Bachelor's Thesis

Modeling Calcium Dynamics in T Cells

submitted to the

Institute of
Analysis and Scientific Computing
TU Wien

under the supervision of

Assistant Prof. Dr. Andreas Körner

by

Ida Hönigmann
Matriculation number: 12002348
TODO Adresszeile1

Vienna, September 20, 2024

Acknowledgement

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Bachelorarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Wien, am 20. September 2024

Ida Hönigmann

Contents

1 Introduction

Research questions:

- Which criteria can distinguish between unactivated, activated and pre-activated cells?
- Do different types of activated cells exists? How are they different?
- With which frequencies does the Calcium concentration repeat after activation?
- Is there a difference in frequencies between mouse and human cells?

2 Optimization Algorithm

An optimization problem is any problem where a function $f : X \to Y$ is given, and we search for the point $x \in X$ such that $f(x)$ is minimal or maximal. Obviously the minimum or maximum must not exist, as the example $f : (0,1) \to \mathbb{R}, x \mapsto x$ demonstrates by not having either. Investigating conditions on X, Y and f such that a minimum or maximum exists is mathematically interesting. However, when implementing an optimization algorithm the true minimum or maximum can sometimes not be found even if it exists and is instead replaced by a sufficiently good approximation.

2.1 Gradient Descent

An iterative algorithm for finding the minimum of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is gradient descent. As the name suggests it uses information of the gradient ∇f. Locally the negative gradient always points into the direction of greatest descent. The idea is to follow this direction for the next guess of the minimum. The pseudocode of this approach is given below.

Algorithm 1: Gradient Descent

```
input  : $f : \mathbb{R}^n \to \mathbb{R}$ ... differentiable, $x_0 \in \mathbb{R}^n$
output: $x \in \mathbb{R}^n$
begin
    for $n = 0$ to $max\_iterations$ do
        if improvement is smaller than threshold then
            break
        end
        set or calculate step size $\gamma_n$
        $x_{n+1} = x_n - \gamma_n \nabla f(x_n)$
    end
    $x = x_n$
end
```

If we consider a function with a local but not global minimum gradient descent might not converge to the optimum. An example of such a function can be seen in figure 2.1 along with the first values x_n of gradient descent for a starting value not converging to the global minimum.

Improvements can be made by choosing good step sizes, starting value or by starting with different values and comparing the results.

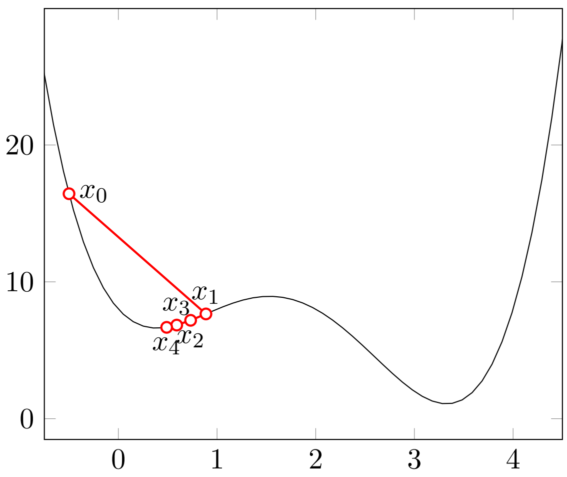

Figure 2.1: The function has two local minimums. For this starting value and step size gradient descent approaches the local, but not global minimum.

2.2 Least Square Problem Algorithms

We now focus on the Least Square Problem and give an introduction into various algorithms used.

In the example dealt with in this work we are given some data points $((x_k, y_k))_{k\in\{1,2,...,n\}}$ and want to find a close approximation in the form of a function $g(x, a_1, a_2, ..., a_m)$ where for every $a = (a_1, ..., a_m)$ we have a function $g_a(x) : \mathbb{R} \to \mathbb{R}, x \mapsto g(x, a_1, ..., a_m)$. Searching for a good approximation can be reformulated as searching for the minimum of $r(a) := \sum_{k=1}^{n} |g_a(x_k) - b_k|^2$ or any other error function. This form of optimization problem is called the Least Square Problem.

First we want to first think about some variations of the problem. Easiest to solve are linear problems. These can be formulated as minimize $||Ax - b||^2$ and solved using Calculus by $x = (A^T A)^{-1} A^T b$ if the rank of A is full.

Often we want to constrain the search for a minimum under some property. For linear problems we can find a formulation as

$$\text{minimize } ||Ax - b||^2 \text{ subject to } Cx = d.$$

Finding a solution can be done by minimizing $||Ax - b||^2 + \lambda||Cx - d||^2$ for very large λ.

General least square problems are formally given a as a residual function $r_f(x)$ which tells us whether a function f is a good approximation at the point x. We therefore want to find a way to minimize $||r(x)||^2$.

2.2.1 Gauss–Newton Algorithm

The idea behind this algorithm is that it is easy to find the intersection with zero of a linear function. If we linearize $r : \mathbb{R}^n \to \mathbb{R}^m$ locally we can approximate the root by finding it of the linear approximating function. This is demonstrated in figure 2.2.

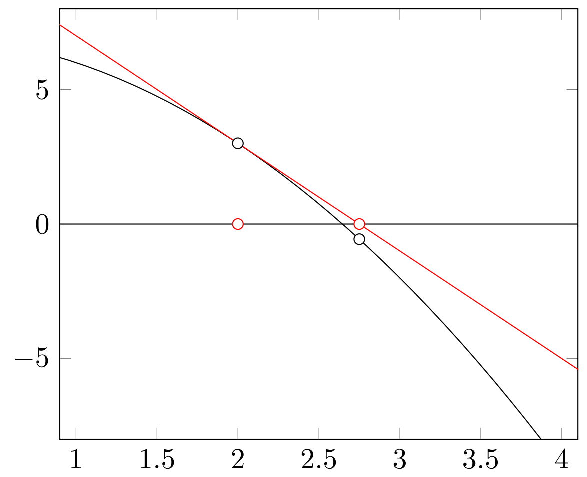

Figure 2.2: By approximating the black function by a line an approximation of the root has been found.

Iterating this step of linear approximating gives us the Gauss-Newton Method. In figure 2.3 we can see that indeed x_n seems to converge towards the root of the function.

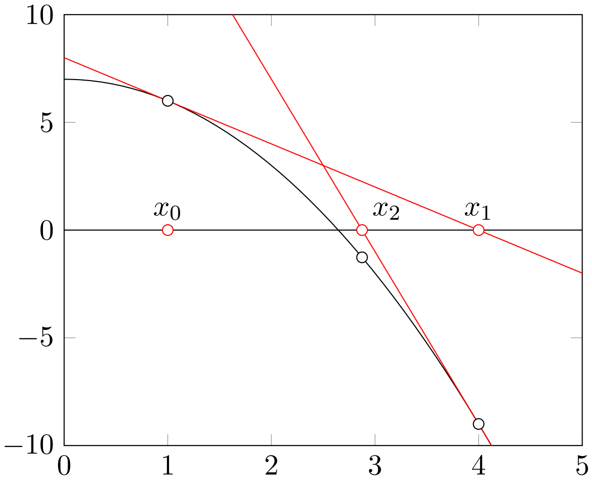

Figure 2.3: Iteratively applying linear approximation gives the Gauss-Newton Method for approximating the root.

Define Dr as the Jacobian matrix $\left(\frac{\partial r_i}{\partial x_j}\right)_{ij}$. Using Taylor's theorem we get the linear approximation

$$r(x) = r(a) + Dr(a)(x-a) + h(x)(x-a) \approx r(a) + Dr(a)(x-a) \text{ with } \lim_{x\to a} h(x) = 0.$$

Rewriting this as $r(x) \approx Ax - b$ where $A := Dr(a)$ and $b := Dr(a)a - r(a)$ gives us the algorithm for this method. As $Dr \in \mathbb{R}^{n\times m}$ we solve $Dr^T Drx = Dr^T b$ in order to get a system with square matrix. If $n = m$ we can skip this step and get the so-called Newton algorithm as a variant.

Algorithm 2: Gauss-Newton

```
input  : $r : \mathbb{R}^n \to \mathbb{R}^m$ ... differentiable, $x_0 \in \mathbb{R}^n$
output: $x \in \mathbb{R}^n$

begin
    for $n = 0$ to $max\_iterations$ do
        if $||r(x_n)||^2$ close enough to zero or $||x_n - x_{n-1}||$ is too small then
            break
        end
        Calculate $A_n := Dr(x_n)$
        Calculate $b_n := A_n x_n - r(x_n)$
        Solve $A_n^T A_n x_{n+1} = A_n^T b_n$
    end
    $x := x_n$
end
```

Gauss-Newton is guaranteed to find a local minimum x if r is twice continuously differentiable in an open convex set including x, Dr has a full rank and the initial value is close enough to x.

