Image Processing and other bioinformatic tools for Neurobiology

Bildbearbeitung und andere bioinformatische Werkzeuge für die Neurobiologie

Doctoral thesis for a doctoral degree
at the Graduate School of Life Sciences,
Julius-Maximilians-Universität Würzburg,
Section Biomedicine

submitted by

Juan Pablo Prada Salcedo

from

Bogota, Colombia

Würzburg 2017

cc BY NC SA

Submitted on: **………………………………………………………..……..**

Office stamp

Members of the *Promotionskomitee*:

Chairperson: Prof. Dr. Christian Janzen

Primary Supervisor: Prof. Dr. Thomas Dandekar

Supervisor (Second): Dr. Robert Blum

Supervisor (Third): Prof. Dr. Martin Heisenberg

Date of Public Defence: **…………………………………………….…………**

Date of Receipt of Certificates: **……………………………………………….**

Affidavit

I hereby confirm that my thesis entitled Image Processing and other bioinformatic tools for Neurobiology is the result of my own work. I did not receive any help or support from commercial consultants. All sources and / or materials applied are listed and specified in the thesis.

Furthermore, I confirm that this thesis has not yet been submitted as part of another examination process neither in identical nor in similar form.

Würzburg, 09.11.2017

Place, Date Signature

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, die Dissertation Bildbearbeitung und andere bioinformatische Werkzeuge für die Neurobiologie eigenständig, d.h. insbesondere selbständig und ohne Hilfe eines kommerziellen Promotionsberaters, angefertigt und keine anderen als die von mir angegebenen Quellen und Hilfsmittel verwendet zu haben.

Ich erkläre außerdem, dass die Dissertation weder in gleicher noch in ähnlicher Form bereits in einem anderen Prüfungsverfahren vorgelegen hat.

Würzburg, 09.11.2017

Ort, Datum Unterschrift

Acknowledgements

I want to thank all the people who collaborated in this project, some of them directly and others on a different level.

I thank to my parents, Daniel Hernando y Margarita, for taking care of me always, even now that we are so far. Thanks to my brother Luis Daniel, for he is an example of integrity and good scientific work.

I want to thank Professor Thomas Dandekar. He is the best adviser I could have asked for, an amazing scientist, and an even better person. Also Dr. Robert Blum who was always working side by side with me, almost more as a colleague than as a supervisor.

Thanks to Anna-Maria for her support and company. Thanks to all my friends in Wuerzburg, Connie, Aitor, Ivo, Diego and others. Many thanks to Paola for her sincere friendship and for the thorough revision to this manuscript.

To end up, I thank the GSLS not only for the funding of the project but for the whole accompaniment during the same. Dr. Gabriele Blum-Oehler was always supportive and patient, helping me all along the way.

I hope this work is helpful to other people and that I can further contribute to science and to create a better world, so all the effort of the people mentioned before was not in vain.

