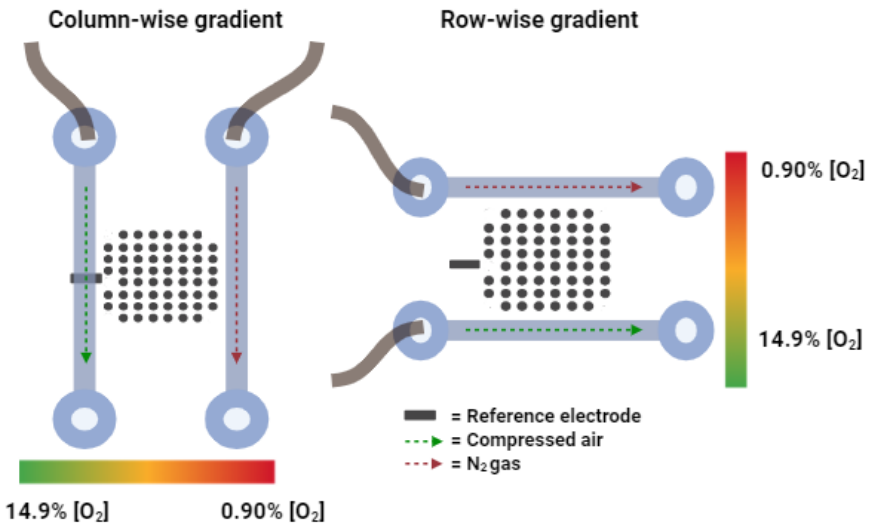


Fig. 3: The direction of the gradient depends on the orientation of the oxygen gradient generator. If it is placed perpendicular with respect to the reference electrode, the different concentrations of oxygen flow from left to right. This creates a column-wise gradient. If the generator is placed parallel to the reference electrode, the oxygen levels travel from bottom to top, creating a row-wise gradient.

minimum amount of required spikes during baseline are depicted.

For the fluorescence images the maximum amount of counted dead and apoptotic cells during baseline was 20 cells/hour per column. Overexposed images were also not taken into account any further.

Electrophysiological Analysis

A MATLAB script was made [16] for analysing the electrophysiological data. In short this script calculates the normalized array-wide fire rate (AWFR) per oxygen concentration over 36 hours. This column-wide fire rate (CWFR) is the sum of the number of APs of all electrodes for a certain column in one hour time bins. The CWFR was normalized to its mean value during baseline to avoid the large spread associated with firing rates that widely differed between cultures. Finally, CWFR curves were averaged across all cultures to assess whether and how activity was related to local oxygen concentrations (n = 7). In the same manner as described above the electrode-wide fire rate (EWFR) was analysed per experiment. See Appendix V-C to find the following scripts.

Cell Viability Analysis

A further developed MATLAB script from Taxis et al. [18] was used to automatically count the number of stained apoptotic and dead cells in the acquired images per column. First an images of 3264x3264 pixels was divided into eight equal parts of 408x3264 pixels, then the image was filtered using a 2D moving average. For each pixel an average was taken of 41x41 pixels surrounding that pixel. If local maxima in the filtered image exceeded a threshold, set at 70% of the mean green and red intensity, it was counted as an apoptotic cell or dead cell. In the end you will end up with two matrices of 144x8. Due to environmental lights some images were overexposed and gained an average blue value of above 1, these values

TABLE I: The minimum amount of spikes per column or row over the first 6 hours that meet the requirement of 50% of the electrodes being active.

	Number of Electrodes	Number of Spikes (6h Baseline)
Column Wise Gradient:		
Col 1	5	240
Col 2-7	8 / Col	480
Col 8	6	360
Row Wise Gradient:		
Row 1,8	6 / Row	360
Row 2-4, 6-7	8 / Row	480
Row 5	7	360

were then excluded from the analysis by changing them to Not a Number (NaN). After this the data of both counts was pinned to 1 hour points by getting the average of 4 images and checked if the count exceeded 20 cells/hour during baseline. If this was the case the whole column was set to NaN. In the end this resulted in a matrix of 36x8, where the columns correspond to a specific oxygen concentration and the rows to the duration of the experiment. If there were three of less images left the mean was still taken of the remaining images. Finally all experiments were compiled together, resulting in a 180x8 matrix. In short the average values corresponding to a specific hour and oxygen concentration are summed up for all experiments and averaged anew, creating again a 36x8 matrix. This was then plotted with standard deviations to get a better estimation of the behaviour of neuronal cells at local oxygen concentrations over 36 hours (n = 6). See Appendix V-D to find the following scripts.

Statistical Analysis

Both electrophysiological and fluorescence data were tested on normality using the Shapiro-Wilk test. When normally distributed, ANOVA was used to test if the differences between oxygen concentrations were significant for the electrophysiological data and the correlation coefficient was determined to look at the relationship of the cell viability with respect to the oxygen concentrations. A p-value less than 0.05 indicated a significant variance and the data was then presented as mean $\pm$ SD. If the group values were not normal, a Kruskal-Wallis test was applied and the data was shown as median $\pm$ interquartile ranges (IQR) between 25$\%$ and 75$\%$.

III. RESULTS

Neuronal networks cultured on top of a MEA were exposed to normoxic conditions for 6 hours, followed by 24 hours of gradient and an additional 6 hours of normoxia. During this measurement electrophysiological recordings were acquired alongside fluorescence imaging of dead and apoptotic cells.

A. Electrophysiological Results

In total ten experiments were conducted, but only some CWFRs of seven experiments were used in the final analysis. Two whole experiments were excluded due to technical malfunctions and one was excluded because of extreme artifacts. For the seven other experiment an overview of all inactive electrodes and the final number of included experiments used per oxygen concentration are depicted in Table II.

The results are shown in Figure 4. In all separate experiments the CWFR decreased with the onset of the gradient, where the activity remained below baseline or in some cases of less severe hypoxia, stayed around the baseline. In the first hours of the gradient there is a noticeable distinction between severities of hypoxia. If there is less oxygen available, the CWFR has a steeper decline. This progression was also perceived when all seven experiments were pooled together, where there were statistically significant differences amongst the concentrations between hours six to twelve during the gradient (Figure 4A & B, ANOVA p $<$0.001). Nonetheless this clear separation between concentrations becomes less coherent after around 14 hours and eventually leads to a very large increase of activity, with an exception to the CWFRs at 14.90% and 0.90% oxygen.

This large increase could be the result of medium evaporation or disinhibition. For one experiment the volume of evaporation was estimated by measuring the weight before and afterwards. This resulted to be around 1 mL of medium with respect to at total volume of 1.4 mL after 42 hours. To look in what manner this phenomenon translated electrophysiologically, the firing rate per electrode was analysed. In this experiment the EWFR showed a synchronous pattern per column as well as through all oxygen concentrations of a large increase in activity above baseline after around six hours of the initiated gradient, which was eventually followed by a decrease in activity. This large simultaneous increase in activity at all levels of hypoxia with respect to baseline was also visible in other experiments, where some EWFRs even decrease to zero after a period of time. However in some experiments one or two electrodes digressed from the rest during the first hours of the gradient. These electrodes tended to have an increased activity earlier on, which could be the result of disinibition. See Appendix V-A for all the separate results per experiment.

TABLE II: The amount of inactive electrodes per oxygen concentration. An electrode is inactive when the spike rate is less than 20 spikes/hour during baseline. Only experiment 5_O was exposed to a row-wise gradient. At 8.90% [O_2] the amount of electrodes necessary to be perceived as active are rounded down for this experiment. So only 3/7 electrodes need to be active in order to be used. The same goes for the column wise experiments at 14.9% [O_2], were there only need to be 2/5 active electrodes. Red indicated inactive electrodes were left out of the analysis.

Experiment:	**14.9% [O_2]**	**12.9% [O_2]**	**10.9% [O_2]**	**8.90% [O_2]**	**6.90% [O_2]**	**4.90% [O_2]**	**2.90% [O_2]**	**0.90% [O_2]**
3_O	3/5	4/8	3/8	2/8	2/8	2/8	3/8	2/6
4_O	**5/5**	4/8	0/8	0/8	0/8	0/8	3/8	**6/6**
5_O	**6/6**	**8/8**	4/8	4/7	4/8	**6/8**	**6/8**	**5/6**
6_O	1/5	3/8	4/8	4/8	**5/8**	3/8	2/8	**4/6**
7_O	1/5	3/8	1/8	1/8	1/8	2/8	1/8	**4/6**
8_O	2/5	3/8	0/8	1/8	1/8	0/8	1/8	3/6
3_L	**5/5**	**8/8**	**5/8**	4/8	3/8	2/8	4/8	1/6
n =	**4**	**5**	**6**	**7**	**6**	**6**	**6**	**3**

B. Cell Viability Results

The final data analysis of the fluorescence images was pooled from six experiments. Four experiments had technical difficulties, whereas some columns of the resulting six experiments did not meet the requirement of a maximum of 20 counted cells per hour during baseline or were not usable due to artifacts. See Appendix V-A for all the separate results per experiment.

Images after the first 15 minutes of normoxia, gradient and again normoxia are represented in Figure 5A. After time passes the fluorescence of both the PI and CAS3/7 increases. The results from the analysis are shown in Figure 5B. Individual oxygen concentrations show an increase in apoptotic and dead cells over time, where the apoptotic cell count stays higher than the dead cell count. As the severity of hypoxia increases, the amount of apoptotic and dead cells increases as well. The correlation between the amount of dead and apoptotic cells with respect to the oxygen concentration are depicted in Figure

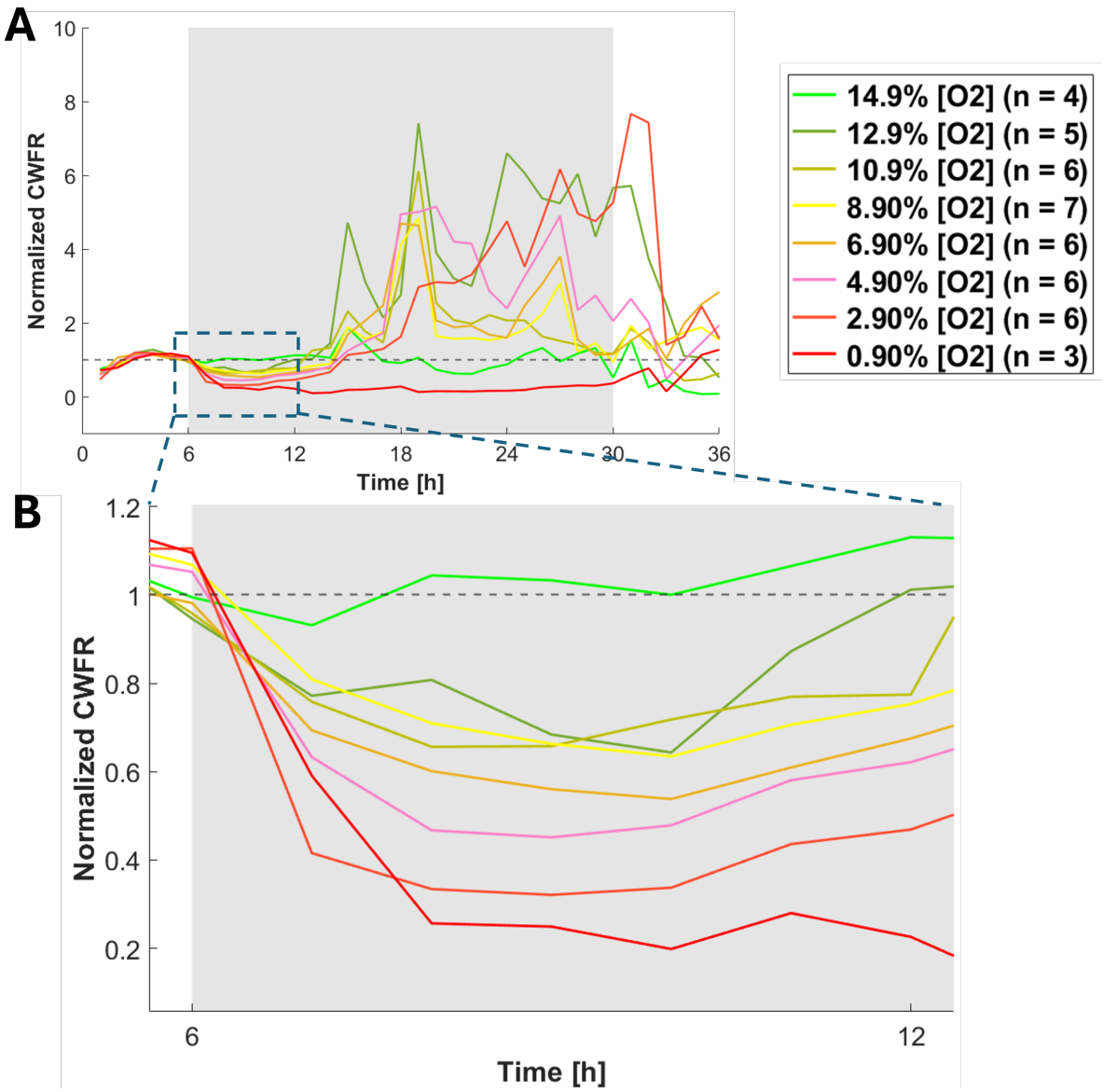


Fig. 4: Multi-electrode array (MEA) system recorded electrophysiological activity. The normalized sum of the number of APs per column in one hour time bins, also referred to as the column-wide fire rate (CWFR), over 36 hours is shown in (A) pooled from a total of seven experiments, where some columns were excluded due to not meeting the requirement of a minimum of 50% activity. (B) zooms in on the behaviour of the activity during the first six hours where the gradient is established. The correlation between the severities of hypoxia are significant ($p < 0.001$). In both graphs the colors of the lines correspond to a specific oxygen concentration, as shown in the legend.

6. A strong negative correlation is perceived with high statistical significance ($p < 0.05$ and $p < 0.001$, $R = -0.80$ and $R = -0.93$) [19].

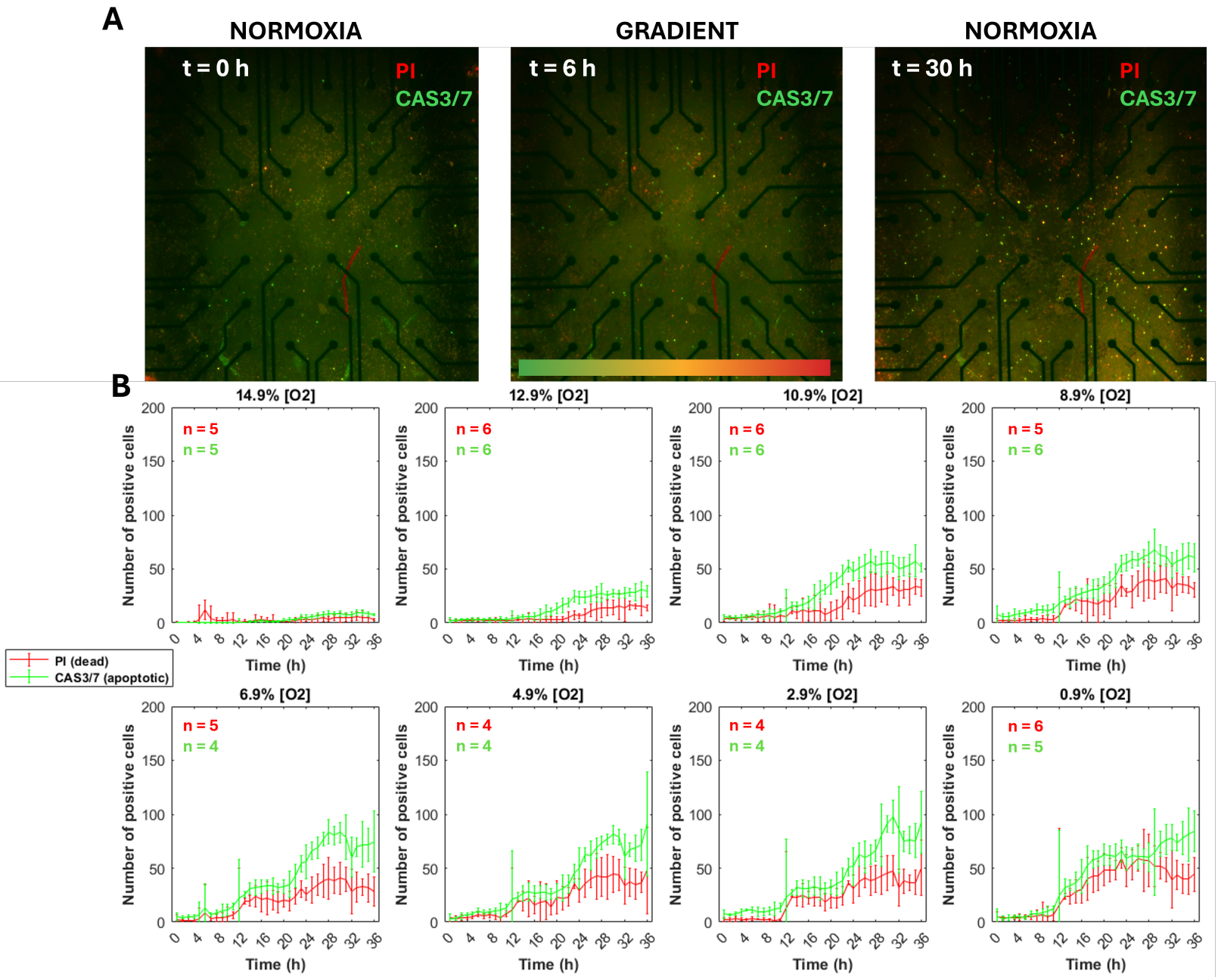


Fig. 5: Example photographs during the different stages of experiment 3_O are shown in (A). Rat cortical cultures are stained with PI (red) and CAS3/7 (green), 10 minutes before the start of the measurement. Apoptotic and dead cells were automatically detected and counted per column by a custom algorithm. The three images represent the first 15 minutes of the stage that they are in, where the severity of hypoxia increases from left to right during the gradient. (B) shows the progression of the cell count for both apoptotic and dead cells over time per oxygen concentration. The amount of experiments used for each column are illustrated in the graph itself, where the color corresponds to the staining used (red = PI and green = CAS3/7). The data is shown as mean $\pm$ STD.