For the example demonstrated in figure 2.4 we can see that choosing a particular starting value leads to a loop in which only two points are explored as possible roots. More extreme examples exists in which Gauss-Newton gets increasingly further away from the root, due to an increasingly flat incline the further we get from the root. One example of such a function can be seen in figure 2.5.

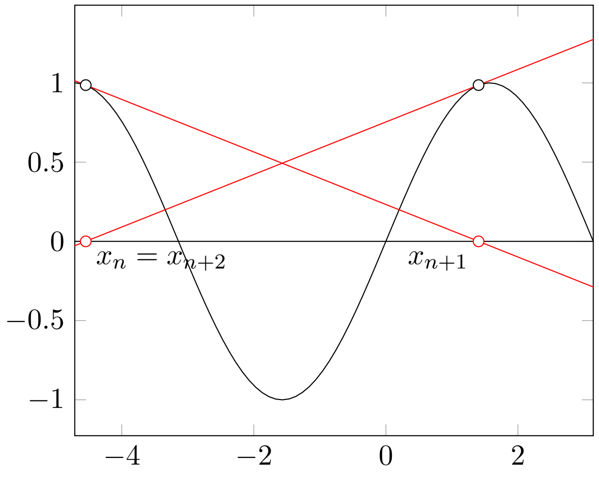

Figure 2.4: For a poor choice of starting values Gauss-Newton can never find the root of the function $\sin(x)$.

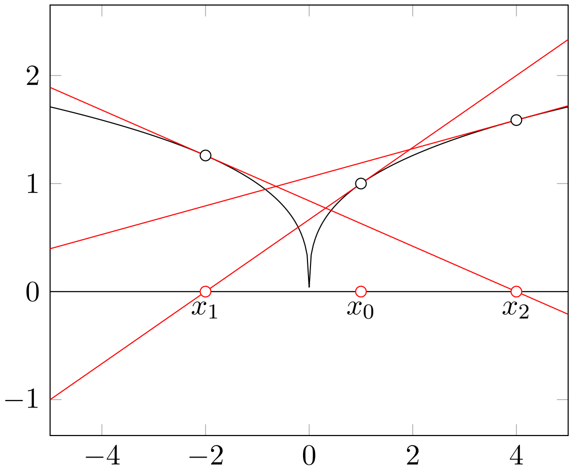

Figure 2.5: Finding the root of the function $\sqrt[3]{|x|}$ using Gauss-Newton is only possible if the starting value x_0 is chosen as 0, which is the root. For any other value we have that the guess gets further and further away. Indeed for any x_n we have $x_{n+1} = -2x_n$.

Gauss-Newton has two problems, the starting value being too far from the root and the Dr not having full rank. This can be combated using the technique of dampening. Instead of moving the new guess all the way to the root of the linear approximation we only move part of the way. How much can be determined by a dampening factor λ_n or a constant λ.

2.2.2 Levenberg-Marquardt Algorithm

This section is following section 18.3 of the book Introduction to Applied Linear Algebra by Stephen Boyd and Lieven Vandenberghe[BV18].

As stated above a shortcoming of Gauss-Newton is that for x far from x_n we must not have that $r(x) \approx r(x_n) + Dr(x_n)(x - x_n) =: \hat{r}(x, x_n)$. Levenberg-Marquardt addresses this by minimizing $||\hat{r}(x, x_n)||^2 + \lambda_n ||x - x_n||^2$. The first is the same as above, while the second objective expresses our desire to not stray away too much from the region where we trust the linear approximation. The parameter λ_n is a positive parameter specifying how far the trusted region extends.

Writing the above idea as a single squared norm to minimize gives us the problem

$$\text{minimize} \quad \left|\left| \begin{pmatrix} Dr(x_n) \\ \sqrt{\lambda_n} I \end{pmatrix} x - \begin{pmatrix} Dr(x_n)x_n - r(x_n) \\ \sqrt{\lambda_n} x_n \end{pmatrix} \right|\right|^2 .$$

We observe that as λ_n is positive the left matrix has full rank. From this it follows that a unique solution exists.

The change of including λ_n translates into the algorithm as replacing solving $A_n^T A_n x_{n+1} = A_n^T b_n$ in Gauss-Newton by solving $A_n^T A_n z + \lambda_n z = A_n^T b_n + \lambda_n x_n$.

The question of how to choose λ_n arises. If too small x_{n+1} can be too far from x_n to trust the approximation. If too big the convergence will be slow. If in the previous step the objective $||r(x_n)||^2$ decreased we decrease λ_{n+1} slightly. If the last step was not successful λ_n was too small. Therefore we increase λ_{n+1}.

Pseudo code of the resulting Levenberg-Marquardt algorithm is shown below. The stopping criteria of $||2Dr(x_n)^T r(x_n)||$ being too small is known as the optimality condition. It is derived from the fact that $2Dr(x)^T r(x) = \nabla||r(x)||^2 = 0$ holds for any x minimizing $||r(x)||^2$. Note that this condition can be met for points other that the minimum.

Algorithm 3: Levenberg-Marquardt

```
input  : $r : \mathbb{R}^n \to \mathbb{R}^m$ ... differentiable, $x_0 \in \mathbb{R}^n$, $\lambda_0 > 0$
output: $x \in \mathbb{R}^n$
begin
    for $n = 0$ to $max\_iterations$ do
        Calculate $A_n := Dr(x_n)$
        Calculate $b_n := A_n x_n - r(x_n)$
        if $||r(x_n)||^2$ close enough to zero or $||2A^T r(x_n)||$ is too small then
            break
        end
        Solve $(A_n^T A_n + \lambda_n)z = A_n^T b_n + \lambda_n x_n$
        if $||r(z)||^2 < ||r(x_n)||^2$ then
            $x_{n+1} := z$
            $\lambda_{n+1} := 0.8\lambda_n$
        else
            $x_{n+1} := x_n$
            $\lambda_{n+1} := 2\lambda_n$
        end
    end
    $x := x_n$
end
```

Coming back to the example where Gauss-Newton failed we once again consider the function from figure 2.5. In comparison this time we are able to find a good approximation of the root using Levenberg-Marquardt. A few steps are demonstrated in figure 2.6.

Further improvements to this algorithm can be made using good starting values perhaps from the output of other algorithms or by letting the algorithm run multiple times with different starting values and comparing the results.

2.2.3 Algorithms for Bounded Least Square Problems

The algorithms described above do not consider bounds. For bounded problems two algorithms know as Trust Region Reflective Algorithm and Dogleg Algorithm with Rectangular Trust Regions can be used.

Trust Region Reflective gets its name from the use of trust regions as in Levenberg-Marquardt as well as reflecting along the bounds[BCL99]. If an iterative x_n lands outside

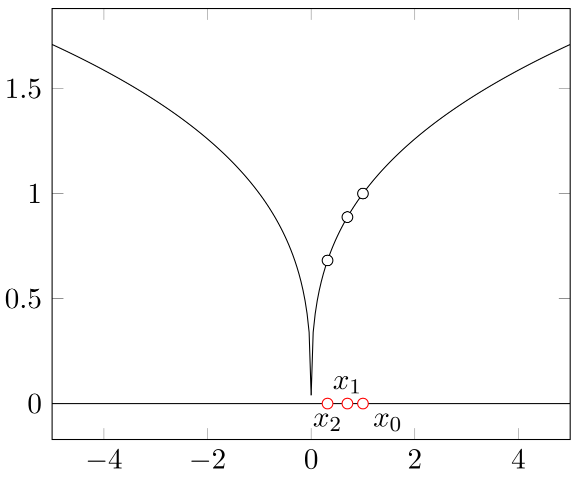

Figure 2.6: In comparison to Gauss-Newton, Levenberg-Marquardt is able to find the root of the function $\sqrt[3]{|x|}$. From the starting value $x_0 := 1$ and using $\lambda_0 := 1$ the guesses x_n move towards the root $x = 0$.

the bounds set, it is replaced by a reflected value within the bounds. This ensures each iterative is feasible as a solution.

As the name suggests the Dogleg Algorithm with Rectangular Trust Regions uses rectangular trust regions as opposed to ellipsoids.[VL04] As the bounds are specified as a rectangle to stay within, this results in the intersection of trust region and bounds to be rectangular. The resulting minimizing problem is solved with an adequate algorithm[NW99].

