Using Activity-Related Signal Changes for Automated Cell Segmentation and Signal Extraction

Research Project for M. Sc. Neuroscience March – May 2024

Julian Webb

University of Freiburg,
University Clinic of Freiburg, Department of Neurosurgery

Supervisors:
Christophe Petry, Vatsalkumar Jariwala, Dr.-Ing. Kevin Joseph, Prof. Jürgen Beck

Table of Contents

Abstract

In this project, an automatic method was developed to extract the signals of glioblastoma cells from a live cell recording monitoring intracellular calcium concentration. Unlike other methods, it uses the signal throughout time to determine which parts of the image belong to the same cells. The benefit of this method is eliminating the human bias of manual cell segmentation and facilitating the process of analyzing the recordings. It works by first dividing the image into a grid of regions of interest (ROIs) – like a multielectrode array. Then, for each ROI, a signal is extracted based on the pixels it contains. Next, to remove low-frequency trends in the signal, which are not due to changes in calcium concentration in the cells, the signal is detrended by use of a rolling mean. After removing the ROIs which do not contain cells, the ROIs are clustered based on their signal and location, using an algorithm such as K-Means or agglomerative clustering. Ideally, each cluster contains one cell. Finally, a representative signal is extracted from each cluster, which can be used for further analysis. Depending on the image, the algorithm identifies ~50-75% of cells correctly. This shows that the method has merit but needs to be further refined, potentially by improving the removal of empty ROIs and the way signals are compared.

Abbreviations

ROI: Region of Interest
GBM: Glioblastoma
STD: standard deviation

Introduction

Glioblastoma (GBM) is an aggressive type of cancer that occurs in the brain and spinal cord. It originates from astrocytes (Mayo Clinic, 2024), which are a type of glial cell in the brain. Despite decades of research, its treatment hasn't improved. To evade therapies and ensure progression, it heavily relies on resistant cellular networks (Osswald et al., 2015), which among other things are used for resource trafficking and communication (Winkler & Wick, 2018). Since calcium is an essential second messenger in biology, many cancers have mutations in their calcium signaling apparatus that allow their uncontrolled growth. Since it has been shown that calcium signaling in GBM networks has biological relevance (Hausmann et al., 2023; Robil et al., 2015), understanding the specifics of it will likely help with developing new therapeutic strategies (Leclerc et al., 2016; Monteith et al., 2012, 2017; Stewart et al., 2015).

To measure the calcium concentrations in live cells, genetically encoded calcium indicators (GECIs) like GCaMP6f are employed (Miyawaki et al., 1997). These indicators change their fluorescence properties upon binding to calcium. Then, to visualize the fluorescent indicators, fluorescence microscopy is used. This involves illuminating the sample with a specific wavelength of light that excites the calcium indicator. The emitted light is then collected and detected, often with a camera or photomultiplier tube. While this method was only used on these types of images here, it can in principle be used for any imaging modalities which measure a change in activity over time.

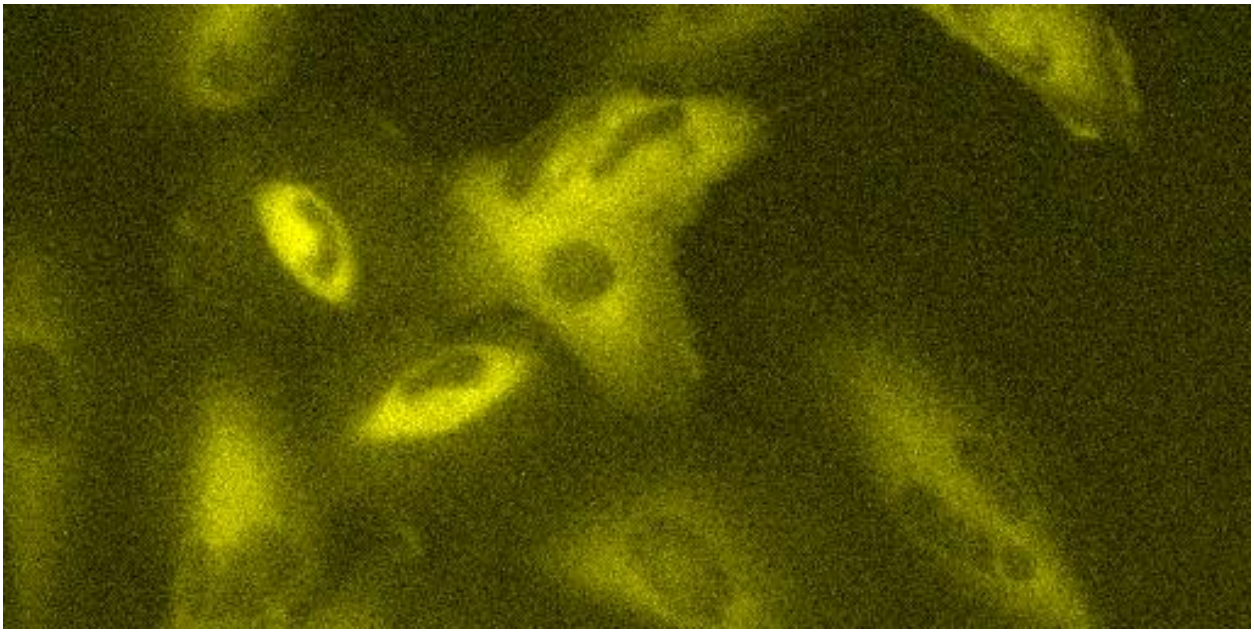

Figure 1 – Calcium Image of Glioblastoma Cell Culture (512 x 256 pixels)

By repeatedly making these images (Figure 1) with a certain sampling rate, a series of images, i.e. a video, is created. The sampling rate must be sufficiently higher than the speed of the cell activity. Since this is comparatively low for glioblastoma cells, a sampling rate of around 1 Hz was employed here.

For further analysis, the signal of each individual cell should be extracted. There are various attempts to do this:

- Manually, one would do this by using a software, such as ImageJ, to outline the borders of each cell, creating *regions of interest (ROIs)*, and then extracting the signal of each ROI. Unfortunately, this approach is very tedious, time-consuming, and introduces variability due to human error.
- There are also automated solutions using computer vision to detect the GBM cells. However, they are difficult to get working in this scenario. This is due to the highly varying sizes and shapes of the glioblastoma *multiform* cells and because the calcium labeling with GCaMP6f is not uniform across cells.

In this project, this procedure is automated, using a different paradigm. Instead of just using the 2D image properties of a single frame, the 3rd dimension, which is the signal throughout time, will also be used to detect the different cells. This enables comparing the signals of certain regions to determine whether they belong to the same cell, rather than just using the raw data of one frame.

The basic idea is to create a grid of ROIs, similar to a multi-electrode-array, where each ROI contains many pixels. Then, the signals of each pixel in a ROI are summarized into a single signal. These signals are then clustered based on their similarity, using an algorithm such as K-Means or agglomerative clustering. The goal is to end up with a cluster for each cell. The signals can then be extracted from each cluster to get the cell signals. Figure 2 shows an overview of the steps.

To test whether this approach has any validity, it was tested on an artificially generated image with no noise and perfectly similar cell signals. Here, it perfectly clusters the artificial cells.

With a real image, it correctly identifies many of the cells correctly but also tends to erroneously divide larger cells into 2 or more clusters, and group multiple smaller cells into a single cluster. With more experimentation, the approach likely be further improved.