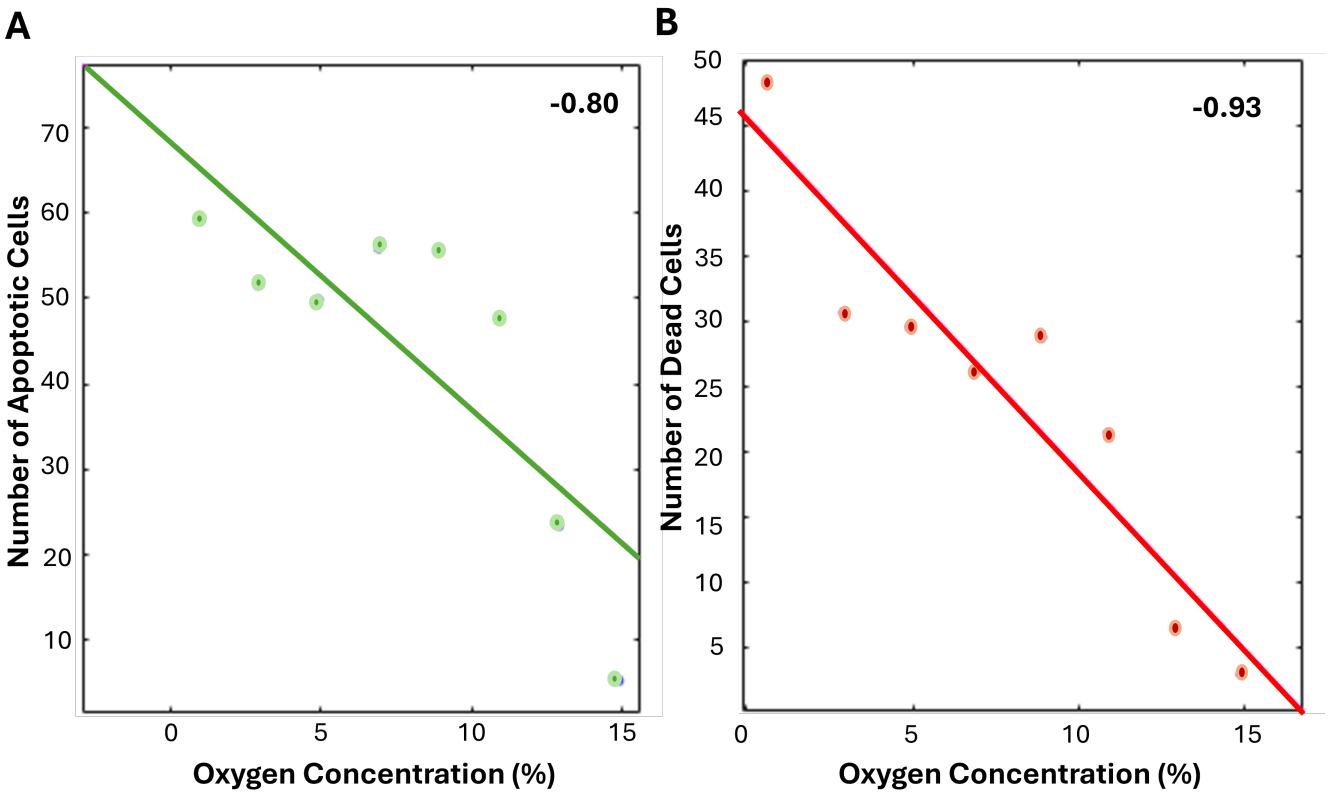


Fig. 6: (A & B) show the negative correlation between the apoptotic cell count (left) and dead cell count (right) with respect to the oxygen gradient. These cell counts are a combination of all six experiments at t = 24h. The scatter plot for the apoptotic count holds an R = -0.80 and the scatter plot for the dead cell count holds an R = -0.93. Both correlation coefficients also gave rise to p-values. For the apoptotic count $p < 0.05$ and for the dead count $p < 0.001$ suggesting a high statistical significance in both cases.

IV. DISCUSSION

In the present study, the effect of an oxygen gradient on the activity and cell viability on neuronal cell cultures was investigated. When the gradient was established the CWFR showed significant differences between oxygen concentrations, where the activity decreased as the severity of hypoxia increased. The amount of apoptotic and dead cells increased as the as the oxygen concentration decreased.

The neuronal activity in the penumbra after a stroke is strongly reduced due to synaptic failure [9]. Which is also reflected in the normalized CWFR during the first two hours when the gradient is applied. The more severe hypoxia gets, the steeper the decrease in activity is. Something that is also observed when neuronal cultures are exposed to constant depths of hypoxia for periods of 48 hours [9]. When both results were compared, there were almost no visible differences in the amount of activity after the decrease of the CWFR and the AWFR for different levels of hypoxia. The only noticeable difference was at an oxygen and nitrogen ratio of 10:90, where the CWFR was slightly lower than the AWFR. Naturally not all concentrations were tested in that article, but the depths of hypoxia that were tested did not show extensive differences with respect to a more *in vivo* like model of the penumbra.

Another observation of this experiment that resembles the trajectory of the discussed article [9] is the increasing CWFR towards baseline after around 4 hours of the gradient [14]. This is visible in all but two oxygen concentrations, namely the columns that had the most and the least amount of oxygen. In these experiments done by Le Feber et al. [14] partial recovery of activity above the baseline was observed after 4-6 hours of hypoxia in some cases at different constant hypoxic depths. They related this to an increased activity in 1-3 electrodes once hypoxia was initiated. Likewise behaviour was observed in the EWFR in this experiment, where some electrodes tended to increase in activity once the gradient was initiated. This phenomenon could be caused by disinhibition. When there is synaptic failure, tonically firing neurons that were previously inhibited become disinhibited, meaning that they will fire more rapidly once the gradient is initiated and lead to an increased EWFR. Another article suggest that disinhibition is regulated by a subset of highly active inhibitory interneurons, that express vasoactive intestinal peptide (VIP), to uphold cortical excitability during a stroke [20]. Which could also result in an increased EWFR. However this does not explain why the column exposed to the most severe hypoxia shows no recovery to baseline like the rest. One reason could be that the neurons have been damaged irreversibly as in the core. But in contrast to cell death in the core that only takes minutes and is passive [7], this cell death is much slower and programmed, which was concluded from the cell count of both dead and apoptotic cells. In all columns the apoptotic count stays higher during the experiment than the dead cell count, indicating that the neurons died via apoptosis rather than necrosis.

Nonetheless disinhibition does not explain the high increase of activity between 14-33 hours. What could result in these observation is medium evaporation in the culture chamber. In the experiments that were conducted, a gradient generator device was used that covered the cells as apposed to a watertight foil that is used in less recent studies to mimic the penumbra [14] [9] [18]. This device consists of elements glued together and was cleaned by soaking it in 70% ethanol (EtOH). EtOH is known to dissolve adhesives [21], which could result in the device not being airtight anymore, accelerating the process of evaporation. From the results of the EWFRs of experiments 2_L and 3_L, where it was known that medium evaporation had occurred after weighing the volume at the beginning and the end of the experiment, synchronous activity in all electrodes was observed. The EWFRs showed simultaneous peaks in activity followed by a decrease to zero after a period of time. If we translate this to the other experiments we also saw a large increase in activity, where the peaks coincided between concentrations, but not all of them digressed to zero afterwards. This could mean that the medium was not fully evaporated or other artifacts played a roll, such as debris moving around. This displacement of debris was also visible in the fluorescent images.

Aside from the electrophysiology, the fluorescence imaging showed that with increasing severity of hypoxia both apoptotic and dead cell count increased as well, which aligns with the results from a paper by Taxis et al [18]. They observed that more apoptosis occurred when there was less activity in the cultures during hypoxia at a constant depth, with an oxygen and nitrogen ratio of 10:90. If there was more residual activity during hypoxia the amount of apoptosis was small. Even though the pattern of progression is similar, the amount of apoptotic cells counted in this research are present in a fewer extent. This could be because of persisting activity and neurotrophic factors released by the least severe hypoxic area of the cell culture [22]. Nevertheless a more likely explanation is that there were more cells overall. In order to say this for certain, both counts should be normalized.

However, all these findings should be interpreted with caution. The most evident limitation of this experiment is the large amount of medium evaporation, which causes debris to move around in the culture chamber and the microscope to be out of focus. These consequences make both electrophysiological data and fluorescence imaging during this period less reliable.

Aside from tuning the analysis to undermine these artifacts, one solution to this problem would be to shorten the duration of the measurement, but this would not minimize the amount of evaporation. To do so, the design or the assembly of the generator lid should be modified, which could also be combined with reducing the contact between the device and the cells. This could barely be avoided during the assembly of the generator, resulting in compressed cells in the area where the inserts were. These modifications would entail adding footholds outside the area of the electrodes to indicate that the bottom of the MEA has been reached, making the culture chamber less flexible by fabricating it out of biocompatible PDMS-PEG [23] and using another type of glue for the assembly that cannot be dissolved by EtOH.

In conclusion, once neuronal cultures are exposed to a gradient and the severity of hypoxia becomes greater, the CWFR decreases more rapidly and the amount of apoptosis increases. These findings follow a similar trajectory as previous studies, where neuronal cultures were exposed to constant hypoxic depths. This suggest that there is no difference in the behaviour of neuronal cells between exposure to a gradient or a constant depth of hypoxia, contrary to what was hypothesised. Either way this research has brought new insight into modelling the penumbra with the use of an oxygen gradient rather than a constant oxygen concentration.

REFERENCES

[1] World Health Organization. The top 10 causes of death; 2020. Last accessed 3 May 2024.

[2] World Stroke Organization. Incidence and Prevalence for all Ischemic Stroke; 2022. Last accessed 3 May 2024.

[3] Kuriakose D, Xiao Z. Pathophysiology and Treatment of Stroke: Present Status and Future Perspectives. 2020 International Journal of Molecular Sciences; 21:7609. doi:10.3390/ijms21207609.

[4] Santiago J, Kreutzer J, Bossink E, Kallio P, le Feber J. Oxygen gradient generator to improve in vitro modeling of ischemic stroke. 2023 Frontiers in Neuroscience; 17:Not available. doi:10.3389/fnins.2023.1110083.

[5] Meredith L Turetz MD RGCM. Neurobiology of disease. Elsevier ; 2011.

[6] Fink SL, Cookson BT. Apoptosis, Pyroptosis, and necrosis: Mechanistic description of dead and dying eukaryotic cells. 2005 Infection and immunity; 73(4):1907-16. doi:10.1128/iai.73.4.1907-1916.2005.

[7] Radak D, Katsiki N, Resanovic I, Jovanovic A, Sudar-Milovanovic E, Zafirovic S, et al. Apoptosis and Acute Brain Ischemia in Ischemic Stroke. 2017 Current Vascular Pharmacology; 15:115-22. doi:10.2174/1570161115666161104095522.

[8] Sodaei F, Shahmaei V. Identification of penumbra in acute ischemic stroke using multimodal MR imaging analysis: A case report study. 2020 Radiology Case Reports; 15:2041-6. doi:10.1016/j.radcr.2020.07.066.

[9] le Feber J, Erkamp N, van Putten MJAM, Hofmeijer J. Loss and recovery of functional connectivity in cultured cortical networks exposed to hypoxia. 2017 Journal of Neurophysiology; 118:394-403. doi:10.1152/jn.00098.2017.

[10] Datta A, Park JE, Li X, Zhang H, Ho ZS, Heese K, et al. Phenotyping of an in vitro model of ischemic penumbra by ITRAQ-Based shotgun quantitative proteomics. 2009 Journal of proteome research; 9(1):472-84. doi:10.1021/pr900829h.

[11] Dwyer TA, Earl DE, Wang L. The utility of a new in vitro model of the Stroke Penumbra. 2008 The journal of neuroscience; 28(26):6537-8. doi:10.1523/jneurosci.1841-08.2008.

[12] ir A S Pires Monteiro. Human neuronal in-vitro model of the ischemic penumbra ; 2019.

[13] Monteiro SP, Voogd E, Muzzi L, Vecchis GD, Mossink B, Levers M, et al. Neuroprotective effect of hypoxic preconditioning and neuronal activation in a in vitro human model of the ischemic penumbra. 2021 Journal of Neural Engineering; 18:036016. doi:10.1088/1741-2552/abe68a.

[14] Feber JL, Pavlidou ST, Erkamp N, Van Putten MJAM, Hofmeijer J. Progression of neuronal damage in an in vitro model of the ischemic penumbra. 2016 PloS one; 11(2):e0147231. doi:10.1371/journal.pone.0147231.

[15] Muzzi L, Hassink G, Levers M, Jansman, Hofmeijer J, Van Putten M, et al. Mild stimulation improves neuronal survival in an in vitro model of the ischemic penumbra. 2019 Journal of neural engineering; 17(1):016001. doi:10.1088/1741-2552/ab51d4.

[16] Kulta O. Ischemic penumbra-on-chip: studying the role of oxygen gradient. Tampere University ; 2023.

[17] Kwon J, Elgawish MS, Shim S. Bleaching-Resistant Super-Resolution fluorescence microscopy. 2022 Advanced science; 9(9). doi:10.1002/advs.202101817.

[18] Di Bordonia E Valnigra DT, Hassink GC, Levers MR, Frega M, Hofmeijer J, Van Putten MJAM, et al. The Association between Hypoxia-Induced Low Activity and Apoptosis Strongly Resembles That between TTX-Induced Silencing and Apoptosis. 2022 International journal of molecular sciences; 23(5):2754. doi:10.3390/ijms23052754.

[19] Akoglu H. User's guide to correlation coefficients. 2018 Turkish journal of emergency medicine/Turkiye acil tıp dergisi; 18(3):91-3. doi:10.1016/j.tjem.2018.08.001.

[20] Motaharinia M, Gerrow K, Boghozian R, White E, Choi SE, Delaney KR, et al. Longitudinal functional imaging of VIP interneurons reveals sup-population specific effects of stroke that are rescued with chemogenetic therapy. 2021 Nature communications; 12(1). doi:10.1038/s41467-021-26405-6.

[21] Iliev G, Hardan L, Kassis C, Bourgi R, Cuevas-Suárez CE, Lukomska-Szymanska M, et al. Shelf Life and Storage Conditions of universal adhesives: A literature review. 2021 Polymers; 13(16):2708. doi:10.3390/polym13162708.

[22] Warm D, Bassetti D, Schroer J, Luhmann HJ, Sinning A. Spontaneous activity predicts survival of developing cortical neurons. 2022 Frontiers in cell and developmental biology; 10. doi:10.3389/fcell.2022.937761.

[23] Mair DB, Williams MAC, Chen JF, Goldstein A, Wu A, Lee PHU, et al. PDMS–PEG block copolymer and pretreatment for arresting drug absorption in microphysiological devices. 2022 ACS applied materials interfaces; 14(34):38541-9. doi:10.1021/acsami.2c10669.

V. APPENDIX

A. RESULTS PER EXPERIMENT

EXPERIMENT 3 (Oskari)

The electrophysiological data is visible in Figure 7A and B, whereas the raw data is shown in Figure 7C. At the end of the experiment the CWFR starts to increase.

The fluorescence data is visible in Figure 8. At the end of the measurement the focus of the microscope shifted, which is visible in Figure 8A and also translates to the graphs after the apoptotic and dead cells have been counted (Figure 8B). This could be happening due to medium evaporation.

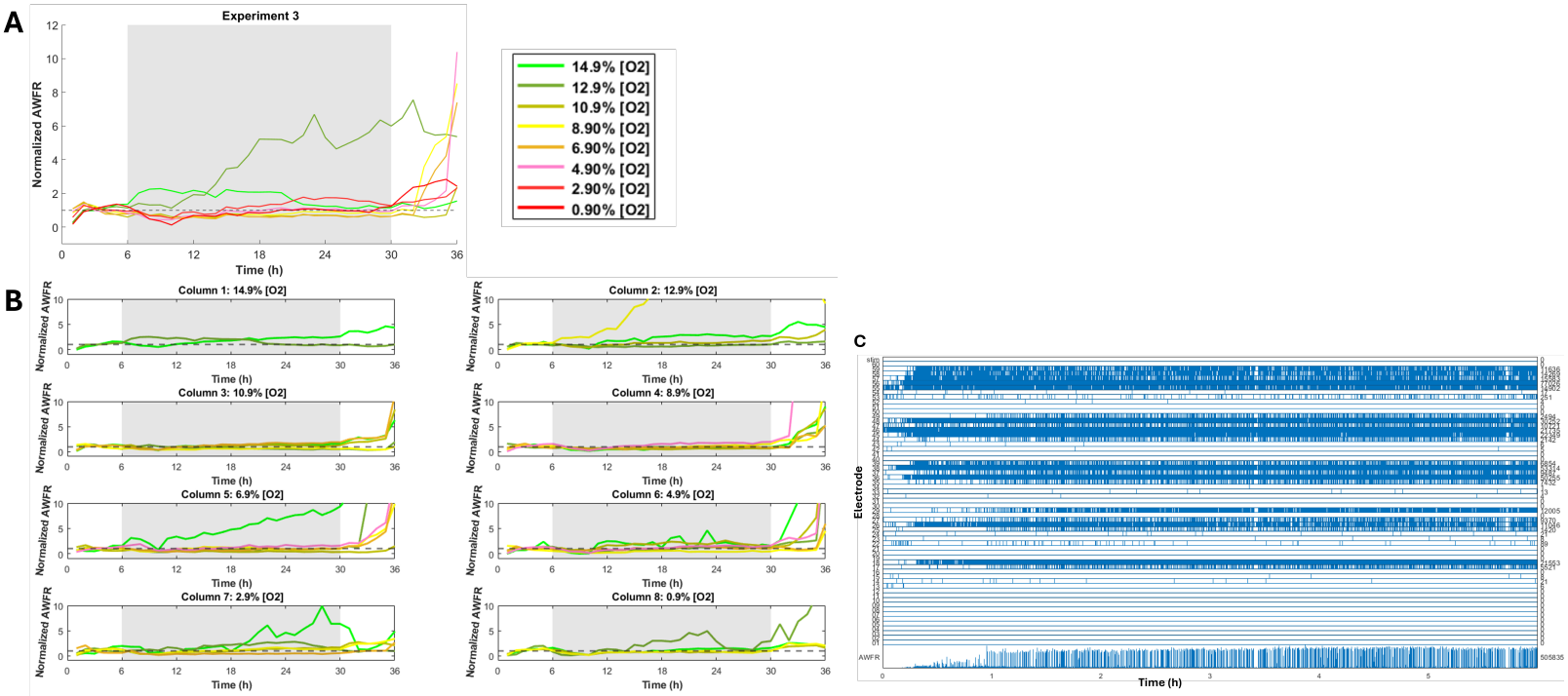


Fig. 7: (A) depicts the electrophysiological data of experiment 3, normalized over the first 6 hours of the measurement. Every line resembles a different oxygen concentrations from 14.9% to 0.90%. In (B) every electrode from a specific concentration is extracted and individually visualised in the same manner as the data from (A). Rasterplot (C) shows the activity of every electrode in the first 6 hours, where the neuronal cells are exposed to a normoxic environment.

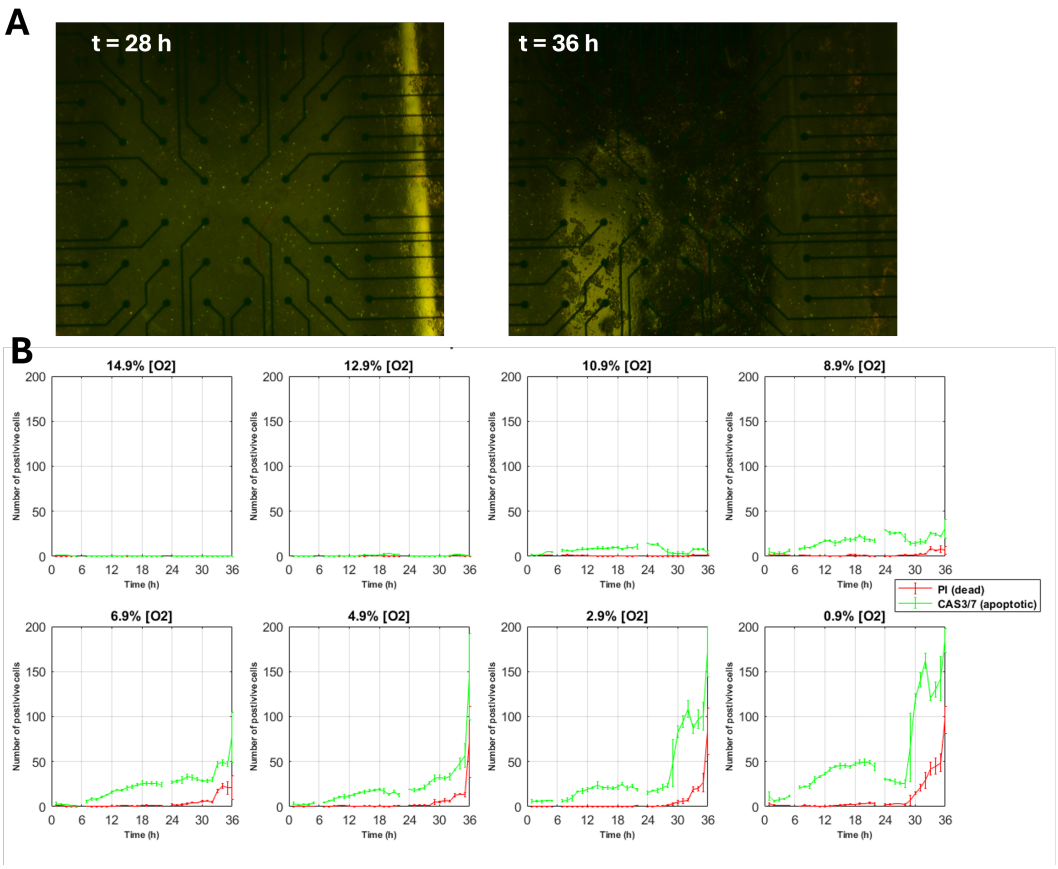


Fig. 8: An example of fluorescence images taken during measurement 3_O are depicted in (A). There is a shift in focus, due to medium evaporation probably. The graphs below that resemble the apoptotic (green) and dead (red) cell count at different oxygen concentrations (B).