In Python the library SciPy provides the three methods Levenberg-Marquardt, Trust Region Reflective and Dogleg with Rectangular Trust Regions for solving Least Square Problems.

3 T Cells, Calcium Concentration

Lymphocytes form a key component of the immune system. T cells are a type of lymphocyte and are responsible for responding to viruses, fungi, allergens and tumours. Different subtypes of t cells exist, that perform various responsibilities. They are transported throughout the body via the lymphatic system and blood.[KCF18]

Precursor cells are formed in the bone marrow. Once they are transported to the thymus they undergo maturation and selection to become t cells. Each cell forms receptors, called t cell receptors (TCR), that respond to one particular out of many (10^6–10^9) possible short pieces of proteins, called peptides. These peptides are attached to the major histocompatibility complex (MHC) present on antigens and antigen presenting cells (APC). Important aspects of the selection are ensuring that the t cells react to foreign peptides, but not to those present on the body's own cells.[AH24]

In positive selection cells in the thymus present peptides on their MHC. If a t cell is unable to bind, it will undergo apoptosis, a type of cell death. T cells which were able to bind receive survival signals. Negative selection verifies that t cells will not attack the body's own cells. This is done by only selecting t cells which only bind moderately to the peptides presented, as a strong bond suggests that these t cells would have a high likelihood of being reactive to own cells.[Hag18] If a t cell passed both the positive and negative selection it is transported to the periphery.

There are multiple types of peripheral t cells. Native t cells respond to new antigens. Cytotoxic t cells kill cells which present peptides on their MHC compatible with the t cells TCR. Helper T cells activate other parts of the immune response. Memory t cells shorten the reaction time when the same antigen is encountered again at a later point in time. Suppressor t cells moderate the immune response.[Gan97]

3.1 Components of a T Cell

T cell components relevant in activation and subsequent changes in intracellular Ca^{2+} are listed below and schematically shown in figure 3.1.

- **T cell receptor (TCR):** Receptor on the cell surface that can recognize peptides. By the simultaneous triggering of the TCR and co-stimulator signalling is induced that leads to activation.
- **Co-stimulator:** A stimulation of co-stimulatory molecules is necessary in order for signalling to occur as part of activation.
- **Endoplasmic reticulum (ER):** A series of connected sacs in the cytoplasm that is attached to the nucleus. Important functions are folding, modification and transportation of proteins.[Rog24]

- **Ca^{2+} permeable ion channel on the ER:** There are several Ca^{2+} channels present on the ER. Some receptors are responsible for releasing Ca^{2+} into the cytoplasm, when the intracellular Ca^{2+} concentration is low. [SB16]
- **Ca^{2+} storage in the ER:** Ca^{2+} is stored in the ER and can be released by Ca^{2+} permeable ion channels on the ER.
- **Cytoplasm:** The semi-fluid substance enclosed in the plasm membrane. It contains organelles, ions, proteins and molecules.
- **Stromal interaction molecule (STIM):** If the Ca^{2+} storage in the ER is depleted STIM proteins cluster where the ER is in the vicinity of the plasm membrane and assembles CRAC, which then leads to uptake in extracellular Ca^{2+}. [SB16]
- **Plasm membrane:** A semipermeable structure forming the wall of the cell made up of lipids and proteins. Ion channels and transport proteins allow certain substances to move through.[Gan12]
- **Ca^{2+} release activated Ca^{2+} channel (CRAC):** Opened after a decrease in ER stored Ca^{2+} is sensed by STIM, these channels intake Ca^{2+} from outside the cell.[SI13]
- **Cytoskeleton:** A system of fibres within the cell, that allows it to change shape and move.[Gan12]
- **Nucleus:** An organelle that stores most of the DNA, controls cell growth and cell division. A double membrane separates it from the cytoplasm.[CA22]

Relevant components of APC are the

- **Major histocompatibility complex (MHC)**, which can present peptides, and the
- **Co-stimulator**, which can form a bond with the co-stimulator on a t cell.

Both are present on the surface of the APC.

3.2 Activation

Activation is necessary for t cells to divide and perform their functions.[Gan97]

When a native t cell encounters a peptide on an APC that is compatible, a bond is formed between the TCR on the t cell and the peptide-MHC complex on the APC. This recognition can be triggered by less than ten molecules of foreign substance and is therefore described as near perfect. Sufficiently long contact is necessary between the APC and the t cell in order for the t cell to activate. The role of contact time in t cell activation is modelled by Morgan et al..[ML23].

The presence of co-stimulatory molecules is needed for proper activation. The bond between the co-stimulatory molecules on the t cell and APC plays a role in signalling. Ca^{2+} signals play a vital part in t cell activation.

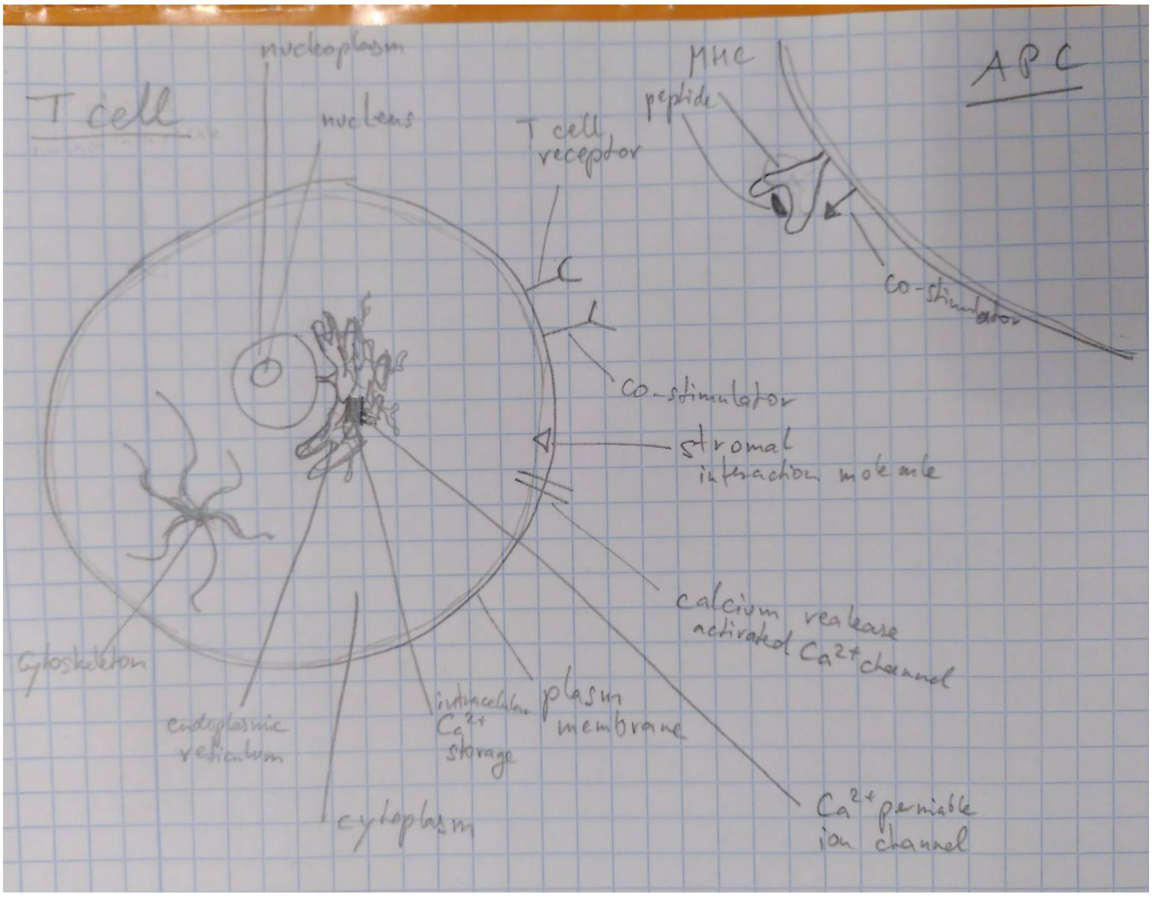
T cell
nucleoplasm
nucleus
T cell receptor
MHC
peptide
APC
co-stimulator
co-stimulator
stromal interaction molecule
calcium realease activated Ca^{2+} channel
cytoskeleton
endoplasmic reticulum
intracellular Ca^{2+} storage
plasm membrane
cytoplasm
Ca^{2+} permiable ion channel

Figure 3.1: Schematic view of a t cell and antigen presenting cell, with all relevant components.

