Correlation Analysis of Calcium Signalling Networks in Living Cells

Erik Nilsson

January 2009-01-16

Master of Science Thesis in Engineering Physics at KTH
Supervisor: Docent Per Uhlén Karolinska Institutet (KI)
Examiner: Professor Hjalmar Brismar KTH/KI

TRITA-FYS-2009:01    ISSN 0280-316X    ISRN KTH/FYS/-09:01-SE
To Anna

Nothing is worth more than this day.

Johann Wolfgang von Goethe
Abstract

In living cells, calcium ions (Ca\textsuperscript{2+}) play an important role as an intracellular second messenger. It mediates the regulation of cellular processes such as gene expression, initiation of vesicle fusion in synapses, is used in muscle contraction and is believed to play a fundamental role in synaptic plasticity as a molecular substrate for learning. The Ca\textsuperscript{2+} signals are created by the fact that the concentration of Ca\textsuperscript{2+} in the cytosol is four orders of magnitude lower than in the extracellular fluid as well as in cytoplasmic compartments such as the endoplasmic reticulum (ER). This enables fast increments in the cytosol concentration, which is regulated back to normal concentration by different mechanisms. In this project, the connection between Ca\textsuperscript{2+} signals of different cells was analysed using different correlation techniques: cross-correlation of continuous signals and digitalised signals. Therefore a software tool was developed in MATLAB, which takes Ca\textsuperscript{2+} recordings from time-lapse fluorescence microscopy as input and calculates the pairwise correlation for all cells. The software was tested by using previous data from experiments with embryonic stem cells from mouse (mES) and human (hES) as well as data from recordings done as part of the project.

The study shows that the mathematical method of cross-correlation can successfully be applied to quantitative and qualitative analysis of Ca\textsuperscript{2+} signals. Furthermore, there exist strongly correlated cells in colonies of mES cells and hES cells. We suggest the synchronisation is achieved by physical coupling implicating a decrease of correlation as the distance increases for strong correlations. In addition, the lag used by the cross-correlation function (an effective phase shift) decreases as the correlation coefficient increases and increases as the intercellular distance increases for high correlation coefficients. Interestingly, the number of cells included in small scale clusters of strongly correlated cells is significantly larger for differentiating mES cells than for proliferating mES cells. In a broader perspective, the developed software might be used in for instance analysis of cellular electrical activity and shows the relevance of applying methods from the exact sciences to biology.
Sammanfattning

## CONTENTS

1  **INTRODUCTION** ........................................................................................................ 1
   1.1  **AIM** ...................................................................................................................... 1
   1.2  **BACKGROUND** .................................................................................................... 1
      1.2.1  **Biology** .......................................................................................................... 1
      1.2.2  **Mathematics** .................................................................................................. 4
      1.2.3  **Imaging principles** .......................................................................................... 8
2  **MATERIALS AND METHODS** .................................................................................... 11
   2.1  **COMPUTER SOFTWARE** ...................................................................................... 11
   2.2  **EXPERIMENTS** .................................................................................................... 11
      2.2.1  **Calcium recordings** ........................................................................................... 11
      2.2.2  **Previous experiments with stem cells from mouse** .................................................. 11
      2.2.3  **Previous experiments with stem cells from human** ................................................ 12
      2.2.4  **Experiments carried through as a part of the project** .............................................. 12
3  **RESULTS** .................................................................................................................... 13
   3.1  **ANALYSIS OF PREVIOUS STUDIES** .................................................................... 13
   3.2  **THE SOFTWARE TOOL** ......................................................................................... 13
      3.2.1  **Correlation methods** .......................................................................................... 15
      3.2.2  **Visualisation of correlations** .............................................................................. 18
   3.3  **ANALYSIS OF PREVIOUS CALCIUM RECORDINGS** .................................................. 20
      3.3.1  **Differentiating stem cells from mouse** ................................................................. 20
      3.3.2  **Proliferating stem cells from mouse** ..................................................................... 23
      3.3.3  **Human stem cells** ............................................................................................ 27
   3.4  **ANALYSIS OF EXPERIMENTS CARRIED THROUGH AS A PART OF THE PROJECT** ...... 28
4  **DISCUSSION** ................................................................................................................ 30
   4.1  **THE DEVELOPED SOFTWARE** ............................................................................... 30
   4.2  **FILTERING ASPECTS - THE CUT OFF** ................................................................... 30
   4.3  **GENERAL PHYSICAL PRINCIPLES** ...................................................................... 32
5  **CONCLUSION** ............................................................................................................... 33
6  **FUTURE PERSPECTIVES** .............................................................................................. 34
   6.1  **TECHNICAL IMPROVEMENTS OF THE SOFTWARE** ............................................... 34
   6.2  **NOVEL AREAS OF USE** ....................................................................................... 34
7  **ACKNOWLEDGMENT** .................................................................................................... 35
8  **APPENDIX** ................................................................................................................... 36
9  **REFERENCES** ............................................................................................................... 60
1 Introduction

1.1 Aim

The aim of this project is to develop a tool for computer based correlation analysis of calcium signalling cells. In order to do this, a number of correlation methods are implemented. Using the tool, networks of Ca$^{2+}$ signalling cells are studied and analysed. The data is recorded in real experiments with different types of living cells using time-laps fluorescence microscopy. Previous data is used as well as data recorded as a part of the project.

1.2 Background

1.2.1 Biology

The human body, as well as all living organisms, is constituted by cells. The life processes of the cells are highly dependent on predictable and stable conditions and uses different control systems to regulate the conditions within the cell membrane. This phenomenon is called homeostasis and includes the regulation of ion concentration. Keeping different parameters around steady state enables signalling pathways by shifting the levels from equilibrium. For example the physiological important ions potassium (K$^+$), sodium (Na$^+$), calcium (Ca$^{2+}$) and chlorine (Cl$^-$) are kept at certain resting concentrations in the cell. This in turn implies a stable electrical resting potential according to the Goldman Hodgkin Katz equation. By shifting the conductance of the cell membrane for different ions (first Na$^+$ and then K$^+$ in the Hodgkin-Huxley model of the squid), an electrical impulse called an action potential is generated. Following this, regulating proteins called ion pumps restore the ion concentrations and equilibrium is reached. The electrical action potential is an intercellular signal only occurring in the nervous system. There is also a large number of intracellular signalling pathways, where for example proteins called transcription factors regulate gene expression as a final result to ligands (stimulating molecules) binding to receptors on the cell membrane. An especially important second messenger is Ca$^{2+}$ that controls almost everything that we do. During fertilisation, mammalian eggs generate low frequency Ca$^{2+}$ signals to initiate development including cell cleavage and nuclear envelope breakdown. This is followed by other Ca$^{2+}$ signals during embryonic pattern formation, where it is believed that Ca$^{2+}$ act during the specification of the dorsoventral axis. At a later stage in the development, Ca$^{2+}$ is used in the process of differentiation of cells, for example in nerve and muscle cells. Being differentiated, most cells do not divide and proliferate. Some cells though, maintain the
possibility to enter the cell cycle in response to for instance growth factors. \( \text{Ca}^{2+} \) is one of the most important regulators of proliferation. The main role of \( \text{Ca}^{2+} \) in the process of proliferation is to activate transcription factors in the cytoplasm and the nucleus. At last, \( \text{Ca}^{2+} \) is involved in apoptosis (programmed cell death).

The generation of \( \text{Ca}^{2+} \) signals is enabled by the fact that the cytosolic concentration of \( \text{Ca}^{2+} \) is four orders of magnitude smaller than the extracellular concentration. In addition, the \( \text{Ca}^{2+} \) concentration in intracellular compartments such as the endoplasmic reticulum, is in the same order of magnitude as the extracellular concentration. The process of \( \text{Ca}^{2+} \) signalling can be divided into four parts:

1) Triggering by a stimulus, for example a ligand. Concentration around 100 nM.
2) Influx of \( \text{Ca}^{2+} \) into the cytoplasm. Concentration around 500-1000 nM.
3) Signal transduction, where the \( \text{Ca}^{2+} \) signal is translated into physiological changes such as gene expression.
4) The removal of \( \text{Ca}^{2+} \) from the cytosol by pumps and exchangers.

Figure 1 (by Per Uhlén) shows a schematic simplified picture of the regulation of \( \text{Ca}^{2+} \). The red part of the signal is the rising part, corresponding to number 1 and 2 in the list above. The blue part on the other hand corresponds to number 3 and 4 in the list and constitutes the falling part. What happens in the red part is that some stimulating molecule binds to the G-protein coupled receptor (GPCR), which activates phospholipase C (PLC) that produces inostol-triphosphate (InsP\(_3\)). InsP\(_3\) diffuses through the cytosol and activates the InsP\(_3\) receptor (InsP\(_3\)R) in the endoplasmic reticulum (ER) or the sarcoplasmic reticulum (SR). The result of binding to the InsP\(_3\)R is dependent on the \( \text{Ca}^{2+} \) concentration. In order to open, \( \text{Ca}^{2+} \) is needed, leading to positive feedback. In some cell types, the InsP\(_3\)R is exchanged for ryanodine receptors (RYR). \( \text{Ca}^{2+} \) is also flowing into the cytosol from the extracellular fluid through voltage and receptor operated channels (VOC and ROC). When the internal stores are empty with \( \text{Ca}^{2+} \), so called store-operated channels (SOC) are opened.

What happens in the blue part is that \( \text{Ca}^{2+} \) pumps in the plasma membrane driven by ATP (PMCA) and in the SR/ER (SERCA) start to pump \( \text{Ca}^{2+} \) out from the cytosol and thereby re-establish the equilibrium concentration.

The process above may happen only once, which constitutes a transient called a \( \text{Ca}^{2+} \) spike. If the process is repeated in the time plane, a \( \text{Ca}^{2+} \) oscillation is induced. Repeating the rise and fall in space, a \( \text{Ca}^{2+} \) wave is created. The waves are created when \( \text{Ca}^{2+} \) diffuses to areas
with lower concentration, which excites the InsP3R and thereby increases the local concentration. The intracellular Ca\(^{2+}\) can propagate further to neighbouring cells through gap junctions and thereby induce intercellular waves. Gap junctions are intercellular channels that allow direct transfer of electrical current and small molecules (<1 kDa) between adjacent cells. There are at least 20 genes coding for gap junctions and Connexin43 is one of the most studied. It is not clear exactly how the signals are transmitted through the gap junctions. It might be by Ca\(^{2+}\) itself or by InsP3.\(^5\) The physiological relevance of these waves has been shown in for example lung epithelial cells, where the waves stimulate the beat frequency of cilia that remove inhaled contaminants.\(^6\)

![Figure 1](image.png)

**Figure 1** Schematic simplified picture of Ca\(^{2+}\) regulation. The red parts belong to the rising phase and the blue parts to the falling phase. © Per Uhlén 2008.

The Ca\(^{2+}\) signals transmit information that is to be decoded by the cell. In electrical intercellular neuronal transmission with action potentials, the signals are frequency modulated and the amplitude constant. The Ca\(^{2+}\) signals on the other hand can be FM as well as AM and everything in between. For instance differential gene transcription in B lymphocytes (the antibody-producing cells of the immune system) has been shown to be amplitude dependent.
Concerning the frequency, it has for example been shown that the ubiquitous Ca\(^{2+}\) calmodulin kinase II (CaMKII; an enzyme believed to play a critical role in long-term potentiation and thereby memory as well\(^8\)) is highly frequency dependent.\(^9\)

Ca\(^{2+}\) signalling is utilised in probably all cell types, and a previous study\(^10\) shows that Ca\(^{2+}\) oscillations are of great importance in the regulation of proliferation in mouse embryonic stem cells. Stem cells are by definition cells with the ability to go through numerous cycles of cell division remaining in the undifferentiated state and with the capacity to differentiate to any specialised cell type. Progenitors are cells that just like stem cells have the ability to differentiate into a specific cell type, but that on the other hand is forced to differentiate to a special cell type (not totipotent).

### 1.2.2 Mathematics

#### 1.2.2.1 Stochastic processes and correlation

A time series is a set of data points \(\{X_t, t \in T\}\) with corresponding time points \(t\), at which the data was recorded. The set \(T\) can either be discret or continuous. An example of a discret time series is the strength of fluorescence due to calcium concentration measured at distinct time points. In order to analyse such a time series, it is advantageous to implement a stochastic model, since the future values are not fully predictable. This leads to the definition of a multidimensional random variable and a stochastic process.

**Definition 1**

A real n-dimensional random variable is a real-valued function \((X_1, X_2, \ldots, X_n)\) defined on a sample space and mapping \(\mathbb{R}^n\).

**Definition 2**

A stochastic process is a family of random variables \(\{X_t, t \in T\}\) on a probability space.\(^11\)

At every single time point \(t_j\), the stochastic process is a random variable \(X(t_j)\). Therefore, the stochastic process can be seen as an n-dimensional random variable for time points \(\{t_1, t_2, \ldots, t_n\}\). In this model, a time series is seen as a realisation of a stochastic process.

**Definition 3**
The functions \( \{ X(\omega), \omega \in \Omega \} \) on \( T \) are known as the realisations of the stochastic process \( \{ X, t \in T \} \). The set of all possible realisations is called the ensemble of the process.