Methods

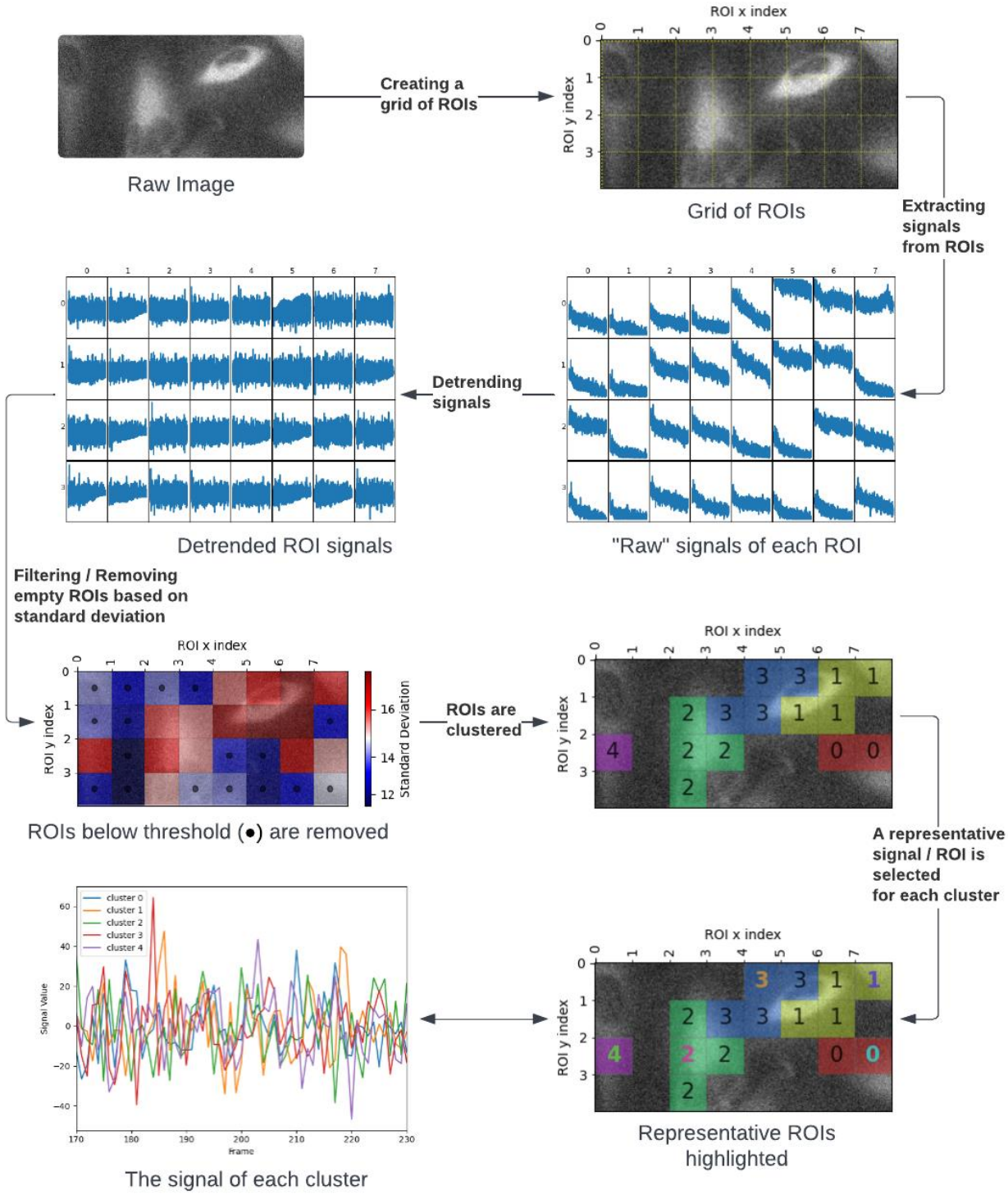

Figure 2 - An overview of the processing steps

Digital Image Representation

To process the image, it is essential to understand how it is represented digitally.

When loading the image into Python, it is represented as a 3-dimensional array. Two of the dimensions are the width and height of the image in pixels. The last dimension is the number of frames which is equivalent to the length of the recording times the sampling rate. Each entry $e_{f,x,y}$[1] in that array represents the luminosity of the pixel at position (x, y) in frame f. This is typically a positive floating-point number ranging from 0 to 255.[2]

Thus, as a first step, the width and height, i.e. the image dimensions, and the number of frames are read out because they will be needed throughout the code.

Generating Regions of Interest (ROIs)

To reduce computational complexity, the pixels are grouped into equally large rectangles (typically squares), termed "regions of interest". The height and width in pixels are specified by the experimenter. Then, based on the size, the algorithm generates a grid of ROIs. Each ROI has a horizontal (x) index and a vertical (y) index.

Extracting ROI signals

Based on the grid of ROIs, the pixel values of each ROI should be summarized into one signal. Since a ROI contains $n \times m$ pixels, there are endless ways of doing this.

Here, two were tried:

1. **Average value:** Summarizing all the pixels' signals into one signal, by taking the average for each frame.
2. **Highest amplitude:** Selecting the pixel with the highest amplitude. This means for each ROI, the pixel with the highest value in any frame is found and that pixel's entire signal is used as the ROI's signal.

Either way, there's with a signal for each ROI, which will be used for further processing.

Detrending ROI Signals

To account for low-frequency trends in the luminosity which are not due to changes in calcium concentration, the signal is detrended by subtracting a rolling mean from the original values. To do this, the experimenter decides on a window size, which corresponds to a number of frames. For every frame f, the mean of the signal around f is calculated

[1] Note that the order of indexing the frames and coordinates varies throughout the code
[2] Different scales are taken care of before clustering

based on the window size and is then subtracted from the signal value at f. E.g., if window size of 60 frames is chosen and the sampling rate is 1Hz, the mean of the signal in the interval $[f-30, f+30)$ is calculated and subtracted from the value at frame f.

Removing Empty ROIs

As you can see in Figure 2 *Grid of ROIs*, there are many "empty" ROIs, which do not overlap with any cell. To remove these, the experimenter specifies a standard deviation (STD) threshold. The STD is calculated for the detrended signal of each ROI and the ROIs which are below this threshold are excluded from further analysis. This procedure is based on the assumption that active cells will have a more variable signal than the background and thus only the background will be filtered out.

Clustering ROIs

To create a cluster of ROIs for each cell, based on the list of mostly filled ROIs and their detrended signals, two different clustering paradigms were tried:

1. Hierarchical clustering (agglomerative clustering)
2. Partitioning clustering (K-Means)

These will be discussed in the sections below.

Beside the signals, it is also useful to factor in the location of the ROIs, so that ROIs which are too far apart don't get clustered together. The reasoning is that cells only have a limited size and ROIs that are further apart than the maximum cell size cannot belong to the same call. This will be done in the some of the variants below.

Hierarchical Clustering (Agglomerative Clustering)

Hierarchical clustering works based on a proximity matrix containing the proximity[3] of each pair of samples. Based on this matrix, in each step the samples with the highest similarity are merged (agglomerative clustering) or the samples with the highest dissimilarity are divided (divisive clustering) (Everitt et al., 2011).

Here, the samples are the filled ROIs, and the matrix entries are the pairwise proximities between each pair of ROIs. The procedure consists of first computing a proximity matrix and then clustering based on it. The main argument for using this form of clustering is that the proximity can be freely computed, based on what works best. This allows for decreasing the similarity when ROIs are further apart.

[3] Proximity is used as an umbrella term for similarity, dissimilarity, or distance.

Computing Proximity

In this step, the desired result is a symmetric $n \times n$ matrix of pairwise proximities between each ROI, where n is the number of ROIs (Table 1). If the values represent dissimilarity, the diagonal should be 0 because there is no dissimilarity of a ROI with itself. The signals and spatial location of the ROIs should be considered, so that ROIs which are too far apart don't get merged.

Table 1 - Partial ROI dissimilarity matrix. Since the matrix is symmetric, only one half is shown.

	ROI (0;2)	**ROI (0;6)**	**ROI (2;0)**	**ROI (2;2)**
ROI (0;2)	0.0			
ROI (0;6)	1.0	0.0		
ROI (2;0)	0.94	2.0	0.0	
ROI (2;2)	0.98	1.0	0.98	0.0

There's an endless number of different possible measures to compare the signals. The most common one is the Euclidian distance. However, here, the cosine similarity, which doesn't take the amplitude of the signals into account, is used.

When using cosine similarity, the signals are interpreted as vectors in an f-dimensional space, where f is the number of frames. The cosine similarity is the angle between these vectors. Formally, it's defined as $\frac{R_1 \cdot R_2}{||R_1|| \cdot ||R_2||}$, i.e. the dot product of the signal vectors divided by the product of their magnitudes. It is bounded in the interval $[-1, 1]$ (Prakash, 2023).

Because, for later analysis, dissimilarity rather than a similarity is desired, the cosine similarity is subtracted from 1. It results in a dissimilarity in the interval $[0, 2]$. This can be efficiently computed for the whole signals data structure with the scikit-learn library.

To avoid clustering ROIs which are spatially too far apart, a ROIs dissimilarity with all ROIs which are not within a certain spatial range is set to the maximum value (2). This range is specified by the experimenter.

Clustering

Next, based on the pairwise dissimilarities, the ROIs are clustered. Each ROI starts out as its own cluster and in each step, the most similar clusters are combined. This can easily be done in Python with the following function: `scipy.cluster.hierarchy.linkage(distances)`.

This process can be visualized as a dendrogram (Figure 3). It shows in which order the clusters were merged and, on the y-axis, one can see how high the dissimilarity was for merging them.