An increase of Ca^{2+} in t cells during activation is caused by the stimulation of Ca^{2+} permeable ion channel receptors on the ER membrane. Ca^{2+} is released from the ER into the cytoplasm. Additionally, this decrease in Ca^{2+} is sensed by STIM, which leads to an influx of Ca^{2+} through plasma membrane CRAC channels.[SKJ09]

As the intracellular Ca^{2+} concentration is dependent on the interaction between Ca^{2+} sources and sinks, a variety of different forms in Ca^{2+} concentration have been observed. Examples are infrequent spikes, sustained oscillations and plateaus.[Lew01]

Intercellular Ca^{2+} increase together with other signals lead to a redistribution of receptors, signalling molecules and organelles.[JRB14]

4 Data

From section 3.2, we gather that analysing the intracellular Ca^{2+} concentration gives us good insight in whether and when a cell activates. Additionally, it can be measured relatively easily by the method described in this chapter.

4.1 Structure of Data

First we describe the structure of the data this work uses.

The data matrix has one row for each tracked particle and frame combination. In this context cells are called particles as the recording might feature non-cells that are detected as a cell and recorded in the data set. The information stored for each particle and frame combination is described in detail in table 4.1.

Name	Data Type	Description
x	float64	Position of particle in pixels along the horizontal axis
y	float64	Position of particle in pixels along the vertical axis
frame	int32	Number of frame, with frame rate of 1 frame per second
mass short	float64	Brightness of cell in 340nm channel
bg short	float64	Background in 340nm channel
mass long	float64	Brightness of cell in 380nm channel
bg long	float64	Background in 380nm channel
ratio	float64	Calculated as mass short divided by mass long
particle	int32	Identification for each particle

Table 4.1: Description and data type of all columns present in the data matrix.

One recording can have between 500 and 10000 particles and is between 700 and 1000 frames long, which corresponds to between about 11 and 17 minutes. The ratio recorded is typically between 0 and 5.

Four recordings where generated, with two each from human and mouse cells. For each cell type a positive and negative control was measured. In a positive control the conditions are such, that in theory every cell should activate, while in negative control the conditions are such, that none should activate. Due to stress on the cells caused by the movement or changes in temperature and other factors a few cells will activate before the recording starts, during the recording in the negative control or not activate at all in the positive control, regardless of the conditions.

4.2 Jurkat Cells, 5c.c7 primary mouse T cells and Fura-2

The prototypical cell line to study T cell signalling is the Jurkat cell line.[ML23] It was obtained from the blood of a boy with T cell leukaemia.[SSB77] Different cell lines within the Jurkat family are described by Abraham and Weiss.[AW04] They provide a timeline of discoveries linked to Jurkat cells and t cell receptor signalling.

Another type of T cells used in signalling studies are gathered from mice. [Additional information]

In order to be able to measure the intracellular Ca^{2+} concentration of cells they can be labelled with Fura-2. This method provides a way to record the Ca^{2+} concentration of multiple cells over a time period.[MMS17] Challenges encountered when using Fura-2 on certain cell types are described by Roe, Lemasters and Herman along with their respective solutions.[RLH90]

4.3 Measuring Calcium Concentration

After the cells have been labelled with Fura-2, a recording of up to 15 to 20 minute can be generated. To achieve this the cells and stimulant are photographed at both 340nm and 380nm wavelength once per second. The resolution of the images are 1.6um per pixel. By calculating the ratio of the two images at each pixel the Ca^{2+} concentration can be observed. An exemplary resulting image showing the ratio is shown in figure 4.1. The T cells appear a lighter shade than the background when activated and darker when not activated.

To activate the cells in the duration of the recording they are transferred to a plate covered with replicas of the MHC-peptide complex normally present on APCs. This plate is then recorded as described above. For a negative control the plate is not covered with peptides, while for the positive control the peptide covering on the plate is very dense. Recordings of different densities in peptides lead to activation of a percentage of t cells.

4.4 Processing

To track single t cells moving around during the video the sum of the 340nm and 380nm image of each second is calculated. This image provides the basis for separating t cells from the background. On this image all t cells will appear similarly light in colour. Therefore, it is used to track the movement of cells. Each cell is numbered, such that the same cell will have the same number during the video. For some cells the trajectory tracking is not perfect, resulting in a split of the numbering into multiple numbers for the same cell. The position and shade during both 340nm and 380nm as well as the ratio of each particle and each frame is then recorded into the data structure used in this work. The first roughly 50 frames at the start of the recording are discarded due to the video being out of focus. Additionally, cells only appearing in fewer than 300 frames are discarded as they most likely represent trajectories incorrectly tracked or split. The resulting data is then stored in a matrix structured as described in table 4.1.

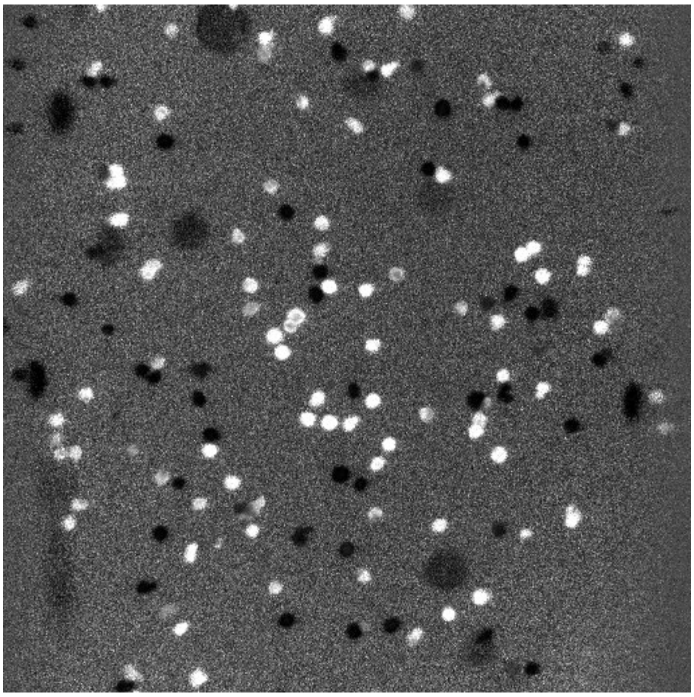

Figure 4.1: Single frame showing the ratio of the 340nm and 380nm images from a recording of human Jurkat cells. Activated cells appear lighter, unactivated cells darker than the background. Big dark circles are out of focus cells that have not yet settled on to the plate.

5 Approximating the Calcium Concentration

If we have a look at the typical trajectory of the calcium concentration in activated and unactivated cells, shown in figure 5.1, we can see differences emerging. For one the maximum concentration value reached by most activated cells is higher. Another distinguishing feature is the presence of a steep incline at the moment of activation.

By modelling the time series with a function incorporating features such as the increase, maximum value and oscillations present in the decrease afterwards, we can extract these features more easily. By doing this, using approximation methods from chapter 2, we want to answer the research questions from the introduction.

5.1 Approximation Function

From studying the data in the two control groups we find to expect a function close to

$$f_{unac}(x) := u \tag{5.1}$$

for unactivated cells and

$$f_{ac}(x) := \begin{cases} \frac{a-u}{1+e^{-k_1(x-w_1)}} + u & \textbf{if } x <= t \\ \frac{a-d}{1+e^{-k_2(x-w_2)}} + d & \textbf{else} \end{cases} \tag{5.2}$$

for activated cells. The parameters can be understood as described in table ??.

u ...	average value before activation,
a ...	value reached at the peak of activation,
d ...	average value after activation,
k_1 ...	steepness of increase,
k_2 ...	steepness of decrease,
w_1 ...	time point at which the increase happens,
w_2 ...	time point at which the decrease happens,
t ...	time point at which the increase ends, and the decrease starts,

Table 5.1: List of parameters and their interpretation.

Figure 5.2 shows how the above functions 5.1 and 5.2 look and how the relations to the parameters are in unactivated and activated cells.

For our model to make sense we have to impose some conditions onto the parameters. We expect $0 \leq u \leq d \leq a$, $w_1 \leq t \leq w_2$, $k_1 > 0$ and $k_2 < 0$.