Contents

1 Introduction 1

2 Neuronal calcium activity assisted detection 13

2.1 Introduction 13

2.2 Methods and Results 15

2.3 Discussion 38

3 Deep Learning for biological image analysis 43

3.1 Introduction 43

3.2 Methods and Results 45

3.3 Discussion 59

4 Biological systems simulations on inflammation 63

4.1 Introduction 63

4.2 Methods and Results 68

4.3 Discussion 84

5 Protein folding and theoretical implications from Bioinformatics 91

5.1 Introduction 91

5.2 Results and Methods 94

5.3 Discussion 103

6 General discussion 107

6.1 Conclusions 112

6.2 Discussion remarks on the PhD thesis system 113

Bibliography 115

A Calcium Activity Tool - User manual 125

List of Figures

1.1 Definition of biology 3

2.1 Mexican hat mother wavelet 19

2.2 Example of a signal and its wavelet transform 21

2.3 Algorithm basic features 23

2.4 Pipeline of the tool 24

2.5 User interface of the tool 24

2.6 User parameter characterization 26

2.7 Calcium spiking event detection 28

2.8 Activity profile of cultured hippocampal neurons 30

2.9 Noise analysis 31

2.10 Activity detection under blockade conditions 33

2.11 Events detection and estimation of signal fluctuation 34

2.12 Study of local-spontaneous activity 36
2.13 Performance under low SNR conditions 37

3.1 Deep Learning architecture used 45
3.2 Structure of a multilayer neural network 47
3.3 Three common activation functions 48
3.4 Auto-encoder structure 52
3.5 Stacked auto-encoder structure 53
3.6 Image example on its original and its processed versions 54
3.7 Close view to an image example on its original and its processed versions 55
3.8 Some patch examples of the original images 56
3.9 Some patch examples of the filtered images 57
3.10 Learning curves on the original data set 58
3.11 Learning curves on the filtered data set 58
3.12 Result of an image tested for segmentation 60

4.1 Schematic TNFR1 pre-assembly and cluster formation after ligand binding 67
4.2 Experimental observations of TNFR1 and TNFR2 affinities for TNF . . 69
4.3 Set of reactions of TNFR1 72
4.4 Set of reactions of TNFR2 73
4.5 Contact Map of the TNF model 73
4.6 Percentage of dimmers in time for several values of the activation constant 75

4.7 Percentage of dimmers with respect to the diffusion constant 76

4.8 Receptor 1 multimmer formation as function of TNF concentration . . . 77

4.9 Receptor 2 multimmer formation as function of TNF concentration . . . 78

4.10 Receptor 2 multimmer formation as function of TNF concentration . . . 79

4.11 Receptor 1 multimmer formation as function of KD_{pR1_m} 80

4.12 Receptor 2 multimmer formation as function of KD_{pR2_m} 81

4.13 Receptor 1 multimmer formation as function of KD_{pR1_m} for several values of receptor 1 population 82

4.14 Receptor 2 multimmer formation as function of KD_{pR2_m} for several values of receptor 2 population 83

5.1 Protein folding Funnel 96

5.2 Unfolded chain 97

5.3 Folded Chain 97

5.4 Contact matrix 98

5.5 Optimal fold 101

5.6 Optimal matrix 102

Abstract

Neurobiology is widely supported by bioinformatics. Due to the big amount of data generated from the biological side a computational approach is required. This thesis presents four different cases of bioinformatic tools applied to the service of Neurobiology.

The first two tools presented belong to the field of image processing. In the first case, we make use of an algorithm based on the wavelet transformation to assess calcium activity events in cultured neurons. We designed an open source tool to assist neurobiology researchers in the analysis of calcium imaging videos. Such analysis is usually done manually which is time consuming and highly subjective. Our tool speeds up the work and offers the possibility of an unbiased detection of the calcium events. Even more important is that our algorithm not only detects the neuron spiking activity but also local spontaneous activity which is normally discarded because it is considered irrelevant. We showed that this activity is determinant in the calcium dynamics in neurons and it is involved in important functions like signal modulation and memory and learning.

The second project is a segmentation task. In our case we are interested in segmenting the neuron nuclei in electron microscopy images of *c.elegans*. Marking these structures is necessary in order to reconstruct the connectome of the organism. *C. elegans* is a great study case due to the simplicity of its nervous system (only 502 neurons). This worm, despite its simplicity has taught us a lot about neuronal mechanisms. There is still a lot of information we can extract from the *c. elegans*, therein lies the importance of reconstructing its connectome. There is a current version of the *c. elegans* connectome but it was done by hand and on a single subject which leaves

a big room for errors. By automatizing the segmentation of the electron microscopy images we guarantee an unbiased approach and we will be able to verify the connectome on several subjects.

For the third project we moved from image processing applications to biological modeling. Because of the high complexity of even small biological systems it is necessary to analyze them with the help of computational tools. The term *in silico* was coined to refer to such computational models of biological systems. We designed an *in silico* model of the TNF (Tumor necrosis factor) ligand and its two principal receptors. This biological system is of high relevance because it is involved in the inflammation process. Inflammation is of most importance as protection mechanism but it can also lead to complicated diseases (e.g. cancer). Chronic inflammation processes can be particularly dangerous in the brain. In order to better understand the dynamics that govern the TNF system we created a model using the BioNetGen language. This is a rule based language that allows one to simulate systems where multiple agents are governed by a single rule. Using our model we characterized the TNF system and hypothesized about the relation of the ligand with each of the two receptors. Our hypotheses can be later used to define drug targets in the system or possible treatments for chronic inflammation or lack of the inflammatory response.