EXPERIMENT 4 (Oskari)

The electrophysiological data is visible in Figure 9A and B, whereas the raw data is shown in Figure 9C. Between 21-24 hours there is a large increase in activity in all columns.

The fluorescence data is visible in Figure 10. During the measurement air bubbles started to form, which are visible in Figure 10A indicated by the orange rectangles. This also translated to the graphs after the apoptotic and dead cells have been counted (Figure 10B) between 21-24 hours. This could be happening due to medium evaporation.

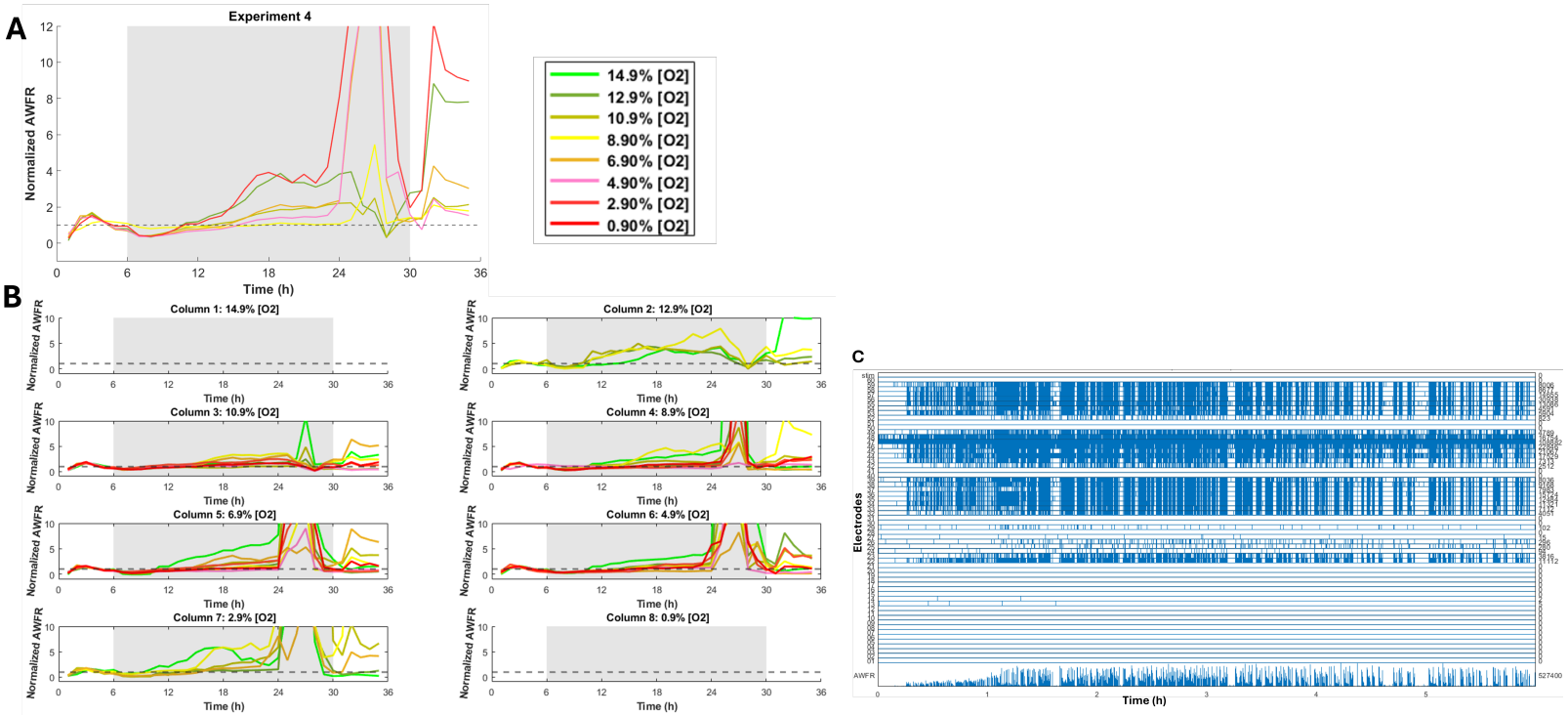


Fig. 9: (A) depicts the electrophysiological data of experiment 4, normalized over the first 6 hours of the measurement. Every line resembles a different oxygen concentrations from 14.9% to 0.90%. In (B) every electrode from a specific concentration is extracted and individually visualised in the same manner as the data from (A). Rasterplot (C) shows the activity of every electrode in the first 6 hours, where the neuronal cells are exposed to a normoxic environment.

EXPERIMENT 5 (Oskari)

The electrophysiological data is visible in Figure 11A and B, whereas the raw data is shown in Figure 11C. After 12 hours the CWFR starts to increase drastically in all rows.

The fluorescence data is visible in Figure 12. During the measurement debris between the cells started to dislocate, which is visible in Figure 12A indicated by the orange rectangles. This also translated to the graphs after the apoptotic and dead cells have been counted (Figure 12B) in the most severe case of hypoxia. This could be happening due to medium evaporation and row 7 was set to NaN during the analysis because of this.

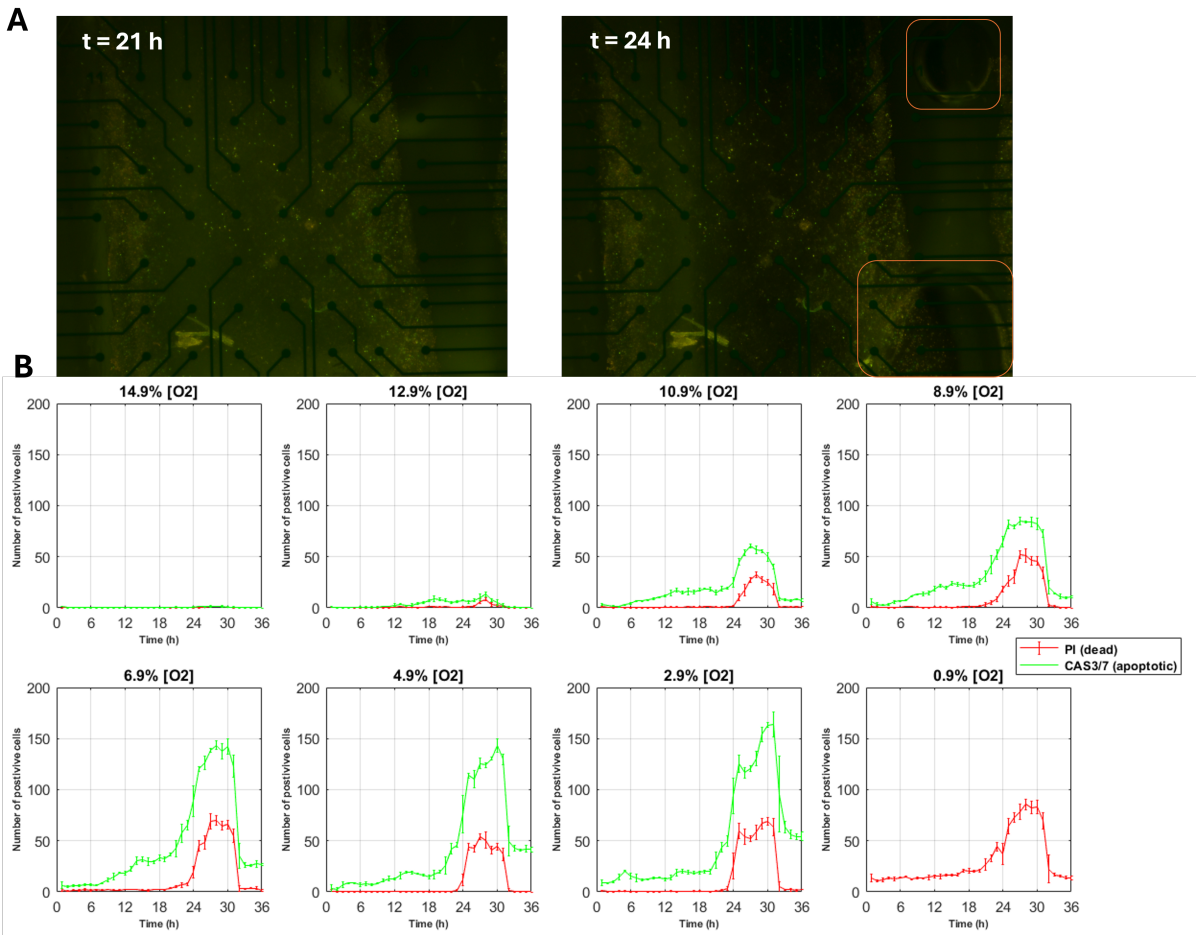


Fig. 10: An example of fluorescence images taken during measurement 4_O are depicted in (A). The orange rectangles show air bubbles in the medium, which causes these high counts after 21 hours. The graphs below that resemble the apoptotic (green) and dead (red) cell count at different oxygen concentrations (B).

EXPERIMENT 6 (Oskari)

The electrophysiological data is visible in Figure 13A and B, whereas the raw data is shown in Figure 13C. There were no fluorescence images taken, due to technical issues.

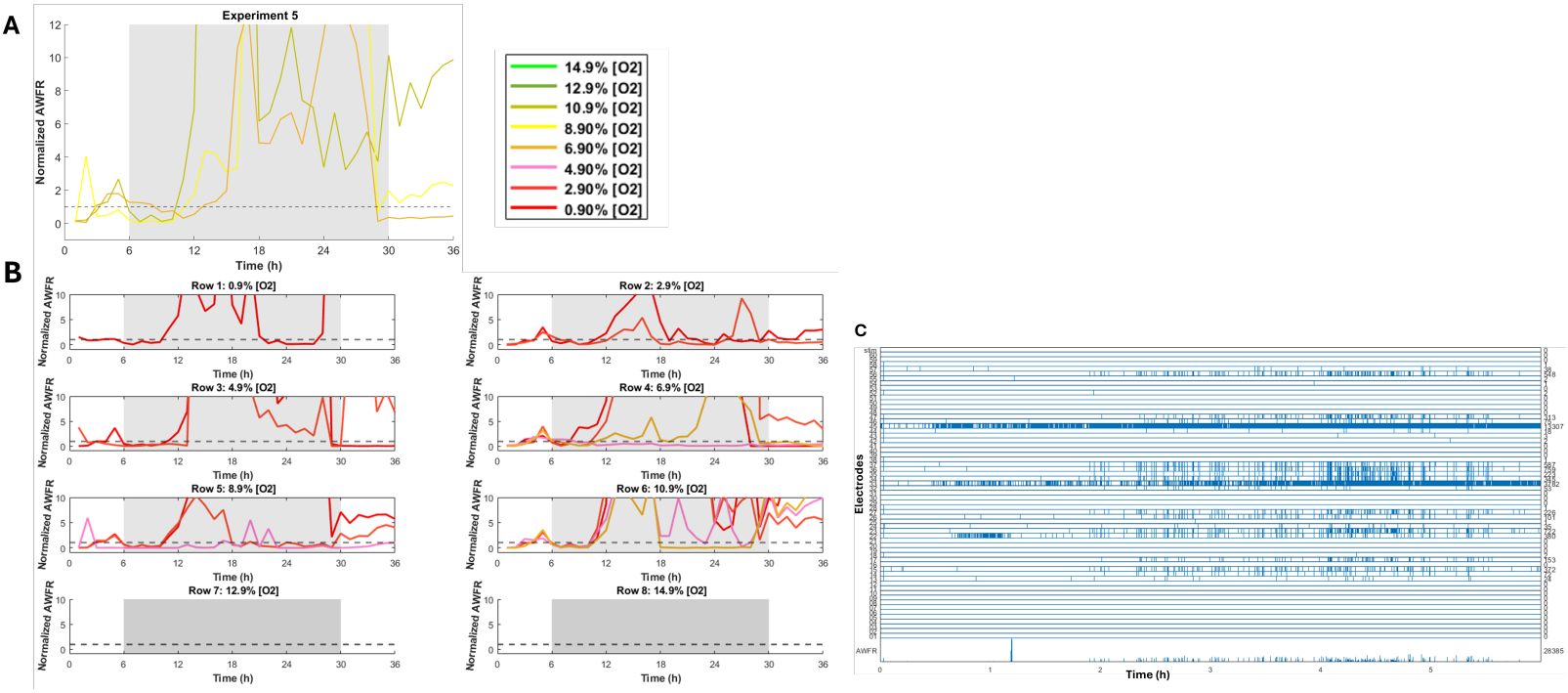


Fig. 11: (A) depicts the electrophysiological data of experiment 5, normalized over the first 6 hours of the measurement. Every line resembles a different oxygen concentrations from 14.9% to 0.90%. In (B) every electrode from a specific concentration is extracted and individually visualised in the same manner as the data from (A). Rasterplot (C) shows the activity of every electrode in the first 6 hours, where the neuronal cells are exposed to a normoxic environment.

EXPERIMENT 7 (Oskari)

The electrophysiological data is visible in Figure 14A and B, whereas the raw data is shown in Figure 14C. There were no fluorescence images taken, due to technical issues.

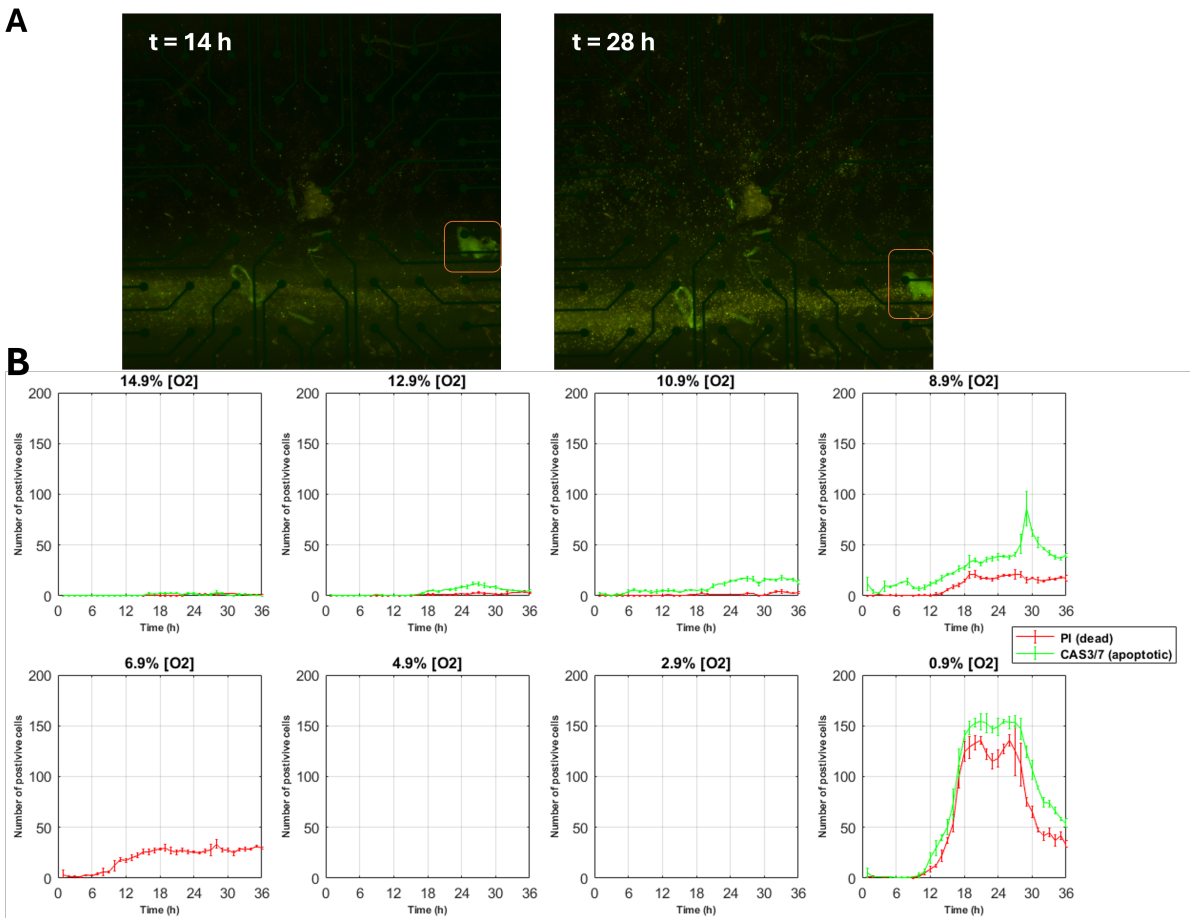


Fig. 12: An example of fluorescence images taken during measurement 5_O are depicted in (A). The orange rectangles show a dislocation in debris. The graphs below that resemble the apoptotic (green) and dead (red) cell count at different oxygen concentrations (B). Row 7 was set to NaN, because of compressed cells by the inserts.

EXPERIMENT 8 (Oskari)

The electrophysiological data is visible in Figure 15A and B, whereas the raw data is shown in Figure 15C. Between 12-20 hours there is a large increase in CWFR, due to moving debris.

The fluorescence data is visible in Figure 16. During the measurement there was a change in lighting and debris started to move around, which is visible in Figure 16A indicated by the orange rectangles. This also translated to the graphs after the apoptotic and dead cells have been counted (Figure 16B) between 11-19 hours. This could be happening due to medium evaporation.

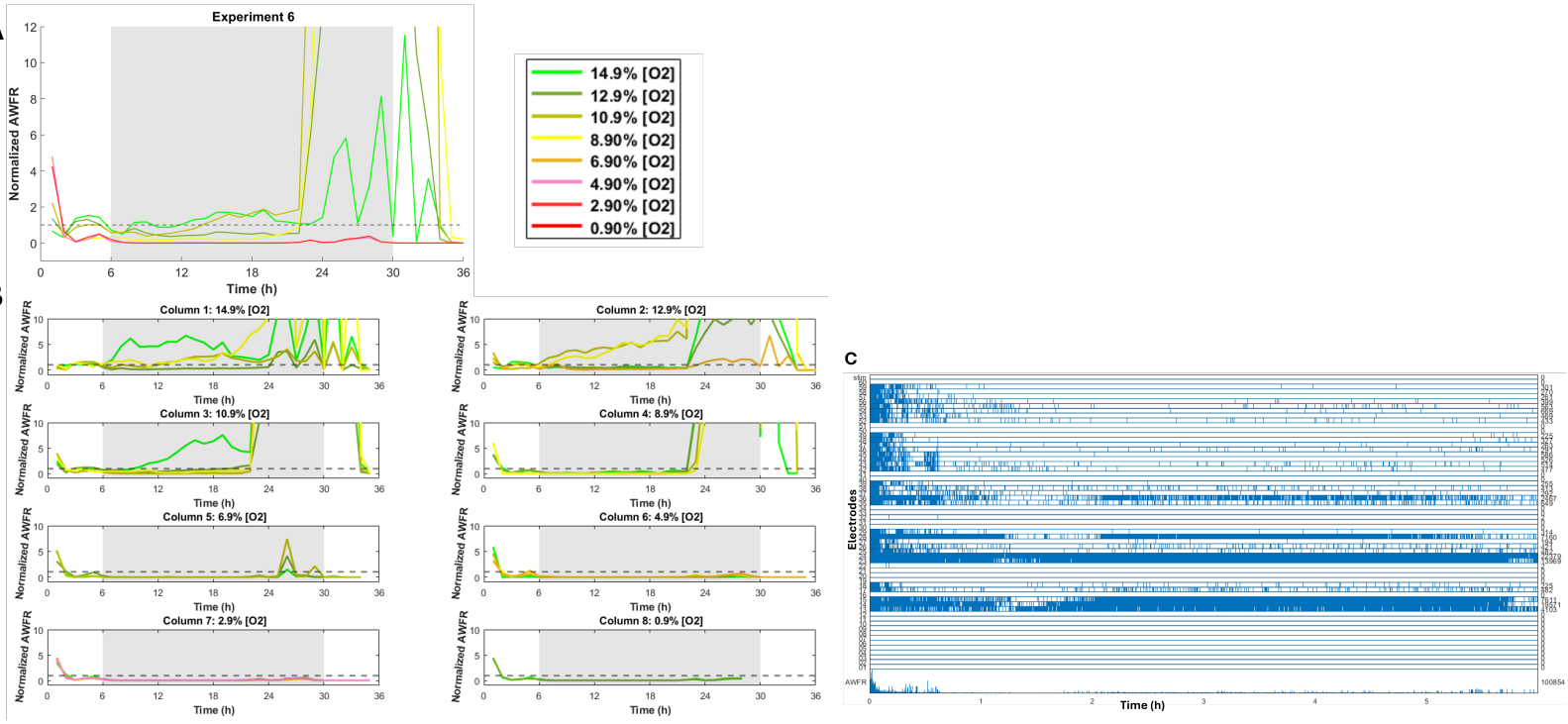


Fig. 13: (A) depicts the electrophysiological data of experiment 6, normalized over the first 6 hours of the measurement. Every line resembles a different oxygen concentrations from 14.9% to 0.90%. In (B) every electrode from a specific concentration is extracted and individually visualised in the same manner as the data from (A). Rasterplot (C) shows the activity of every electrode in the first 6 hours, where the neuronal cells are exposed to a normoxic environment.