There are multiple ways in which the parameters of f_{ac} can be chosen to get a function similar to f_{unac}. If w_1 is very large or $u \approx d \approx a$ then f_{ac} approaches a constant value of

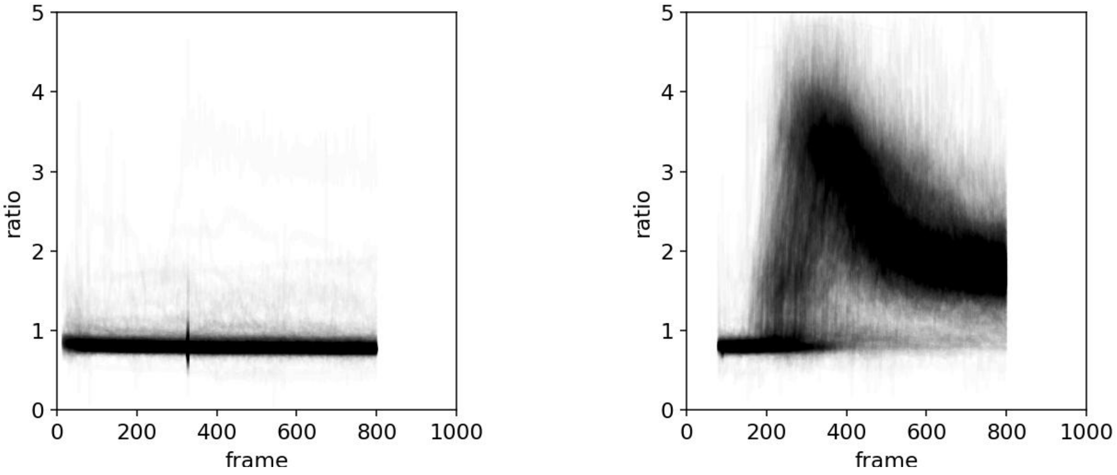

Figure 5.1: Two plots of the overlapping calcium concentration time series of cells. On the left a negative control and on the right a positive control of mouse cells.

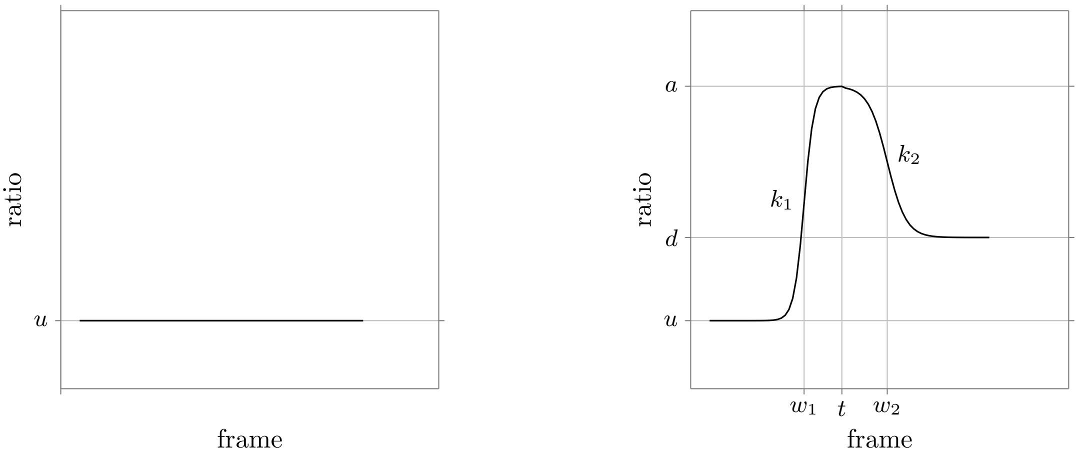

Figure 5.2: Left shows the function f_{unac} defined in 5.1 with the parameter u. The right shows the function f_{ac} defined in 5.2 with the parameters u, d, a, w_1, t, w_2, k_1 and k_2.

u, thus approximating f_{unac}. If the approximation of a cell has parameters with w_1 very large or $u \approx d \approx a$ we can therefore expect it to be of an unactivated cell. Otherwise, it is more probable to be activated.

5.2 Implementation

Now that we have defined our model functions we will implement a routine that fits such a f_{ac}-function through the data of a cell.

First we describe a routine which handles reading the data, some necessary preprocessing steps and saving of the resulting parameter lists. The approximation itself is wrapped in the function `particle_to_parameters`, which takes a (frame, ratio)-matrix and returns a parameter list.

Algorithm 4: Main

```
   output: parameter matrix
1  begin
2      read data
3      filter data
4      for each single particle do
5          particle data := (frame, ratio) columns of this particle
6          if length of particle data is too short then
7              skip
8          end
9          parameters := approximate(particle data)
10         show ratio data and approximation
11         save parameters
12     end
13 end
```

Filtering the data is necessary as the ratio can be very large if the denominator can be small. Values are therefore cropped to lie within $[0, 5]$. Any values higher than 5 are almost certainly caused by measurement errors.

The visualization of line 10 generates images such as figure 5.3.

The above-mentioned approximation function is described in pseudocode below.

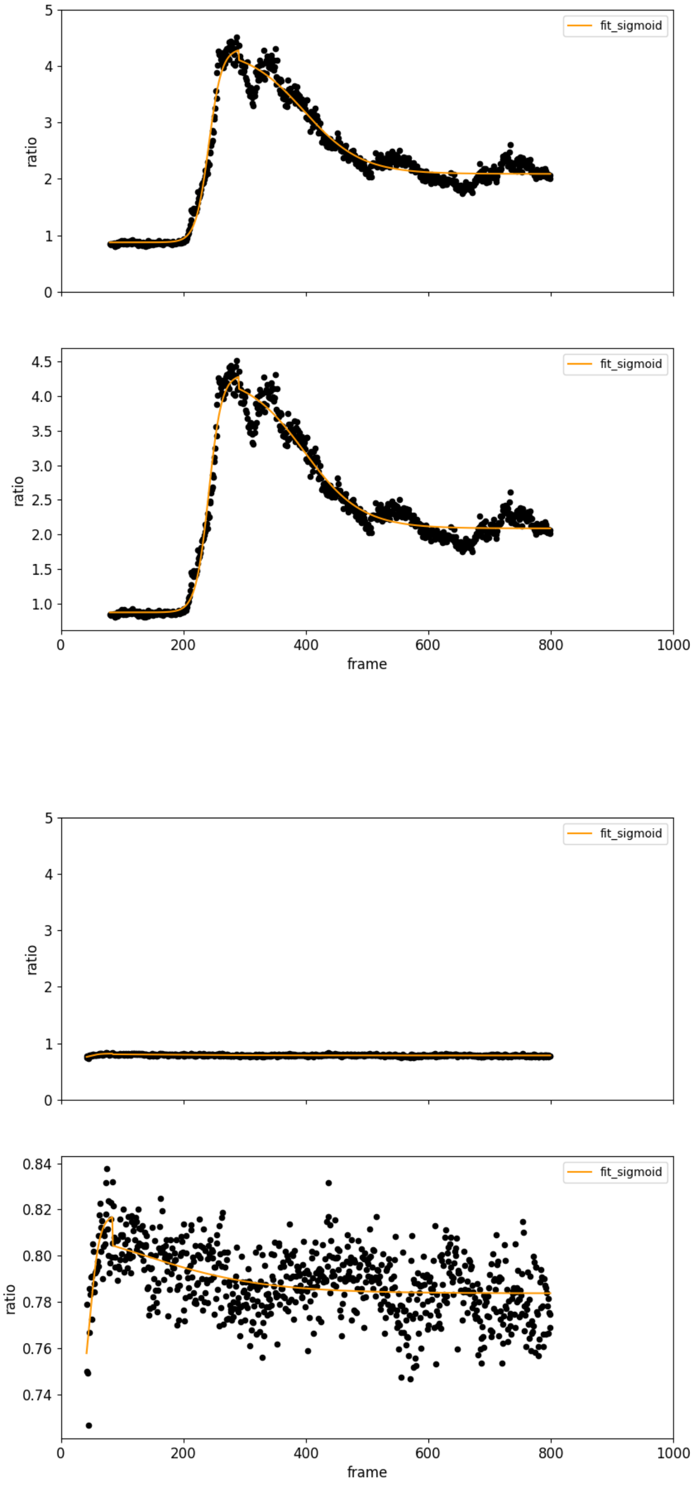

Figure 5.3: The upper plot shows the data in black and approximation in orange of an activated cell. The first plot is scaled from 0 to 5, the second one is scaled to fit the data. The lower plot shows the same of an unactivated cell.