The final project deals with the protein folding problem. In our organism proteins are folded all the time, because only in their folded conformation are proteins capable of doing their job (with some very few exceptions). This folding process presents a great challenge for science because it has been shown to be an NP problem. NP means non deterministic Polynomial time problem. This basically means that this kind of problems cannot be efficiently solved. Nevertheless, somehow the body is capable of folding a protein in just milliseconds. This phenomenon puzzles not only biologists but also mathematicians. In mathematics NP problems have been studied for a long time and it is known that given the solution to one NP problem we could solve many of them (i.e. NP-complete problems). If we manage to understand how nature solves the protein folding problem then we might be able to apply this solution to many other problems. Our research intends to contribute to this discussion. Unfortunately, not to explain how nature solves the protein folding problem, but to explain that it does not solve the problem at all. This seems contradictory since I just mentioned that the body folds proteins all the time, but our hypothesis is that the organisms have learned to solve a simplified version of the NP problem. Nature does not solve the protein folding problem in its full complexity. It simply solves a small instance of the problem. An instance which is as simple as a convex optimization problem. We formulate the protein folding problem as an optimization problem to illustrate our claim and present some toy examples to illustrate the formulation.

If our hypothesis is true, it means that protein folding is a simple problem. So we just need to understand and model the conditions of the vicinity inside the cell at the moment the folding process occurs. Once we understand this starting conformation and its influence in the folding process we will be able to design treatments for amyloid diseases such as Alzheimer's and Parkinson's.

In summary this thesis project contributes to the neurobiology research field from four different fronts. Two are practical contributions with immediate benefits, such as the calcium imaging video analysis tool and the TNF *in silico* model. The neuron nuclei segmentation is a contribution for the near future. A step towards the full annotation of the *c. elegans* connectome and later for the reconstruction of the connectome of other species. And finally, the protein folding project is a first impulse to change the way we conceive the protein folding process in nature. We try to point future research in a novel direction, where the amino code is not the most relevant characteristic of the process but the conditions within the cell.

Zusammenfassung

Neurobiologie wird durch Bioinformatik unterstützt, aufgrund der großen Datenmengen, die von biologischer Seite her anfallen, bedarf es eines rechnerischen Ansatzes, um diese Daten sinnvoll zu interpretieren. Im Rahmen der vorliegenden Dissertation werden vier Werkzeuge aus dem Bereich der Bioinformatik für die Anwendung in der Neurobiologie vorgestellt.

Die ersten beiden Werkzeuge gehören zum Bereich der digitalen Bildverarbeitung. Das erste Werkzeug nutzt einen Algorithmus basierend auf der Wavelet-Transformation, um Calciumaktivität in Neuronenkulturen zu bewerten. Hierzu wurde Open-Source-Software entwickelt, die Neurobiologen bei der Analyse von Videoaufnahmen unterstützt. Diese Analyse wird herkömmlicherweise manuell vorgenommen, sodass der Prozess zeitintensiv und sehr subjektiv ist. Die entwickelte Software beschleunigt den Arbeitsprozess und ermöglicht eine unverzerrte Detektion der Ereignisse in Bezug auf Calcium. Von noch größerer Bedeutsamkeit ist die Tatsache, dass der entwickelte Algorithmus nicht nur neuronale Spiking-Aktivität detektiert, sondern auch lokale Spontanaktivität, die herkömmlicherweise als irrelevant betrachtet und daher verworfen wird. Wir konnten zeigen, dass diese Spontanaktivität hohe Relevanz für die Dynamik von Calcium in den Neuronen besitzt und wahrscheinlich an wichtigen Funktionen beteiligt ist, wie der Signalmodulation, Lernen und Gedächtnis.