**Example 1**
An example of a stochastic process is a sinus wave with random phase, whereas all other parameters are deterministic. \( X(\Phi, t) = a \sin(2\pi ft + \Phi) \), where \( \Phi \) is a random variable. A realisation of this process is \( x_i(t) = a \sin(2\pi ft + \varphi_i) \). The set of all possible realisations is the ensemble of \( X(\Phi,t) \). Fixing the time on the other hand results in a random variable \( X(t_j) \). Finally, fixing the time as well as the realisation results in a real number \( X(\varphi_i,t_j) = x_i(t_j) \).

In order to characterise a stochastic process, its expectation value is important.

**Definition 4**
For a continuous stochastic process, its expectation value is defined as \( m_X(t) = E(X(t)) \)

\[
= \int_{-\infty}^{\infty} x f_X(x) \, dx , \text{ where } f_X(x) \text{ is the probability density function of } X(t).
\]

Now consider a stochastic process \( X(t) \). To determine the correlation between two time points in the process, the autocorrelation function is calculated.

**Definition 5**
The autocorrelation function of a stochastic process is defined as \( r_X(t_1,t_2) = E[X(t_1)X(t_2)] \). Similarly, the covariance function is defined as \( \text{Cov}_X(t_1,t_2) = E[(X(t_1) - m_X(t_1))(X(t_2) - m_X(t_2))] \). A normalised version of the covariance function is the correlation function \( \rho_X(t_1,t_2) = \frac{\text{Cov}_X(t_1,t_2)}{\sigma_X(t_1)\sigma_X(t_2)} \). Here, \( \sigma_X(t_1) \) and \( \sigma_X(t_2) \) are the square root of \( \text{Cov}_X(t_1,t_1) \) and \( \text{Cov}_X(t_2,t_2) \). Fixing the values of \( t_1 \) and \( t_2 \) results in the well known correlation coefficient (also known as the Pearson correlation).

Now suppose there are two stochastic processes and one would like to determine the correlation between them. This can be done by exchanging \( X \) for \( Y \) in **Definition 5**.
**Definition 6**

The cross-correlation function is defined as \( r_{YX}(t_1,t_2) = E[Y(t_1)X(t_2)] \) and the cross-covariance function as \( \text{Cov}_{YX}(t_1,t_2) = E[(Y(t_1) - m_Y(t_1))(X(t_2) - m_X(t_2))] \). The normalised version of the cross-covariance function is \( \rho_{YX}(t_1,t_2) = \frac{\text{Cov}_{YX}(t_1,t_2)}{\sigma_Y(t_1)\sigma_X(t_2)} \). Again, fixing \( t_1 \) and \( t_2 \) results in the correlation coefficient.\(^\text{12}\)

It should be noted that the cross-correlation function can be defined for two functions as well and seen as a variant of convolution. Seeing it this way, the process of cross-correlation is to find the lag where the overlap of the two functions is the greatest.\(^\text{13}\)

The correlation coefficient defined above is, as a sequel to the normalisation, a real number between -1 and 1. It is a measure of the strength of linear relation between two variables, for instance the strength of fluorescence due to \( \text{Ca}^{2+} \) concentration in two different cells at some time points. A correlation coefficient equal to zero means there is no linear relation between the variables, whereas a coefficient equal to -1 or 1 means the relation is perfect. The sign contains information about the direction of the relation. A negative coefficient means the second variable decreases as the first one increases and vice versa. It is very important to understand that the correlation coefficient is only a measure of the strength of linear dependence and hence not non-linear.\(^\text{14}\) Furthermore, correlation does not imply causation, in the same way as two independent random variables are not correlated, even though they are not necessarily independent if they are uncorrelated.\(^\text{15}\)

Stationary stochastic processes belong to a special class of processes in which the expectation value is constant and the autocorrelation function is only dependent on the distance between two time points.

**Definition 7**

A continuous stochastic process \( X(t) \) is stationary if the following is fulfilled.

\[
\begin{align*}
E(X(t)) &= m_X \text{ constant} \\
E[X(t+\tau)X(t)] &= r_X(\tau) \text{ independent of } t
\end{align*}
\]

NB: the above definition is sometimes used for weak stationary stochastic processes.

Another important notion is that of ergodicity, which is defined in **Definition 8**.
Definition 8

A stationary stochastic process is said to be ergodic if all its statistical properties can be inferred from a single realisation of the process.

The notions of stationarity and ergodicity are very important, since they can often be applied to real models. Modelling a time series as a non-ergodic process would demand a large number of realisations to find its statistical properties. It is often impossible to record such a large number of realisations and for example a cell cannot be loaded with dyes and recorded repeatedly. Therefore, one often assumes that a given time series is a realisation of a stationary and ergodic stochastic process.\textsuperscript{16} If the data has a trend or season component, these can often be removed, yielding a new time series that can be seen as a realisation of a stationary and ergodic process.\textsuperscript{17}

Being presented to a time series that can be modelled as a realisation of an ergodic and stationary stochastic process, it is possible to estimate the expectation value, autocorrelation function etc. in a good way. The expectation value is estimated as the sample mean value.

$$\hat{m}_x = \frac{1}{N} \sum_{n=0}^{N-1} x(n)$$

The autocorrelation function can be estimated either as

$$\hat{r}_{x,x}(k) = \frac{1}{N} \sum_{n=0}^{N-k-1} x(n+k)x(n)$$

or as

$$\hat{r}_{x,x}(k) = \frac{1}{N-k} \sum_{n=0}^{N-k-1} x(n+k)x(n)$$

The former is called biased and the latter unbiased, since the unbiased estimation yields the correct expectation value and the biased do not when the expectation values of their stochastic variables are calculated. The cross-covariance and cross-correlation functions are estimated in a similar fashion.\textsuperscript{18}

In this project, the numerical computing environment and programming language MATLAB (Mathworks, Natick, MA, USA) is used. MATLAB estimates the cross-covariance function (here called $xcov$) as

$$c_{xy}(m) = \begin{cases} \sum_{n=0}^{N-1} x(n+m) - \frac{1}{N} \sum_{i=0}^{N-1} x_i \left( y_{n}^* - \frac{1}{N} \sum_{i=0}^{N-1} y_{n_i}^* \right), & m \geq 0 \\ c_{yx(-m)}, & m < 0 \end{cases}$$
The normalisation coefficient is chosen in order to regain the correlation coefficient when specifying the lag.\textsuperscript{19}

1.2.2.2 Statistical significance

An important question to consider is whether the correlation coefficients found between signals are statistically significant; that is there is an actual linear relationship. To test this, one establishes a null hypothesis implying a coefficient equal to a certain cut off. Normally, it is assumed that the cut off is zero, meaning there is no linear dependence at all. In general this might not be true, since studying cells with high level of similarities should imply a certain level of correlation. Furthermore, one can argue that a strong correlation coefficient should be in the interval $0.7 < r < 1$ as absolute value. A problem with the significance test is that the sampling distribution is rather complex compared to the distribution for e.g. the mean. The convergence of the sampling distribution towards the normal distribution is slow for increasing number of time points $N$. When the null hypothesis includes a correlation coefficient not equal to zero, a transform must be used in order to make the distribution approximately normal. The transform used is the so called Fisher transform.

$$Z = \frac{1}{2} \ln \left( \frac{1+r_{xy}}{1-r_{xy}} \right)$$

The variance of the $Z$ distribution is equal to $1/(N-3)$, and a $z$-statistics test can be carried through as

$$z = \frac{Z - Z(\rho)}{\sqrt{1/(N-3)}}$$

Where $Z(\rho)$ is the Fisher transform of the cut off. In this project, a confidence level of 95\% is implemented. This implies that in order to be significant larger than a certain cut off, the $z$ value above must be larger or less than 1.96 and -1.96 respectively.\textsuperscript{20}

1.2.3 Imaging principles

The concentration of Ca$^{2+}$ can be measured experimentally by loading cells with Ca$^{2+}$ sensitive fluorescent dyes such as Fluo3/AM and Fura2/AM. The dyes permeate the cell membrane and bind to Ca$^{2+}$ due to its high affinity and undergo a structural change that affects the cell membrane permeability, implying no loss of dye to the extracellular fluid. By challenging the loaded cells with light, the fluorescent molecules absorb photon energy and as a sequel to this emit light with longer wavelength from an excited electronic state. The loss of energy implied in the increase of wavelength is a so called Stokes shift and is due to loss of
vibrational energy. Fluo3/AM has an absorption wavelength at 506 nm and an emission wavelength at 526 nm.

The fluorescent light emitted by the fluorophore can be detected using either ordinary wide field microscopy or confocal microscopy. The common way is to use the latter, and that is also the case in this project. Confocal microscopy is in contrary to conventional microscopy a technique where only a small spot of the sample is imaged by using a pinhole to eliminate out of focus light. By using this imaging paradigm, a higher resolution is gained (the point spread function for confocal microscopy is the square of the point spread function for wide field microscopy) as well as the possibility to optically section the sample and to construct three dimensional images. See Figure 2 below by Kjell Carlsson. Here the beam splitter reflects light with the excitation wavelength and transmits light with the fluorescence wavelength.

The imaging method described above is limited in quality by the laws of physics. First of all, so called bleaching is a problem where fluorescent molecules undergo conformational changes resulting in the destruction of the fluorophore by the excitation light. Another difficulty is stated in the sampling theorem. This says that: “In order to be able to correctly recreate an image with (spatial or temporal) frequencies up to \( \nu_{\text{max}} \), sampling must be carried out at a frequency of at least 2\( \nu_{\text{max}} \) (that is, two sampling points per period)”. Following this, in order to faithfully record a \( \text{Ca}^{2+} \) oscillation of a given frequency, there is a quantitative

![Figure 2](image-url)
demand on the sampling frequency that must be fulfilled. Optimally, one would like to take pictures infinitely often resulting in an infinite sampling frequency. This is not possible though, both for practical reasons as well as due to the bleaching process. 22
2 Materials and methods

2.1 Computer software

Data recordings from Ca\(^{2+}\) imaging experiments results in image stacks available for further processing. The software ImageJ is used. The use of ImageJ results in data files in .dat format (other formats are possible as well; the developed program uses this however). The numerical computing environment and programming language MATLAB (Mathworks, Natick, MA, USA) is used in this project. The software developed consists of two main programs: CorrelationAnalysis and CorrelationVisualisation. As a help some further function files are used: correlationmax.m, correlationbinary.m, correlationamp.m and plotnetwork.m. The previously developed program PickCells is also included in the toolkit.

2.2 Experiments

2.2.1 Calcium recordings

The Ca\(^{2+}\) imaging experiments were done at two different microscopes. One is a Zeiss Axiovert 100 M fluorescence microscope with 25X 0.8 NA water objective (Carl Zeiss, Göttingen, Germany) in a QE-1 tempered cell chamber at 37 °C (Warner Instruments, Hamden CT, USA) using MetaFluor v6.3r6 (Molecular Dynamics Corporation, GE Healthcare Bio-Sciences, Uppsala, Sweden) as the image acquisition software. The camera was a thermoelectrically cooled Cascade II 512, a 512x512 pixel resolution 16 bit EMCCD camera (Photometrics, Tucson AZ, USA). Excitation at 495 nm was done with a filter wheel (Lambda 10-3 with SmartShutter, Sutter Instrument, Novato CA, USA). A distance of 1 pixel corresponds to a physical distance of 0.6719 \(\mu\)m.

The other microscope used is an upright Zeiss LSM510 Meta confocal 2-photon laser scanning microscope. The laser is a Coherent Chameleon Ultra II and the microscope is equipped with a Sutter MPC-385-2 micromanipulator. The distance of one pixel corresponds to different physical distances depending on which optical settings that are used (e.g. zoom). A zoom equal to 1 corresponds to 0.82 \(\mu\)m and 0.7 to 1.2 \(\mu\)m.

2.2.2 Previous experiments with stem cells from mouse

Calcium recordings were done at the Zeiss Axiovert 100 M fluorescence microscope (differentiating mES) as well as the Zeiss LSM 510 Meta confocal 2-photon laser scanning
microscope zoom 0.7 (proliferating mES) with cells loaded with the fluorescent dye Fluo3/AM (Invitrogen) placed in an incubator for approximately 30 minutes. The cells used in the experiments were neurally differentiated mouse embryonic stem (mES) cells and proliferating mES cells. The differentiating cells were prepared from undifferentiated mES cell line R1. The undifferentiated mES cell line had been maintained in a serum replacement medium containing leukaemia inhibitory factor (LIF) to prevent differentiation. Neural differentiation was induced according to a previous protocol.  

2.2.3 Previous experiments with stem cells from human

Calcium recordings were done at the Zeiss Axiovert 100 M fluorescence microscope with cells loaded with the fluorescent dye Fluo3/AM (Invitrogen) placed in an incubator for approximately 30 minutes. Undifferentiated human ES cell lines H9 and HS181 were cultured on mitotically inactivated human foreskin fibroblasts. Medium was changed daily. Neural differentiation of hES cells was induced by means of coculture on PA6 stromal cells, according to Perrier et al. 2004 [ref Perrier, Studer PNAS 2004].

2.2.4 Experiments carried through as a part of the project

Calcium recordings were done at the Zeiss LSM 510 with cells loaded with the fluorescent dye Fluo3/AM (Invitrogen) placed in an incubator for approximately 30 minutes. Cells from the kidney (KUA), cervix (HeLa) and urine bladder (T24) as well as mES cells were used.
3 Results

3.1 Analysis of previous studies

In order to analyse what has been done in the field a number of scientific articles were studied. The question to answer is whether there are any studies on correlation in biology in general and on correlation between calcium signalling cells in particular. In addition, the aim was to investigate what methods that had been used and the quality of the methods.