Algorithm 5: Approximate

```
  input  : particle data as (frame, ratio) matrix
  output: parameters
1 begin
2     set boundaries for parameters
3     set start values for parameters
4     use Trust Region Reflective Algorithm with boundaries and start values to get
        parameters
5     calculate corresponding approximation and add as fit_sigmoid columns to data
        matrix
6     return paramters
7 end
```

The parameters of f_{ac} used in the approximation are not independent of each other if we choose t to be the point at which $(a-u)/(1+e^{-k_1(x-w_1)})$ almost reaches the value a. We choose $t := w_1 - log(1/0.99-1)/k_1 = w_1 - log(1/99)/k_1$ as the function has had 99% of the increase of the sigmoid curve up to this point.

Setting the boundaries in line 2 is non-trivial. We have noted that a condition such as $0 \leq u \leq d \leq a$, $w_1 \leq t \leq w_2$, $k_1 > 0$ and $k_2 < 0$ are expected. We want to impose them using boundaries in which the parameters must lie. However, boundaries for each parameter must not depend on other parameters. We can circumvent this by changing the parameters to be relative to each other. As $u \leq d \leq a$ we choose to use the three parameters $u, d-u$ and $a-d$. We can then set the lower boundary to be 0 which ensures

$$0 \leq u \qquad \wedge \qquad 0 \leq d-u \implies d \geq u \qquad \wedge \qquad 0 \leq a-d \implies a \geq d$$
$$\implies 0 \leq u \leq d \leq a.$$

Using the same method, we choose the parameters $w_1 - start$ and $w_2 - w_1$, where $start$ is the first frame in which the particle was tracked. The resulting boundaries are described in table ??, where we set min val, max val and median val as the minimum, maximum and median of the particles' ratio data respectively while start and end is the first and last frame where data was recorded for this particle.

The condition $t \leq w_2$ can be violated, but it is ensured that at least $w_1 \leq w_2$.

The other conditions are met as $k_1 \in [0.05, 10] \implies k_1 > 0$ while $k_2 \in [-1, -0.01] \implies k_2 < 0$ and

$$t = w_1 - \underbrace{log(1/99)/k_1}_{<0} \geq w_1.$$

Starting values can have a big impact on the approximation reached by the algorithm. We want to choose starting values close to the expected resulting parameters. By choosing

parameter	lower bound	upper bound	starting value
u	min val	max val	min val
$d - u$	0	max val	median val - min val
$a - d$	0	max val	max val - median val
$w_1 - start$	0	end - start	0
$w_2 - w_1$	0	end - start	(end - start) / 2
k_1	0.05	10	0.1
k_2	-1	-0.01	-0.03
d	min val	2 max val	median val
a	min val	3 max val	max val
w_1	start	end	start
w_2	start	2 end - 2 start	(start + end)/2

Table 5.2: Upper and lower bounds as well as starting value for each of the parameters. The boundaries and starting values of d, a, w_1 and w_2 are derived from the parameters used in the implementation of the approximation, shown above the double line.

the starting value of $w_1 - start$ as 0, which corresponds to choosing $w_1 = start$, we favour the first increase in the data to be the point of activation. Otherwise, we are more likely to mistake an oscillation later in the data as the activation point. As we do not know when the activation happens when setting the boundaries we guess that w_2 will lie somewhere in the middle. Therefore we choose $(end - start)/2$ as the starting value for $w_2 - w_1$. The other starting values are chosen as we expect u to be low, a to be high, d to lie somewhere in the middle. Experimenting showed that k_1 often has a value around 0.1 while k_2 lies around -0.03.

This work uses the python scipy function `scipy.optimize.curve_fit(function, xdata, ydata, p0=starting_values, method='trf', bounds=(lower_bounds, upper_bounds))` as it provides all the necessary functionality. The method parameter `trf` stands for Trust Region Reflective, as described in section 2.2.3.

5.3 Analysis of the Approximation

We now give data on the parameters found from the above approximation. Some statistics are found in table ??. Figure 5.4 shows the distribution of the resulting parameters of the approximation. From the figure it seems the differences between activated and unactivated cells is biggest in the parameters activated value a and decreased value d and the steepness of increase k_1.

As the datasets are not perfectly labelled, meaning there are activated cells in the negative control and vice versa, we have relatively high standard deviation.

We can use the mean and standard deviation of each of the parameters to find data points that can be considered outliers. We expect wrongly-labelled data, e.g. activated cells in the negative control, to be an outlier in the parameters a and d. However, activated cells in the positive control might have a decreased value d that is very low, around u.

		Positive Control		Negative Control		
	Parameter	Average	Standard Deviation	Average	Standard Deviation	Difference
human cells	u	0.663	0.521	0.613	0.311	0.05
	a	2.808	0.461	0.923	0.669	1.885
	d	1.937	0.491	0.685	0.412	1.252
	w_1	142.228	124.012	171.062	131.563	-28.834
	w_2	445.386	185.971	478.843	190.792	-33.457
	k_1	0.263	0.428	0.524	0.963	-0.261
	k_2	-0.059	0.164	-0.163	0.292	0.104
mouse cells	u	0.889	0.27	0.79	0.093	0.099
	a	2.9	0.907	0.876	0.186	2.024
	d	1.749	0.407	0.804	0.129	0.945
	w_1	295.809	77.207	100.712	112.352	195.097
	w_2	469.952	105.375	304.283	179.834	165.669
	k_1	0.15	0.409	1.161	1.235	-1.011
	k_2	-0.1	0.195	-0.133	0.267	0.033

Table 5.3: Statistics of the parameters retrieved from approximating the human cell data.

This makes it difficult to distinguish activated from unactivated cells when looking at the parameter d. Therefore, we choose a as the only parameter when filtering for these kinds of outliers.

A particle from the positive control dataset that has a value in parameter a higher than the median should still be classified as activated. Only a value lower than some threshold indicates an unactivated cell. The same holds for values of a lower than the median in the negative control dataset. In short, we want to filter out particles with a high value of a in the negative control and those with a low value of a in the positive control dataset. The question of how to choose the threshold will be discussed next.

As we do not have information on what percentage of cells behaved correctly in the positive and negative control we do not have enough information to choose threshold values without guessing. Instead, we can manipulate the threshold as a multiple of the standard deviation until we filter out incorrectly labelled data, but would filter out correctly labelled data points if we increase the value. This trial and error approach led to different values for each of the four control datasets, which can be seen in table ??.

dataset	lower bound	upper bound
human positive	mean -3 std $= 1.582$	∞
human negative	$-\infty$	mean $+0.5$ std $= 1.306$
mouse positive	mean -2 std $= 1.41$	∞
mouse negative	$-\infty$	mean $+3$ std $= 1.445$

Table 5.4: Thresholds in outlier detection in the different datasets.

Naturally we can use the same outlier detection with different parameters to find particles where the approximation failed to yield a good result.

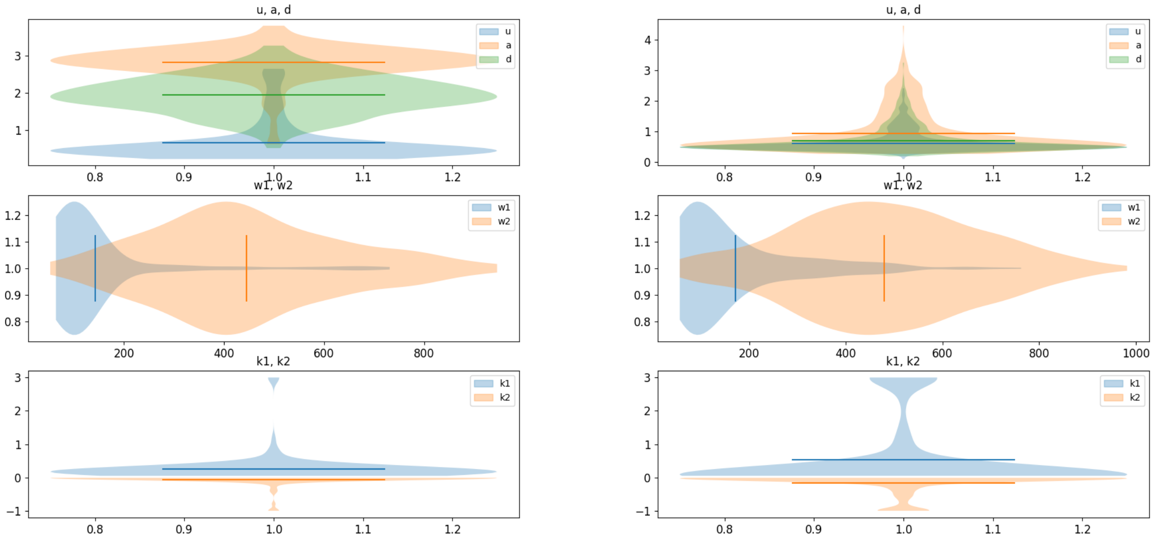

Figure 5.4: Violin plots of parameters u, a, d, w_1, w_2, k_1 and k_2 from the approximations. The parameters of the positive control are on the left and those of the negative control are on the right. Both are of human cells.