EXPERIMENT 1 (Luna)

The electrophysiological data is visible in Figure 17A and B, whereas the raw data is shown in Figure 17C. After around 6 hours the data starts to increase. Data not used because of extreme artifacts.

The fluorescence data is visible in Figure 18. During the measurement there was a change in focus and light, which is visible in Figure 18A indicated by the orange rectangle. This also translated to the graphs after the apoptotic and dead cells have been counted (Figure 18B) at around 6 hours. This could be happening due to medium evaporation or artifacts, row 6 was set to NaN.

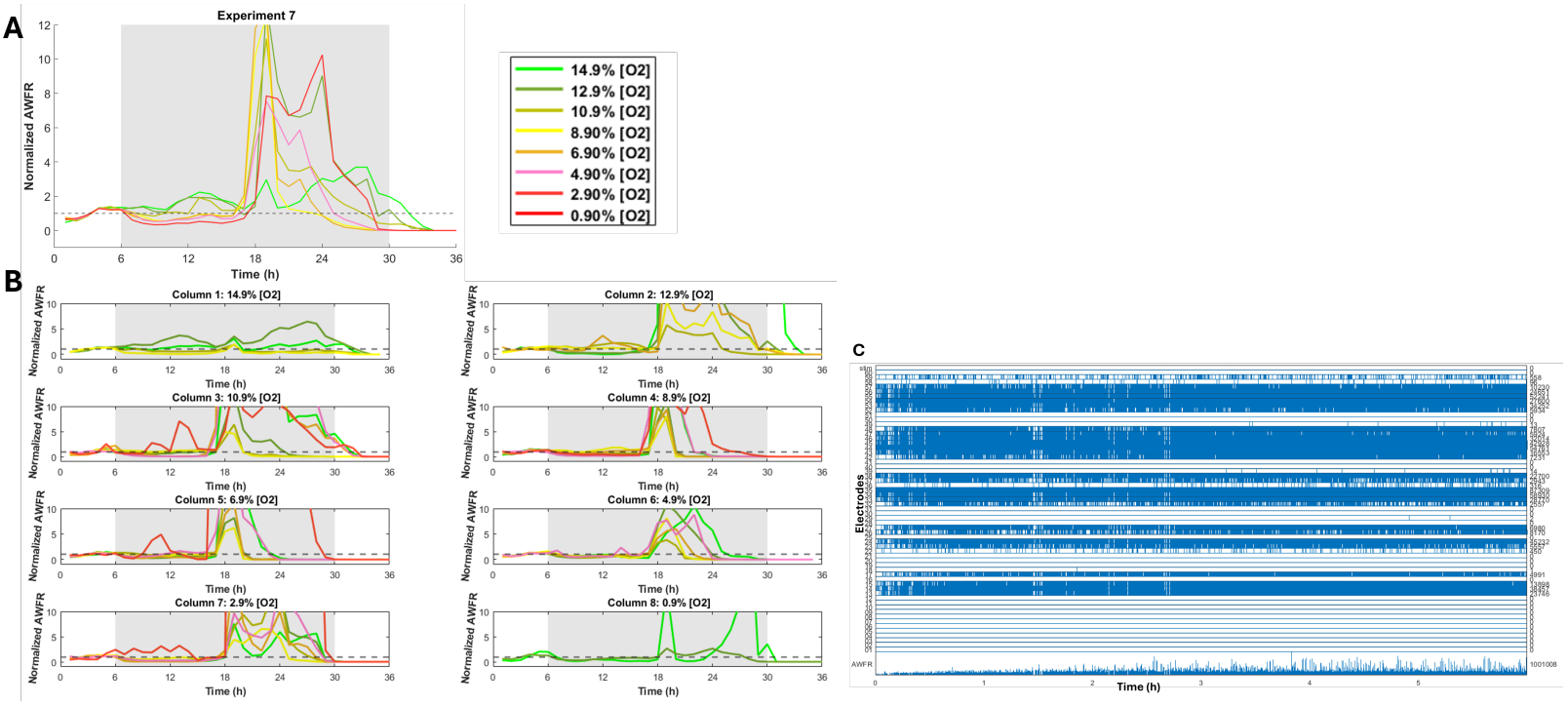


Fig. 14: (A) depicts the electrophysiological data of experiment 7, normalized over the first 6 hours of the measurement. Every line resembles a different oxygen concentrations from 14.9% to 0.90%. In (B) every electrode from a specific concentration is extracted and individually visualised in the same manner as the data from (A). Rasterplot (C) shows the activity of every electrode in the first 6 hours, where the neuronal cells are exposed to a normoxic environment.

EXPERIMENT 2 (Luna)

The electrophysiological data is visible in Figure 19A and B, whereas the raw data is shown in Figure 19C. After around 14 hours there is no more activity in all the columns, probably due to medium evaporation.

The fluorescence data is visible in Figure 20. During the measurement there was a lot of medium evaporation, because there was no medium left in the end, which is visible in Figure 20A. This does not really reflect in the data after analysis, because the cell count does not increase by much except in column 6.90 %(Figure 20B).

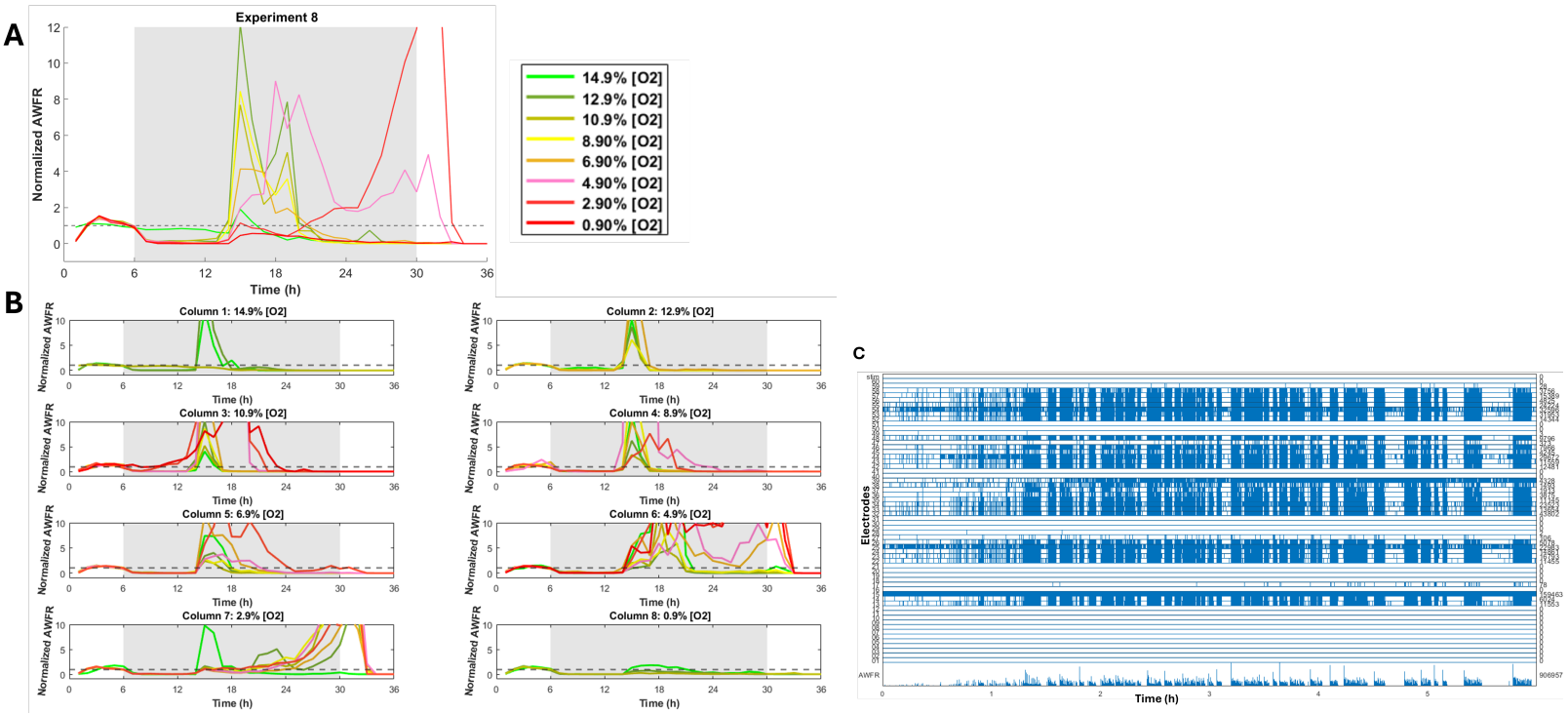


Fig. 15: (A) depicts the electrophysiological data of experiment 8, normalized over the first 6 hours of the measurement. Every line resembles a different oxygen concentrations from 14.9% to 0.90%. At 12 hours the data increases due to movin debris. In (B) every electrode from a specific concentration is extracted and individually visualised in the same manner as the data from (A). Rasterplot (C) shows the activity of every electrode in the first 6 hours, where the neuronal cells are exposed to a normoxic environment.

EXPERIMENT 3 (Luna)

The electrophysiological data is visible in Figure 21A and B, whereas the raw data is shown in Figure 21C. At 33 hours there is a huge decrease in activity, but it is unknown how that were to happen.

The fluorescence data is visible in Figure 22. During the measurement there was a lot of medium evaporation, but noting really changed during the measurement which is visible in Figure 22A and B. Maybe because of compressed cells due to the gradient inserts.

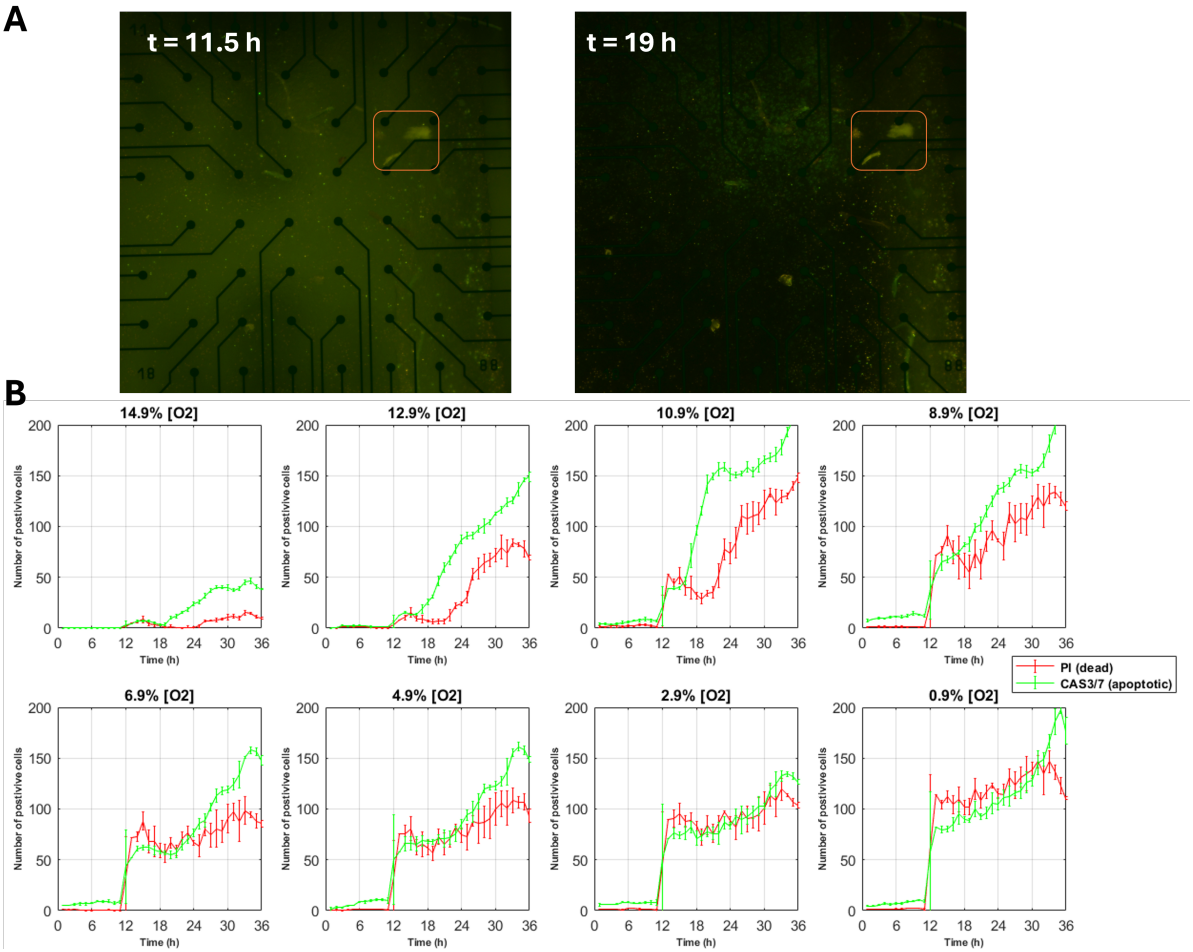


Fig. 16: An example of fluorescence images taken during measurement 8_O are depicted in (A). These orange rectangles depict the dislocation of debris in the cell culture, which translates to the graphs in (B). These graphs resemble the apoptotic (green) and dead (red) cell count at different oxygen concentrations.

EXPERIMENT 4 (Luna)

The electrophysiological data is visible in Figure 23A and B, whereas the raw data is shown in Figure 23C. There is only activity in the 2nd, 3rd and 4th column, due to a constant hypoxic environment the whole experiment.

The fluorescence data is visible in Figure 24. During the measurement the cells were subjected to only nitrogen for 18 hours, see Figure 24. In Figure 24B you can see a similar trend in amount of dead cells in columns 3, 4 and 5, but not in the rest. This is probably due to compressing cells by the gradient insert

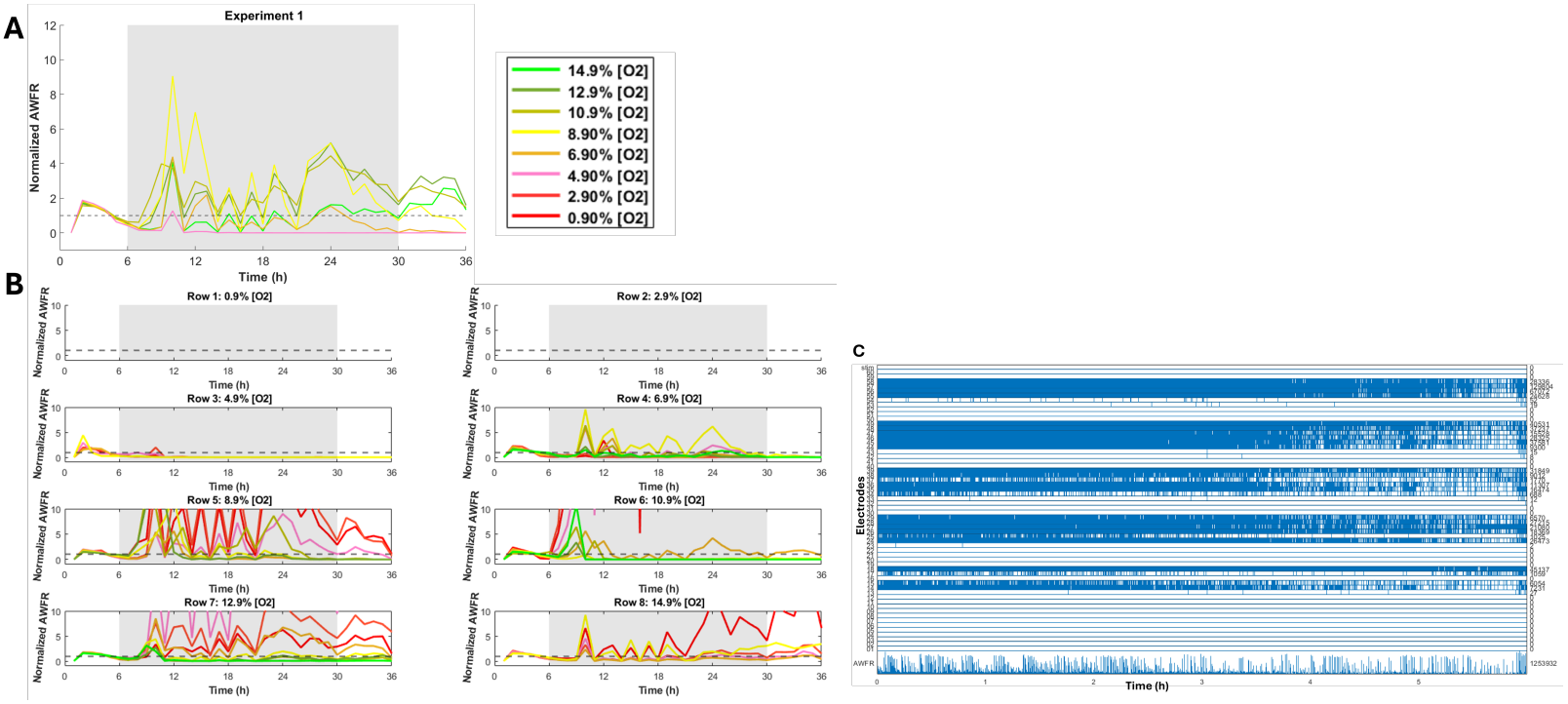


Fig. 17: (A) depicts the electrophysiological data of experiment 1, normalized over the first 6 hours of the measurement. Every line resembles a different oxygen concentrations from 14.9% to 0.90%. In (B) every electrode from a specific concentration is extracted and individually visualised in the same manner as the data from (A). Rasterplot (C) shows the activity of every electrode in the first 6 hours, where the neuronal cells are exposed to a normoxic environment. This experiment was not used in the analysis because of these extreme artifacts.

B. EXPERIMENTAL NOTES

These notes are only from the experiments that were done after those of Oskari Kulta. If you wish to see those notes go to his thesis [16].

EXPERIMENT 1: (*15-05-2024 / 21-05-2024*)
MEA_40409
Cell culture 18-04-2024
Gradient insert = Chip 1
Gradient direction = Row wise

TABLE III: Timestamps of the start and the end of experiment 1.

Start flow:	16:29 (15-05-2024)
Start recording:	16:38
Start timelapse:	16:33 (n = 19-162)
Stop everything:	09:08 (17-05-2024)

TABLE IV: The length during which cells are exposed to the three different phases of experiment 1.