The conclusion is that the stringency and amount of quantitative reasoning spans from rather low to very high. There seems to be the same problem with coherence concerning terminology as was pointed out already in 1967. In for instance an article with the title Correlation of NADH and Ca$^{2+}$ signals in mouse pancreatic acinar cells no quantitative correlation analysis is carried through. In articles by Moreaux et al., Weliky et al. and Mandeville et al., although not treating correlation between calcium signalling cells, correlation analysis is done properly. What is important is that the authors clearly define what sort of correlation is used. The two most relevant studies are the ones by Yuste et al. In these, the correlation between Ca$^{2+}$ signalling cells is considered. This is achieved by considering the onset of activity; that is the time point for the initiation of a Ca$^{2+}$ spike. Vectors with such time points are compared in pairs. Therefore, the signals are digitalised and only the onset of activity is taken care of.

3.2 The software tool

In order to analyse and calculate the pair wise correlations, a computer software tool was developed in MATLAB. Different methods for correlation analysis were implemented: continuous correlation with lag and without lag, binary correlation with lag and without lag as well as simple correlation and amplitude correlation. The working process of analysing the correlations between calcium signals of different cells can be formalised to an algorithm.

1) Open the image files from Ca$^{2+}$ recordings in ImageJ. If the experiment was done using the 2-photon microscope, ImageJ reads the LSM-file. If the experiment was done using the Zeiss microscope, the files must be converted to some readable format (e.g. jpeg using some re-naming software to simplify the process), then imported to ImageJ as an image stack. It should be noted that recordings from other microscope types can be analysed as well, since ImageJ only requires the image stack.
2) The next step is to generate the data files to be read by the developed software. This is done using the tool ROI manager (included in ImageJ), where regions of interest (ROIs) are chosen and saved. The mean intensity is measured for the different images and ROIs. See Figure 3 for an example of ROIs. Then, the settings in Set Measurements under Analyze must be changed. Make sure that only Center of Mass, Mean Gray Value and Invert Y coordinates are marked. Then the mean values of intensity are measured using Multi Measure. Save the resulting file in .dat format.

![Figure 3](image.png) Real image of differentiating human embryonic stem cells with ROIs from a Ca$^{2+}$ experiment.

3) Now the program CorrelationAnalysis can be used for calculation of the correlation coefficients between signals. The program reads a file named oscillationdata.dat and saves a file named correlationdata.dat. Therefore, the .dat file with data from ImageJ should be renamed and placed in the correct directory. Do also put a % sign in front of the comments to make the file directly loadable by MATLAB. In the program, the indices of cells and part of the signals to analyse can be set. It is also possible to control the selection of oscillating cells by going through the program PickCells. Furthermore, different physical parameters such as sampling time is set and which correlation method to use. See Figure 4 for one part of the graphical interface.
4) The file generated by CorrelationAnalysis contains important information, including the correlation matrix, distances and coordinates and can be visualised by different means in CorrelationVisualisation.

3.2.1 Correlation methods

First of all, the signals to analyse can be trend corrected if so is required. This is done by selecting the degree of polynomial to fit the data points and then subtract this from the actual recording. It might for instance be valuable to correct for focus shift and bleaching by subtracting a linear function if the original signal has a negative slope superimposed on the actual oscillating signal.

In the program CorrelationAnalysis, it is possible to digitalise the signals or to keep them continuous. The process of making the signals binary is initiated by setting a threshold, above which a 1 is set for corresponding time points and below a 0. The threshold is set to the arithmetic mean of the 20 first data points. If this value turns out to high or to low the threshold is adjusted to the arithmetic mean of the entire signal. The reason for this is that the
MATLAB function `corrcoef` that calculates the correlation coefficient results in NaN (not a number) if the signals only consist of ones or zeros.

Being presented to possibly trend corrected and digitalised signals, the program calculates a correlation matrix where the value at index (i,j) corresponds to the correlation coefficient between cell i and cell j. The matrix is obviously symmetric and contains ones in the diagonal. Each value of the matrix is calculated by calling the built in function `corrcoef`, that estimates the correlation coefficient between two stochastic processes as mentioned in the mathematical background. Before calculating the actual coefficient, the signals can be shifted in order to find the maximum value of correlation. This is done by calling the built in function `xcov`, that estimates the cross-covariance function. The maximum lag used by `xcov` is set to the diffusion time with a security factor. Thus, the function is not allowed to shift the signal a time step larger than the maximum time the process of diffusion requires. The maximum lag is calculated by the following relationship.

\[
t_{\text{max}} = 2 \frac{s^2}{2D}
\]

Here the leading 2 is the security factor that is supposed to partially take care of diffusion not occurring in the shortest direction. D is the diffusion coefficient and s the physical distance between the cells (centres of mass for corresponding ROIs). In this program, D is set to 283 µm²/s and is used in a study by N. Hernjak et al. for InsP³ [31]. The reason why the D for InsP³ and not Ca²⁺ was chosen is that it implies the possibility of smaller lags and since Ca²⁺ is bound to buffers to a large extent. In addition, a threshold is set to half the signal length, thereby preventing a lag larger than that.

The lag resulting in the maximum correlation is chosen and one of the signals is shifted in the corresponding direction (right or left depending on the sign). Those parts that do not overlap are deleted and the correlation coefficient is calculated by the `corrcoef` function. Figures 5-7 show the original signals (4), shifted signals (5) and the correlation as a function of lag (6) for actual Ca²⁺ recordings.
Figure 5 Two $\text{Ca}^{2+}$ signals before shifting.

Figure 6 The same $\text{Ca}^{2+}$ signals as in Figure 3 but shifted. Note the shift to the left of the green graph.
Two more correlation methods were implemented. These include amplitude correlation and simple binary correlation. The first method calculates the ratio of the total area under the signals; that is the integral in time. A correlation coefficient equal to one means the areas of the two signals are the same. The other method calculates the correlation coefficient between two signals by adding the binary vectors and dividing the number of twos with the length of the vectors. It should be noted that these methods are not covered by the statistics introduced in the mathematical background.

3.2.2 Visualisation of correlations

The file saved by CorrelationAnalysis named correlationdata.dat can be further analysed using the program CorrelationVisualisation. The user can choose between either visualising the results from one experiment or from many experiments. In order to succeed with the latter, there must be data files named correlationdata1.dat, correlationdata2.dat, ..., correlationdataN.dat if N different experiments are to be compared. Doing this, a box plot and scatter plot is plotted for the different experiments for all cells and for cells above a certain cut off.
If only one experiment is to be chosen, further plots are available. For instance it is possible to plot the correlation coefficients as a function of distance, networks of strongly correlated cells, the distance dependence of the lags chosen in CorrelationAnalysis and the correlation coefficients as a function of index in the correlation matrix etc. The graphical interface with possible elections is shown in Figure 8.

![Visualisation parameters](image)

**Figure 8** Graphical interface in CorrelationVisualisation where different parameters are set.

As can be seen in Figure 8, it is possible to set a decrement factor for correction for multiple input disturbance ($0<\gamma<1$). This is a correction function, which reduces the cut off as a function of distance to a minimum of $(1-\gamma) \times \text{cut off}$. The cut off is the null hypothesis and is intended to be used in order to find relevant correlation coefficients. The correction function is a quadratic function taking care of the fact that at a certain distance the number of cells in between increases as a quadratic function of the distance (proportional to the area). Therefore,
\( \gamma \) is a measure of the impact the disturbances of these intermediate cells. The function is as follows:

\[
f(s) = \frac{\gamma \text{cutoff}}{s_{\text{max}} - s_{\text{min}}^2} (s^2 - s_{\text{min}}^2)
\]

Here, \( s_{\text{min}} \) is the minimum distance present in the data and \( s_{\text{max}} \) is the maximum ditto. This representation is phenomenological, but could in principle be interpreted with physical parameters including the density of gap junctions etc. This is however unnecessary, since the model is very approximate and does only take care of one specific part of the distance dependence. By setting \( \gamma = 0 \) the effect of the function is neglected.

### 3.3 Analysis of previous calcium recordings

In order to test the program previous recordings of Ca\(^{2+} \) signals were analysed.

#### 3.3.1 Differentiating stem cells from mouse

Calcium recordings from differentiating stem cells in mouse (mES13, mES23 and mES33 a-b)\(^{32} \) were analysed using the programs CorrelationAnalysis and CorrelationVisualisation.

Both continuous and digitalised signals were considered with lags as well as no lags. Before analysing the correlations, only the spontaneously oscillatory part of the signals were chosen and then trend corrected using a first degree polynomial. A cell was considered to have a oscillating Ca\(^{2+} \) signal if it had at least two transients 15 \% above the baseline. In order to pick out the cells with relevant correlation, a cut off was set to 0.7 for both the continuous and the binary case when using lag and 0.5 when using no lag. In addition, the cut off was decreased as a quadratic function of distance to a minimum of 85 \% of the cut off for maximum distance. For the digitalisation, a threshold of 15 \% was used.

Comparing the different correlation methods it could be concluded that the mean correlation is lower for digitalised signals compared to continuous and signals with no lag compared to with lag. The distance dependence for the correlation coefficients was more marked (that is the correlation decreases as the distance increases) for the continuous signals than for the binary signals (except for mES23a where the correlations were almost the same) when considering with lag and above the cut off. Generally, the distance dependence is also more marked when only considering cells above cut off than all cells (except for mES13a binary with lag and mES33b binary without lag). Table 1 displays the correlation between correlation coefficients and distance for the different experiments and methods.
Table 1 Correlation coefficient between correlation coefficient and distance for different methods and cells.

<table>
<thead>
<tr>
<th>Method</th>
<th>mES13a</th>
<th>mES23a</th>
<th>mES33a</th>
<th>mES13b</th>
<th>mES23b</th>
<th>mES33b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous with lag all cells</td>
<td>0.41</td>
<td>0.35</td>
<td>0.13</td>
<td>0.18</td>
<td>0.29</td>
<td>0.028</td>
</tr>
<tr>
<td>Continuous with lag cells above cut off</td>
<td>-0.63</td>
<td>-0.5</td>
<td>-0.27</td>
<td>-0.46</td>
<td>-0.5</td>
<td>-0.26</td>
</tr>
<tr>
<td>Continuous no lag all cells</td>
<td>-0.013</td>
<td>0.0066</td>
<td>-0.002</td>
<td>-0.071</td>
<td>0.041</td>
<td>-0.15</td>
</tr>
<tr>
<td>Continuous no lag cells above cut off</td>
<td>-0.26</td>
<td>-0.34</td>
<td>-0.11</td>
<td>-0.33</td>
<td>-0.18</td>
<td>-0.17</td>
</tr>
<tr>
<td>Binary with lag all cells</td>
<td>0.43</td>
<td>0.39</td>
<td>0.23</td>
<td>0.19</td>
<td>0.2</td>
<td>0.13</td>
</tr>
<tr>
<td>Binary with lag cells above cut off</td>
<td>0.46</td>
<td>-0.49</td>
<td>-0.017</td>
<td>-0.009</td>
<td>-0.04</td>
<td>-0.031</td>
</tr>
<tr>
<td>Binary no lag all cells</td>
<td>-0.0053</td>
<td>0.0047</td>
<td>-0.003</td>
<td>-0.027</td>
<td>-0.0081</td>
<td>-0.11</td>
</tr>
<tr>
<td>Binary no lag cells above cut off</td>
<td>-0.15</td>
<td>-0.29</td>
<td>-0.4</td>
<td>-0.13</td>
<td>-0.074</td>
<td>0.041</td>
</tr>
</tbody>
</table>

Figure 9 shows the dependence of correlation on distance with cut off and linear fit for the continuous signals with lag from mES13b.

Figure 9 Plot of correlation coefficients as a function of distance including linear fit for all cells, cut off function and linear fit for cells above cut off. Continuous signals with lag from mES13b.

To control whether the cross-correlation method with lag really works as it should (that is takes care of phase shifts due to distance), the lag was plotted as a function of distance and correlation for all correlation coefficients and for those above the cut off.
First of all, the graph for lag as a function of distance is limited by a quadratic function, since the lags are set to follow the diffusion time. Furthermore, correlating the lag with distance yields the results shown in Table 2.

<table>
<thead>
<tr>
<th></th>
<th>mES13a</th>
<th>mES23a</th>
<th>mES33a</th>
<th>mES13b</th>
<th>mES23b</th>
<th>mES33b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous with lag all cells</td>
<td>0.63</td>
<td>0.65</td>
<td>0.57</td>
<td>0.7</td>
<td>0.62</td>
<td>0.62</td>
</tr>
<tr>
<td>Continuous with lag cells above cut off</td>
<td>0.46</td>
<td>0.66</td>
<td>0.32</td>
<td>0.74</td>
<td>0.44</td>
<td>0.36</td>
</tr>
<tr>
<td>Binary with lag all cells</td>
<td>0.6</td>
<td>0.63</td>
<td>0.62</td>
<td>0.65</td>
<td>0.65</td>
<td>0.59</td>
</tr>
<tr>
<td>Binary with lag cells above cut off</td>
<td>0.46</td>
<td>0.61</td>
<td>0.62</td>
<td>0.42</td>
<td>0.42</td>
<td>0.43</td>
</tr>
</tbody>
</table>

**Table 2** Correlation coefficients between lag and distance for different methods and cells.

As can be seen, the lag clearly increases as the distance increases.

Plotting the lag as a function of correlation coefficients reveals which signals that needed to be shifted in order to find the maximum correlation coefficient. Correlating the lag with correlation coefficients yields the following results seen in Table 3.