These results will be used in [TODO] to remove wrongly-labelled data from the datasets.

5.4 Adding Oscillation in the Decrease

In order to answer the research questions concerned with the oscillations happening in the decrease of the Ca^{2+} concentration we want to model them as well. We use a method often used when analysing oscillating data, called Fourier Transformation.

Many applications are concerned with cyclic temporal data. Examples are sound waves, seismic data or oscillations of a skyscraper in strong wind. This data can be represented as a function of amplitude over time. Most of the time we are not interested in the amplitude at a specific point in time, as a temporal shift would represent very similar information. Such a shift is demonstrated in figure 5.5.

As the function of sound waves or oscillations in the Ca^{2+} concentration in t cells is almost cyclic we might be interested in a decomposition into simple cyclic function, such as sine. We can then analyse the most prominent frequencies and their respective amplitudes. This gives a representation of the data, that can be easier to interpret. Fast Fourier Transformation (FFT) is an algorithm that transforms temporal data into such a representation of a weighted sum of sines.

As the oscillations happen in the decrease of the Ca^{2+} concentration we apply FFT to that part of the data. We can then filter out the 10 frequencies with the highest amplitudes and use them to further analyse the oscillations. This gives an even better approximation of the data, which can be seen in figure 5.6.

We can store the data gathered using the FFT as a list of frequencies and corresponding amplitudes.

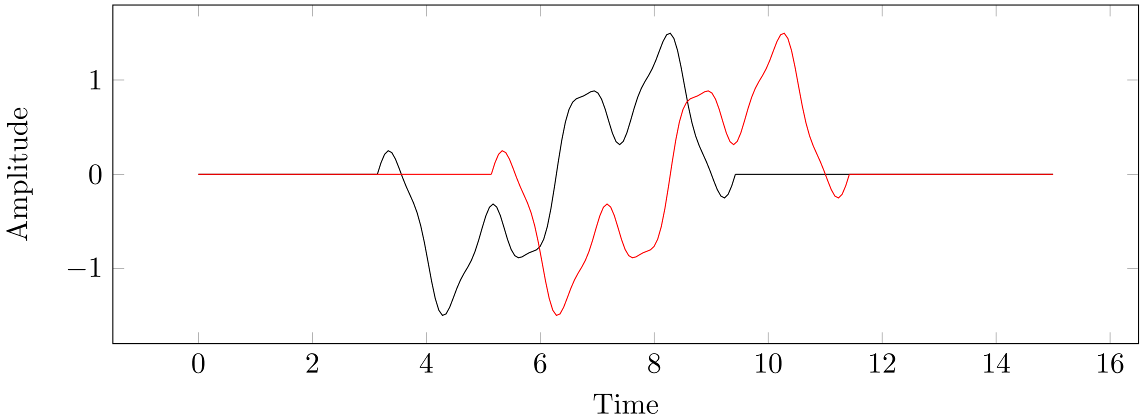

Figure 5.5: Two signals that differ by a temporal shift.

[TODO analyse frequencies and amplitudes]

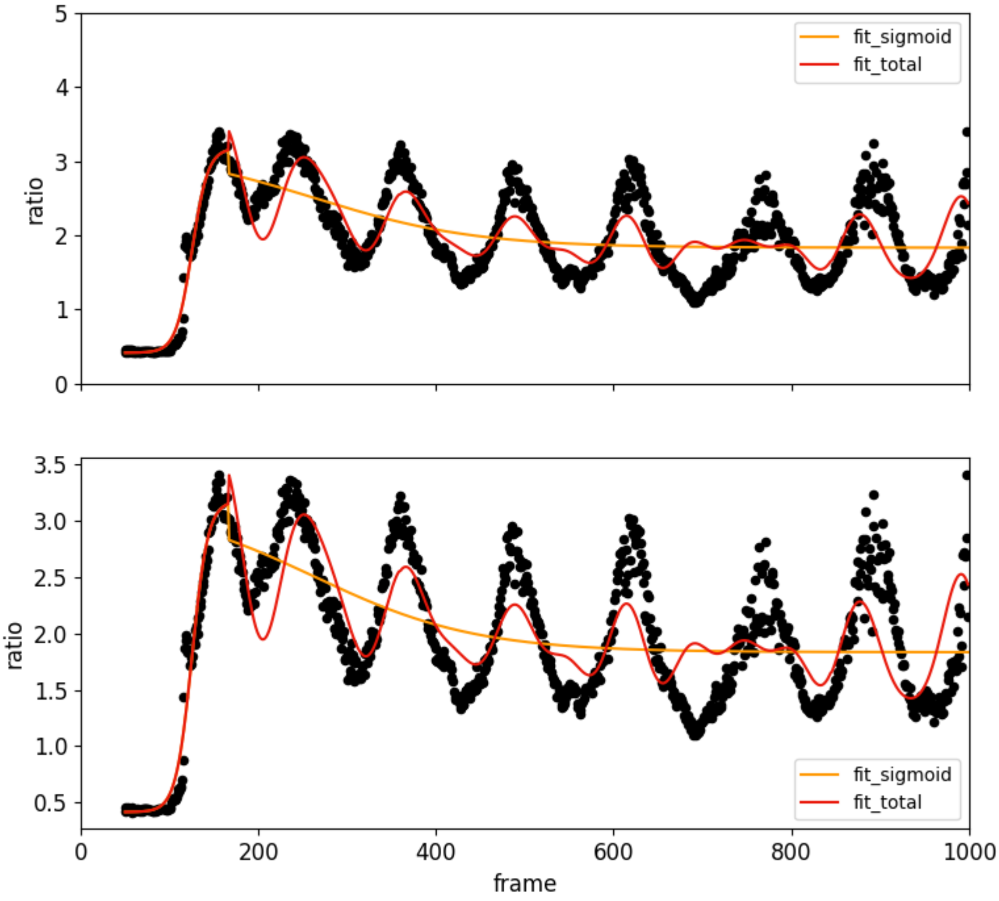

Figure 5.6: The data of an activated cell with heavy oscillation is shown in black, simple approximation in orange and the approximation with FFT added in red. The first plot is scaled from 0 to 5, the second one is scaled to fit the data.

6 Clustering

The objective of classification is to find assignments between data points and categories. For some applications this can be done by taking correctly labelled data and comparing a new data point to the data points in different categories to see which category best fits. One such algorithm is k-nearest-neighbour. In our context the issue with this approach is that the data is only labelled as to which experiment it came from, e.g. positive control in human cells, negative control in mouse cells. However, as noted before not all cells from these experiments behaved as we expected them to, e.g. some activated in the negative control or did not activate in the positive control. Therefore, we choose to use a clustering algorithm which does not have the need for classified training data.

6.1 Gaussian Mixture Model

This section follows the article Gaussian Mixture Model by Reynolds [Rey+09].

Often gathered observations are distributed as a normal distribution. These distributions have a density function

$$g(x|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

with parameters D as the dimension, μ as the mean vector and Σ as the covariance matrix.

As we are concerned with clustering data we expect the observed data from different clusters to have different parameters in the normal distribution they come from. Assuming we have n different data sources gives us normal distributions $g_i(x|\mu_i, \Sigma_i)$ where $i = 1, ..., n$. Additionally, we might have more data points being generated from some normal distributions while less from others. We can express this using another weight parameter w_i with $i = 1, ..., n$. To normalize the weights we set the constraint $\sum w_i = 1$.

The distribution describing the entire dataset now can be described with the distribution

$$p(x) = \sum_{i=1}^{n} w_i g(x|\mu_i, \Sigma_i). \tag{6.1}$$

Gaussian Mixture Model is a method to retrieve these parameters w_i, μ_i and Σ_i for some D dimensional data points generated from n normal distributions.

From these parameters it is easy to cluster the data as we know where data points from the different clusters are expected to lie.

From equation 6.1 we expect every Σ_i to be independent of each other. In the context of Gaussian Mixture Models this is called having a full covariance matrix. However, we can eliminate some of the variables in the covariance matrix if we choose a diagonal covariance matrix. Additionally, we might specify to use the same covariance matrix for all i, which is called tied in this context.

Choosing a full covariance matrix is not necessary even if the data is expected to have statistically independent features, as the overall density is compromised from multiple normal distributions with diagonal Σ_i. This enables us to model correlations between features.