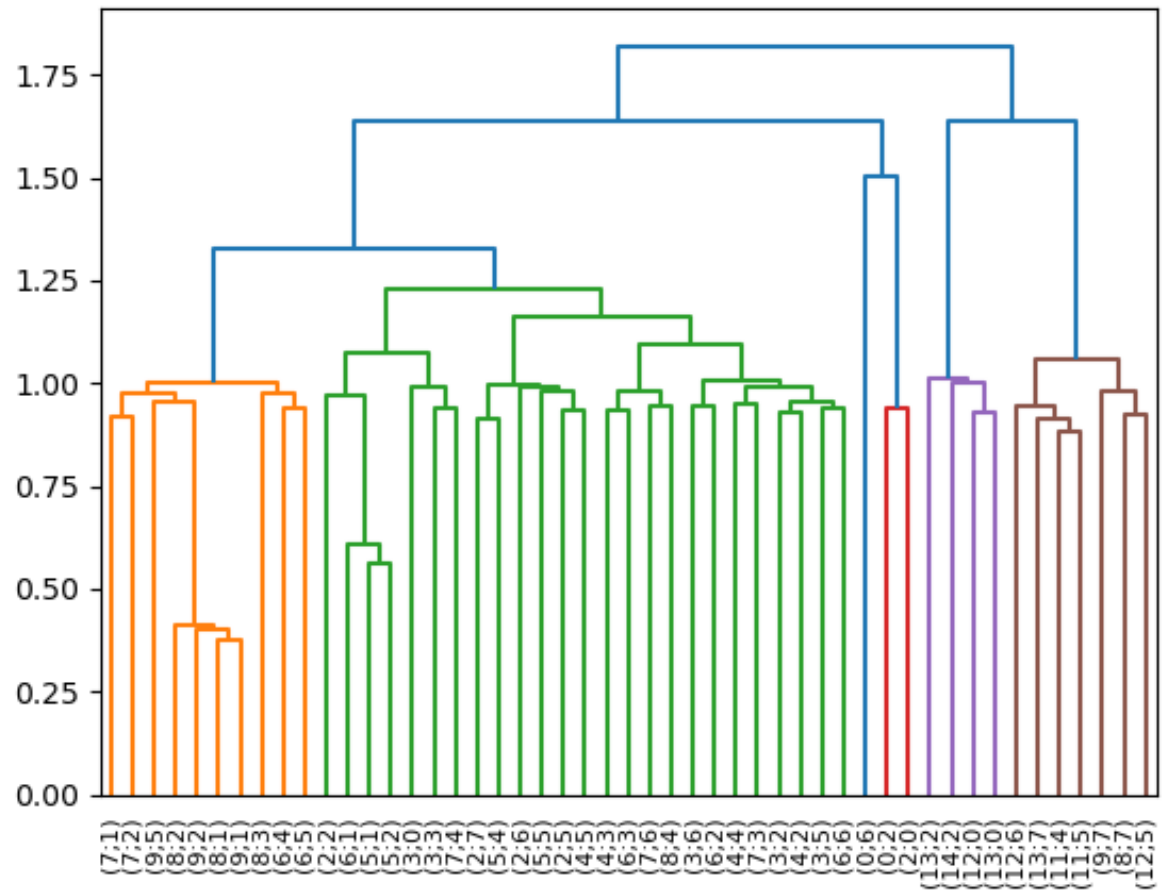

Figure 3 - Agglomerative Clustering Dendrogram

Partition Clustering (K-Means)

K-Means doesn't work based on a proximity matrix but rather based on directly computing the Euclidian distance between samples and centroids. K samples are chosen to be the initial centroids. Then each sample is assigned to the nearest centroid based on the Euclidian distance. Now, the centroids are recalculated based on the mean of the samples in its cluster. Then, in each iteration the samples are reassigned to the nearest centroid and the centroids are recomputed. This is repeated until convergence or for a maximum number of iterations (Everitt et al., 2011).

Basic K-Means

In this case, the samples are the ROIs. In the most basic form, their signals are compared, and a "centroid signal" is computed based on the means of the signals. Here, this is referred to as "basic K-Means".

Spatial, Weighted K-Means

To factor in the spatial location of the ROIs, they need to be added as additional variables to the samples. However, this requires some more advanced adjustments.

Adjusting for Image Resolution

The computed spatial distance between the ROIs should not depend on the resolution of the image, but only on the real-world spatial distance between the ROIs. However, if there are two recordings of identical cells but one had a higher resolution, then the difference in pixels would be higher in the high-res image.

To avoid this, the location of the ROIs won't be measured in pixels, but instead in millimeters (mm). To compute this, the experimenter needs to specify the resolution r of the image in $\frac{pixels}{mm}$. The width measured in millimeters can be calculated by dividing the width measured in pixels by the resolution:

$$w_{mm} = {}^{w_{pixels}}\Big/_{r\,\frac{pixels}{mm}}$$

The same is done for the height.

Now, the center can be computed based on the spatial location on the image, rather than the pixel location. This way the Euclidian distance between two ROIs will remain unaffected by the resolution.

Adjusting for Number of Frames and Spatial Weight

Now that the center of the ROIs is known, it will be included in the variables of each ROI. That means each ROI will be represented by a vector of variables:

$$(x, y, s_0, \dots, s_{F-1})$$

Where x and y are the x- and y-coordinates of ROI's center, s_f is the signal value at frame f and F is the number of frames.

Thus, when K-Means computes the overall Euclidian distance[4] between two ROIs, it will use the following formula, based on the definition of the Euclidian distance for ROIs a and b:

[4] There are three different distances which are considered. Here, they are called "spatial distance" which refers to the distance between the centers of the ROIs, "signal distance" which is the dissimilarity of the ROI's signals, and "overall/ROI distance/dissimilarity" which is some form of combination of the two.

$$E(a,b) = \sqrt{(x^{(a)} - x^{(b)})^2 + (y^{(a)} - y^{(b)})^2 + \sum_{f=0}^{F-1} \left(s_f^{(a)} - s_f^{(b)}\right)^2}$$

From the formula, it becomes clear that the location only makes up the first two summands, but the signal makes up a much larger number of summands, depending on the number of frames F. This means that the signal distance will be weighed much more heavily than the location, especially if there are many frames, i.e. the recording is long. This is undesirable, as the location has the same importance, no matter how long the recording is.

Further, it would be useful to introduce a factor that lets the experimenter weigh how much they want to take the location vs. the signal into account, to see what optimizes the results.

Thus, the first two summands are scaled up by F and and a weight w. Since there are two coordinates, the product is divided by 2. That gives us $j = \frac{n \cdot w}{2}$:

$$j\left(\left(x^{(a)} - x^{(b)}\right)^2 + \left(y^{(a)} - y^{(b)}\right)^2\right)$$

If $w = 1$, the signal and spatial distance will be weighed equally, if $w > 1$, the spatial distance will be weighed more, and it will be weighed less if $w < 1$.

To implement this in Python, the actual center values x and y should be multiplied, so j would be brought into the parentheses:

$$\left(\sqrt{j}x^{(a)} - \sqrt{j}x^{(b)}\right)^2 + \left(\sqrt{j}y^{(a)} - \sqrt{j}y^{(b)}\right)^2$$

That leads to the next iteration of the formula:

$$E(a,b) = \sqrt{j(x^{(a)} - x^{(b)})^2 + j(y^{(a)} - y^{(b)})^2 + \sum_{f=0}^{F-1} \left(s_f^{(a)} - s_f^{(b)}\right)^2}$$

Thereby, the number of frames has been compensated, and a scalar has been introduced that lets the experimenter regulate how much they want to take the location of the ROIs into account.

Adjusting for Signal Amplitude

The last factor that must be adjusted for is the amplitude and the numerical representation of the signal. If the amplitudes of the signals are higher, then the signal distances will also be higher. However, this will again bias the overall distance towards the signal distance. Thus, it needs to be adjusted for.

As an example, say there are two signals a and b and which are represented in three different ways:

1. Normalized: Values are represented as decimal point numbers in the range $[0, 1]$.
2. 8-bit: Values are represented in the range $[0, 2^8[$.
3. 16-bit: Values are represented in the range $[0, 2^{16}[$.

Figure 4 shows a plot of the signals and the Euclidian distances between them in the different representations. It shows that the higher the amplitude is, the higher the distance is. Even with the same number of bits, the signal amplitudes vary based on the saturation during the recording and the amplitude of the spikes of the neurons. Again, this is undesirable because this means the signal distance is unproportionally high in comparison to the spatial distance.

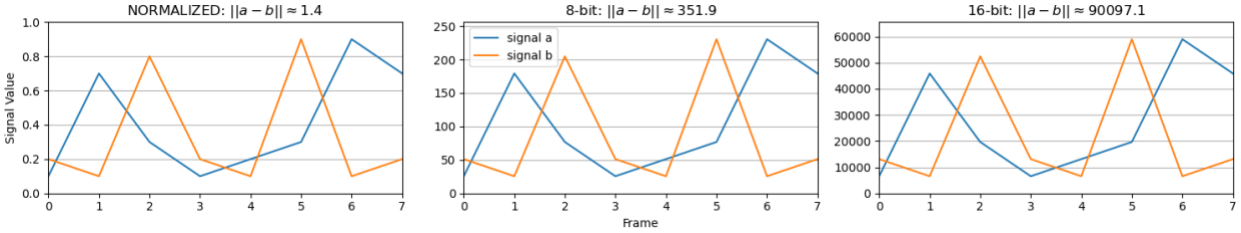

Figure 4 - Equivalent signals with varying amplitudes and their distances

To compensate for this, the overall maximum value in any of the absolute signals is found, and all the signals are normalized by dividing by this one value. That way, all the signals are normalized between -1 and 1 but are still on the same scale.