<table>
<thead>
<tr>
<th></th>
<th>mES13a</th>
<th>mES23a</th>
<th>mES33a</th>
<th>mES13b</th>
<th>mES23b</th>
<th>mES33b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous with lag all cells</td>
<td>0.27</td>
<td>0.21</td>
<td>-0.11</td>
<td>0.054</td>
<td>0.11</td>
<td>-0.12</td>
</tr>
<tr>
<td>Continuous with lag cells above cut off</td>
<td>-0.31</td>
<td>-0.34</td>
<td>-0.26</td>
<td>-0.38</td>
<td>-0.23</td>
<td>-0.1</td>
</tr>
<tr>
<td>Binary with lag all cells</td>
<td>0.27</td>
<td>0.26</td>
<td>0.12</td>
<td>0.08</td>
<td>-0.037</td>
<td>-0.014</td>
</tr>
<tr>
<td>Binary with lag cells above cut off</td>
<td>-0.18</td>
<td>-0.19</td>
<td>-0.32</td>
<td>-0.11</td>
<td>-0.19</td>
<td>-0.042</td>
</tr>
</tbody>
</table>

**Table 3** Correlation coefficients between lag and correlation coefficients for different methods and cells.

Only considering cells above cut off shows that the trend is negative for continuous as well as binary signals; that is the larger the correlation is the smaller is the lag.

Plotting the network of pair wise correlations above cut off for differentiating cells in the physical plane results in figures such as those in Figure 10 (mES13a, mES23a, mES33a, mES13b, mES23b and mES33b continuous signals with lag).
In all six cell groups, there are small clusters with high correlation.

### 3.3.2 Proliferating stem cells from mouse

In the same way as for the differentiating stem cells above, Ca\(^{2+}\) recordings from proliferating stem cells (mES0 and mES4)\(^{33}\) were analysed. The same conclusions as above could be made.
concerning the different correlation methods when it comes to the mean correlation coefficient. Also with these cells, the distance dependence of the correlation coefficients is more marked for the cells above the cut off compared to all. Comparing the continuous signals with the binary signals, the distance dependence is approximately the same using lag and cells above cut off. The correlation coefficients between correlation and distance are shown in Table 4.

<table>
<thead>
<tr>
<th>Method</th>
<th>mES0</th>
<th>mES4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous with lag all cells</td>
<td>0.36</td>
<td>0.38</td>
</tr>
<tr>
<td>Continuous with lag cells above cut off</td>
<td>-0.34</td>
<td>-0.21</td>
</tr>
<tr>
<td>Continuous no lag all cells</td>
<td>0.0020</td>
<td>0.0042</td>
</tr>
<tr>
<td>Continuous no lag cells above cut off</td>
<td>-0.14</td>
<td>-0.2</td>
</tr>
<tr>
<td>Binary with lag all cells</td>
<td>0.28</td>
<td>0.28</td>
</tr>
<tr>
<td>Binary with lag cells above cut off</td>
<td>-0.24</td>
<td>-0.25</td>
</tr>
<tr>
<td>Binary no lag all cells</td>
<td>-0.0091</td>
<td>-0.016</td>
</tr>
<tr>
<td>Binary no lag cells above cut off</td>
<td>-0.4</td>
<td>-0.21</td>
</tr>
</tbody>
</table>

**Table 4** Correlation coefficients between correlation coefficient and distance for different methods and cells.

The same control of the functionality of the cross-correlation method with lag was carried through for the proliferating cells as for the differentiating cells. The correlation between lag and distance are shown in Table 5.

<table>
<thead>
<tr>
<th>Method</th>
<th>mES0</th>
<th>mES4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous with lag all cells</td>
<td>0.43</td>
<td>0.49</td>
</tr>
<tr>
<td>Continuous with lag cells above cut off</td>
<td>0.17</td>
<td>0.29</td>
</tr>
<tr>
<td>Binary with lag all cells</td>
<td>0.42</td>
<td>0.5</td>
</tr>
<tr>
<td>Binary with lag cells above cut off</td>
<td>0.31</td>
<td>0.29</td>
</tr>
</tbody>
</table>

**Table 5** Correlation coefficient between lag and distance for different methods and cells.

Clearly, the lag increases with increasing distance.

The correlation between lag and correlation coefficients are shown in Table 6.

<table>
<thead>
<tr>
<th>Method</th>
<th>mES0</th>
<th>mES4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous with lag all cells</td>
<td>0.26</td>
<td>0.23</td>
</tr>
<tr>
<td>Continuous with lag cells above cut off</td>
<td>-0.089</td>
<td>-0.13</td>
</tr>
<tr>
<td>Binary with lag all cells</td>
<td>0.2</td>
<td>0.13</td>
</tr>
<tr>
<td>Binary with lag cells above cut off</td>
<td>0.0032</td>
<td>-0.0026</td>
</tr>
</tbody>
</table>

**Table 6** Correlation coefficient between lag and correlation coefficient for different methods and cells.
Only considering cells above cut off shows that the trend is negative; that is the larger the correlation is the smaller the lag is.

Plotting the network of pair wise correlations above cut off for proliferating cells in the physical plane results in figures such as those in Figure 11 (mES0 and mES4 continuous signals with lag).

**Figure 11** Correlation network for continuous signals with lag above cut off for mES0 and mES4.

In both cell groups, the clusters of cells with high correlation are rather large. Figure 12 below shows box plots combined with scatter plots for mES13-33 a-b (differentiating stem cells) and mES0 and mES4 (proliferating stem cells) for all cells and for cells significantly above cut off 0.7. Clearly, nothing can be said about significant differences comparing differentiating to proliferating cells.
Figure 12 Scatter plot and box plot for the different cells using continuous signals with lag for all cells and above cut off. In order of: mES13-33 a-b and mES0 and mES4.

Studying the network plots above, it seems like there are more small scale clusters in the differentiating cells than in the proliferating cells. This difference can be quantified by exactly defining the notion of small scale clusters.

**Definition 9**

A small scale cluster is a group of cells containing at least three cells and with no pairwise distance larger than approximately five standard cell lengths.

Doing the quantification according to Definition 9 with standard cell length approximately 15 μm and thereby cluster length 70 μm yields the following results shown in Figure 13 (only experiments with more than 60% active cells are considered). It should be noted that it is the total number of cells included in small scale clusters that is shown. Even though clusters were often connected to each other, all cells fulfilling Definition 9 were accounted for.
The number of cells in small scale clusters is clearly higher for the differentiating cells compared to the proliferating cells. Calculating the arithmetic means and standard deviations of the two groups yields the results in Table 7. It can be seen that the arithmetic means are significantly higher for the differentiating cells compared to the proliferating cells for cut off 0.5 as well as 0.7.

<table>
<thead>
<tr>
<th></th>
<th>Differentiating cells</th>
<th>Proliferating cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean with cut off 0.7 (%)</td>
<td>29.96</td>
<td>1.16</td>
</tr>
<tr>
<td>Std with cut off 0.7 (%)</td>
<td>15.27</td>
<td>1.83</td>
</tr>
<tr>
<td>Mean with cut off 0.5 (%)</td>
<td>53.05</td>
<td>10.54</td>
</tr>
<tr>
<td>Std with cut off 0.5 (%)</td>
<td>5.04</td>
<td>5.25</td>
</tr>
</tbody>
</table>

Table 7 Arithmetic means and standard deviations for the number of cells included in small scale clusters.

### 3.3.3 Human stem cells

Calcium recordings from differentiating (early and late stage) and proliferating stem cells from human\(^4\) were analysed using the programs CorrelationAnalysis and CorrelationVisualisation. Only continuous signals with lag were considered. Before analysing the correlations, only the spontaneously oscillatory part of the signals were chosen and then trend corrected using a first degree polynomial. A cell was considered to have an oscillating
Ca\textsuperscript{2+} signal if it had at least two transients 15\% above the baseline. In order to filter out the cells with relevant correlation, a cut off was set to 0.7. In addition, the cut off was decreased as a quadratic function of distance to a minimum of 85\% of the cut off for maximum distance. Further filtering was carried through by demanding at least 55 cells, 13\% active cells and an experimental length of at least 790 s.

The same measures were taken as for the mouse embryonic stem cells in order to analyse the correlation among cells. As a result, the mean correlation coefficient between correlation coefficient and distance (all cells/above cut off ± standard deviation) was for the differentiating cells 0.31±0.12/-0.34±0.17 (17 experiments) and for the proliferating cells 0.24±0.16/-0.64±0.48 (2 experiments). Clearly, the mean correlation coefficient decreases and changes sign when considering cells above the cut off compared to all cells. The mean correlation coefficient between lag and distance was 0.55±0.043/0.45±0.095 for the differentiating cells and 0.58±0.16/0.62±0.43 for the proliferating cells. Obviously, the trend is positive; that is the longer the intercellular distance, the longer must the signals be shifted to get the maximum correlation. The mean correlation coefficient between lag and correlation coefficient was 0.15±0.10/-0.21±0.17 for the differentiating cells and 0.05±0.31/-0.57±0.42 for the proliferating cells. As can be seen, the mean correlation coefficient changes sign to negative when considering the cells above the cut off. This means that the lag decreases as the correlation coefficient increases, with the implication that strongly correlated signals are weakly phase shifted.

As a contrast to the mouse embryonic stem cells, the Ca\textsuperscript{2+} experiments with human embryonic stem cells did not exhibit any significant difference between the differentiating and the proliferating cells concerning the number of cells included in small scale clusters of strongly correlated cells. The mean number of cells (in percent of total number of active cells cut off 0.7/cut off 0.5 ± standard deviation) was 11.3±14.1/37.7 ±18.3 for the differentiating cells and 19.0±26.9/40.8±42.2 for the proliferating cells. Clearly, nothing can be said about differences because of the large standard deviations.

### 3.4 Analysis of experiments carried through as a part of the project

To test the correlation method on experimental data, a series of experiments with different kinds of cells was performed (see Materials and methods). In these experiments a cut was made, thereby creating a cell free lane that separates two domains of living cells. A previous study\textsuperscript{35} investigates similar properties as here, although no formal cross-correlation analysis is done. The purpose of these experiments is to compare the correlation across the cut to the
correlation amongst the cells on the same side of the cut. The correlation across the cut should be significantly lower compared to the correlation amongst cells on the same side according to the hypothesis that high synchronisation of Ca$^{2+}$ signals is due to physical coupling of gap junctions. By making a cut, existing gap junctions coupling the cells are removed.

The results of the experiments were not supporting the hypothesis. The main problem was to record spontaneous activity, not being a result of the stress that the cut causes as well as silent cells. In Figure 14, an image from an experiment with mES cells is shown.

Figure 14 Ca$^{2+}$ experiment with mES cells with a diagonal cut creating a cell free lane.
4 Discussion

4.1 The developed software

The developed software implements correlation analysis of continuous signals as well as digitalised signals. This is a great advantage, since the analysis of correlation of Ca\textsuperscript{2+} oscillations requires a broad spectrum of methods, being a poorly understood field of research. Furthermore, the lag used by the covariance function is limited by physical arguments including the maximum diffusion length. The correlation matrix created by the program is visualised in different ways, including correlation network plots, distance dependence of the correlation coefficients, the distance dependence of the lag etc. Taken together, this implementation of the correlation analysis method is a strong tool, enabling qualitative as well as quantitative analysis of Ca\textsuperscript{2+} experiments.

The quality of the correlation analysis is highly dependent on the data. If not being careful with choosing active cell traces, there is a significant risk of being presented to highly correlated signals (false positives), since signals of silent cells are strongly linear dependent. This problem can be solved by going through all cell traces manually and selecting the active traces with the program PickCells. This selection is not fully objective, since many traces are superimposed on polynomials of different degrees, resulting in difficulties in choosing peaks above the baseline with a certain amount. A more experimentally oriented problem is the sampling frequency. In order to fully correlate the signals of different cells, it is important to increase the temporal resolution; that is the number of images taken in the microscopy setup every point of time.

Another disadvantage with the program is the time it consumes. Using hundreds of cells as input implies rather long processing times. This is especially relevant for the correlation network plot, where the algorithm includes plotting lines between all cell pairs above the cut off.

4.2 Filtering aspects - the cut off

The working hypothesis of this project has been that cells being active with Ca\textsuperscript{2+} oscillations show higher correlation if being physically connected by some means. Possible connections include secreted molecules such as ATP and InsP\textsubscript{3} as well as direct contacts such as gap junctions. A central problem of analysing the results of correlation analysis is to decide which correlation coefficients are relevant; that is to decide the cut off. This is exceedingly
important, since for instance the distance dependence of the correlation coefficients is highly dependent on the cut off. By statistical arguments, it is reasonable to set the cut off around 0.7, since that is the domain where the linear relationship is reasonable strong. This rule of thumb is to be combined with some biological data. By making the assumption that highly synchronised cells are physically connected by gap junctions, the cut off can be set to some mean value of the correlation coefficient across a cut made in a cell culture. This is reasonable, since that correlation coefficient should only include that part of linear dependence that can be deduced from cellular similarities etc. Correlation coefficients above that value (present prior to the cut), manifest some other synchronisation factor resulting from gap junctions.

A possible approach is to consider the distribution of the correlation coefficients. If all correlation coefficient were equally common, the distribution should be uniform. This is not the case though, since more signals are moderately correlated (around 0.2-0.6) than not correlated at all or very strongly correlated (>0.7). By studying the distribution of a large number of experiments, it might be possible to find a natural cut off, given all distributions have some parameters in common. Figure 15 shows the distribution for an experiment with mouse embryonic stem cells. The data is distributed in a histogram with 100 equally spaced bins.