The question now is how we can derive the parameters w_i, μ_i and Σ_i. We choose the approach which chooses the parameters where the likelihood that the data was generated by these parameters is maximal. This is known as maximum likelihood estimation. The likelihood can be expressed as

$$L(w_i, \mu_i, \Sigma_i | X) = p(X | w_i, \mu_i, \Sigma_i) = \prod_{t=1}^{n} p(x_t | w_i, \mu_i, \Sigma_i)$$

with $X = (x_1, ..., x_n)$ being the recorded data. As $L(w_i, \mu_i, \Sigma_i | X)$ is non-linear in the parameters deriving the maximum is not trivial. Instead, we use an iterative approach which approaches the solution. Define $\lambda = (w_i, \mu_i, \Sigma_i)$. Simplifying to a diagonal covariance matrix gives us the iterative algorithm where we define the successor values $\bar{\cdot}$ as

$$\begin{aligned} Pr(i|x_t, \lambda) &:= \frac{w_i g(x_t|\mu_i, \Sigma_i)}{\sum_{k=1}^{n} w_k g(x_t|\mu_k, \Sigma_k)} \\ \bar{w_i} &:= \frac{1}{n} \sum_{t=1}^{n} Pr(i|x_t, \lambda) \\ \bar{\mu_i} &:= \frac{\sum_{t=1}^{n} Pr(i|x_t, \lambda) x_t}{\sum_{t=1}^{n} Pr(i|x_t, \lambda)} \\ \bar{\sigma_i^2} &:= \frac{\sum_{t=1}^{n} Pr(i|x_t, \lambda) x_t^2}{\sum_{t=1}^{n} Pr(i|x_t, \lambda)} - \bar{\mu_i^2}. \end{aligned}$$

for w_i, μ_i and σ_i^2 respectively. One can show that with this iteration rule we have $p(X|\bar{\lambda}) \geq p(X|\lambda)$. The value $Pr(i|x_t, \lambda)$ is known as the a posteriori probability for the i-th component.

6.2 Implementation

Python offers an implementation of Gaussian Mixture Model with the sklearn package. The function with parameters relevant to us is `sklearn.mixture.GaussianMixture(n_components, covariance_type)`. The number of components `n_components` can be any positive integer. The `covariance_type` can be one of `'full'`, `'tied'`, `'diag'` or `'spherical'` and describes what type of covariance matrix is used.

[TODO

describe separating algorithm

which parameters to use?

visualize]

```
Algorithm 6: Separate
  input  : approximation parameters of all particle data sets
  output: assignments to different clusters
begin
    initialize Gaussian Mixture by specifying n_components and covariance_type
    apply Gaussian Mixture to approximation parameters of all particle data sets
    assign particles to clusters according to Gaussian Mixture results
    compare asignments from Gaussian Mixture to those of the data set the data
      stems from
end
```

When comparing different covariance types in the Gaussian Mixture we see that using `‘diag’` we have the lowest error rate. The details are shown in table ??.

`full`: 13.23%	`tied`: 12.7%
`diag`: 7.17%	`spherical`: 31.52%

Table 6.1: Error as a percentage of particles being assigned the wrong component.

7 Results

8 Conclusion

List of Figures

List of Tables

List of Algorithms

Bibliography

[AH24] K Maude Ashby and Kristin A Hogquist. "A guide to thymic selection of T cells". In: *Nature Reviews Immunology* 24.2 (2024), pp. 103–117.

[AW04] Robert T Abraham and Arthur Weiss. "Jurkat T cells and development of the T-cell receptor signalling paradigm". In: *Nature reviews immunology* 4.4 (2004), pp. 301–308.

[BCL99] Mary Ann Branch, Thomas F Coleman, and Yuying Li. "A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems". In: *SIAM Journal on Scientific Computing* 21.1 (1999), pp. 1–23.

[BV18] Stephen Boyd and Lieven Vandenberghe. "Levenberg–Marquardt algorithm". eng. In: *Introduction to Applied Linear Algebra.* Cambridge University Press, 2018, pp. 391–399. ISBN: 781316518960.

[CA22] Geoffrey M Cooper and Kenneth Adams. "The Nucleus". eng. In: *The cell: a molecular approach.* 19. edition. Oxford University Press, 2022, pp. 336–364. ISBN: 9780197583722.

[Gan12] William F. Ganong. "Overview of Cellular Physiology in Medical Physiology". eng. In: *Review of medical physiology.* 24. edition. Stamford, Conn: McGraw-Hill, 2012, pp. 35–66. ISBN: 9780071780032.

[Gan97] William F. Ganong. "Circulating Body Fluids". eng. In: *Review of medical physiology.* 18. ed. Stamford, Conn: Appleton & Lange, 1997, pp. 486–488. ISBN: 9780838584439.

[Hag18] Kimberly Hagel. *Positive and Negative Selection of T Cells.* 2018. URL: https://immunobites.com/2018/08/20/positive-and-negative-selection-of-t-cells/ (visited on 06/21/2024).

[JRB14] Noah Joseph, Barak Reicher, and Mira Barda-Saad. "The calcium feedback loop and T cell activation: how cytoskeleton networks control intracellular calcium flux". In: *Biochimica et Biophysica Acta (BBA)-Biomembranes* 1838.2 (2014), pp. 557–568.

[KCF18] Brahma V Kumar, Thomas J Connors, and Donna L Farber. "Human T cell development, localization, and function throughout life". In: *Immunity* 48.2 (2018), pp. 202–213.

[Lew01] Richard S Lewis. "Calcium Signaling Mechanisms in T Lymphocytes". In: *Annual Review of Immunology* 19.Volume 19, 2001 (2001), pp. 497–521. ISSN: 1545-3278. DOI: https://doi.org/10.1146/annurev.immunol.19.1.497. URL: https://www.annualreviews.org/content/journals/10.1146/annurev.immunol.19.1.497.

[ML23] Jonathan Morgan and Alan E Lindsay. “Modulation of antigen discrimination by duration of immune contacts in a kinetic proofreading model of T cell activation with extreme statistics”. In: *PLOS Computational Biology* 19.8 (2023), e1011216.

[MMS17] Magdiel Martínez, Namyr A Martínez, and Walter I Silva. “Measurement of the intracellular calcium concentration with Fura-2 AM using a fluorescence plate reader”. In: *Bio-protocol* 7.14 (2017), e2411–e2411.

[NW99] Jorge Nocedal and Stephen J. Wrigh. “The Dogleg Method”. eng. In: *Numerical Optimization*. Springer, 1999, pp. 73–76.

[Rey+09] Douglas A Reynolds et al. “Gaussian mixture models.” In: *Encyclopedia of biometrics* 741.659-663 (2009).

[RLH90] MW Roe, JJ Lemasters, and B Herman. “Assessment of Fura-2 for measurements of cytosolic free calcium”. In: *Cell calcium* 11.2-3 (1990), pp. 63–73.

[Rog24] Kara Rogers. *endoplasmic reticulum*. 2024. URL: https://www.britannica.com/science/endoplasmic-reticulum (visited on 06/23/2024).

[SB16] Dianne S. Schwarz and Michael D. Blower. “The endoplasmic reticulum: structure, function and response to cellular signaling”. In: *Cellular and Molecular Life Sciences* 73 (2016), pp. 79–94. DOI: https://doi.org/10.1007/s00018-015-2052-6. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4700099/.

[SI13] Peter B Stathopulos and Mitsuhiko Ikura. “Structural aspects of calcium-release activated calcium channel function”. In: *Channels* 7.5 (2013). PMID: 24213636, pp. 344–353. DOI: 10.4161/chan.26734. eprint: https://doi.org/10.4161/chan.26734. URL: https://doi.org/10.4161/chan.26734.

[SKJ09] Jennifer E Smith-Garvin, Gary A Koretzky, and Martha S Jordan. “T cell activation”. In: *Annual review of immunology* 27 (2009), pp. 591–619.

[SSB77] Ulrich Schneider, Hans-Ulrich Schwenk, and Georg Bornkamm. “Characterization of EBV-genome negative “null” and “T” cell lines derived from children with acute lymphoblastic leukemia and leukemic transformed non-Hodgkin lymphoma”. In: *International journal of cancer* 19.5 (1977), pp. 621–626.

[VL04] C Voglis and IE Lagaris. “A rectangular trust region dogleg approach for unconstrained and bound constrained nonlinear optimization”. In: *WSEAS International Conference on Applied Mathematics*. Vol. 7. 2004.