In terms of our formula, the detrended signal value s_f is replaced with $g_f = {}^{s_f}/_{p}$ where p is the overall maximum amplitude and g_f is the signal normalized with p. The last summand of the formula thus transforms:

$$\sum_{f=0}^{F-1}\left(s_f^{(a)} - s_f^{(b)}\right)^2 \rightarrow \sum_{f=0}^{F-1}\left(g_f^{(a)} - g_f^{(b)}\right)^2$$

Final Formula

The following formula for the Euclidian distance results from accumulating these changes into one equation:

$$E(a,b) = \sqrt{j(x^{(a)} - x^{(b)})^2 + j(y^{(a)} - y^{(b)})^2 + \sum_{f=0}^{F-1} \left(g_f^{(a)} - g_f^{(b)}\right)^2}$$

a, b: the ROIs
x, y: The x- and y-coordinates of the ROI's center
$j = \frac{n \cdot w}{2}$: The adjustment for the number of frames F and the spatial weight w
$g_f = \frac{s_f}{p}$: The normalized signal in frame f
s_f: The detrended signal in frame f
p: The overall maximum value of any signal

Conclusively, to adjust the values, the center coordinates must be represented in spatial units (e.g. millimeters), multiplied with $\sqrt{j}$, and the detrended signals must be divided by the overall maximum value in any signal p.

Extracting Signals from Clusters

Now that the ROIs have been clustered, no matter which method was used, there's a list of k clusters which each are made up of a set of ROIs. Optimally, each cluster should represent one cell. Since the desired outcome for further analysis is one signal for each cell, a signal must be extracted from each cluster.

Since each ROI has its own signal, there are many ways a signal could be chosen or created. It's important to consider that many ROIs are only partially filled with a cell and the cell's activity can vary significantly on different parts of its soma.

To find the ROI on the part of the cell, which is most active, the highest peak, i.e. the maximum value, of any signal in that cluster, is found. This whole signal is then used as the representative signal for that cluster. The ROI that belongs to that signal is termed the "representative" ROI for that cluster.

Results

Generating Regions of Interest (ROIs)

Analyzing single pixel signals would lead to very high computational complexity and at the same time is not necessary. For instance, the typical an image would have dimension of $2048 \cdot 1536 = 3{,}145{,}728$ pixels. Sampling for 30 minutes with a sampling rate of $1Hz$ would result in 1800 frames. In the clustering paradigm, this means there are $3{,}145{,}728$ samples and 1800 variables which even a high-end computer can't handle.

To avoid this, the ROIs are created. In general, a smaller size, i.e. a higher resolution, increases the computational complexity but the size also needs to be small enough, such that ROIs don't tend to cover multiple cells. This is because later each ROI will be assigned to one cell. There is no correct solution for this if a ROI overlaps with multiple cells. Based on these considerations, the experimenter chooses a size that suits their image.

In Figure 5, you can see a grid of ROIs superimposed on the image.

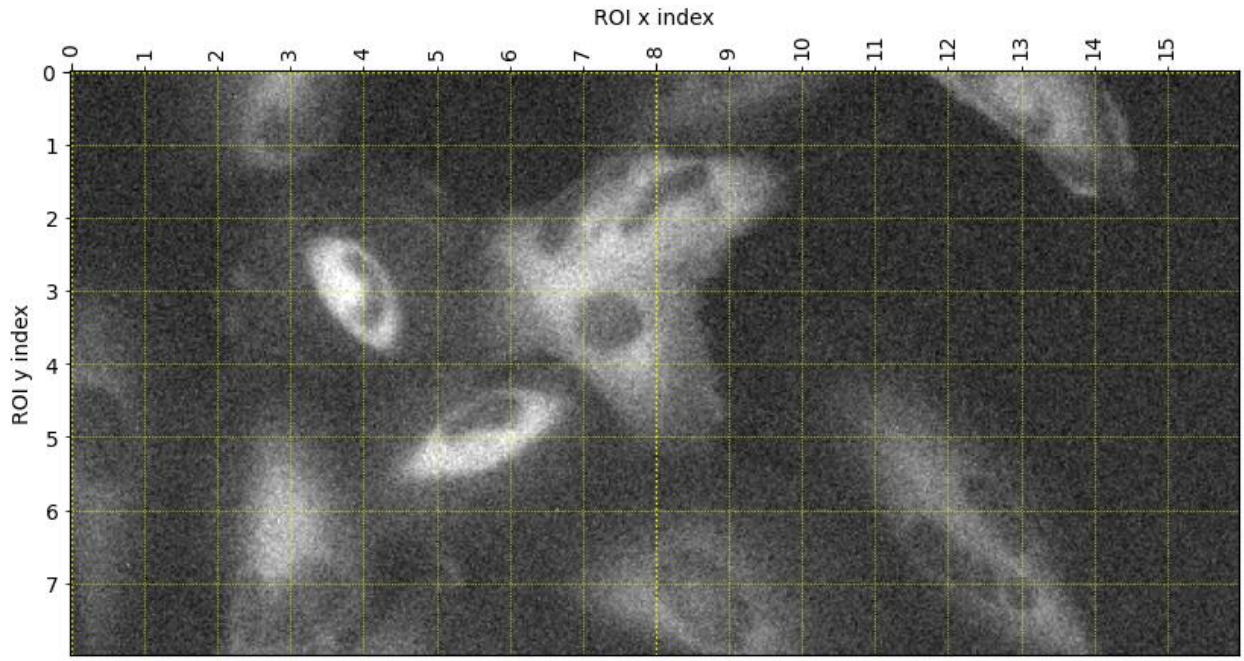

Figure 5 - Grid of ROIs with size 32 x 32 pixels on the image

Extracting ROI signals

If the average value for each frame is used, ROIs at the edge of a cell have a very low amplitude signal because they are mostly filled with background. Even if a ROI is fully covered by a cell, it's possible that only a part of that cell lights up. These factors often result

in the ROI's signal being lower than would be ideal for clustering, which is why this approach usually doesn't yield the best results.

This changes when the highest amplitude is used. Then, even ROIs that only partially cover an active cell will normally get their signal from the part that is covering the cell, because this will most likely have the highest peak. This means a partially filled ROI will still have a strong signal. Hence, the highest amplitude approach was used to extract ROI signals, which indeed helped the downstream processing. Figure 6 shows each ROI with a "raw" signal extracted using the highest amplitude. This signal will be used for further processing.

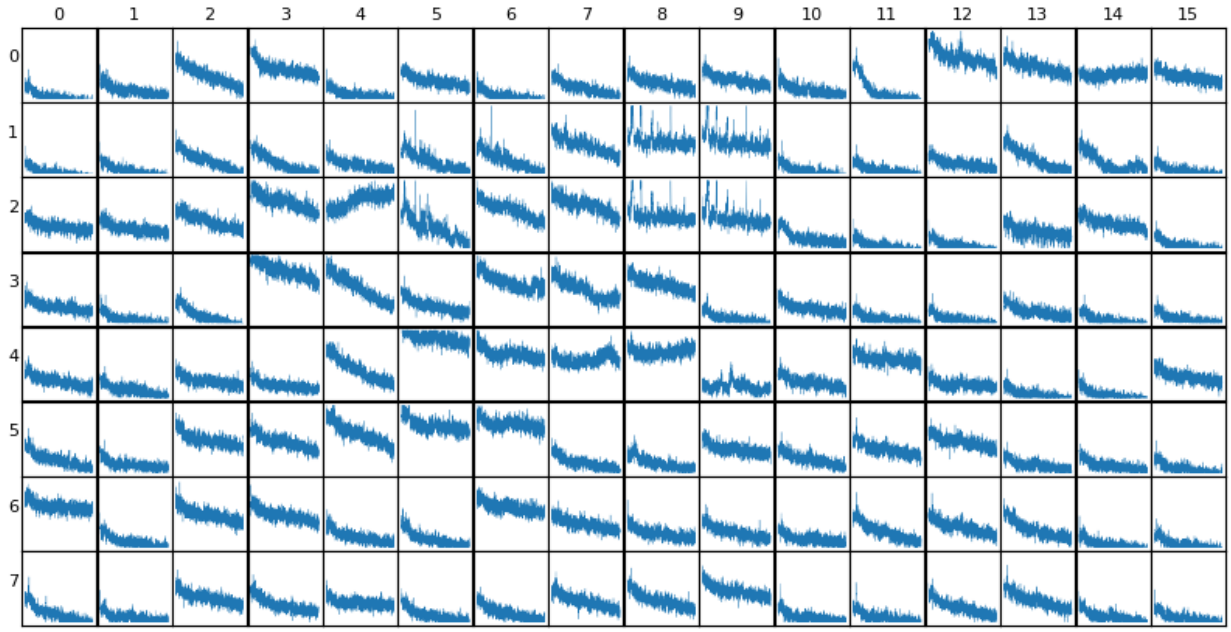

Figure 6 - The signals of the ROIs in the grid. The labels represent the x- and y-indexes.