Beim zweiten Projekt handelt es sich um eine Segmentierungsaufgabe. Wir sind daran interessiert, die neuronalen Zellkerne in elektromikroskopischen Aufnahmen des C.elegans zu segmentieren. Die Kennzeichnung dieser Struktur ist notwendig, um das Konnektom dieses Organismus zu rekonstruieren. Als Studienobjekt eignet sich C.elegans aufgrund der

Simplizität seines Nervensystems (er besteht lediglich aus 502 Neuronen). Trotz der Simplizität des Nervensystems dieses Wurms konnten wichtige Erkenntnisse im Hinblick auf neuronale Mechanismen durch die Untersuchung dieses Modellorganismus gewonnen werden. Daher ist die Bestimmung des Konnektoms bedeutsam. Es existiert bereits eine Version des Konnektoms, doch diese wurde händig für lediglich ein Subjekt rekonstruiert und ist daher möglicherweise fehlerbehaftet. Die automatisierte Segmentierung der elektronenmikroskopischen Aufnahmen ermöglicht einen weniger verzerrten Ansatz, der zudem die Verifizierung an mehreren Subjekten gestattet.

Das dritte Projekt dieser Dissertation ist ein Projekt zur Modellierung und Simulation eines biologischen Systems. Aufgrund der hohen Komplexität selbst kleinster biologischer Systeme ist die computergestützte Analyse notwendig. Der Begriff in silico wurde für die computergestützte Simulation biologischer Systeme geprägt. Wir haben ein in silico Modell des TNF (Tumornekrosefaktor) Ligand und seiner zwei Hauptrezeptoren entwickelt. Dieses biologische System ist von hoher Bedeutsamkeit, da es am Entzündungsprozess beteiligt ist, der höchste Wichtigkeit als Schutzmechanismus hat, aber es kann auch komplizierte Erkrankungen auslösen (beispielsweise Krebs), falls es zu einer chronischen Entzündungsreaktion kommt. Derartige Entzündungsprozesse können besonders gefährlich im Gehirn sein. Um die Dynamiken besser zu verstehen, die das TNF System leiten, haben wir ein Modell mittels der BioNetGen Sprache erstellt. Diese regelbasierte Sprache ermöglicht es ein System zu simulieren, in dem multiple Agenten geleitet werden von einer Regel. Mithilfe unseres Modells charakterisieren wir das TNF System und stellen Hypothesen über die Beziehung des Liganden mit den beiden Rezeptoren auf. Diese Hypothesen können später genutzt werden, um mögliche Ziele im System für Arzneimittel, mögliche Behandlungen für chronische Entzündungen oder das Fehlen einer Entzündungsreaktion zu bestimmen.

Im abschießenden Projekt wird das Proteinfaltungsproblem behandelt. In unserem Organismus werden ständig Proteine gefaltet, denn nur im gefalteten Zustand können sie ihrer Aufgabe nachkommen (mit sehr wenigen

Ausnahmen). Dieser Faltungsprozess stellt eine große Herausforderung für die Wissenschaft dar, weil gezeigt wurde, dass der Faltungsprozess ein NP Problem ist. NP steht dabei für nichtdeterministisch polynomielles Zeitproblem. Dies bedeutet im Grunde, dass es nicht effizient gelöst werden kann. Nichtsdestotrotz ist der Körper in der Lage, ein Protein in Millisekunden zu falten. Dieses Phänomen stellt nicht nur Biologen sondern auch Mathematiker vor Rätsel. In der Mathematik wurde diese Probleme schon lange studiert und es ist bekannt, dass die Kenntnis der Lösung eines NP Problems die Lösung vieler bedeuten würde (insbesondere NP-kompletter Probleme). Daher ist die Idee, dass viele Probleme gelöst werden könnten, durch das Verständnis davon, wie die Natur das Problem löst. Unsere Forschung zielt darauf ab, zu dieser Diskussion beizutragen, allerdings nicht durch die Erklärung davon, wie die Natur das Problem löst, sondern durch die Erklärung, dass die Natur das Problem nicht löst. Dies scheint zunächst widersprüchlich, da der Körper ständig Proteine faltet. Unsere Hypothese besagt jedoch, dass der Organismus gelernt hat, eine vereinfachte Version des NP Problems zu lösen. Die Natur löst das Problem nicht in seiner vollen Komplexität, sondern nur eine kleine Instanz davon. Eine Instanz, die ein konvexes Optimierungsproblem darstellt. Wir formulieren das Proteinfaltungsproblem als konvexes Optimierungsproblem und zur Illustrierung unserer Behauptung nutzen wir theoretische Beispiele.