Timeslot (h):	**Gasflow:**
10.30	Normoxia
24	Gradient
6	Normoxia

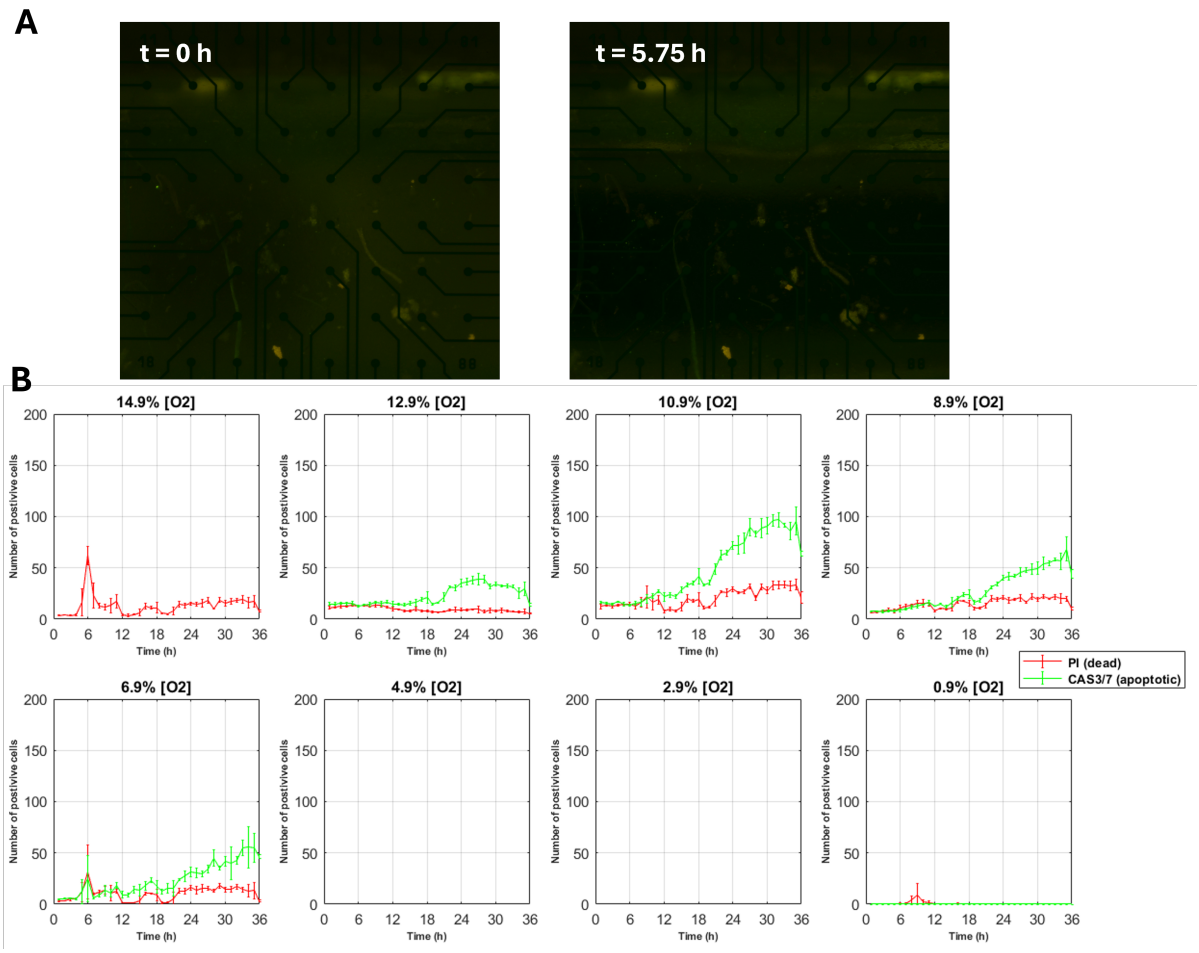


Fig. 18: An example of fluorescence images taken during measurement 1_L are depicted in (A). The graphs below that resemble the apoptotic (green) and dead (red) cell count at different oxygen concentrations (B). Row 6 was set to NaN, because of artifacts.

TABLE V: Characteristics of the MEA before experiment 1.

MEA_40409	
Activity:	Active, but less than initial check. Rows 1&2 not very active
Bursting:	Synchronous bursting
Grounded:	58

Notes:

- B1 = COM9 & B2 = COM6
- Lots of debris on cells
- Be careful putting the box over the setup, can cause things to move around and lose focus in image.
- These starting times are the new times, because the computer froze after 30 minutes into the experiment. Needed to be rebooted. Could be because the Matlab file was not open as main screen.

Checkup next day: *16-05-2024*
Everything still works. Electrode 86 maybe inactive.

End 36h experiment: *17-05-2024*
Refreshed 0.5 ml medium to culture.

Checkup 3 days later: *21-05-2024*
Gradient device has touched the cells, resulting in compressed and detached cells. Also electrode 77 was grounded.

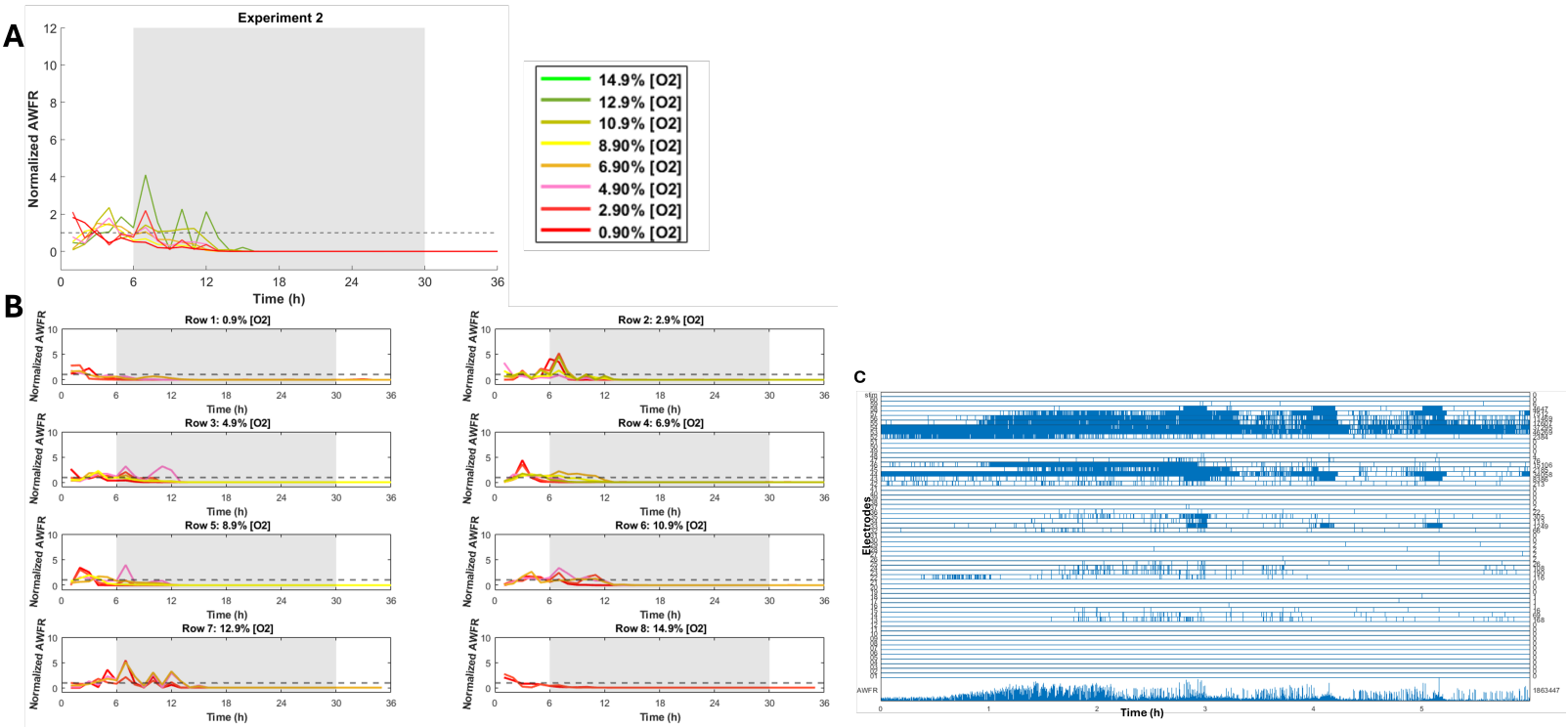


Fig. 19: (A) depicts the electrophysiological data of experiment 2, normalized over the first 6 hours of the measurement. Every line resembles a different oxygen concentrations from 14.9% to 0.90%. In (B) every electrode from a specific concentration is extracted and individually visualised in the same manner as the data from (A). Rasterplot (C) shows the activity of every electrode in the first 6 hours, where the neuronal cells are exposed to a normoxic environment.

TABLE VI: Timestamps of the start and the end of the checkup experiment 1.

Start flow:	14:09
Start recording:	14:13
Start timelapse:	14:11 (n = 5)
Stop everything:	15:17

EXPERIMENT 2 MEA_40417
Cell culture 18-04-2024
Gradient insert = Chip 2
Gradient direction = Row wise

TABLE VII: Timestamps of the start and the end of experiment 2.

Start flow:	15:17 (22-05-2024)
Start recording:	15:24
Start timelapse:	16:15 (n = 82)
Stop everything:	15:37 (14-05-2024)

Notes:

- Gradient device was placed at a slight angle w.r.t. the MEA
- Used red lidlock instead of the black one

Checkup next day: *23-05-2024*
Switch to nitrogen was not made, therefore no gradient was applied. We stopped the flow of B2 and checked if the switch worked. It did, so we put programmed LabView to immediately go to nitrogen. Because of this the experiment lasted 8 hours longer. B2 started at 9:37.

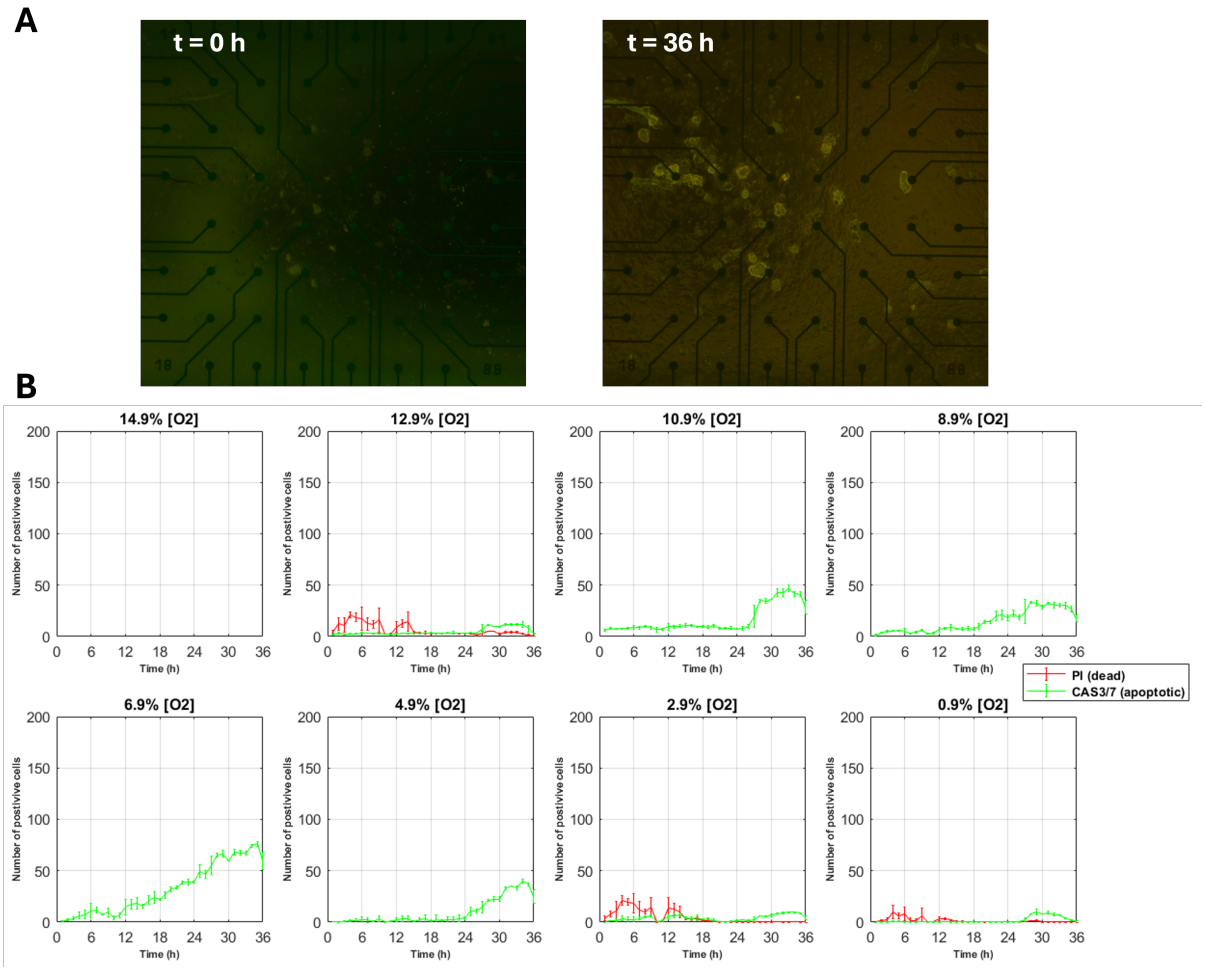


Fig. 20: An example of fluorescence images taken during measurement 2_L are depicted in (A). The graphs below that resemble the apoptotic (green) and dead (red) cell count at different oxygen concentrations (B). At the end of the experiment everything was evaporated.

TABLE VIII: The length during which cells are exposed to the three different phases of experiment 2.

Timeslot (h):	Gasflow:
11.45	Normoxia
24	Gradient
6	Normoxia

End 36h experiment: *24-05-2024*
All medium evaporated!

Checkup 3 days later: *27-05-2024*
Did not do it, because the cells were dead.

EXPERIMENT 3 MEA_34266
Cell culture 16-05-2024
Gradient insert = Chip 4
Gradient direction = Column wise

Notes:

- Weight MEA before $\approx$ 2.892 grams and after $\approx$ 1.856 grams. So almost 1 ml of medium evaporated in a time slot of

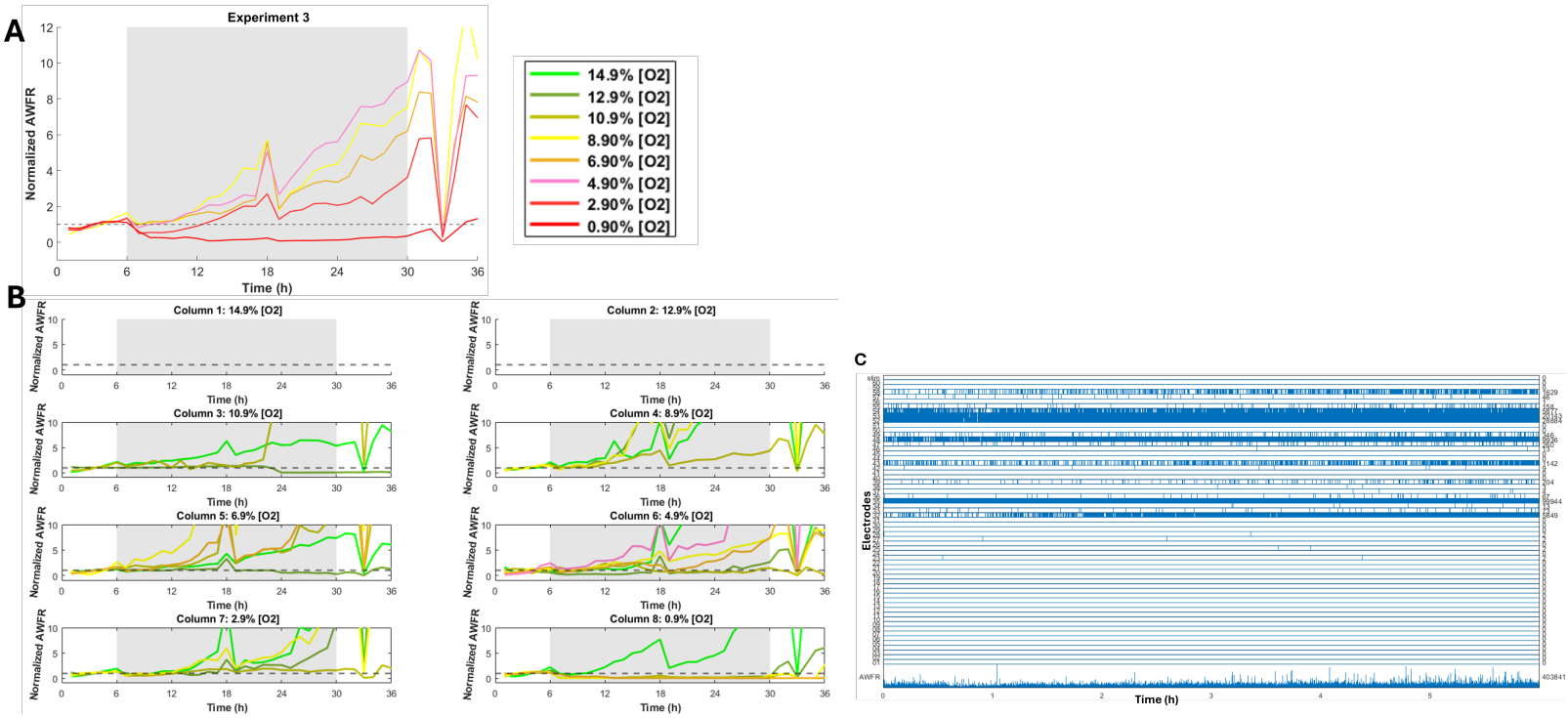


Fig. 21: (A) depicts the electrophysiological data of experiment 3, normalized over the first 6 hours of the measurement. Every line resembles a different oxygen concentrations from 14.9% to 0.90%. In (B) every electrode from a specific concentration is extracted and individually visualised in the same manner as the data from (A). Rasterplot (C) shows the activity of every electrode in the first 6 hours, where the neuronal cells are exposed to a normoxic environment.

TABLE IX: Characteristics of the MEA before experiment 2.

MEA_40417	
Activity:	Row 1-4 were not very active.
Bursting:	Synchronous bursting
Grounded:	-

roughly 42 h.

- The tubing of the flows was connected in the flow hood before putting it in the MEA recording setup to try and avoid pushing the cells to hard. The tubes were first inserted in the generator and then placed in the culture well. In the other 2 experiments we inserted the flow tubes after placing the generator on the cells, fastening it with a lidlock and placing it in the setup .

Checkup next day: *04-06-2024*
Everything still works

End 36h experiment: *05-06-2024*
Looks like medium was gone, but the part where the cells were still held some medium due to the gradient inserts. Refreshed cells with 1 ml medium

Checkup 2 days later: *07-06-2024*
Under the microscope and by eye you can see that the inserts left a mark on the cells, which means that the cells in those parts got squished and died probably.