Detrending ROI Signals

As you can see in *Figure 6*, most signals tend to decrease or increase over time. The signals should represent the changes in calcium concentrations in the cell. However, not all changes in fluorophore brightness can be attributed to changes in calcium level. There are other factors like fluorophore bleaching or other transient changes in brightness. These changes are usually slower and need to be filtered out so that only the fast calcium concentration related changes remain. This is necessary for two reasons:

- ROIs which don't contain a cell will be filtered out later based on their STD. A low frequency trend will inflate the STD.
- Later when clustering the ROIs, their signals are compared. The comparison is likely to be more meaningful if they have a common baseline and only the changes in calcium concentration are represented.

To account for this, the signal is detrended by using the rolling mean. Figure 7 that the detrending procedure removes the low-frequency trend while maintaining the higher-frequency changes in luminance.

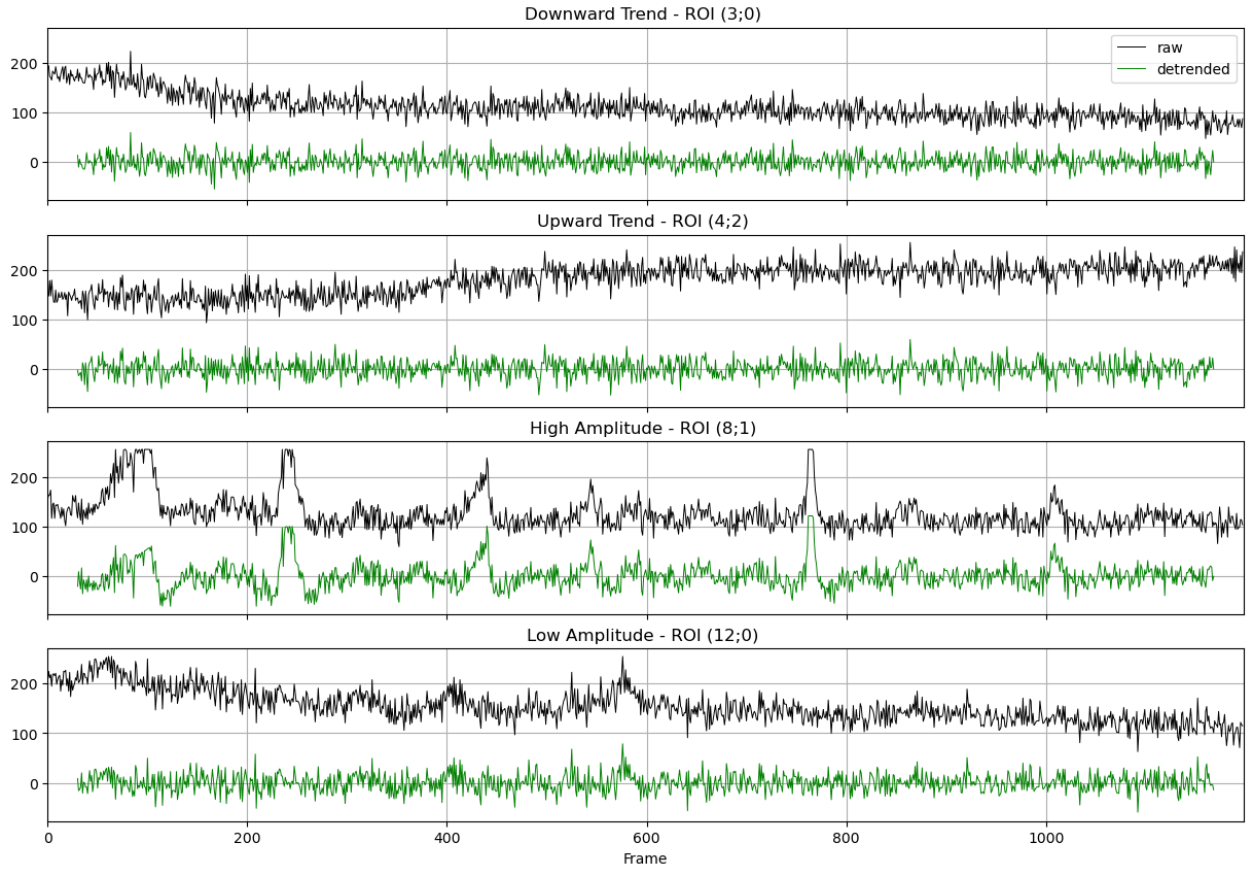

Figure 7 - ROI signals before and after detrending

Removing Empty ROIs

Empty ROIs need to be removed because only cells with calcium activity are interesting. Also, they should be removed at this step because they are problematic for the clustering mainly for two reasons:

1. They increase the computational complexity, simply because there are more ROIs.
2. They interfere with the quality of the clustering results in various ways. If not removed, the clustering algorithm should make one huge cluster of empty ROIs. However, this isn't what happens in practice. One reason is empty ROIs which are far apart won't get clustered together because the spatial location of each ROI will be factored in later. Also, empty ROIs still have a signal with a significant amplitude and their signals aren't necessarily similar which means they often don't get clustered together.

This makes it essential to remove these empty ROIs. Figure 8 shows the ROIs that are removed. Many of the empty ROIs are removed and most of the filled ROIs are kept. However, when compared with Figure 5, one can see that there are also filled ROIs which are erroneously removed. E.g. on the left side of the image, ROIs with index $y = 0$ and $x = [3, 7]$ contain a cell but all but one get removed. On the flip side, ROIs such as the one at $(x = 14, y = 2)$ are completely empty but get kept anyways.

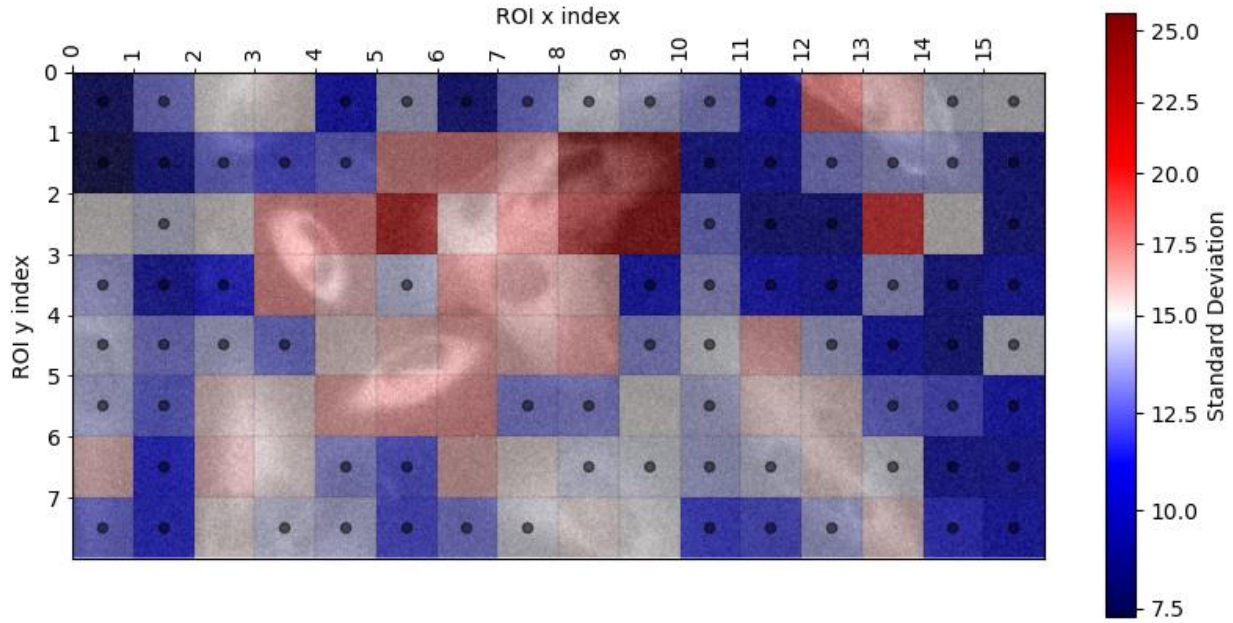

Figure 8 - The standard Deviations of the ROIs. ROIs marked with a black dot are below the threshold of 15 and get removed.

It is important to note that one can't simply fix this by adjusting the STD threshold. If raised, the empty ROIs will get filtered out, but even more ROIs with minor signals that are not strong enough will get filtered out as well. If lowered, more filled ROIs will be kept but also more empty ROIs will be considered for processing. Thus, the experimenter must find the threshold with the best balance between keeping too many empty ROIs and removing too many filled ROIs.

It's also important to note that the erroneously removed ROIs do contain cells which are active. E.g., the cell on the left mentioned before, while not the most active, does fire ~5 times throughout the recording. In this case, the empty ROIs' signals have a higher STD than the filled ROIs'. The reason seems to be that the background contains some pixels which are quite active, and these represent the signal of the ROI because the maximum amplitude pixel is selected.