![Distribution of correlation coefficients](image)

**Figure 15** Distribution of correlation coefficients for an experiment with mES cells. 100 equally spaced bins are used for the partitioning.
4.3 General physical principles

A question of deepest relevance concerning Ca\(^{2+}\) oscillations is the origin of the phenomenon. The wave like behaviour is often explained by combining different parts of the so called signalling toolkit, but more seldom explained in any deeper fashion. Our hypothesis is that it is the very existence of gap junctions that results in the oscillations. The line of argument is as follows. All individual cells have a system regulating the Ca\(^{2+}\) concentration. This system includes pumps and channels as discussed in the biological introduction to this paper and is intended to keep the concentration constant in order to minimise the total energy as a function of the special set of regulating tools. By challenging the individual cells by some agonist or by disturbing the Ca\(^{2+}\) concentration directly, the regulating system is forced to be active. The system is reasonable robust and fast, resulting in the suppression of the concentration disturbance. Connecting the cells with gap junctions on the other hand implies a new situation, where the regulating system of each cell is dependent on the system of adjacent cells. According to our hypothesis, this decreases the robustness of the regulating system leading to oscillations when trying to keep the Ca\(^{2+}\) concentration at a constant level after a disturbance. In general physical terms, this means that a break of symmetry is needed in order of a disturbance from equilibrium to result in oscillations. Nature has, in the process of evolution, learned to exploit these oscillations as a part of a complex and ubiquitous signalling system.
5 Conclusion

In the present paper, it has been shown that the method of correlation analysis can be applied to the field of Ca$^{2+}$ signalling. We have implemented the cross-covariance function in a MATLAB program, including means of visualising the results of the correlation analysis. Several experimental parameters and theoretical analysis parameters can be set by the user.

By applying the software to previous Ca$^{2+}$ experiments, we have shown that there are cells with very high degree of correlation. In experiments with mouse embryonic stem cells, we have shown that the number of cells included in small scale clusters with strongly correlated cells is significantly larger in the differentiating cells than the proliferating cells for two choices of the cut off (0.5 and 0.7). In the same experiments, correlation analysis and filtering out cells with high correlations (above 0.7), results in a negative correlation between the correlation coefficients and the distances. Furthermore, the lag is negatively correlated with the correlation coefficient for all experiments. These results might suggest some kind of cooperation between differentiating mouse embryonic stem cells via physical couplings that is not as apparent in proliferating cells.

In experiments with human embryonic stem cells, the use of the developed software revealed cells with high correlation coefficients. We have shown that the correlation coefficients decrease as the distance increases above a cut off equal to 0.7. Further, the time lag (and thus the effective phase shift in time) increases as the distance increases and decreases as the correlation coefficients increases. Taken together, this might suggest that the synchronisation is achieved through physical connections. Important to note however, is that the results with the human embryonic stem cells are rather poor with large standard deviations.
6 Future perspectives

In this study, the developed software has been briefly applied to Ca$^{2+}$ experiments. In order to really test the method in general and the software in specific, more serious quantitative analysis must be performed using the program. In more detail, a thorough study comparing the results of a large number of experiments using continuous and digitalised signals with and without lag would be advantageous.

6.1 Technical improvements of the software

An obvious improvement of the software is the correlation network plot algorithm. It is at the present time consuming and can be made more efficient in that sense. The very same algorithm can be made better by superimposing the correlation network plot on the actual microscope picture to enhance the visibility. Another technical improvement is the possibility to export analysis parameters to other calculation programs than MATLAB, for instance EXCEL. This might include the possibility to save a file containing different forms of analysis results to be used in further processing in any format.

In addition, the correlation methods comparing the amplitude of the signals must be improved. Up to now, this method cannot be used with any statistics and is not reliable because of the non-linear relation between actual Ca$^{2+}$ concentration and fluorescence intensity. The covariance function is scale independent, and it is thus irrelevant whether the actual Ca$^{2+}$ concentration or the fluorescence intensities are compared. This is however not the case for the amplitude correlation method, where the total area of the signals are compared. A possible improvement is to normalise all signals and to compare some of the dominant peaks.

6.2 Novel areas of use

The method in general and the software in specific described in this paper can be used in other applications than just correlation analysis of Ca$^{2+}$ signals. These novel areas might include electrical activity or other signalling molecules. In the future, new fluorescent dyes as well as completely new imaging paradigms will appear, making it possible to use the present method in further fields of biological sciences.
7 Acknowledgment

I want to show my gratitude to my supervisor Docent Per Uhlén and co-supervisor PhD Seth Malmersjö at the Division of Molecular Neurobiology at the Department of Molecular Biochemistry and Biophysics at Karolinska Institutet, Stockholm. Thank you for your help, advice and nice interesting discussions. I also want to thank PhD students Marie Karlsson and Paola Rebellato for kindly helping me with the experiments and MSc Hampus Sunner for previous work in the field and interesting discussions.

Last but not least I want to thank Professor Hjalmar Brismar for making me interested in the field of biological physics and for being the examiner of my MSc thesis.
The source code for the MATLAB programs CorrelationAnalysis.m, correlationamp.m, correlationbinary.m, correlationmax.m, CorrelationVisualisation.m and plotnetwork.m follows below.

```
% Correlation Analysis of Calcium Oscillations
% by Erik Nilsson MSc Engineering Physics
% Version 1.0
% Copyright 2008 by Erik Nilsson All Rights Reserved

% Program file: CorrelationAnalysis.m
% Related needed files: correlationmax.m; correlationbinary.m; correlationamp.m; PickCells.m; CorrelationVisualisation
% Required file: oscillationdata.dat
% Creates file: correlationdata.dat
% 1st version: 2008-12-12, Karolinska Institutet, Stockholm, Sweden
% Contact: ernil@kth.se
% Last updated: 2008-11-18

% - - - Main file for the CorrelationAnalysis program - - -

%Users choice of parameter values

clear

%Indices of cells. all=all cells, i j k... or n: N indices.
%Sampling time(s).
%Length scale (m). MetaFluor 0.6719*10^-6, LSM 0.82*10^-6 zoom 1 and 1.2*10^-6 zoom 0.7.

%Correlation alternative. continuous/1, binary/2 or amplitude/3.
%Binary correlation alternative. maximum/1 or simple/2.
%Binal threshold (factor).
%Test binarisation and plot the result. yes/1 or no/0.
%Lag alternative. no/0 or maximum/1 lag.
%Test trendcorrection and plot the result. yes/1 or no/0.
%Apply trendcorrection. yes/1 or no/0.

%Default values of the parameter choice.

fid=fopen('CorrelationAnalysis.ini');
param=[];
```
if fid=-1
    for i=1:11
        tempparam=fgetl(fid);
        param=strvcat(param, tempparam);
    end
    cellstr(param);
    defaultanswer1={deblank(param(1,:)), deblank(param(2,:)), deblank(param(3,:))};
    defaultanswer2={deblank(param(4,:)), deblank(param(5,:)), deblank(param(6,:))},
    deblank(param(7,:)), deblank(param(8,:)), deblank(param(9,:)),
    deblank(param(10,:)), deblank(param(11,:))};
    fclose(fid)
else
    disp('Empty file! Using default values for CorrelationAnalysis.');
    defaultanswer1={'all','5','0.6719*10^(-6)'};
    defaultanswer2={'continuous','maximum','1.15','no','maximum','yes','no','1'};
end
%Parameter choice.
answer1=inputdlg(prompt1,title1,1,defaultanswer1);
%
%---------------------------------------------------
%Set the indices of the cells that should be analysed.
if ~strcmpi(char(answer1(1)),'ALL')
    indexofcells=str2num(char(answer1(1)));
end
%Set the length scale used in the experiment. Physical distance between two
%pixels in meters.
scale=str2num(char(answer1(3)));
%Samplingtime in seconds. How often a picture is taken.
samplingtime=str2num(char(answer1(2)));

%Data from ImageJ Multi Measure with one file containing time points,
%intensities and coordinates for ROI.
load oscillationdata.dat
expdata=oscillationdata;
%Timepoints in (s).
oldtime=expdata(:,1);
time=samplingtime*(expdata(:,1)-expdata(1,1));
%Remove the timepoints.
expdata(:,1)=[];
M2=size(expdata);
N=M2(2)/3;
T=M2(1);
xcoordinates=ones(1,N);
ycoordinates=ones(1,N);
xcoordinates2=[];
ycoordinates2=[];
celldata=ones(T,N);
for i=1:N
    j=3*i;
    k=3*i-1;
    l=3*i-2;
    xtemp=expdata(l,k);
    ytemp=expdata(l,j);
tempcell=expdata(:,1);
xcoordinates(i)=xtemp;
ycoordinates(i)=ytemp;
celldata(:,i)=tempcell;
end
celldataold=celldata;

%Information box 1.
h2=helpdlg('Select region and/or press key to analyze.', 'Region Selection');
uiwait(h2);

%Part of signals choice.
if ~strcmpi(char(answer1(1)),'ALL')
    numberofcells=length(indexofcells);
else
    indexofcells=1:N;
    numberofcells=N;
end
cla
for i=1:numberofcells
    j=indexofcells(i);
    y=celldata(:,j);
    plot(time,y), hold on
    title('Fluorescence intensity as a function of time for the chosen cells')
    xlabel('Time (s)')
    ylabel('Fluorescence intensity (a.u)')
end
zoomHold = waitforbuttonpress;
%Time indices for the signals.
t1=1; %First index.
t2=length(time); %Last index.
while ~zoomHold
    point1=get(gca, 'CurrentPoint');
    finalRect=rbbox;
    point2=get(gca, 'CurrentPoint');
    x1=min(point1(1,1),point2(1,1));
    x2=max(point2(1,1),point2(1,1));
    for t=1:length(time)-1
        if x1>=time(t) && x1<time(t+1)
            t1=t;
        end
        if x2>time(t) && x2<=time(t+1)
            t2=t;
        end
    end
    cla
    for i=1:numberofcells
        y=celldata(t1:t2,j);
        plot(time(t1:t2),y),hold on
        title('Fluorescence intensity as a function of time for the chosen cells')
        xlabel('Time (s)')
        ylabel('Fluorescence intensity (a.u)')
    end
    zoomHold = waitforbuttonpress;
end
close
%New signals.
time=time(t1:t2);
celldata=celldata(t1:t2,:);

%User can pick the cells that oscillate.
selectionButton1=questdlg('Do you want to pick cells?','How to proceed','Yes','No','Yes');
switch selectionButton1,
    case 'Yes',
    %Save signal data to a file.
    celldata_new=[];
    for i=1:numberofcells
        index=indexofcells(i);
        celldata_new=[celldata_new celldata(:,index)];
    end
    celldata=celldata_new;
to_pick_from=[time, celldata];
save signaldata.txt to_pick_from -ascii
    disp('Use signaldata.txt in PickCells.m')
%Pick cells with PickCells.m
PickCells
% Read the new cell indices.
fi = fopen('signaldata.txt', 'r');
fi2 = fopen('temp.dat', 'w');  % temp file
i = 0;
while ~feof(fi)
    n = fgetl(fi);
    if length(find(n == '%')) > 0, i = i + 1; end
    if length(find(n == '%')) == 0, fprintf(fi2, [n ' ']); end
end  %Put line without % in temp file
fclose(fi)
fclose(fi2)
fi = fopen('temp.dat', 'r');
indiceofcells = fgetl(fi);
indiceofcells = str2num(indiceofcells)
indiceofcells = indiceofcells(2:end);
fclose(fi)
delete('temp.dat')
numberofcells = length(indiceofcells);
% Save data for the picked cells in oscillationdata_new.dat
expdata_new = [];
for i = 1:numberofcells
    j = indiceofcells(i);
    signaltemp = celldataold(:, j);
    xtemp = xcoordinates(j) * ones(length(oldtime), 1);
    ytemp = ycoordinates(j) * ones(length(oldtime), 1);
    expdata_new = [expdata_new; signaltemp; xtemp; ytemp];
end
% Put the time vector first.
expdata_new = [oldtime; expdata_new];
% Save the data.
save oscillationdata_new.dat expdata_new -ascii
close
% If the user do not want to pick cells.
case 'NO',
    close
end
% Parameter choice.
answer2 = inputdlg(prompt2, title2, 1, defaultanswer2);
% Save the parameter choice.
y = [answer1; answer2];
file1 = fopen('CorrelationAnalysis.ini', 'wt');
fprintf(file1, '%s
', y{:});
close('all');
disp('Parameter choice saved in CorrelationAnalysis.ini')
%
% Set the correlation method that should be used. 1 = correlationmax,
% 2 = correlationbinary, 3 = correlationamp.
if strcmpi(char(answer2(1)), 'continuous')
corralt = 1; 
elseif strcmpi(char(answer2(1)), 'binary')
corralt = 2;
elseif strcmpi(char(answer2(1)), 'amplitude')
corralt = 3;
else
    corralt = str2num(char(answer2(1))); 
end
% Set the binary correlation method that should be used. 1 = correlationmax,
% 2 = correlationsimple.
if strcmpi(char(answer2(2)), 'maximum')
corraltbin = 1;
elseif strcmpi(char(answer2(2)), 'simple')
corraltbin=2;
else
corraltbin=str2num(char(answer2(2)));
end
%Set the binary threshold factor.
threshold=str2num(char(answer2(3)));
%Set whether binarisation should be shown or not.
if strcmpi(char(answer2(4)),'YES')
    PLOT=1;
elseif strcmpi(char(answer2(4)),'NO')
    PLOT=0;
else
    PLOT=str2num(char(answer2(4)));
end
%Set whether binarisation should be shown or not.
if strcmpi(char(answer2(4)),'YES')
    PLOT=1;
elseif strcmpi(char(answer2(4)),'NO')
    PLOT=0;
else
    PLOT=str2num(char(answer2(4)));
end
%Set the lag used in cross-covariance correlation. 0=no lag, 1=maximum lag
%dependent on physical distance.
if strcmpi(char(answer2(5)),'no')
    lagalt=0;
elseif strcmpi(char(answer2(5)),'maximum')
    lagalt=1;
else
    lagalt=str2num(char(answer2(5)));%
end
%Set whether trendcorrection should be implemented and tested or not in a
%figure. 0=NO, 1=YES.
if strcmpi(char(answer2(6)),'YES')
    trendcorrection1=1;
elseif strcmpi(char(answer2(6)),'NO')
    trendcorrection1=0;
else
    trendcorrection1=str2num(char(answer2(6)));%
end
%Set whether trendcorrection should be implemented and used or not in a
%figure. 0=NO, 1=YES.
if strcmpi(char(answer2(7)),'YES')
    trendcorrection2=1;
elseif strcmpi(char(answer2(7)),'NO')
    trendcorrection2=0;
else
    trendcorrection2=str2num(char(answer2(7)));%
end
%Set the degree of polynomial that should be used for trendcorrection.
trendcorrectiondeg=str2num(char(answer2(8)));%
%
%--------------------------------------------------------
%Test trendcorrection and plot the result.
if trendcorrection1==1 && trendcorrection2~=1 && corralt~=3
    celldataold=celldata;
    figure
    for i=1:numberofcells;
        j=indexofcells(i);
        y=celldata(:,j);
        plot(time,y), hold on
        title('Fluorescence intensity as a function of time before trendcorrection')
    xlabel('Time (s)')
    ylabel('Fluorescence intensity (a.u)')
    end
end
figure
for i=1:numberofcells;
    j=indexofcells(i);
    y=celldata(:,j);
    fitpolynom=polyfit(time,y,trendcorrectiondeg);
    y=y-polyval(fitpolynom,time);
    plot(time,y), hold on
    title('Fluorescence intensity as a function of time after trendcorrection')
end
xlabel('Time (s)')
ylabel('Fluorescence intensity (a.u)')