EXPERIMENT 4 MEA_37049
Cell culture 16-05-2024
Gradient insert = Chip 4
Gradient direction = Column wise

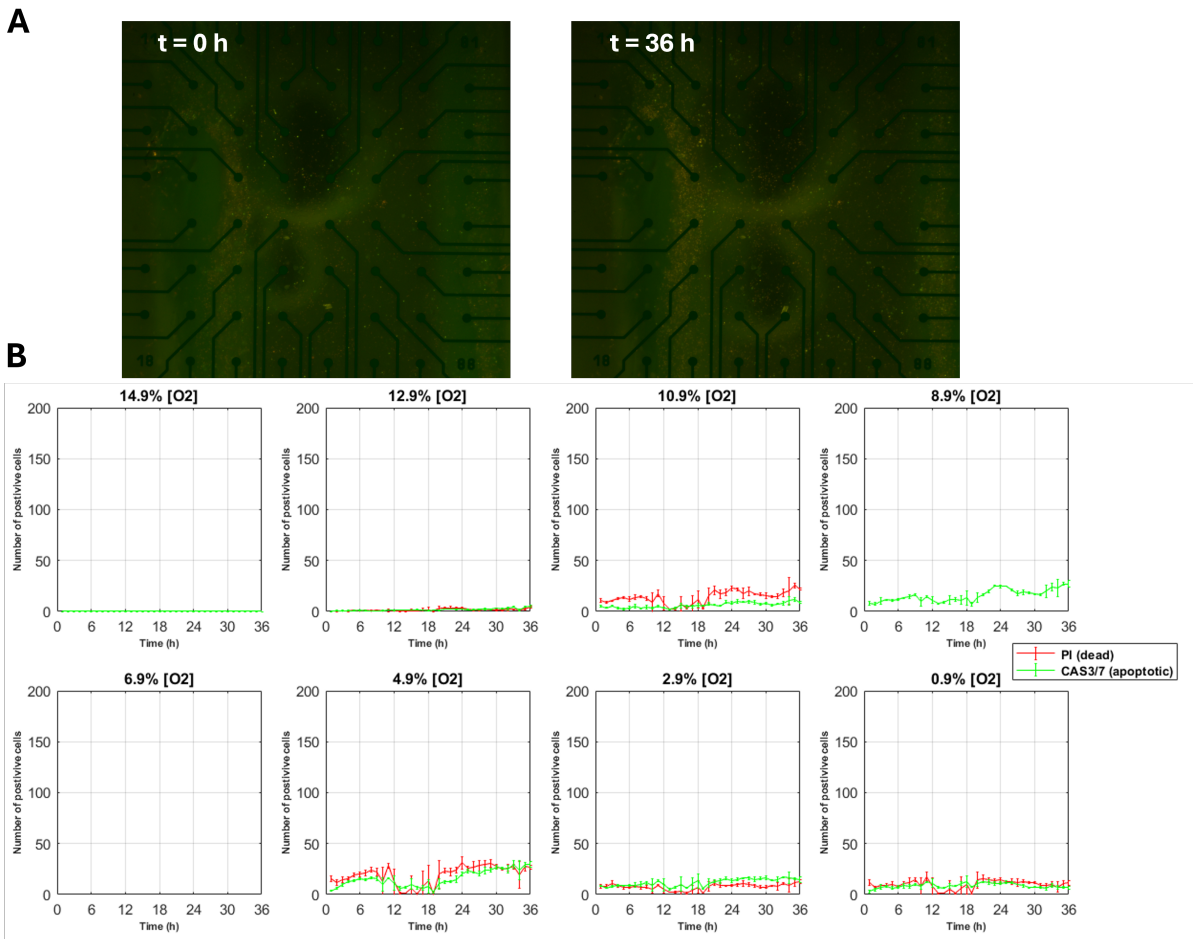


Fig. 22: An example of fluorescence images taken during measurement 3_L are depicted in (A). The graphs below that resemble the apoptotic (green) and dead (red) cell count at different oxygen concentrations (B). There is almost no change between these concentrations.

TABLE X: The length during which cells are exposed to the three different phases of the experiment 2.

Timeslot (h):	Gasflow:
1 min	Normoxia
24	Gradient
6	Normoxia

Notes:

- Weight before $\approx$ 1.931 grams

Checkup next day: *06-06-2024*

Stopped measurement, because the gasflows were not set properly. Through both inserts only nitrogen flowed through.

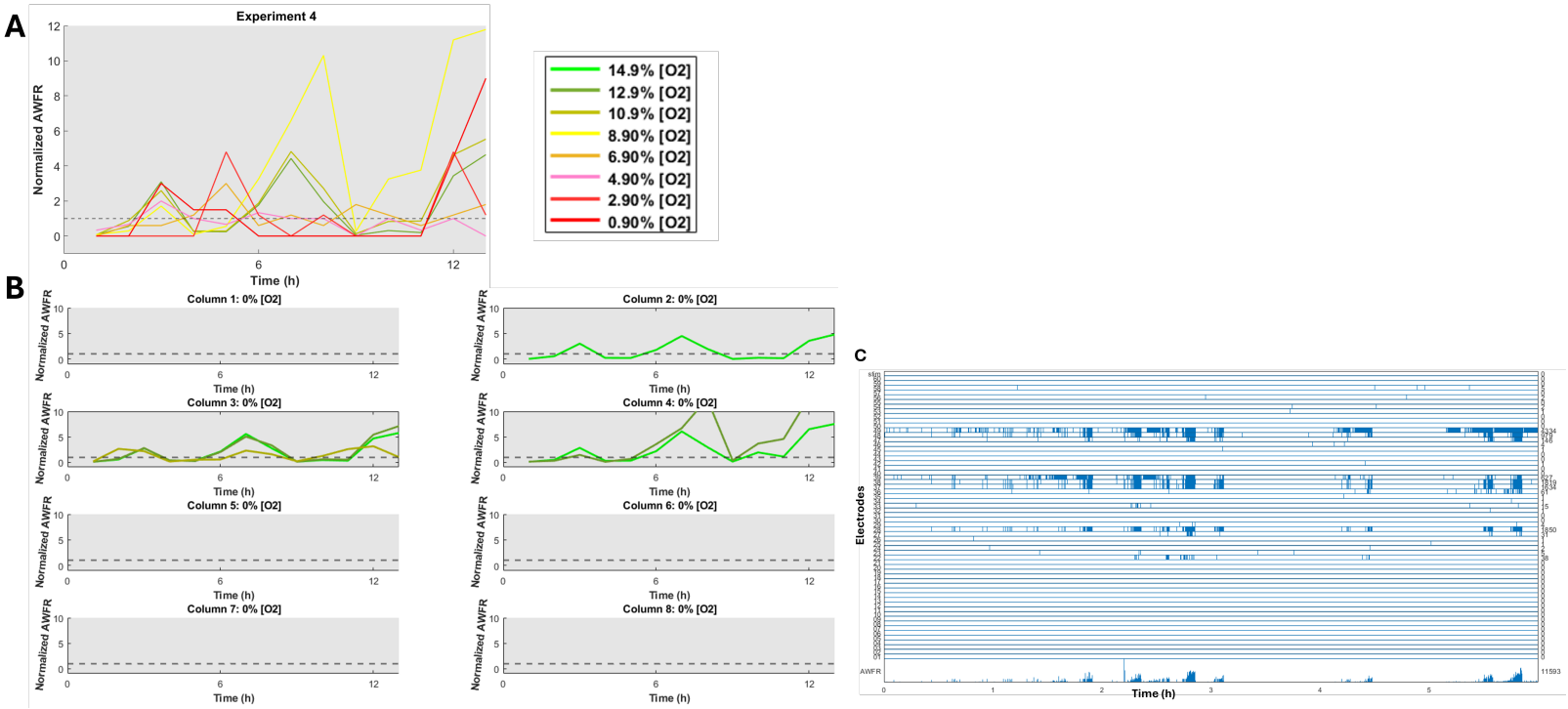


Fig. 23: (A) depicts the electrophysiological data of experiment 4, normalized over the first 6 hours of the measurement. The MEA was exposed to 0.90% oxygen throughout the whole experiment. In (B) every electrode from a specific column is extracted and individually visualised in the same manner as the data from (A). Rasterplot (C) shows the activity of every electrode in the first 6 hours.

TABLE XI: Timestamps of the start and the end of experiment 3.

Start flow:	15:45 (03-06-2024)
Start recording:	16:06
Start timelapse:	16:05 (n = 48-191)
Stop everything:	09:54 (05-06-2024)

C. MEA SCRIPTS

SINGULAR EXPERIMENT: PLOT CWFR PER GRADIENT

```
%% Info script
% Plots spikes normalized over first six hours per column in one plot.
% Can be done for both column or row wise gradient data.
%% Load MEA data colomn wise gradient
clear; clc; close all
[filename, path] = uigetfile('C:\Users\lunah\OneDrive\Documenten\MATLAB\Matlab Utwente\Bachelor CNPH\
    Artifects removed\', 'multiselect','on');
C = []; T = [];

% A single file, or a list of files
if class(filename) == 'char'
    files = cell(1, 1);
    files{1} = [char(path) char(filename)];
else
    files = cell(length(filename), 1);
    for i=1:length(filename)
       files{i} = [char(path) char(filename(i))];
    end
end

% Read every file and combine
load(files{1}); T = [T; data.Ts]; C = [C; data.Cs];
for i = 2:size(files, 1)
    fprintf('loading...\n')
```

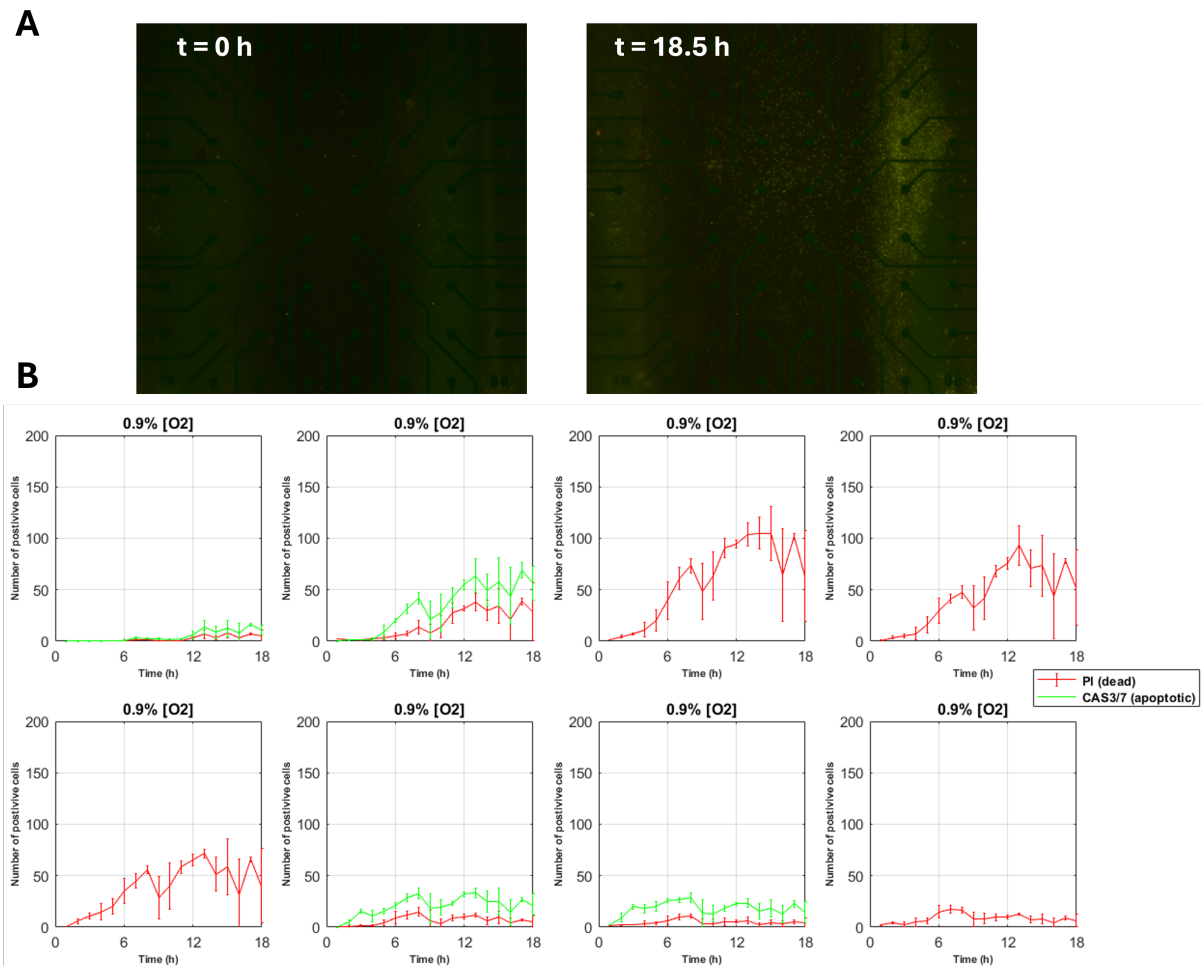


Fig. 24: An example of fluorescence images taken during measurement 4_L are depicted in (A). The graphs below that resemble the apoptotic (green) and dead (red) cell count during the most severe hypoxia (B).

TABLE XII: The length during which cells are exposed to the three different phases of experiment 3.

Timeslot (h):	Gasflow:
11.15	Normoxia
24	Gradient
6	Normoxia

```
    load(files{i});
    T = [T; data.Ts];
    C = [C; data.Cs];
end
% Variables
Timebin = 1;                                    % In hours
T = T/(25000*3600);                             % convert timestamps to hours (sample freq = 25kHz)
%% Analyze per column
clear time
First6Hours = 6;
for n = 1:8
    selectionIndex_C = find(C > (n)*10 & C < ((n+1)*10) - 1);                                    % C between
        11 and 18 for n = 1
    Tselection_C = T(selectionIndex_C);                                                          % Contains
        only timestamps in column n
    for m = 1:max(Tselection_C)/Timebin
        NumberSpikes_colomn(m,n) = (length (find (Tselection_C > (m-1)*Timebin & Tselection_C < m*Timebin
            )));
        time(m) = m*Timebin;
```

TABLE XIII: Characteristics of the MEA before experiment 3.

MEA_34266	
Activity:	Low activity but it came back.
Bursting:	Synchronous bursting
Grounded:	43 and 58

TABLE XIV: Timestamps of the start and the end of the checkup of experiment 3.

Start flow:	9:18
Start recording:	9:21
Start timelapse:	9:33 (n = 18)
Stop everything:	10:21

```
        colomn_time = time';                                                    % Transpose
            of time from row to colomn vector
    end
    % Check if at least 50% of the electrodes per column are active (1 spike/3 min)
    if length(NumberSpikes_colomn) >= First6Hours
        sixhourspikes = sum(NumberSpikes_colomn(1:First6Hours, n));
        if n == 1 && sixhourspikes < 240                                % 2/5 electrodes active
            NumberSpikes_colomn(:,n) = NaN;
        elseif n >= 2 && n <= 4 && sixhourspikes < 480                  % 4/8 electrodes active
            NumberSpikes_colomn(:,n) = NaN;
        elseif n == 8 && sixhourspikes < 360                            % 3/6 electrodes active
            NumberSpikes_colomn(:,n) = NaN;
        end
    else
        disp(['Not enough data for the first 6 hours for column ' num2str(n)]);
        % Skip to the next electrode if there no data for first 6 hours
        NumberSpikes_colomn(:,n) = NaN;
    end
end
% Check if NumberSpikes_colomn is 36 rows
if size(NumberSpikes_colomn, 1) < 36
    rowsToAdd = 36 - size(NumberSpikes_colomn, 1);
    NaNcol = NaN(rowsToAdd, size(NumberSpikes_colomn, 2));
    NumberSpikes_colomn = [NumberSpikes_colomn; NaNcol];
elseif size(NumberSpikes_colomn, 1) > 36
    NumberSpikes_colomn = NumberSpikes_colomn(1:36, :);
end
% Check if time is the same length as NumberSpikes_colomn
if size(colomn_time, 1) < size(NumberSpikes_colomn, 1)
    add = size(NumberSpikes_colomn, 1) - size(colomn_time, 1);
    nancol = NaN(add,size(colomn_time, 2));
    colomn_time = [colomn_time; nancol];
elseif size(colomn_time, 1) > size(NumberSpikes_colomn, 1)
    colomn_time = colomn_time(1:36, :);
end

% save non normalized data
raw_numberspikes_col = [];
raw_numberspikes_col = [raw_numberspikes_col, NumberSpikes_colomn];
% Normalizing with first 6 hours
NumberSpikes_two_C = [];
err = [];
avg_baseline_C_values = [];
for n = 1:8
    sixhours_baseline_C = 0;
    for m = 1:6                                                             % The first
        six hours
        spikesperhour_C = NumberSpikes_colomn(m,n);                         % First row
            an then column
        sixhours_baseline_C = sixhours_baseline_C + spikesperhour_C;        % Summing
            the six hours
    end
    avg_baseline_C = sixhours_baseline_C/6;                                 % Average
        of first six hours
    avg_baseline_C_values = [avg_baseline_C_values, avg_baseline_C];
```

TABLE XV: Timestamps of the start and the end of experiment 4.

Start flow:	14:31 (05-06-2024)
Start recording:	14:30
Start timelapse:	15:16 (n = 43)
Stop everything:	09:12 (06-06-2024)

TABLE XVI: The length during which cells are exposed to the three different phases of experiment 4.

Timeslot (h):	**Gasflow:**
12.30	Normoxia
24	Gradient
6	Normoxia

```
    NumberSpikes_two_C = [NumberSpikes_two_C,  NumberSpikes_colomn(:,n)/avg_baseline_C];      % Devides
        every spike by the average and add a new column to matrix
    err = [err, std(NumberSpikes_two_C, 0, 2)];                                               % Standard
        deviation per column
end
NumberSpikes_colomn = NumberSpikes_two_C;
figure;
% Grey patch from 6 to 30 hours
hold on
v = [6 -1; 30 -1; 30 12; 6 12];
% v = [0 -1; 14 -1; 14 12; 0 12]; % exp 4 luna
f = [1 2 3 4];
p = patch('Faces',f,'Vertices',v,'FaceAlpha', 0.1, 'EdgeColor','none','LineWidth',0.5);
p.Annotation.LegendInformation.IconDisplayStyle = 'off';
% plot spikes per column
plot(colomn_time, NumberSpikes_colomn, 'LineWidth', 2);
set(gca, 'FontSize', 20)
xticks(0:6:36)
xlabel('Time (h)','fontweight','bold')
ylabel('Normalized CWFR','fontweight','bold')
title('\bf Experiment 8')
hold off
% Colors of the lines from green to red
newcolors_C = [0 1 0;
             0.4660, 0.6740, 0.1880;
             0.75, 0.75, 0;
             1 1 0;
             0.9290 0.6940 0.1250;
             1 0.5 0.8;
             1 0.2 0.1840;
             1 0 0];
colororder(newcolors_C)
xlim([0 36])
% xlim([0 13]) % exp 4 luna
ylim([-1 12])
yline(1,'--', 'LineWidth',1.5)
lgd = legend('\bf 14.9% [O2]','\bf 12.9% [O2]','\bf 10.9% [O2]','\bf 8.90% [O2]','\bf 6.90% [O2]','\bf
    4.90% [O2]','\bf 2.90% [O2]','\bf 0.90% [O2]', 'Location','northeastoutside');
% lgd = legend('\bf 0.90% [O2]','\bf 0.90% [O2]','\bf 0.90% [O2]','\bf 0.90% [O2]','\bf 0.90% [O2]','\bf
    0.90% [O2]','\bf 0.90% [O2]','\bf 0.90% [O2]', 'Location','northeastoutside'); % exp 4 luna
lgd.FontSize = 10;
save("MEA_data_after_analysis\Gradient\exp8_O_plot.mat",'raw_numberspikes_col', 'NumberSpikes_colomn','
    avg_baseline_C_values', '-mat');
%% Load MEA data row wise gradient
[filename, path] = uigetfile('C:\Users\lunah\OneDrive\Documenten\MATLAB\Matlab Utwente\Bachelor CNPH\
    Artifects removed\', 'multiselect','on');
C = []; T = [];
% A single file, or a list of files
if class(filename) == 'char'
    files = cell(1, 1);
    files{1} = [char(path) char(filename)];
else
    files = cell(length(filename), 1);
    for i=1:length(filename)
```

TABLE XVII: Characteristics of the MEA before experiment 4.