Clustering ROIs

After removing the empty ROIs, a list of mostly filled ROIs and their detrended signals is left. Based on these signals, it is computed which cell each ROI belongs to. Since cells only have a limited size, ROIs that are too far apart shouldn't be clustered together. To achieve this, the location of the ROIs can also be factored in.

For this project, two different clustering paradigms were tried: hierarchical clustering with agglomerative clustering and partitioning clustering with K-Means. These will be discussed in the sections below.

Hierarchical Clustering (Agglomerative Clustering)

Computing Proximity

There is an issue with Euclidian distance in this context. Two signals that are equivalent in all but amplitude will have a non-zero Euclidian distance. This might be problematic because a cell that fires may light up more in a certain area which leads to a higher amplitude. Thus, two ROIs covering the same cell might have a similar signal but with a different amplitude. The Euclidian distance will be high, even though the ROIs belong to the same cell. For this reason, the cosine distance was chosen. It doesn't take the amplitude into account.

After the computation, a signal dissimilarity for each pair of ROIs is available. However, ROIs that can't belong to the same cell because they are far apart spatially can still have a low dissimilarity if their signals are similar. Again, this is undesirable because they will be clustered together, even though they can't belong to the same cell.

To avoid this, the dissimilarity is set to the maximum for ROIs not within spatial range. This spatial range is based on the size of the largest cell in the recording, which the experimenter can determine by looking at the image. This way the distance is maximal for ROIs that are too far apart, and they are unlikely to be merged quickly.

However, this strategy has a major drawback: it will have no effect whatsoever on smaller cells that are close together and have a similar signal. This is because the spatial distance will be below the spatial range. For this reason, the clustering actually performs much better if the spatial range is much lower than the actual maximum cell size (around 20% of it). Additionally, this method is quite complicated, which makes it harder to debug and interpret.

Clustering

Next, the algorithm is executed based on a maximum number of clusters, which should equal the number of cells in the image. By specifying this, the algorithm returns a cluster association for each ROI. It can be visualized by plotting it on top of the image, as in Figure 9.

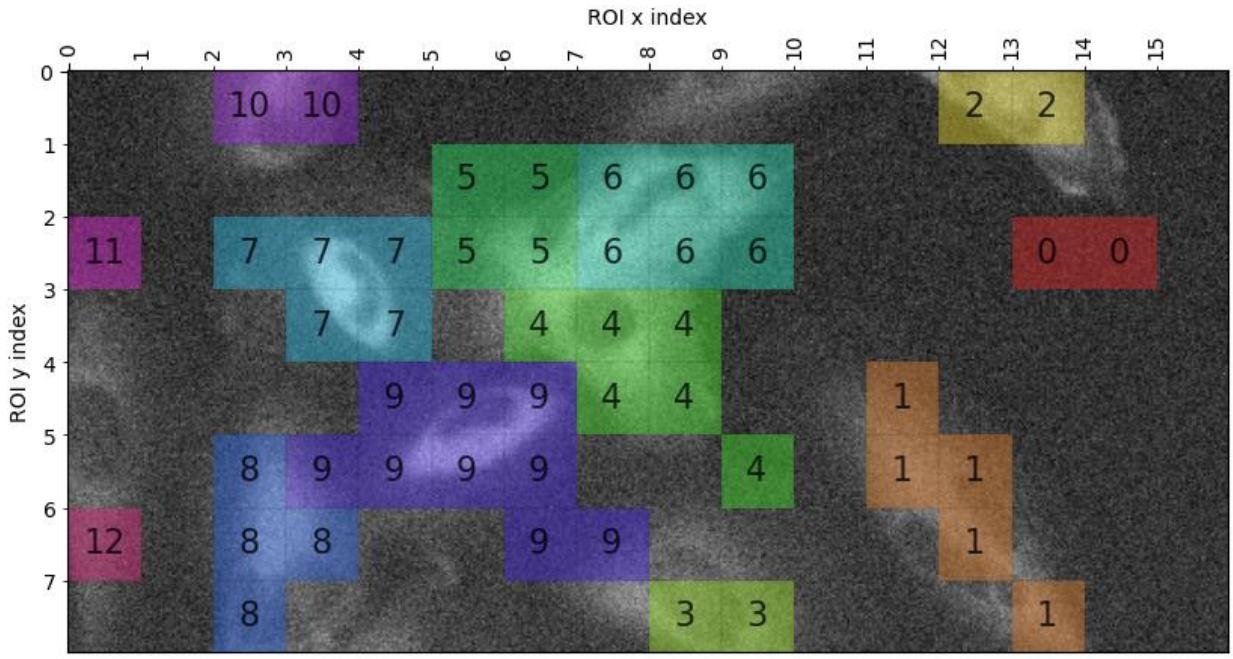

Figure 9 - ROI-cluster associations: which cluster each ROI belongs to

Almost all ROIs are clustered correctly. However, there are two types of errors here:

1. Clusters 0 and 11 consist of empty ROIs and some other clusters (e.g. 4) contain empty ROIs. However, this is an issue with filtering the ROIs rather than clustering.
2. Some clusters (e.g. cluster 9) are too big, covering multiple cells. This is even more prominent in many other images. Vice versa, clusters 8 and 3 are too small – not covering the whole cells.

Larger images with more cells result in the same issues.

Partition Clustering (K-Means)

Basic K-Means

Figure 10 shows that basic K-Means, which doesn't account for location, performs terribly. This is because the ROI's spatial location, which is a valuable information, isn't taken into account at all and the signals' Euclidian distance can be high, even if the signals are from the same cell, as discussed in the section *Computing Proximity*.