selectionButton2=questdlg('Do you want to proceed with this trendcorrection setting?', 'How to proceed', 'Apply', 'Do not apply', 'Apply');
switch selectionButton2,
  case 'Apply',
    close
    close
    trendcorrection1=0;
    trendcorrection2=1;
  case 'Do not apply',
    close
    close
    trendcorrection1=0;
    trendcorrection2=0;
end

if trendcorrection1==1 && corralt==3
  disp('Warning! Trendcorrection cannot be applied together with amplitude correlation.')
elseif trendcorrection2==1 && corralt==3
  disp('Warning! Trendcorrection cannot be applied together with amplitude correlation.')
end

% Show binarisation.
if corralt==2 && PLOT==1
  figure
  for k=1:numberofcells
    h=waitbar(0, 'Please wait... Testing and showing binarisation');
    i=indexofcells(k);
    for l=1:numberofcells
      j=indexofcells(l);
      y1=celldata(:,i);
      y2=celldata(:,j);
      % If trendcorrection should be applied first.
      if trendcorrection2==1
        fitpolynom1=polyfit(time,y1,trendcorrectiondeg);
        y1=y1-polyval(fitpolynom1,time);
        fitpolynom2=polyfit(time,y2,trendcorrectiondeg);
        y2=y2-polyval(fitpolynom2,time);
      end
      if i==j
        T1=t1*samplingtime;
        T2=t2*samplingtime;
        correlationbinary(time,y1,y2,1,0,threshold,PLOT,T1,T2,i,j);
      end
    end
    close(h)
  end
  % User choose whether to apply the binarisation or not.
  selectionButton2=questdlg('Do you want to proceed with this binarisation setting?', 'How to proceed', 'Apply', 'Do not apply', 'Apply');
  switch selectionButton2,
    case 'Apply',
      close
    case 'Do not apply',
      close
      disp('Correlation analysis cancelled! No correlation data saved!')
      break
% If the user do not want to test trendcorrection.
if trendcorrection1~=1 || trendcorrection1==1 && corralt==3

% Calculate the correlation, distance and lag matrix.
correlationmatrix=ones(numberofcells,numberofcells);
distancematrix=zeros(numberofcells,numberofcells);
lagmatrix=zeros(numberofcells,numberofcells);

% k and l are indices for the matrix, whereas i and j are indices for the
% signaldata. This is to enable any choice of cell indices.

h=waitbar(0,'Please wait... Calculating correlation matrix');
for k=1:numberofcells
    waitbar(k/numberofcells)
    i=indexofcells(k);
    for l=k:numberofcells
        j=indexofcells(l);
        if i~=j
            if lagalt==1
                x1=xcoordinates(i); y1=ycoordinates(i);
                x2=xcoordinates(j); y2=ycoordinates(j);
                s=sqrt((x1-x2)^2+(y1-y2)^2);
                s=s*scale;
            end
            % Diffusion coefficient.
            % See Modeling and Analysis of Calcium Signalling Events
            % Leading to Long-Term Depression in Cerebellar Purkinje Cells
            % Hernjak et al. 2005.
            D=283*10^(-12); % m^2/s IP3.
            factor=2;
            maxtime=(factor*s^2)/(2*D);
            maxlags=ceil(maxtime/samplingtime);
            else
                maxlags=0;
            end
            y1=celldata(:,i);
            y2=celldata(:,j);
            if trendcorrection2==1 && corralt~=3
                fitpolynom1=polyfit(time,y1,trendcorrectiondeg);
                y1=y1-polyval(fitpolynom1,time) ;
                fitpolynom2=polyfit(time,y2,trendcorrectiondeg);
                y2=y2-polyval(fitpolynom2,time) ;
            end
            if corralt==1
                % If the first method is chosen.
                Temp=correlationmax(time,y1,y2,maxlags,i,j);
                Rtemp=Temp(1); lagtemp=Temp(2);
                lagmatrix(k,l)=lagtemp*samplingtime;
                lagmatrix(l,k)=lagmatrix(k,l);
                correlationmatrix(k,l)=abs(Rtemp);
                correlationmatrix(l,k)=correlationmatrix(k,l);
            end
            if corralt==2
                % If the second method is chosen.
                T1=t1*samplingtime;
                T2=t2*samplingtime;
                Temp=correlationbinary(time,y1,y2,corraltbin,maxlags,threshold,0,T1,T2,i,j);
                Rtemp=Temp(1); lagtemp=Temp(2);
                lagmatrix(k,l)=lagtemp*samplingtime;
                lagmatrix(l,k)=lagmatrix(k,l);
                correlationmatrix(k,l)=abs(Rtemp);
                correlationmatrix(l,k)=correlationmatrix(k,l);
            end
            if corralt==3
                % If the third method is chosen.
                correlationamp(time,y1,y2);
                correlationmatrix(k,l)=abs(Rtemp);
                correlationmatrix(l,k)=correlationmatrix(k,l);
            end
        end
    end
end
end
%Calculate the distance matrix.
x1=xcoordinates(i); y1=ycoordinates(i);
x2=xcoordinates(j); y2=ycoordinates(j);
disttemp=sqrt((x1-x2)^2+(y1-y2)^2);

distancematrix(k,l)=scale*disttemp;
distancematrix(l,k)=distancematrix(k,l);
end
%
%Prepare for saving the right coordinates.
xcoordinates2=[xcoordinates2 xcoordinates(1,i)];
ycoordinates2=[ycoordinates2 ycoordinates(1,i)];
end

close(h)
Nvector=length(time)*ones(1,numberofcells);

%Save the indices, correlation matrix, distance matrix, lag matrix, coordinates and number of time points.
data_to_save=[indexofcells;correlationmatrix;distancematrix;lagmatrix;xcoordinates2; ycoordinates2;Nvector];
save correlationdata.dat data_to_save -ascii
disp('Correlation data saved in correlationdata.dat');

%Continue with CorrelationVisualisation?
selectionButton3=questdlg('Do you want to proceed with
CorrelationVisualisation?','How to proceed','Yes','No','Yes');
switch selectionButton3,
%If the user wants to continue with CorrelationVisualisation.
case 'Yes',
CorrelationVisualisation
%If the user does not want to continue with CorrelationVisualisation.
case 'No',
close
break
end

end

%correlationamp.m
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%%%%%%%%%%%%%
%              Correlation Analysis of Calcium Oscillations      
%                by  Erik Nilsson MSc Engineering Physics       
%                                                                     
%                             Version 1.0                             
%                                                                     
%           Copyright 2008 by Erik Nilsson All Rights Reserved     
%                                                                     
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%%%%%%%%%%%%%
%                                                                     
%  Program file: CorrelationAnalysis.m                              
%  Related needed files: correlationmax.m; correlationbinary.m;     
%  Required file: oscillationdata.dat                               
%  Creates file: correlationdata.dat                                 
%  1st version: 2008-10-09, Karolinska Institutet, Stockholm, Sweden  
%  Contact: ernil@kth.se                                           
%                                                                     
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%%%%%%%%%%%%%
Amplitude dependent correlation coefficient
Function that calculates the correlation coefficient between two signals
by only taking care of the amplitudes.

Input:
t: the time points of the signals.
y1: the fluorescence intensities for cell i at every time point.
y2: the fluorescence intensities for cell j at every time point.

Output:
z: correlation coefficient between two signals.

% Amplitude dependent correlation coefficient
% Function that calculates the correlation coefficient between two signals
% by only taking care of the amplitudes.
%-------------------------------------------------- ------------------------
% Input:
% t: the time points of the signals.
% y1: the fluorescence intensities for cell i at every time point.
% y2: the fluorescence intensities for cell j at every time point.
% Output:
% z: correlation coefficient between two signals.
%-------------------------------------------------- ------------------------
function z=correlationamp(t,y1,y2)
% Calculate the total areas under the curves.
A1=trapz(t,y1);
A2=trapz(t,y2);
if A1>=A2
 rA=A2/A1;
else
 rA=A1/A2;
end
% Outout from the function file.
z=rA;
% Binary correlation coefficient
% Function that binarise two signals and calculates the correlation
% coefficient between them. If user wants, the binarisation is also plotted.
% %-------------------------------------------------- ------------------------
% %Input:
% t: the time points of the signals.
% y1: the fluorescence intensities for cell i at every time point.
% y2: the fluorescence intensities for cell j at every time point.
% corraltbin: choice of correlation alternative.
% maxlags: the maximum lag used by xcov.m.
% threshold: threshold value used in binarisation.
% PLOT: 0=NO plot, 1=plot.
% t1: start time point for the signals.
% t2: end time point for the signals.
% i1: index for the first cell.
% j1: index for the second cell.
% % %Output:
% z: binary correlation coefficient between two signals and the lag z=[Rmax lag].
% %-------------------------------------------------- ------------------------

function z=correlationbinary(t,y1,y2,corraltbin,maxlags,threshold,PLOT,t1,t2,i1,j1)
Y1=y1; Y2=y2;
N=length(y1);
mean1=mean(y1(1:20));
mean2=mean(y2(1:20));
limit1=threshold*mean1;
limit2=threshold*mean2;
limit12=mean(Y1);
limit22=mean(Y2);
YESval=1; NOval=0;
Maxlag=0;

for i=1:N
  if y1(i)>=limit1
    y1(i)=YESval;
  elseif y1(i)<limit1
    y1(i)=NOval;
  end
  if y2(i)>=limit2
    y2(i)=YESval;
  elseif y2(i)<limit2
    y2(i)=NOval;
  end
end

%If the threshold value was to high.
if (sum(y1)==N*NOval || sum(y1)==N*YESval)
  %Redo the binarisation.
  for i=1:N
    if Y1(i)>=limit12
      y1(i)=YESval;
    elseif Y1(i)<limit12
      y1(i)=NOval;
    end
  end
end

if (sum(y2)==N*NOval || sum(y2)==N*YESval)
  %Redo the binarisation.
  for i=1:N
    if Y2(i)>=limit22
      y2(i)=YESval;
    elseif Y2(i)<limit22
      y2(i)=NOval;
    end
  end
end
\[
y_2(i) = \text{NOval};
\]
\]
\]
\]
\%
\%
\%
\%
\% If the threshold value is still too high.
\% If the first method is chosen.
\% If the second method is chosen.
\% Output from the function file.
\% %correlationmax.m
\% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\%
\% Correlation Analysis of Calcium Oscillations
\% by Erik Nilsson MSc Engineering Physics
\% Version 1.0
\% Copyright 2008 by Erik Nilsson All Rights Reserved
\% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function z=correlationmax(t,y1,y2,maxlags,i1,j1)
Y2=y2;
y2old=y2;
y1old=y1;
told=t;
Maxlag=0;
if maxlags~=0
    if (maxlags/length(y1))>=0.5
        maxlags=floor(0.25*length(y1));
    end
end
%Cross-covariance between the two signals.
[Ryly2,lags]=xcov(y1,y2,maxlags,'coeff');