MEA_37049	
Activity:	Active.
Bursting:	Synchronous bursting
Grounded:	17, 24, 58, 75, 78, 84 and 85

```
       files{i} = [char(path) char(filename(i))];
    end
end

% Read every file and combine
load(files{1}); T = [T; data.Ts]; C = [C; data.Cs];
for i = 2:size(files, 1)
    fprintf('loading...\n')
    load(files{i});
    T = [T; data.Ts];
    C = [C; data.Cs];
end
% Variables
Timebin = 1;                                    % In hours
T = T/(25000*3600);                             % convert timestamps to hours (sample freq = 25kHz)
%% Analyse per row
clear time
First6Hours = 6;
for n = 1:8
    selectindex_R = find(C == (n)+10 | C==(n)+20 | C==(n)+30 | C==(n)+40 | C==(n)+50 | C==(n)+60 | C==(n)
        +70 | C==(n)+80);  % for n=1 takes 11 21 31...
    Tselection_R = T(selectindex_R);
                                                                                                % The time
     that corresponds to the index; contains only timestamps in a row n
    for m = 1:max(Tselection_R)/Timebin
        NumberSpikes_row(m,n) = (length (find (Tselection_R > (m-1)*Timebin & Tselection_R < m*Timebin)))
            ;
        time(m) = m * Timebin;
        Time = time';
    end
    % Check if at least 50% of the electrodes per row are active (1 spike/3 min)
    if length(NumberSpikes_row) >= First6Hours
        sixhourspikes_r = sum(NumberSpikes_row(1:First6Hours, n));
        if n == 1 && n == 5 && n == 8 && sixhourspikes_r < 360                              % 3/6
            electrodes active
            NumberSpikes_row(:,n) = NaN;
        elseif n >= 2 && n <= 4 && n == 6 && n== 7 && sixhourspikes_r < 480                   % 4/8
            electrodes active
            NumberSpikes_row(:,n) = NaN;
        end
    else
        disp(['Not enough data for the first 6 hours for row ' num2str(n)]);
        % Skip to the next electrode if there no data for first 6 hours
        NumberSpikes_row(:,n) = NaN;
    end
end
% Check if NumberSpikes_row has 36 rows
if size(NumberSpikes_row, 1) < 36
    rowsToAdd = 36 - size(NumberSpikes_row, 1);
    NaNRows = NaN(rowsToAdd, size(NumberSpikes_row, 2));
    NumberSpikes_row = [NumberSpikes_row; NaNRows];
elseif size(NumberSpikes_row, 1) > 36
    NumberSpikes_row = NumberSpikes_row(1:36, :);
end
% Check if time is the same length as NumberSpikes_colomn
if size(Time, 1) < size(NumberSpikes_row, 1)
    add = size(NumberSpikes_row, 1) - size(Time, 1);
    nanrow = NaN(add,size(Time, 2));
    Time = [Time; nanrow];
elseif size(Time, 1) > size(NumberSpikes_row, 1)
    Time = Time(1:36, :);
end
NumberSpikes_row = flip(NumberSpikes_row, 2);                           % same concentration per column
     in matrix as column data
```

```
NumberSpikes_row(:, 1) = NaN;
% save non normlized data
raw_numberspikes_row = [];
raw_numberspikes_row = [raw_numberspikes_row, NumberSpikes_row];
% Normalizing with first 6 hours
NumberSpikes_two_R = [];
avg_baseline_R_values = [];
for n = 1:8
    sixhours_baseline_R = 0;
    for m = 1:6                                                                    % The first six
        hours
        spikesperhour_R = NumberSpikes_row(m,n);                                   % First row an
            then column
        sixhours_baseline_R = sixhours_baseline_R + spikesperhour_R;               % Summing the
            six hours
    end
    avg_baseline_R = sixhours_baseline_R/6;                                        % Average of
        first six hours
    avg_baseline_R_values = [avg_baseline_R_values, avg_baseline_R];

    NumberSpikes_two_R = [NumberSpikes_two_R,  NumberSpikes_row(:,n)/avg_baseline_R];    % Devides every
        spike by the average and add a new column
end
NumberSpikes_row = NumberSpikes_two_R;
figure;
% Grey patch from 6 to 30 hours
hold on
v = [6 -1; 30 -1; 30 12; 6 12];
f = [1 2 3 4];
p = patch('Faces',f,'Vertices',v,'FaceAlpha', 0.1, 'EdgeColor','none','LineWidth',0.5);
p.Annotation.LegendInformation.IconDisplayStyle = 'off';
% plot spikes per row
plot(Time, NumberSpikes_row, 'LineWidth', 2)
set(gca, 'FontSize', 20)
xticks(0:6:36)
xlabel('Time (h)','fontweight','bold')
ylabel('Normalized CWFR','fontweight','bold')
title('\bf Experiment 2')
hold off
% Colors of the lines from green to red
newcolors_C = [0 1 0;
               0.4660, 0.6740, 0.1880;
               0.75, 0.75, 0;
               1 1 0;
               0.9290 0.6940 0.1250;
               1 0.5 0.8;
               1 0.2 0.1840;
               1 0 0];
colororder(newcolors_C)
xlim([0 36])
ylim([-1 12])
yline(1,'--', 'LineWidth',1.5)
lgd = legend('\bf 14.9% [O2]','\bf 12.9% [O2]','\bf 10.9% [O2]','\bf 8.90% [O2]','\bf 6.90% [O2]','\bf
    4.90% [O2]','\bf 2.90% [O2]','\bf 0.90% [O2]', 'Location','northeastoutside');
lgd.FontSize = 10;
save("MEA_data_after_analysis\Gradient\exp2_L_plot.mat", 'raw_numberspikes_row', 'NumberSpikes_row','
    avg_baseline_R_values', '-mat');
```

ALL EXPERIMENTS: PLOT CWFR PER GRADIENT

```
% Initiate empty matrices
Spikes_row = [];
% A single file, or a list of files
if ischar(filename)
     files = cell(1, 1);
     files{1} = fullfile(path, filename);
else
     files = cell(length(filename), 1);
     for i = 1:length(filename)
        files{i} = fullfile(path, filename{i});
     end
end
```

```
% Read every file and combine
load(files{1});
Spikes_row = raw_numberspikes_row;
for i = 2:size(files, 1)
    fprintf('loading...\n')
    load(files{i});
    Spikes_row = [Spikes_row; raw_numberspikes_row];
end
%% Load MEA data column wise gradient
[filename, path] = uigetfile('C:\Users\lunah\OneDrive\Documenten\MATLAB\Matlab
   Utwente\Bachelor CNPH\MEA_scripts\MEA_data_after_analysis\Gradient\Column\', '
   multiselect','on');
% Initiate empty matrices
Spikes_col = [];
% A single file, or a list of files
if ischar(filename)
    files = cell(1, 1);
    files{1} = fullfile(path, filename);
else
    files = cell(length(filename), 1);
    for i = 1:length(filename)
       files{i} = fullfile(path, filename{i});
    end
end
% Read every file and combine
load(files{1});
Spikes_col = raw_numberspikes_col;
for i = 2:size(files, 1)
    fprintf('loading...\n')
    load(files{i});
    Spikes_col = [Spikes_col; raw_numberspikes_col];
end
time = [1:36]';
%% Put column and row wise data together
AllNumberSpikes = [];
err = [];
AllNumberSpikes = vertcat(Spikes_col, Spikes_row);
%% Plots all experiments together (row & column data)
NumberSpikes = zeros(36, 8);
% Sum rows to form the new 36x8 matrix
for column = 1:8
    for row = 1:36
        total = 0;
        for exp = 1 : 7
            value = AllNumberSpikes(row + (36 * (exp- 1)), column);
            % Replace NaN with zero before summing
            if isnan(value)
                value = 0;
            end
            total = total + value;
        end
        NumberSpikes(row, column) = total;
    end
end
% Normalizing with first 6 hours
NumberSpikes_two = [];
err = [];
```

```
avg_baseline_values= [];
for n = 1:8
    sixhours_baseline = 0;
    for m = 1:6                                                              %
       The first six hours
        spikesperhour = NumberSpikes(m,n);                                   %
           First takes the row an then the column
        sixhours_baseline = sixhours_baseline + spikesperhour;               %
           Summing the first six hours
    end
    avg_baseline= sixhours_baseline/6;                                       %
       Average of first six hours
    avg_baseline_values = [avg_baseline_values, avg_baseline];

    NumberSpikes_two = [NumberSpikes_two,  NumberSpikes(:,n)/avg_baseline];  %
       Devides every spike by the average and adds a new column to matrix
    err = [err, std(NumberSpikes_two, 0, 2)];                                %
       Standard deviation per [O2]
end
NumberSpikes = NumberSpikes_two;
% Plot data in one figure
figure;
% Grey patch from 6 to 30 hours
hold on
v = [6 -1; 30 -1; 30 10; 6 10];
f = [1 2 3 4];
p = patch('Faces',f,'Vertices',v,'FaceAlpha', 0.1, 'EdgeColor','none','LineWidth'
   ,0.5);
p.Annotation.LegendInformation.IconDisplayStyle = 'off';
plot(time, NumberSpikes, 'LineWidth', 2)
% Colors of the lines from green to red
newcolors_column = [0 1 0;
                    0.4660, 0.6740, 0.1880;
                    0.75, 0.75, 0;
                    1 1 0;
                    0.9290 0.6940 0.1250;
                    1 0.5 0.8;
                    1 0.3 0.1840;
                    1 0 0];
colororder(newcolors_column)
set(gca, 'FontSize', 20)
% for k = 1:8
%     e = errorbar(time, NumberSpikes(:, k), err(:, k), "MarkerSize", 20, 'Color',
   newcolors_column(k, :), "CapSize", 10);
%     e.LineWidth = 2;
% end
xticks(0:6:36)
xlabel('Time [h]','fontweight','bold')
ylabel('Normalized CWFR','fontweight','bold')
hold off
xlim([0 36])
ylim([-1 10])
yline(1,'--', 'LineWidth',1.5)
lgd = legend('\bf 14.9% [O2] (n = 4)','\bf 12.9% [O2] (n = 5)','\bf 10.9% [O2] (n =
   6)','\bf 8.90% [O2] (n = 7)','\bf 6.90% [O2] (n = 6)','\bf 4.90% [O2] (n = 6)','
   \bf 2.90% [O2] (n = 6)','\bf 0.90% [O2] (n = 3)', 'Location','northeastoutside')
   ;
```

```
lgd.FontSize = 20;
save("MEA_data_after_analysis\Gradient\Allexp\plot_gradient_n7.mat",'NumberSpikes',
    'err', 'avg_baseline_values',"-mat" )
```

SUBPLOTS EWFR PER ELECTRODE & PER GRADIENT:

```
C = []; T = [];
% A single file, or a list of files
if class(filename) == 'char'
    files = cell(1, 1);
    files{1} = [char(path) char(filename)];
else
    files = cell(length(filename), 1);
    for i=1:length(filename)
       files{i} = [char(path) char(filename(i))];
    end
end
% Read every file and combine
load(files{1}); T = [T; data.Ts]; C = [C; data.Cs];
for i = 2:size(files, 1)
    fprintf('loading...\n')
    load(files{i});
    T = [T; data.Ts];
    C = [C; data.Cs];
end
% Variables
Timebin = 1;                                                                % In
    hours
T = T/(25000*3600);                                                         %
    convert timestamps to hours (sample freq = 25kHz)
% Colors of the lines from green to red
newcolors_column = [0 1 0;
                    0.4660, 0.6740, 0.1880;
                    0.75, 0.75, 0;
                    1 1 0;
                    0.9290 0.6940 0.1250;
                    1 0.5 0.8;
                    1 0.3 0.1840;
                    1 0 0];
colororder(newcolors_column)
% concentrations of oxygen from high to low in percentages
concentrations = [14.9; 12.9; 10.9; 8.90; 6.90; 4.90; 2.90; 0.90];
%% Analyze data per column
%% Number of spikes per column per electrode over 36 h
% Initialize a dictionary to store avgFirst6Hours
avgFirst6HoursDict = struct();
for n = 1:8
    % Initialize variables for each column
    NumberSpikes_E = [];
    % Spikes per electrode
    for p = 1:8
        selectionIndex_E = find(C == (n)*10 + p);                           % For n
           = 1 and p = 1:8 --> C= 11, 12, 13...
        if isempty(selectionIndex_E)
            disp(['No data for column ' num2str(n) ' and electrode ' num2str(p)]);
            % Skip to the next electrode if there is no data
            continue;
        end
```

```
    Tselection_E = T(selectionIndex_E);                                    %
       Contains only timestamps in column n
    NumberSpikes_elec_col = zeros(floor(max(Tselection_E)/Timebin), 1);
    for q = 1:floor(max(Tselection_E)/Timebin)                             % Floor
       to ensures integer number of bins
        NumberSpikes_elec_col(q) = (length(find(Tselection_E > (q-1)*Timebin &
           Tselection_E < q*Timebin)));
    end
    % Check for enough activity (1 spike per 3 min per electrode)
    First6Hours = 6;
    if length(NumberSpikes_elec_col) >= First6Hours
        sixhourspikes = sum(NumberSpikes_elec_col(1:First6Hours));
        if sixhourspikes < 120                        % 6 hours of baseline, so
           360/3 = 120 spikes/h
            disp(['Column ' num2str(n) ' electrode ' num2str(p) ' is activated
               less than 1 spike/3min in baseline']);
            % Skip to the next electrode if the activity is less than 1 spike/
               min
            continue;
        end
    else
        disp(['Not enough data for the first 6 hours for column ' num2str(n) '
           and electrode ' num2str(p)]);
        % Skip to the next electrode if there no data for first 6 hours
        continue;
    end
    % Normalize over the first 6 hours
    if length(NumberSpikes_elec_col) >= First6Hours
        avgFirst6Hours = sixhourspikes/6;

        % Save avgFirst6Hours in the dictionary
        key = ['n' num2str(n) '_p' num2str(p)];
        avgFirst6HoursDict.(key) = avgFirst6Hours;
        % Normalize the electrode number of spikes by its 6 hour average
        NumberSpikes_elec_col = NumberSpikes_elec_col / avgFirst6Hours;
    else
        disp(['Not enough data for the first 6 hours for column ' num2str(n) '
           and electrode ' num2str(p)]);
        continue;
    end
    % Append the column data
    if isempty(NumberSpikes_E)
        NumberSpikes_E = NumberSpikes_elec_col;
    else
        % Makes sure to handle different lengths (not every column has same
           length)
        len_diff = length(NumberSpikes_elec_col) - size(NumberSpikes_E, 1);
        if len_diff > 0
            NumberSpikes_E = [NumberSpikes_E; zeros(len_diff, size(
               NumberSpikes_E, 2))];
        elseif len_diff < 0
            NumberSpikes_elec_col = [NumberSpikes_elec_col; zeros(-len_diff, 1)
               ];
        end
        NumberSpikes_E = [NumberSpikes_E, NumberSpikes_elec_col];
    end
end
```

```
    % Plot data
    figure(1);
    subplot(4, 2, n);
    plot((1:size(NumberSpikes_E, 1))' * Timebin, NumberSpikes_E, 'LineWidth', 2);
    set(gca, 'FontSize', 10)
    hold on
    xticks(0:6:36)
    xlabel('Time (h)', 'fontweight', 'bold')
    ylabel('Normalized EWFR', 'fontweight', 'bold', 'Rotation', 85)
    title(['\bf Column ' num2str(n) ': ' num2str(concentrations(n)) '% [O2]']);
%     title(['\bf Column ' num2str(n) ': 0% [O2]']);                          % exp 4
    luna
    % Grey patch from 6 to 30 hours
    hold on
    v = [6 -1; 30 -1; 30 10; 6 10];
%     v = [0 -1; 14 -1; 14 10; 0 10];                                          % exp
    4 luna
    f = [1 2 3 4];
    pa = patch('Faces', f, 'Vertices', v, 'FaceAlpha', 0.1, 'EdgeColor', 'none', '
        LineWidth', 0.5);
    pa.Annotation.LegendInformation.IconDisplayStyle = 'off';

    xlim([0 36])
%     xlim([0 13])                                                             % exp
    4 luna
    ylim([-1 10])
    yline(1, '--', 'LineWidth', 1.5)
    hold off
    % change name depending on experiment
    save(fullfile('C:\Users\lunah\OneDrive\Documenten\MATLAB\Matlab Utwente\Bachelor
        CNPH\MEA_scripts\MEA_data_after_analysis\Electrode\Column\', ['
       exp4_L_column' num2str(n) '.mat']), 'NumberSpikes_E', '-mat');
end
% Save avgFirst6HoursDict: normalization factor
save(fullfile('C:\Users\lunah\OneDrive\Documenten\MATLAB\Matlab Utwente\Bachelor
    CNPH\MEA_scripts\MEA_data_after_analysis\Electrode\Column\', '
    exp4_L_avgFirst6HoursDict.mat'), 'avgFirst6HoursDict', '-mat');
%% Load MEA data row wise gradient
[filename, path] = uigetfile('C:\Users\lunah\OneDrive\Documenten\MATLAB\Matlab
    Utwente\Bachelor CNPH\Artifects removed\', 'multiselect','on');
C = []; T = [];
% A single file, or a list of files
if class(filename) == 'char'
    files = cell(1, 1);
    files{1} = [char(path) char(filename)];
else
    files = cell(length(filename), 1);
    for i=1:length(filename)
       files{i} = [char(path) char(filename(i))];
    end
end
% Read every file and combine
load(files{1}); T = [T; data.Ts]; C = [C; data.Cs];
for i = 2:size(files, 1)
    fprintf('loading...\n')
    load(files{i});
    T = [T; data.Ts];
```

```
    C = [C; data.Cs];
end
% Variables
Timebin = 1;                                                    % In
   hours
T = T/(25000*3600);                                             %
   convert timestamps to hours (sample freq = 25kHz)
% concentrations of oxygen from low to high in percentages
concentrations_row = [0.90; 2.90; 4.90; 6.90; 8.90; 10.9; 12.9; 14.9];
%% Analyse data per row:
% Number of spikes per row per electrode over 36 h
% Initialize a dictionary to store avgFirst6Hours_r
avgFirst6Hours_r_Dict = struct();
for n = 1:8
    % Initialize variables for each row
    NumberSpikes_E_R = [];
    % Spikes per electrode
    for p = 1:8
        selectionIndex_E_R = find(C == n + 10*p);                   % For
            n = 1 and p = 1:8 --> C= 11, 21, 31, 41...
        if isempty(selectionIndex_E_R)
            disp(['No data for row ' num2str(n) ' and electrode ' num2str(p)]);
            continue;                                               %
               Skips to the next electrode if no data
        end
        Tselection_E_R = T(selectionIndex_E_R);                     %
           Contains only timestamps in column n
        NumberSpikes_elec_row = zeros(floor(max(Tselection_E_R)/Timebin), 1);
        for q = 1:floor(max(Tselection_E_R)/Timebin)                %
           Floor to ensures integer number of bins
            NumberSpikes_elec_row(q) = (length(find(Tselection_E_R > (q-1)*Timebin &
                Tselection_E_R < q*Timebin)));
        end
        % Check for enough activity (1 spike per 3 min per electrode)
        First6Hours = 6;
        if length(NumberSpikes_elec_row) >= First6Hours
            sixhourspikes_r = sum(NumberSpikes_elec_row(1:First6Hours));
            if sixhourspikes_r < 120                        % 6 hours of baseline, so
                360/3 = 120 spikes/h
                disp(['Row ' num2str(n) ' electrode ' num2str(p) ' is activated less
                    than 1 spike/ 3min in baseline']);
                % Skip to the next electrode if the activity is less than 1 spike/
                   min
                continue;
            end
        else
            disp(['Not enough data for the first 6 hours for row ' num2str(n) ' and
               electrode ' num2str(p)]);
            % Skip to the next electrode if there no data for first 6 hours
            continue;
        end
        % Normalize over the first 6 hours
        if length(NumberSpikes_elec_row) >= First6Hours
            avgFirst6Hours_r = mean(NumberSpikes_elec_row(1:First6Hours));
            % Save avgFirst6Hours in the dictionary
            key = ['n' num2str(n) '_p' num2str(p)];
            avgFirst6Hours_r_Dict.(key) = avgFirst6Hours_r;
```

```
        % Normalize the electrode number of spikes by its average of
        % the first 6 hours
        NumberSpikes_elec_row = NumberSpikes_elec_row / avgFirst6Hours_r;
    else
        disp(['Not enough data for the first 6 hours for row ' num2str(n) ' and
            column ' num2str(p)]);
        continue;
    end
    % Append the row data
    if isempty(NumberSpikes_E_R)
        NumberSpikes_E_R = NumberSpikes_elec_row;
    else
        % Makes sure to handle different lengths
        len_diff = length(NumberSpikes_elec_row) - size(NumberSpikes_E_R, 1);
        if len_diff > 0
            NumberSpikes_E_R = [NumberSpikes_E_R; zeros(len_diff, size(
                NumberSpikes_E_R, 2))];
        elseif len_diff < 0
            NumberSpikes_elec_row = [NumberSpikes_elec_row; zeros(-len_diff, 1)
                ];
        end
        NumberSpikes_E_R = [NumberSpikes_E_R, NumberSpikes_elec_row];
    end
end
% Plot data
figure(3);
subplot(4, 2, n);
plot((1:size(NumberSpikes_E_R, 1))' * Timebin, NumberSpikes_E_R, 'LineWidth', 2)
    ;
set(gca, 'FontSize', 10)
hold on
xticks(0:6:36)
xlabel('Time (h)', 'fontweight', 'bold')
ylabel('Normalized EWFR', 'fontweight', 'bold', 'Rotation', 85)
title("\bf Row " + n + ": " +concentrations_row(n)+ "% [O2]")
% Colors from red to green
newcolors_row = [1 0 0;
    1 0.3 0.1840;
    1 0.5 0.8;
    0.9290 0.6940 0.1250;
    1 1 0;
    0.75 0.75 0;
    0.4660 0.6740 0.1880;
    0 1 0;];
colororder(newcolors_row)
% Grey patch from 6 to 30 hours
hold on
v = [6 -1; 30 -1; 30 10; 6 10];
f = [1 2 3 4];
pa = patch('Faces', f, 'Vertices', v, 'FaceAlpha', 0.1, 'EdgeColor', 'none', '
    LineWidth', 0.5);
pa.Annotation.LegendInformation.IconDisplayStyle = 'off';
xlim([0 36])
ylim([-1 10])
yline(1, '--', 'LineWidth', 1.5)
hold off
% change name depending on experiment
```

```
    save(fullfile('C:\Users\lunah\OneDrive\Documenten\MATLAB\Matlab Utwente\Bachelor
        CNPH\MEA_scripts\MEA_data_after_analysis\Electrode\Row\', ['exp2_L_row'
       num2str(n) '.mat']), 'NumberSpikes_E_R', '-mat');
end
% Save avgFirst6Hours_r_Dict: normalization factor
save(fullfile('C:\Users\lunah\OneDrive\Documenten\MATLAB\Matlab Utwente\Bachelor
   CNPH\MEA_scripts\MEA_data_after_analysis\Electrode\Row\', '
   exp2_L_avgFirst6Hours_rDict.mat'), 'avgFirst6Hours_r_Dict', '-mat');
```