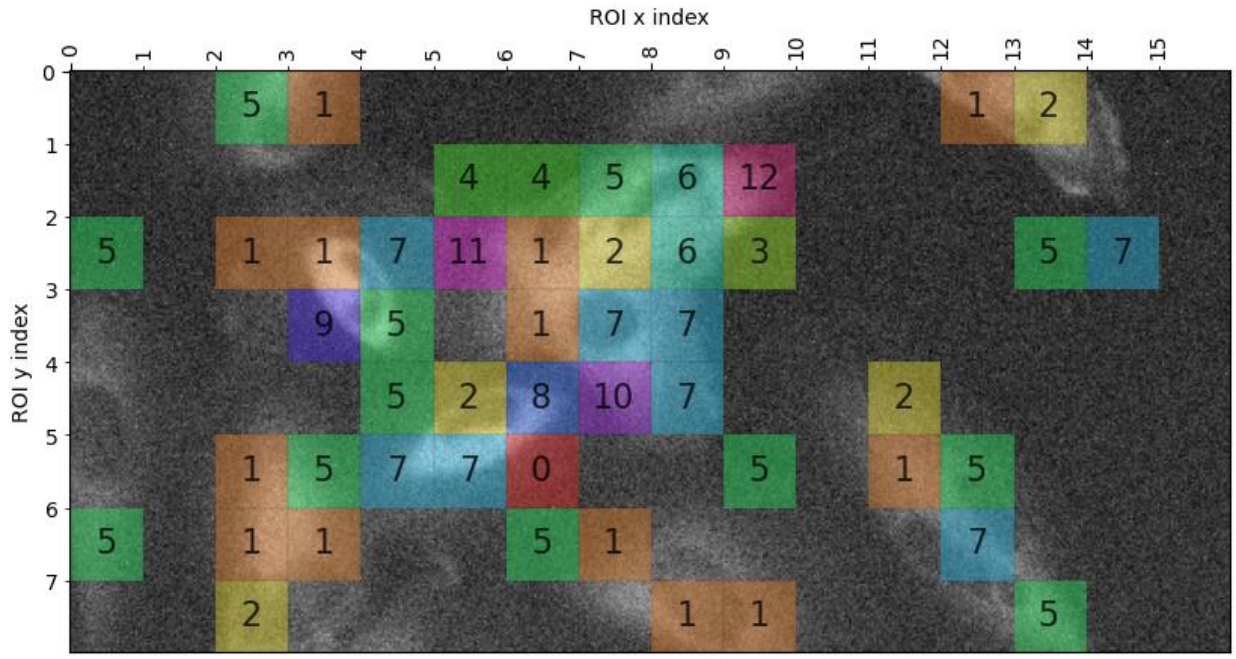

Figure 10 - Clusters of K-Means just based on detrended signals

Spatial, Weighted K-Means

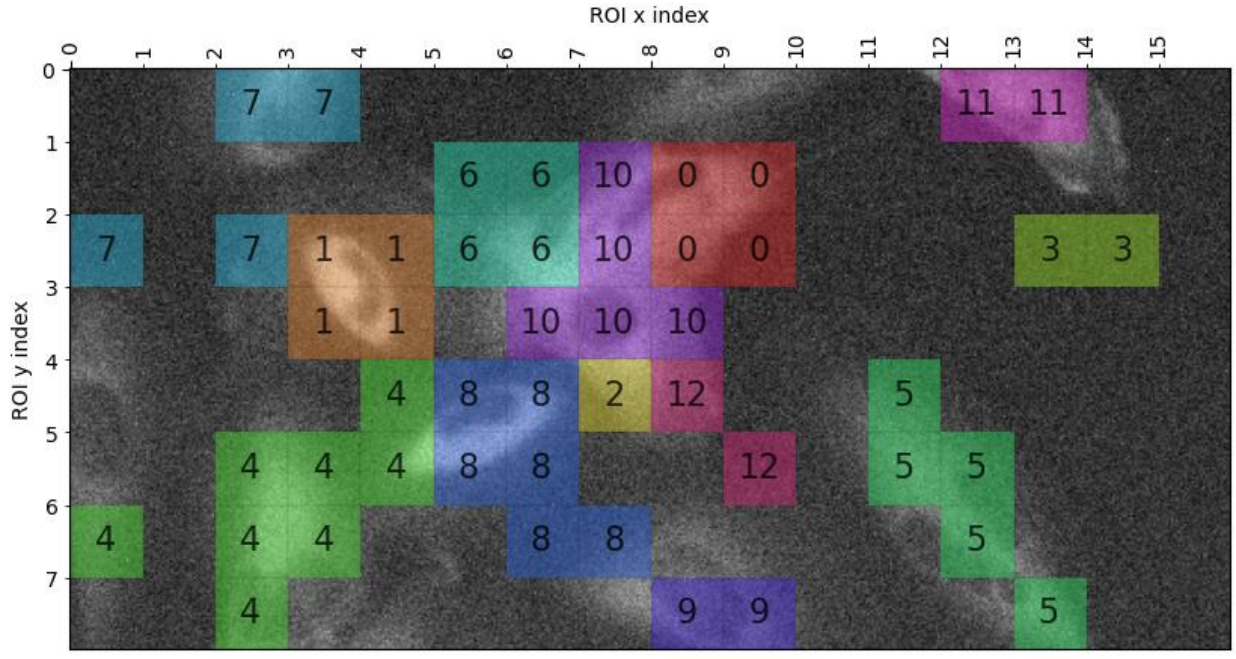

Figure 11 - Clustering Results with spatial, weighted K-Means with spatial weight w=1.5

In Figure 11 shows that spatial, weighted K-Means performs much better than basic K-Means. For this recording, K-Means performs slightly worse than Agglomerative clustering, however, in general their performance is similar.

Extracting Signals from Clusters

Figure 12 shows that selecting the signal with the highest peak tends to select a representative signal that covers the cell. In some cases, e.g. cluster 6, a ROI that is hardly filled with a cell is picked. However, this isn't necessarily wrong if the cell is active around its edge. In this recording, the cell covered by cluster 6 was in fact highly active in its lower right corner, where the signal was selected from.

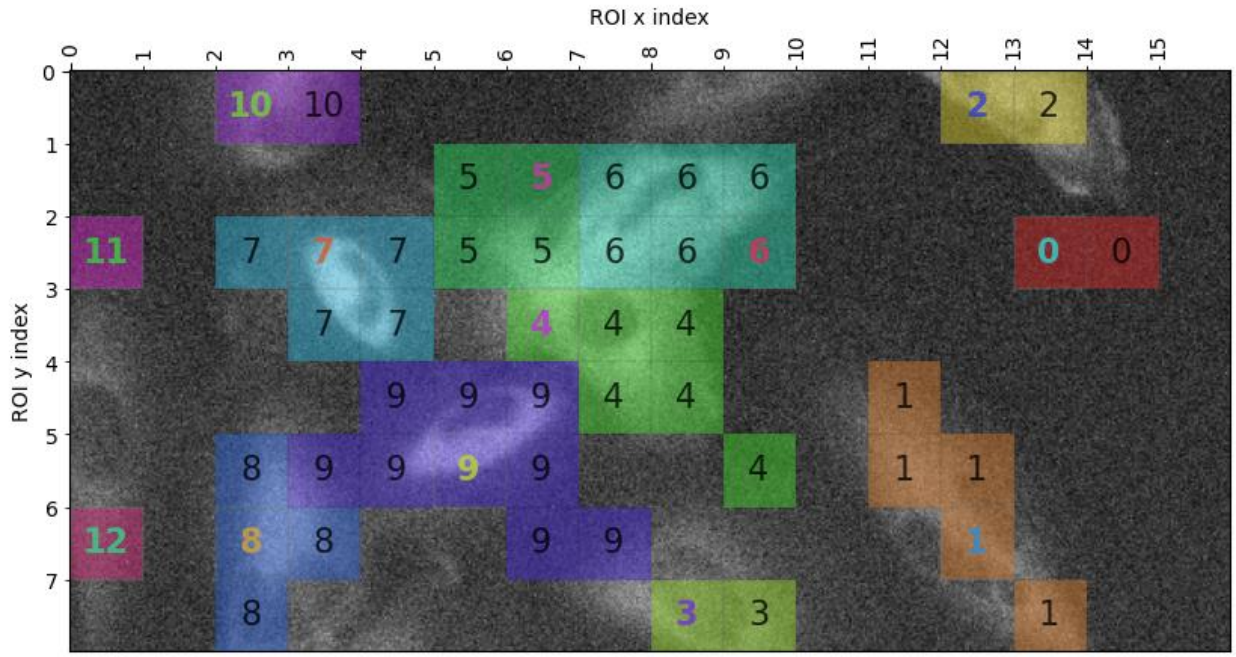

Figure 12- Representative Signals: the ROIs whose signals represent their cluster are highlighted.

What happens occasionally is that clusters which are mostly covering a cell but also contain some empty ROIs select a signal which is from the empty ROIs (Figure 13). This appears to happen because the cell's signal is so weak, thus it is up to the experimenter to decide whether it's a problem that the very low-activity cells aren't recognized.

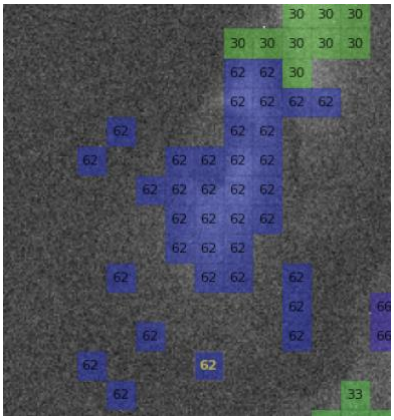

Figure 13 - Representative Signal comes from empty space rather than from cell.

Figure 14 shows the final signals of each cluster, which can be used for further analysis. Optimally, these should represent the detrended signals for each cell. If so desired, it would also be possible to perform the clustering and/or the selection of the representative signal based on the original ROI signals by just changing a few lines of code. That way, one would get the original signals for further analysis.

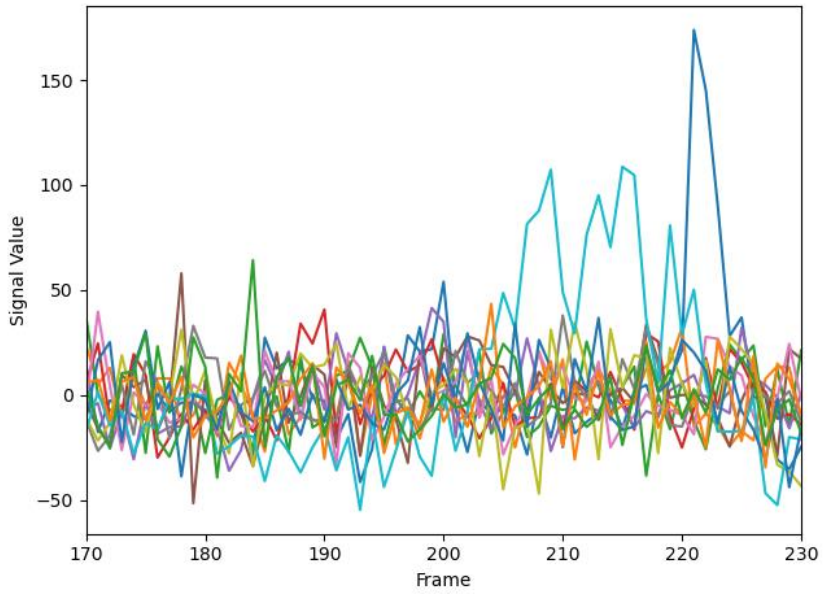

Figure 14 - Signal of each cluster

Key Findings

It has been demonstrated that performing the steps of creating ROIs, extracting and detrending ROI signals, removing the empty ROIs, clustering them, and finally extracting a signal from each cluster produces the desired format of output – a signal for each cluster/cell. Some of the steps can still be improved, to obtain a better result. These are:

1. The **removal of empty ROIs,** where some empty ROIs are kept and filled ROIs are removed,
2. The **clustering** of ROIs, where some clusters contain more, and some contain less than one cell
3. The selection of a **representative signal** for each cluster. Here, ROIs that are not within the cell are selected sometimes. However, this would also be fixed if the ROI removal would be optimized.