%Finds for which lag Maxlag the maximum correlation coefficient is.
N=length(Ryly2);
maxlag=0;
minlag=0;
maxlagindex=1;
minlagindex=1;
for i=1:N
    MAX=max(Ryly2);
    MIN=min(Ryly2);
    if Ryly2(i)==MAX
        maxlag=lags(i);
        maxlagindex=i;
    elseif Ryly2(i)==MIN
        minlag=lags(i);
        minlagindex=i;
    end
end
z=maxlag;
% Output
z: Maximum correlation coefficient between two signals and the lag z=[Rmax lag].
% Input:
% t: the time points of the signals.
% y1: the fluorescence intensities for cell i at every time point.
% y2: the fluorescence intensities for cell j at every time point.
% maxlags: the maximum lag used by xcov.m.
% i1: index for the first cell.
% j1: index for the second cell.
% Output:
% z: maximum correlation coefficient between two signals and the lag z=[Rmax lag].
minlag=lags(i);
minlagindex=i;
end

%Finds the lag with the maximum absolute value.
if abs(Ry1y2(maxlagindex))>=abs(Ry1y2(minlagindex))
    Maxlag=maxlag;
else
    Maxlag=minlag;
end

%Moves the signal y2 in order to get the maximum correlation with y1.
if Maxlag==0
    N1=length(y2);
    Y2=pi*ones(1,N1);
    for i=1:N1
        if Maxlag>0 %If the lag is to the right.
            if i-Maxlag>0
                Y2(i)=y2(i-Maxlag);
            end
        end
        if Maxlag<0 %If the lag is to the left.
            if i-Maxlag<=N1
                Y2(i)=y2(i-Maxlag);
            end
        end
    end
end

%Removes those part of the signals that do not overlap.
%Finds the concerned indices.
indexvector=[];
for i=1:N1
    if Y2(i)==pi
        indexvector=[indexvector i];
    end
end
removeindex=indexvector;
Y2(removeindex)=[];
y1(removeindex)=[];
t(removeindex)=[];
else
    Y2=y2;
end
end

%The maximum correlation coefficient.
M1=corrcoef(y1,Y2);
maxcorrcoeff=M1(1,2);
if isnan(maxcorrcoeff)
    disp(['Warning! correlation coefficient is NaN for cells ',num2str(i1),', and ',num2str(j1),']!')
end

%Output from the function file.
z=[maxcorrcoeff, Maxlag];
% Related needed files: plotnetwork.m;
% Required file: correlationdata.dat
% Possible required files in addition: correlationdata2.dat;,...,
% correlationdataN.dat;
% 1st version: 2008-10-17, Karolinska Institutet, Stockholm, Sweden
% Contact: ernil@kth.se
% Last updated: 2008-10-13

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%%%%%%%%%%%%%
% - - - Main file for the CorrelationVisualisation program - - - %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%%%%%%%%%%%%%
%Users choice of parameter values
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%%%%%%%%%

%-------------------------------------------------- ------------------------
%Users choice of parameter values
%-------------------------------------------------- ------------------------
clear

%Default values of the parameter choice.
fid=fopen('CorrelationVisualisation.ini');
param=[];
if fid==-1
    for i=1:15
        tempparam=fgetl(fid);
        param=strvcat(param, tempparam);
    end
    cellstr(param);
    defaultanswer1={deblank(param(1,:)), deblank(param(2,:)), deblank(param(3,:)),
                    deblank(param(4,:))};
    defaultanswer2={deblank(param(5,:)), deblank(param(6,:)), deblank(param(7,:)),
                    deblank(param(8,:)),
                    deblank(param(9,:)),
                    deblank(param(10,:)),
                    deblank(param(11,:)),
                    deblank(param(12,:)),
                    deblank(param(13,:)),
                    deblank(param(14,:)),
                    deblank(param(15,:))};
    fclose(fid)
else
    disp('Empty file! Using default values for CorrelationVisualisation. ')
    defaultanswer1={'no','0','yes','0.5'};
    defaultanswer2={'5','yes','0.15','no','no','yes','no','no','no','no','no'};
    end
%Parameter choice.
answer1=inputdlg(prompt1,title1,1,defaultanswer1);

%-------------------------------------------------- ------------------------
%Set whether multiple experiments should be compared or not.
if strcmpi(char(answer1(1)),'YES')
    comparison=1;
elseif strcmpi(char(answer1(1)),'NO')
    comparison=0;
else
    comparison=str2num(char(answer1(1)));  
end

%Set the number of experiments.
numberofexperiments=str2num(char(answer1(2)));  

%Set whether just one experiment should be analysed or not.
if strcmpi(char(answer1(3)),'YES')
    analyse=1;
elseif strcmpi(char(answer1(3)),'NO')
    analyse=0;
else
    analyse=str2num(char(answer1(3)));  
end

%Set the correlation cut off value. H0: null hypothesis.
cutoff=str2num(char(answer1(4)));  

%Analysis of one experiment.
if comparison==0 && analyse==1

%Parameter choice.
answer2=inputdlg(prompt2,title2,1,defaultanswer2);
%Save the parameter choice.
y=[answer1; answer2];
file1=fopen('CorrelationVisualisation.ini','wt');
fprintf(file1,'%s
',y{:});
fclose('all');
disp('Parameter choice saved in CorrelationVisualisation.ini')

%-------------------------------------------------- ------------------------

%Set whether the cell network should be plotted or not. 0=NO, 1=YES.
if strcmpi(char(answer2(4)),'YES')  
PLOT1=1;
elseif strcmpi(char(answer2(4)),'NO')
    PLOT1=0;
else
    PLOT1=str2num(char(answer2(4)));  
end

%Set whether the cell network should be plotted with distance restriction or not. 0=NO, distance.
if strcmpi(char(answer2(5)),'NO')
    restriction=0;
else
    restriction=str2num(char(answer2(5)));  
end

%Set whether correlation coefficients as a function of distance should be plotted or not. 0=NO, 1=YES.
if strcmpi(char(answer2(6)),'YES')
    PLOT2=1;
else
    PLOT2=0;
end

elseif strcmpi(char(answer2(6)),'NO')
    PLOT2=0;
else
    PLOT2=str2num(char(answer2(6)));
end

%Sed whether the correlation matrix should be plotted or not. 0=NO, 1=YES.
if strcmpi(char(answer2(7)),'YES')
    PLOT3=1;
elseif strcmpi(char(answer2(7)),'NO')
    PLOT3=0;
else
    PLOT3=str2num(char(answer2(7)));
end

%Sed whether the correlation coefficients should be plotted or not. 0=NO, 1=YES.
if strcmpi(char(answer2(8)),'YES')
    PLOT4=1;
elseif strcmpi(char(answer2(8)),'NO')
    PLOT4=0;
else
    PLOT4=str2num(char(answer2(8)));
end

%Sed whether the lag matrix should be plotted or not. 0=NO, 1=YES.
if strcmpi(char(answer2(9)),'YES')
    PLOT5=1;
elseif strcmpi(char(answer2(9)),'NO')
    PLOT5=0;
else
    PLOT5=str2num(char(answer2(9)));
end

%Sed whether lag as a function of distance should be plotted or not. 0=NO, 1=YES.
if strcmpi(char(answer2(10)),'YES')
    PLOT6=1;
elseif strcmpi(char(answer2(10)),'NO')
    PLOT6=0;
else
    PLOT6=str2num(char(answer2(10)));
end

%Sed whether lag as a function of correlation should be plotted or not. 0=NO, 1=YES.
if strcmpi(char(answer2(11)),'YES')
    PLOT7=1;
elseif strcmpi(char(answer2(11)),'NO')
    PLOT7=0;
else
    PLOT7=str2num(char(answer2(11)));
end

%Sed the maximum width, with which the cell network is plotted.
maxwidth=str2num(char(answer2(11)));

%Sed whether cut off correction for multiple input disturbance should be used or not. 0=NO, 1=YES.
if strcmpi(char(answer2(2)),'YES')
    disturbancecorrection=1;
elseif strcmpi(char(answer2(2)),'NO')
    disturbancecorrection=0;
else
    PLOT7=str2num(char(answer2(2)));
end

%Sed the relevance factor for disturbance correction.
gamma=str2num(char(answer2(3)));

%Reads a file with a matrix containing cell indices, correlation matrix, distance matrix, lag matrix, coordinates and number of time points.
load correlationdata.dat
corrdatala=correlationdata;
M=size(corrdatala);
indexofcells=corrdatala(1,:);
correlationmatrix=corrdatala(2:M(2)+1,:);
distancematrix=corrdatala(M(2)+2:2*M(2)+1,:);
lagmatrix=corrdatala(2*M(2)+2:3*M(2)+1,:);
xcoordinates=corrdatala(end-2,:);
ycoordinates=corrdatala(end-1,:);
Nvector=corrdatala(end,:);

N=Nvector(1);

%Coordinates for the plot window when fitting the cell coordinates.
XY=[xcoordinates;ycoordinates];
Xmax=max(XY(1,:)); Xmin=min(XY(1,:));
Ymax=max(XY(2,:)); Ymin=min(XY(2,:));
Xmarginal=0.1*(Xmax-Xmin);
Ymarginal=0.1*(Ymax-Ymin);
X1=Xmin-Xmarginal; X2=Xmax+Xmarginal;
Y1=Ymin-Ymarginal; Y2=Ymax+Ymarginal;

%Colors and widths of correlation lines.
S=size(XY);
L=S(2);
widths=zeros(L,L);
colors=ones(L*L,3);

%Calculate the largest and smallest distance between cells.
maxdistance=max(max(10^6*distancematrix));
mindistance=0;
distancevector=[];
for i=1:L
  for j=i:L
    if i~=j
      distancevector=[distancevector distancematrix(i,j)];
    end
  end
end
mindistance=10^6*min(distancevector);
XDmarginal=0.1*(maxdistance-mindistance);
XD1=mindistance-XDmarginal;
XD2=maxdistance+XDmarginal;
Smin=mindistance;
Smax=maxdistance;

%Cut off matrix.
if disturbancecorrection==1
  cutoffmatrix=ones(L,L);
  for i=1:L
    for j=i:L
      if i~=j
        s=10^6*distancematrix(i,j);
        increasingfunction=gamma*cutoff/(smax^2-smin^2)*(s^2-smin^2);
        cutofftemp=cutoff-increasingfunction;
        %Statistical significant correlation coefficient if H0: rho=cutoff.
        rho=cutoff;
        %Level of significance 95 %. Fisher's transform is used.
        rho=cutofftemp;
        D=exp(3.92/sqrt(N-3))*((1+rho)/(1-rho));
        r=(D-1)/(D+1);
        cutofftemp=r;
        cutoffmatrix(i,j)=cutofftemp;
        cutoffmatrix(j,i)=cutoffmatrix(i,j);
      end
    end
  end
else
  %Statistical significant correlation coefficient if H0: rho=cutoff.
  %Level of significance 95 %. Fisher's transform is used.
rho=cutoff;
D=exp(3.92/sqrt(N-3))*((1+rho)/(1-rho));
r=(D-1)/(D+1);
cutoff=r;
cutoffmatrix=cutoff*ones(L,L);
end

% Statistical significant correlation coefficient if H0: rho=cutoff.
% Level of significance 95 %. Fisher's transform is used.
rho=cutoff;
D=exp(3.92/sqrt(N-3))*((1+rho)/(1-rho));
r=(D-1)/(D+1);
cutoff=r;