D. PI-CAS3/7 STAINING SCRIPTS

AUTOMATIC COUNT PI-CAS3/7 CELLS PER [O2]:

```
%% PI-CAS3/7 staining count function
% Column wise: Count_C_red, Count_R_red
% Row wise: Count_C_gr, Count_R_gr
% IMPORTANT--> replace which Count and save you need in function depending on column or row wise gradient
    !

function [A, time, red, green, AutoCount, Count_C_red, Count_C_gr, M1, M2] =
    Apoptosis_Analysis_batch_gradient
clc; clf;

%threshold=[20;12];
GeneralFolder = 'C:\Users\lunah\OneDrive\Documenten\MATLAB\Matlab Utwente\Bachelor CNPH\Gradient_cells';

Expt = '4_050624_MEA37049_hypoxia';                                    % Change name to get different
    experiment photos.
SquareSize=3264; drempel_1= 2.3 ; drempel_2= 2.4 ; FluorSize= 20;

ImageFolder = [GeneralFolder '/' Expt '/Photos'];                          % Change name if it is
    different

files = dir([ImageFolder '*/*.jpg']);

figure(1);

MaxLengthFiles = 4 * (6+24+6);                                             % change if experiment took
    longer

if length(files)> MaxLengthFiles
   files = files(1:MaxLengthFiles);
end

% Initialize variables
AutoCount = [];
Count_C_red = [];
Count_C_gr = [];
time = [];
red = [];
green = [];

for n = 1 : length(files)                                                   % use for example 1: 20:
    length(files) when you want to check something (goes much faster)
    disp([ImageFolder '/' files(n).name]);
    A = imread([ImageFolder '/' files(n).name]);

    if exist('x')~=1
        image(3*A);   title('Indicate left border:')                       % makes image brighter
        [x, ~] = ginput(1);                                                % lets you select a point to
             crop the image. Select somewhere in left upper corner
        A = A(1:SquareSize,x+1:x+SquareSize,:);
        MI = (mean(mean(A)));                                              % Mean of column then row
        MI = squeeze(MI(1,1,:));
    else
        A = A(1:SquareSize,x+1:x+SquareSize,:);
        MI = (mean(mean(A)));                                              % Mean of column then row
        MI = squeeze(MI(1,1,:));
    end

    threshold = [drempel_1*MI(1); drempel_2*MI(2)*0.7] + 2;
    MIntensity = mean(MI);

    figure(1);
    image(2*A);
    axis image;
    title([files(n).name ' Mean intensity: ' num2str(MIntensity)]);
    M1(n)=getframe;
    M2(n)=getframe;
    red(n)=mean(mean(A(:,:,1)));
    green(n)=mean(mean(A(:,:,2)));
    time(n)=(n)/4;                                                             % in hours

    % Red/dead cells
    Colour = 1;
```

```
    [h, Section, Filt_Section] = Read_Filter_displayImage(A, FluorSize, Colour);

    % Row wise: [AutoCount(n,1), Count_R_red(n,:)]
    % Column wise: [AutoCount(n,1), Count_C_red(n,:)]
    [AutoCount(n,1), Count_C_red(n,:)] = AutomaticCount(Filt_Section, Colour, threshold, SquareSize,
        FluorSize);


    % Green/ apoptotic cells
    Colour = 2;

    [h, Section, Filt_Section] = Read_Filter_displayImage(A, FluorSize, Colour);

    % Row wise: [AutoCount(n,2), Count_R_gr(n,:)]
    % Column wise: [AutoCount(n,2), Count_C_gr(n,:)]
    [AutoCount(n,2), Count_C_gr(n,:)] = AutomaticCount(Filt_Section, Colour, threshold, SquareSize,
        FluorSize);

    % Checks the average blue value and sets Count_red and Count_gr to NaN if it's above 1
    if MI(3) > 1
        % column
        Count_C_red(n,:) = NaN;
        Count_C_gr(n,:) = NaN;
        % row
%         Count_R_red(n,:) = NaN;
%         Count_R_gr(n,:) = NaN;
    end
end

% figure(2);
% movie(M);
figure(3);
%time=time*2;
plot(time, AutoCount(:,1),'r', time,AutoCount(:,2),'g');
axis([0;40;0;600]);
title(Expt);
xlabel('Time (h)','fontweight','bold','fontsize',9);
ylabel('Number of postivive cells','fontweight','bold','fontsize',9);
grid on;

clear h;

% Whole image cell count
save("Apoptosis_data\Exp4_Luna\AutoCount_exp4_L.mat","AutoCount", '-mat');

% Column gradient cell count
save("Apoptosis_data\Exp4_Luna\Count_C_exp4_L.mat","Count_C_red","Count_C_gr", '-mat');

% Row gradient cell count
% save("Apoptosis_data\Exp2_Luna\Count_R_exp2_L.mat","Count_R_red", "Count_R_gr",'-mat');

%% Local functions
function [h, Section, Filt_Section]=Read_Filter_displayImage(A, FluorSize, Colour)
%% Read, filter and display selected area
Section=A;

kernel = (1/(FluorSize^2))*ones(FluorSize,FluorSize);
Filt_Section = imfilter(Section(:,:,Colour), kernel, 'same');
A=Section;
for p =1:3
    if p~=Colour;
         A(:,:,p)=0;
    end;
end;
h=figure(1);
%% Automatic count function over whole image and per column or row

function [AutoCount, Count_C, Count_R] = AutomaticCount(Filt_Section, Colour, threshold, SquareSize,
    FluorSize)

BW = imregionalmax(Filt_Section);

AutomatischGeteld = 0;
```

```
Coord = zeros(SquareSize);                                                  % empty SquareSize x
    SquareSize matrix for the coordinates of the red & green cells
Count_C = [];
Count_R = [];

for a=1:SquareSize                                                          % b is de horizontale
    component (x coordinate) , a de verticale (y coordinate)
    for b=1:SquareSize
        if (BW(a,b) && (Filt_Section(a,b)>=threshold(Colour)))

            if a <= FluorSize
                    Yrange = (1:a+FluorSize);
            elseif a > SquareSize-FluorSize
                    Yrange = (a-FluorSize:SquareSize);
            else
                    Yrange = (a-FluorSize:a+FluorSize);
            end

            if b <= FluorSize
                    Xrange=(1:b+FluorSize);
            elseif b > SquareSize-FluorSize
                    Xrange=b-FluorSize:SquareSize;
            else
                 Xrange=b-FluorSize:b+FluorSize;
            end

            Size = FluorSize*2;

            if Colour==1
                kleur='r';
            end
            if Colour==2
                kleur='g';
            end

            rectangle('Position',[round(b-((Size-1)/2)) round(a-((Size-1)/2)) Size Size],'Curvature'
                ,[1,1],'EdgeColor',kleur);  % omcirkeld cel rood of groen

            % Count red & green cells
            AutomatischGeteld = AutomatischGeteld+1;
            BW(Yrange,Xrange) = 0;

            % Coordinates red & green cells
            Coord(a,b) = 1;                                                 % Puts 1 in matrix if there is
                 a green or red cell

        end
    end
end

AutoCount = AutomatischGeteld;

% Amount of cells per column or row

width = SquareSize/8;                                                       % 8 colomns/rows with a
    width of 408 pixels

for k = 1:8
    [x , y] = find(Coord == 1);                                             % Gives the x and y
        coordinates of those cells in 2 seperate matrices (remember top left is (0.0))

    % Column wise
    Count_C = [Count_C, length(find(x >= (k-1)*width + 1 & x <= (k)*width))];   % counts number of cells
        in specific column interval i.e. 0:408

    % Row wise
    Count_R = [Count_R, length(find(y >= (k-1)*width + 1 & y <= (k)*width))];   % counts number of cells
        in specific row interval i.e. 0:408
end
```

SINGULAR EXPERIMENT: PLOT COUNT PER [O2]

```
%% plot PI-CAS3/7 data per oxygen concentration
% load data of one or more experiments
% Change from column to row up until 'calulate mean and std',
% also change title of the plots accordingly!
```

```
clear; clc; close all
% Change folder depending on exp
[filename, path] = uigetfile('C:\Users\lunah\OneDrive\Documenten\MATLAB\Matlab Utwente\Bachelor CNPH\
    Staining_scripts\Apoptosis_data\', 'multiselect','on');
% Column
% C_C_red = []; C_C_gr = [];
% Row
C_R_red = []; C_R_gr = [];
% A single file, or a list of files
if ischar(filename)
    files = cell(1, 1);
    files{1} = fullfile(path, filename);
else
    files = cell(length(filename), 1);
    for i = 1:length(filename)
       files{i} = fullfile(path, filename{i});
    end
end
% Read every file and combine
load(files{1});
% Column
% C_red = Count_C_red; C_gr = Count_C_gr;
% Row
C_red = Count_R_red; C_gr = Count_R_gr;
for i = 2:size(files, 1)
    fprintf('loading...\n')
    load(files{i});
    % column
%         C_red = [C_C_red; Count_C_red];
%         C_gr = [C_C_gr; Count_C_gr ];
    % row
        C_red = [C_R_red, Count_R_red];
        C_gr = [C_R_gr, Count_R_gr];
end
% Only for exp 1 and 2 Luna (row wise)
% C_red = flip(Count_R_red, 2);
% C_gr =  flip(Count_R_gr, 2);
% concentrations of oxygen from high to low
concentrations = [14.9; 12.9; 10.9; 8.90; 6.90; 4.90; 2.90; 0.90];
%% Calculate mean and std.
% Ensure that the number of rows in C_red is 144
numRows = size(C_red, 1);
if numRows < 144
    error('Number of rows in C_red is less than 144.');
elseif numRows > 144
    warning('Number of rows in C_red exceeds 144. Truncating to 144 rows.');
    C_red = C_red(1:144, :);
    C_gr = C_gr(1:144, :);
end
% % Make exp 4 Luna 144x8 instead of 76x4
% maxLength = 144;
%
% % Make row and column data the same length
% if size(C_red, 1) < maxLength
%     C_red = [C_red; zeros(maxLength - size(C_red, 1), size(C_red, 2))];
% end
%
% if size(C_gr, 1) < maxLength
%     C_gr = [C_gr; zeros(maxLength - size(C_gr, 1), size(C_gr, 2))];
% end
% Reshape C_red and C_gr to group every 4 rows, 36 times
reshaped_C_red = reshape(C_red, 4, 36, []);
reshaped_C_gr = reshape(C_gr, 4, 36, []);
% Initialize arrays to store the means and standard deviations
mean_C_red = zeros(36, size(C_red, 2));
std_C_red = zeros(36, size(C_red, 2));
mean_C_gr = zeros(36, size(C_gr, 2));
std_C_gr = zeros(36, size(C_gr, 2));
% Compute the mean and standard deviation for every group of 4 rows per column
for col = 1:size(C_red, 2)                                          % iterates over 8 column
    for t = 1:36
        current_block_red = reshaped_C_red(:, t, col);
        current_block_gr = reshaped_C_gr(:, t, col);
        % Compute mean and std ignoring NaNs
        mean_C_red(t, col) = nanmean(current_block_red);            % mean without NaN values
```

```
        std_C_red(t, col) = nanstd(current_block_red);                    % std without NaN values
        mean_C_gr(t, col) = nanmean(current_block_gr);
        std_C_gr(t, col) = nanstd(current_block_gr);
    end
end
%%  Confine to < 120 cells during baseline (20 cells/hour)
% Calculate the sum of the first 6 rows for each column
sumsix_red = nansum(mean_C_red(1:6, :), 1);
sumsix_gr = nansum(mean_C_gr(1:6, :), 1);
% Find the columns where the sum is greater than 120
columnsToReplace_red = sumsix_red > 120;
columnsToReplace_gr = sumsix_gr > 120;
% Replace the entire column with NaN where the condition is true
mean_C_red(:, columnsToReplace_red) = NaN;
std_C_red(:, columnsToReplace_red) = NaN;
mean_C_gr(:, columnsToReplace_gr) = NaN;
std_C_gr(:, columnsToReplace_gr) = NaN;
%% Plot the mean and std per column or row
% Create time vector for 36 time points
time = (1:36) * 4 / 4;                                                    % time vector with intervals
    of 1 hour
figure(1);
for m = 1:8
    subplot(2,4,m)
    errorbar(time, mean_C_red(:, m), std_C_red(:, m), 'r', "MarkerSize", 6, "Color", "r", "CapSize", 1)
    set(gca, 'FontSize', 11)
    hold on
    errorbar(time,mean_C_gr(:, m), std_C_gr(:, m), 'g', "MarkerSize", 6, "Color", "g", "CapSize", 1)
    set(gca, 'FontSize', 11)
    xticks(0:6:36)
    xlim([0 36]);
%     xlim([0 18]); % exp 4 luna
    ylim([0 200]);
    title("\bf" +concentrations(m)+ "% [O2]")
%     title("\bf" +concentrations(8)+ "% [O2]") % exp 4 luna
    sgtitle("\bf Experiment 2")
     if m == 8
        % add a bit space to the figure
        fig = gcf;
        fig.Position(3) = fig.Position(3);
        % add legend
        Lgnd = legend('\bf PI (dead)', '\bf CAS3/7 (apoptotic)');
        Lgnd.Position(1) = 0;
        Lgnd.Position(2) = 0.48;
     end
    xlabel('Time (h)','fontweight','bold','fontsize',8)
    ylabel('Number of postivive cells','fontweight','bold','fontsize',8)
    grid on
    hold off
end
% change filename for different exp!
save("Apoptosis_data\AllExp\mean_std_red&gr_exp2_L.mat", 'mean_C_red', 'std_C_red', 'mean_C_gr','std_C_gr
    ', "-mat");
```

MULTIPLE EXPERIMENTS: PLOT COUNT PER [O2]

```
%% Plot all experiments together
clear; clc; close all;
[filename, path] = uigetfile('C:\Users\lunah\OneDrive\Documenten\MATLAB\Matlab Utwente\Bachelor CNPH\
    Staining_scripts\Apoptosis_data\AllExp\', 'multiselect','on');
% Initiate empty matrices
mean_red = [];
mean_gr = [];
std_red = [];  % if you want to add std together, you need to add the variances = std1^2 + std2^2 and
    then take the squareroot
std_gr = [];
% A single file, or a list of files
if ischar(filename)
    files = cell(1, 1);
    files{1} = fullfile(path, filename);
else
    files = cell(length(filename), 1);
    for i = 1:length(filename)
       files{i} = fullfile(path, filename{i});
    end
end
```

```
% Read every file and combine
load(files{1});
mean_red = mean_C_red;
mean_gr = mean_C_gr;
std_red = std_C_red;
std_gr = std_C_gr;
for i = 2:size(files, 1)
    fprintf('loading...\n')
    load(files{i});
    mean_red = [mean_red; mean_C_red];
    mean_gr = [mean_gr; mean_C_gr];
    std_red = [std_red; std_C_red];
    std_gr = [std_gr; std_C_gr];
end
%% Get mean and std of ALL experiments
% For example takes from exp 1 row 1, from exp 2, row 37, exp 3 row 73....
% they all correspond to the same time.
% Determine the number of experiments
Amount_exp = length(mean_red) / 36;
% Preallocate matrices for average and std deviation
avg_red = zeros(36, 8);
avg_gr = zeros(36, 8);
std_r = zeros(36, 8);
std_g = zeros(36, 8);
for column = 1:8
    for row = 1:36
        % Extract data for the current (row, column) across all experiments
        mean_red_values = mean_red(row:36:end, column);
        mean_gr_values = mean_gr(row:36:end, column);
        std_red_values = std_red(row:36:end, column);
        std_gr_values = std_gr(row:36:end, column);
        % Calculate the mean and std deviation for the current (row, column)
        avg_red(row, column) = nanmean(mean_red_values);
        avg_gr(row, column) = nanmean(mean_gr_values);
        std_r(row, column) = sqrt(nansum(std_red_values .^ 2));
        std_g(row, column) = sqrt(nansum(std_gr_values .^ 2));
     end
end
 %% Plot mean and std
time = ((1:36) * 4 / 4)';                                                    % time vector with
    intervals of 1 hour
figure(1);
% Colors from green to red
newcolors_colomn = [0 1 0;
                    0.4660, 0.6740, 0.1880;
                    0.75, 0.75, 0;
                    1 1 0;
                    0.9290 0.6940 0.1250;
                    1 0.5 0.8;
                    1 0.3 0.1840;
                    1 0 0];
colororder(newcolors_colomn)
% concentrations of oxygen from high to low in percentages
concentrations = [14.9; 12.9; 10.9; 8.90; 6.90; 4.90; 2.90; 0.90];
for m = 1:8
    subplot(2,4,m)
    errorbar(time, avg_red(:, m), std_r(:, m),"-r", "MarkerSize", 6, "Color", "r", "CapSize", 1)
    set(gca, 'FontSize', 11)
    hold on
    errorbar(time, avg_gr(:, m), std_g(:, m),"-g", "MarkerSize", 6, "Color", "g" , "CapSize", 1)
    set(gca, 'FontSize', 11)
    xticks(0:4:36)
    xlim([0 37]);
    ylim([0 200]);
    title( +concentrations(m)+ "% [O2]")
    xlabel('Time (h)','fontweight','bold')
    ylabel('Number of positive cells','fontweight','bold')
    if m == 8
        % add a bit space to the figure
        fig = gcf;
        fig.Position(3) = fig.Position(3);
        % add legend
        Lgnd = legend('\bf PI (dead)', '\bf CAS3/7 (apoptotic)');
        Lgnd.Position(1) = 0;
        Lgnd.Position(2) = 0.48;
```

```
    end
    hold off
end
save("Apoptosis_data\AllExp\mean_std_r&g_all.mat", 'avg_red', 'avg_gr', 'std_r', 'std_g', "-mat")
```