Discussion

Removing Empty ROIs

A major issue with the current state of the algorithm is that ROIs which contain active cells are removed while empty ROIs are being kept, even with an optimal STD threshold. This is problematic because these ROIs interfere with the clustering algorithm. It would be beneficial to improve this.

A promising approach would be to filter out pixels instead of ROIs. This would mean detrending each pixel's value and ignoring pixels which are below a certain threshold. If it works, regions with a high density of filtered pixels should correspond to an active cell. Isolated pixels would correspond with imaging artifacts. This way, single pixels with a high variance signal shouldn't impact the end results.

Signal Comparisons

Figure 10 shows that clustering based solely on the signal performs very poorly. This is somewhat surprising, as one would expect the signals of ROIs overlapping with the same cell to be quite similar. Below, different reasons for this are discussed.

Shifted Signals

One reason for this phenomenon is surely that it takes time for a signal to propagate through the cell. This means that two ROIs on opposite sides of the cell may have a signal which is similar in principle but shifted in time. Figure 15 shows that when this is compared with a distance measure, the resulting distance will be quite high, thus leading to the ROIs not being clustered together. As the speed at which signals propagate through cells varies, this will especially affect cells where propagation is slow.

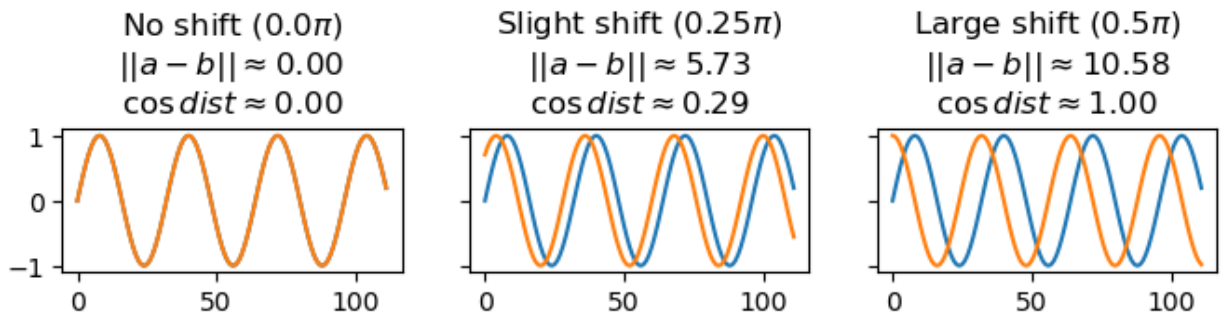

Figure 15 – Distances of shifted Signals

Use of Raw vs. Detrended Signals

Additionally, the signal detrending procedure somewhat distorts the raw signal beyond just removing the luminosity trend. Using the detrended signal is essential for filtering but it's currently also used in the further analysis. It may be worth trying to use the raw ROI signals for clustering and see if this improves the results.

Signal Amplitude

As discussed in the section *Adjusting for Signal Amplitude*, a difference in amplitude for an otherwise equivalent signal leads to a high Euclidian distance. Since the K-Means algorithm uses Euclidian distance, it may be better to normalize each signal with its own maximum value $p^{(a)}$, rather than with the overall maximum value p. This way, areas where a cell lights up less, would be more similar to the area where it lights up more. However, this distorts the signal further and might have other drawbacks. For example, cells that have a higher amplitude in general would be more similar to cells with a lower amplitude.

ROI Signal Selection

In the scope of this project, it has not been thoroughly investigated which exact pixel each ROI is getting its signal from. Usually, a few dispersed, singular pixels vary strongly from their neighbors. It would be important to investigate this further, see why they behave so differently, whether the ROI signal is selected from these pixels, which likely do not reflect the calcium activity of the cell, and investigate how this affects the downstream steps.

Conclusion

In this project, an automatic method to extract the calcium signaling activity from GBM cells from a live-cell recording was developed. This is highly useful to streamline processes, speed up research, and provide a method which is not biased by the human performing it.

The method works based on dividing the image into many regions of interest, computing a signal for each of these, detrending that signal, removing empty ROIs and then clustering the remaining ROIs based on the signal similarity and their location. Each ROI that is covering an individual cell should have highly similar signaling dynamics and should thus be clustered together. Finally, the most representative ROI of each cluster/cell is chosen for further analysis.

In its current state, the method can correctly extract the signals of many of the cells in a recording. While it still makes many mistakes, it is highly likely that it can be improved by improving the following steps. A better filtering of the ROIs might be achieved by filtering based on the pixels, rather than the ROIs and only including regions where the density of filtered pixels is high. A better clustering might be achieved by normalizing each ROI's signal with its own – rather than the overall – maximum signal value.

A limitation of this approach of comparing signals, with regards to the clustering may be the temporal shift in signals due to the time it takes for it to travel through the cell. This seems highly difficult to account for without knowledge about which cells belong to which signals.

Acknowledgements

I would like to thank Chris and Vatsal for supporting me in this project and answering all my questions. I would like to thank Kevin and Vidhya for letting me participate in their NeuroEngineering lab and using the facilities.

References

Everitt, B., Landau, S., Leese, M., & Stahl, D. (2011). *Cluster Analysis* (5th ed.). Wiley.

Hausmann, D., Hoffmann, D. C., Venkataramani, V., Jung, E., Horschitz, S., Tetzlaff, S. K., Jabali, A., Hai, L., Kessler, T., Azoŕin, D. D., Weil, S., Kourtesakis, A., Sievers, P., Habel, A., Breckwoldt, M. O., Karreman, M. A., Ratliff, M., Messmer, J. M., Yang, Y., … Winkler, F. (2023). Autonomous rhythmic activity in glioma networks drives brain tumour growth. *Nature*, *613*(7942), 179–186. https://doi.org/10.1038/s41586-022-05520-4

Leclerc, C., Haeich, J., Aulestia, F. J., Kilhoffer, M.-C., Miller, A. L., Néant, I., Webb, S. E., Schaeffer, E., Junier, M.-P., Chneiweiss, H., & Moreau, M. (2016). Calcium signaling orchestrates glioblastoma development: Facts and conjunctures. *Biochimica et Biophysica Acta*, *1863*(6 Pt B), 1447–1459. https://doi.org/10.1016/j.bbamcr.2016.01.018

Mayo Clinic. (2024, March 7). *Glioblastoma - Overview*. https://www.mayoclinic.org/diseases-conditions/glioblastoma/cdc-20350148

Miyawaki, A., Llopis, J., Heim, R., McCaffery, J. M., Adams, J. A., Ikura, M., & Tsien, R. Y. (1997). Fluorescent indicators for Ca2+ based on green fluorescent proteins and calmodulin. *Nature*, *388*(6645), 882–887. https://doi.org/10.1038/42264

Monteith, G. R., Davis, F. M., & Roberts-Thomson, S. J. (2012). Calcium channels and pumps in cancer: changes and consequences. *The Journal of Biological Chemistry*, *287*(38), 31666–31673. https://doi.org/10.1074/jbc.R112.343061

Monteith, G. R., Prevarskaya, N., & Roberts-Thomson, S. J. (2017). The calcium-cancer signalling nexus. *Nature Reviews. Cancer*, *17*(6), 367–380. https://doi.org/10.1038/nrc.2017.18

Osswald, M., Jung, E., Sahm, F., Solecki, G., Venkataramani, V., Blaes, J., Weil, S., Horstmann, H., Wiestler, B., Syed, M., Huang, L., Ratliff, M., Karimian Jazi, K., Kurz, F. T., Schmenger, T., Lemke, D., Gömmel, M., Pauli, M., Liao, Y., … Winkler, F. (2015). Brain tumour cells interconnect to a functional and resistant network. *Nature*, *528*(7580), 93–98. https://doi.org/10.1038/nature16071

Prakash, A. (2023, September 21). *Unlocking the Power of Cosine Similarity: A Comprehensive Guide to Understanding and Using this Essential Data Science Metric*.

Medium.Com. https://medium.com/@arjunprakash027/understanding-cosine-similarity-a-key-concept-in-data-science-72a0fcc57599

Robil, N., Petel, F., Kilhoffer, M.-C., & Haiech, J. (2015). Glioblastoma and calcium signaling--analysis of calcium toolbox expression. *The International Journal of Developmental Biology*, *59*(7–9), 407–415. https://doi.org/10.1387/ijdb.150200jh

Stewart, T. A., Yapa, K. T. D. S., & Monteith, G. R. (2015). Altered calcium signaling in cancer cells. *Biochimica et Biophysica Acta*, *1848*(10 Pt B), 2502–2511. https://doi.org/10.1016/j.bbamem.2014.08.016

Winkler, F., & Wick, W. (2018). Harmful networks in the brain and beyond. *Science*, *359*(6380), 1100–1101. https://doi.org/10.1126/science.aar5555