% Different plot functions.
if correlationmatrix==correlationmatrix'
  if PLOT1==1
    if correlationmatrix
      % Calculate the widths and colors of the lines and plot.
      indexmatrix1=[];
      correlationvector1=[];
      distancevector1=[];
      correlationvector2=[];
      distancevector2=[];
      for i=1:L
        for j=i:L
          if i~=j
            corrcoff=correlationmatrix(i,j);
            correlationvector1=[correlationvector1 corrcoff];
            indexmatrix1=[indexmatrix1; i j];
            indexmatrix2=indexmatrix1;
          end
        end
      end
      % Sort the correlation coefficients.
      [sortedvector oldindex]=sort(correlationvector1);
      for i=1:length(oldindex)
        indexmatrix2(i,:)=indexmatrix1(oldindex(i),:);
      end
      h=waitbar(0,'Please wait... Plotting correlation network');
      figure
      for m=1:length(sortedvector)
        waitbar(m/length(sortedvector))
        corrcoff=sortedvector(m);
        i=indexmatrix2(m,1);
        j=indexmatrix2(m,2);
        jetcolors=colormap(jet);
        colorsize=size(jetcolors);
        if corrcoff>=cutoffmatrix(i,j)
          if cutoff~=-1
            indextransform=1-(1-corrcoff)/(1-cutoffmatrix(i,j));
            colorindex=ceil(indextransform*colorsize(1));
          else
            disp('Cut off set to 1!')
            break
          end
          color=jetcolors(colorindex,:);
          width=maxwidth*corrcoff;
          A1=[color width];
          u=(i-1)*L+j;
          colors(u,1)=A1(1);
          colors(u,2)=A1(2);
          colors(u,3)=A1(3);
          % Plot lines between different cells, but not the same.
          if i~=j
            if restriction==0
              if 10^6*distancematrix(i,j)<=restriction
                if restriction
                  % Other plot functions.
                  % End of script.
plotnetwork( width, colors(u,:), [XY(1,i) XY(2,i)],
[XY(1,j) XY(2,j)], [X1 Y1], [X2 Y2], corrcoff ), hold on
    colorbar
caxis([cutoff 1])
end
else
    plotnetwork( width, colors(u,:), [XY(1,i) XY(2,i)],
[XY(1,j) XY(2,j)], [X1 Y1], [X2 Y2], corrcoff ), hold on
    colorbar
caxis([cutoff 1])
end
end
end
close(h)
end
if PLOT2==1
%Preparing vectors.
cutoffvector=[];
correlationvector1=[];
distancevector1=[];
correlationvector2=[];
distancevector2=[];
h=waitbar(0,'Please wait... Plotting distance dependence for
correlations');
for i=1:L
    waitbar(i/L)
    for j=i:L
        if i~=j
            cutoffvector=[cutoffvector cutoffmatrix(i,j)];
corrcoff=correlationmatrix(i,j);
distance=10^6*distancematrix(i,j);
distancevector1=[distancevector1 distance];
correlationvector1=[correlationvector1 corrcoff];
        if corrcoff>=cutoffmatrix(i,j)
            distancevector2=[distancevector2 distance];
correlationvector2=[correlationvector2 corrcoff];
        end
    end
end
close(h)
figure
xline=smin:0.1:smax;
parabolafit=polyfit(distancevector1,cutoffvector,2);
parabola=polyval(parabolafit,xline);
plot(distancevector1,correlationvector1 ,
'k.',xline,parabola,'r'), hold
on
title('Correlation coefficient and cut off as a function of distance
and trends for distance dependence');
xlabel('Distance (microns)')
ylabel('Correlation coefficient')
plot(xline,parabola,'r',distancevector1,cutoffvector,'go'), hold on
linearfit1=polyfit(distancevector1,correlationvector1,1);
y1=polyval(linearfit1,distancevector1);
if length(correlationvector2)>1
    linearfit2=polyfit(distancevector2,correlationvector2,1);
y2=polyval(linearfit2,distancevector2);
end
plot(distancevector1,y1,distancevector2,y2);
legend('Data points', 'Cut off function', 'Linear fit for all
points', 'Linear fit for higher than cut off')
r1=corrcoef(correlationvector1,distancevector1);
r2=corrcoef(correlationvector2,distancevector2);
textlength=0.22*(max(distancevector1)-min(distancevector1));
text(smax-textlength,0.05,['Correlation for all data points:
',num2str(r1(1,2))])
text(smax-textlength,0.02,['Correlation for data points above cut
off: ',num2str(r2(1,2))])
else
    plot(distancevector1,y1)
    legend('Data points','Cut off function','Linear fit for all points')
    r1=corrcoef(correlationvector1,distancevector1);
    textlength=0.22*(max(distancevector1)-min(distancevector1));
    text(smax-textlength,0.05,['Correlation for all data points: ','num2str(r1(1,2))'])
end

end

if PLOT3==1
    figure
    pcolor(correlationmatrix)
    colorbar
    title('Correlation coefficient for different indices');
    xlabel('Cell i')
    ylabel('Cell j')
end

if PLOT4==1
    correlationvector1=[];
    h=waitbar(0,'Please wait... Scatter plot and box plot');
    for i=1:L
        waitbar(i/L)
        for j=i:L
            if i~=j
                corrcoff=abs(correlationmatrix(i,j));
                correlationvector1=[correlationvector1 corrcoff];
            end
        end
    end
    close(h)
    figure
    xline=0:0.001:2;
    yline=cutoff;
    plot(xline,yline,'r'), hold on
    plot(1,correlationvector1,'k.'), hold on
    boxplot(correlationvector1)
    axis([0 2 0 1])
    title('Correlation coefficients and boxplot for data');
    xlabel('Experiment 1')
    ylabel('Correlation coefficients 0<r<1')
end

if PLOT5==1
    figure
    pcolor(lagmatrix)
    colorbar
    title('Lags for different indices');
    xlabel('Cell i')
    ylabel('Cell j')
end

if PLOT6==1
    lagvector1=[];
    distancevector1=[];
    h=waitbar(0,'Please wait... Plotting lag as a function of distance for all cells');
    for i=1:L
        waitbar(i/L)
        for j=i:L
            if i~=j
                lag=abs(lagmatrix(i,j));
                distance=10^6*distancematrix(i,j);
                lagvector1=[lagvector1 lag];
                distancevector1=[distancevector1 distance];
            end
        end
    end
    close(h)
    linearfit1=polyfit(distancevector1,lagvector1,1);
y1 = polyval(linearfit1, distancevector1);
figure
plot(distancevector1, lagvector1, 'k.'), hold on
plot(distancevector1, y1, 'r-')
title('Lag as a function of distance');
xlabel('Distance (microns)');
ylabel('Lag (s)');
legend('Datapoints', 'Linear fit')
r1 = corrcoef(distancevector1, lagvector1);
textlength = 0.3 * (max(distancevector1) - min(distancevector1) - 20, ['Correlation: ', num2str(r1(1,2))])
lagvector2 = [];
distancevector2 = [];
h = waitbar(0, 'Please wait... Plotting lag as a function of distance for cells above cut off');
for i = 1:L
    waitbar(i/L)
    for j = i:L
        if i ~= j
            corrcoff = correlationmatrix(i, j);
            distance = 10^6 * distancematrix(i, j);
            lag = abs(lagmatrix(i, j));
            if corrcoff >= cutoffmatrix(i, j)
                lagvector2 = [lagvector2, lag];
                distancevector2 = [distancevector2, distance];
            end
        end
    end
end
close(h)
linearfit2 = polyfit(distancevector2, lagvector2, 1);
y2 = polyval(linearfit2, distancevector2);
figure
plot(distancevector2, y2, 'r-')
title('Lag as a function of distance for correlations significantly higher than cut off');
xlabel('Distance (microns)');
ylabel('Lag (s)');
legend('Datapoints', 'Linear fit')
r2 = corrcoef(distancevector2, lagvector2);
textlength = 0.3 * (max(distancevector2) - min(distancevector2) - 20, ['Correlation: ', num2str(r2(1,2))])
end
if PLOT7 == 1
    lagvector1 = [];
correlationvector1 = [];
h = waitbar(0, 'Please wait... Plotting lag as a function of correlation for all cells');
for i = 1:L
    waitbar(i/L)
    for j = i:L
        if i ~= j
            lag = abs(lagmatrix(i, j));
            corrcoff = correlationmatrix(i, j);
            lagvector1 = [lagvector1, lag];
correlationvector1 = [correlationvector1, corrcoff];
        end
    end
end
close(h)
linearfit1 = polyfit(correlationvector1, lagvector1, 1);
y1 = polyval(linearfit1, correlationvector1);
figure
plot(correlationvector1, y1, 'k.', correlationvector1, y1)
title('Lag as a function of correlation');
xlabel('Correlation')
ylabel('Lag (s)')
legend('Data points', 'Linear fit')
r1=corrcoef(correlationvector1, lagvector1);
textlength=0.3*(max(correlationvector1) - min(correlationvector1));
text(max(correlationvector1) - textlength, min(lagvector1) - 20,
    ['Correlation: ', num2str(r1(1,2))])
lagvector2=[];
correlationvector2=[];
h=waitbar(0, 'Please wait... Plotting lag as a function of correlation
for cells above cut off');
for i=1:L
    waitbar(i/L)
    for j=i:L
        if i~=j
            corrcoff=corrcoefmatrix(i, j);
            lag=abs(lagmatrix(i, j));
            if corrcoff>cutoffmatrix(i, j)
                lagvector2=[lagvector2 lag];
                correlationvector2=[correlationvector2 corrcoff];
            end
        end
    end
end
close(h)
linearfit2=polyfit(correlationvector2, lagvector2, 1);
y2=polyval(linearfit2, correlationvector2);
figure
plot(correlationvector2, lagvector2, 'k.', correlationvector2, y2)
title('Lag as a function of correlation for correlations significantly
higher than cut off.');
xlabel('Correlation')
ylabel('Lag (s)')
legend('Data points', 'Linear fit')
r2=corrcoef(correlationvector2, lagvector2);
textlength=0.3*(max(correlationvector2) - min(correlationvector2));
text(max(correlationvector2) - textlength, min(lagvector2) - 20,
    ['Correlation: ', num2str(r2(1,2))])
else
    disp('Warning! The correlation matrix is not symmetric! Check for NaNs!')
end
end
%Comparison of different experiments.
if comparison==1 && analyse==0
    % Reads a file with a matrix containing cell indices, correlation matrix,
    % distance matrix, lagmatrix, coordinates and number of time points.
    figure
    xline=0:0.001:(2*numberofexperiments+1);
yline=cutoff;
    plot(xline, yline, 'r'), hold on
    for i=1:numberofexperiments
        filename=['correlationdata', num2str(i), '.dat'];
data=load(filename);
M=size(data);
correlationmatrix=data(2:M(2)+1,:);
    if correlationmatrix==correlationmatrix'
correlationvector1=[];
correlationvector2=[];
    for k=1:M(2)
        for l=k:M(2)
            if k~=1
                corrcoff=abs(correlationmatrix(k, l));
correlationvector1=[correlationvector1; corrcoff];
            if corrcoff>cutoff
                correlationvector2=[correlationvector2; corrcoff];
            end
        end
    end
end

77
end
end
plot(i,correlationvector1(:,1), 'ro'), hold on
plot(i+numberofexperiments,correlationvector2(:,1), 'bo'), hold on
boxplot(correlationvector1, 'positions', i, 'widths', 0.5, 'labels', {''}, 'notch', 'on'), hold on
boxplot(correlationvector2, 'positions', (i+numberofexperiments), 'widths', 0.5, 'labels', {''}, 'notch', 'on'), hold on
axis([0 (2*numberofexperiments+1) -0.1 1.1])
title('Correlation coefficients for different experiments')
xlabel('Different experiments')
ylabel('Correlation coefficients 0<r<1')
else
disp('Warning! The correlation matrix is not symmetric! Check for NaNs!')
end
end
end
% plotnetwork.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Visualisation of correlation in Calcium Signalling Networks

by Erik Nilsson MSc Engineering Physics

Version 1.0

Copyright 2008 by Erik Nilsson All Rights Reserved

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Program file: CorrelationVisualisation.m
Related needed files: plotnetwork.m;
Required file: correlationdata.dat
1st version: 2008-10-09, Karolinska Institutet, Stockholm, Sweden

Contact: ernil@kth.se

Last updated: 2008-10-09

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

- - - Main file for the CorrelationVisualisation program - - -
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Plot correlation network
Function that plots a straight line between cell i and cell j with width and color
depending on correlation coefficient.

Input:
width: the width of the line depending on correlation.
color: the color of the line depending on correlation.
p1: vector with coordinates for cell i p1=[x1 y1]
p2: vector with coordinates for cell j p2=[x2 y2]
p3: vector with coordinates for the left most part of plot window p3=[xmin ymin]
%p4: vector with coordinates for the right most part of plot window p4=[xmax ymax]
%corrcoff: correlation coefficient between cell i and j (0<corrcoff<1).

%Output:
%Plots a straight line between cell i and cell j with width and color
%depending on correlation coefficient.

%--------------------------------------------------------
function z=plotnetwork(width,colors,p1,p2,p3,p4,corrcoff)

x1=p1(1);y1=p1(2);
x2=p2(1);y2=p2(2);
xmin=p3(1);ymin=p3(2);
xmax=p4(1);ymax=p4(2);

%Coordinates for text.
X=(x1+x2)/2;
Y=(y1+y2)/2;

%Slope of line and intercept.
if x1~=x2
    k=(y2-y1)/(x2-x1);
m=y1-k*x1;
end
if x1<x2
    x=x1:0.01:x2;
y=k*x+m;
end
if x1>x2
    x=x2:0.01:x1;
y=k*x+m;
end

if x1==x2
    P1=plot(x,y,'color','r');
    axis([xmin xmax ymin ymax])
    set(P1,'Color',colors,'LineWidth',width);
    P2=plot(x1,y1,'color','r');
    set(P2,'Color',colors,'LineStyle','o');
    P3=plot(x2,y2,'color','r');
    set(P3,'Color',colors,'LineStyle','o');
else
    text(X,Y,num2str(corrcoff))
end
xlabel('x coordinates')
ylabel('y coordinates')
title('Plot of network with strongly correlated cells')
end
9 References

6. A.B. Lansley & M.J. Sanderson *Regulation of airway ciliary activity by Ca2+: Simultaneous measurement of beat frequency and intracellular Ca2+* 1999 Biophysical Journal
7. M.J. Berridge *The AM and FM of calcium signalling* 1997 Nature
8. Memories Are Made of This: Modeling the CaMKII Molecular Switch 2005 PLoS Biology
19. Mathwork’s documentation on MATLAB in general and cross-covariance function xcov in particular.
24 J.C. Shaw On the application of correlation theory to signal analysis 1967 Medical and Biological Engineering
28 J.T.H. Mandeville, R.N. Ghosh & F.R. Maxfield Intracellular Calcium Levels Correlate with Speed and Persistent Forward Motion in Migrating Neutrophils 2005 Biophysical Journal
30 A. Aguilo, T.H. Schwartz, V.S. Kumar, Z.A. Peterlin, A. Tsiola, E. Soriano & R. Yuste Involvement of Cajal-Retzius Neurons in Spontaneous Correlated Activity of Embryonic and Postnatal Layer 1 from Wild-Type and Reeler Mice 1999 Journal of Neuroscience
32 Unpublished data provided by S. Malmersjö 070207 (a) & 070208 (b) embryonic stem cells from mouse day 13 (a) & 14 (b)
33 Unpublished data provided by S. Malmersjö 2007-2008 embryonic stem cells from mouse
34 Unpublished data provided by S. Malmersjö 2007-2008 embryonic stem cells from human