Wenn die Hypothese zutrifft, bedeutet dies, dass das Proteinfaltungsproblem ein einfaches ist und wir müssen lediglich die Ausgangskonstellation der Umgebung in der Zelle verstehen und modellieren, in dem Moment in dem die Faltung passiert. Sobald wir die Ausgangskonstellation und den Einfluss auf den Faltungsprozess verstehen, können wir Behandlungen für Amyloid-Krankheiten, wie Alzheimer-Demenz und Morbus Parkinson entwickeln.

Zusammenfassend trägt die vorliegende Dissertation zu neurobiologischer Forschung durch vier Ansätze bei. Zwei sind praktische Beiträge mit sofortigem Nutzen für die Forschung, dazu zählen das Videoanalyse Tool für Calcium Aufnahmen und das TNF in silico Modell. Die neuronale

Zellkernsegmentierung ist ein Beitrag für die nahe Zukunft ? ein Schritt zur Vervollständigung des Konnektoms des C.elegans und langfristig zur Rekonstruktion der Konnektome anderer Spezies. Und schließlich ist das Proteinfaltungsprojekt ein erster Impuls den Proteinfaltungsprozess anders zu denken. Wir versuchen zukünftige Forschung in eine neue Richtung zu lenken, wobei nicht der Aminosäurecode das relevanteste Charakteristikum des Prozesses ist, sondern vielmehr die Bedingungen innerhalb der Zelle.

1

Introduction

> "I just wondered how things were put together.
> Or what laws or rules govern a situation,
> or if there are theorems about what one can't or can do.
> Mainly because I wanted to know myself."
>
> *Claude Shannon*

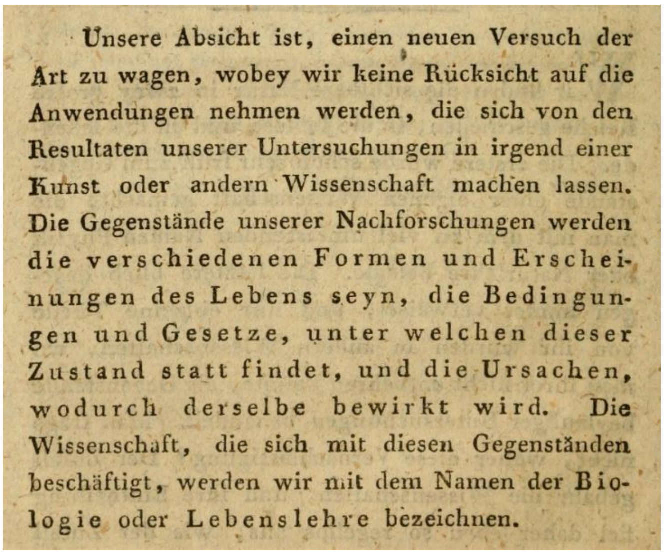
As rational self-conscious beings it is simply natural for us to try to understand and explain the world. By accumulating experience human beings have tried to explain the world since the beginning of its existence as a species 200 thousand years ago. Even before having a writing system, knowledge was already accumulated and communicated orally, for example in the case of maize domestication in Mexico [1]. Obviously those early stages were quite obtuse and the knowledge was extremely shallow, as a matter of fact, I believe is precisely this lack of knowledge which leads to the origin of religion. Imagine the early human beings trying to understand climate calamities that even nowadays we cannot fully understand. These early humans were so decided to explain these calamities that in the absence of a better explanation they chose to believe in a divine power. Religion is in essence not as far from science as normally depicted. They both are human social constructions with the purpose of understanding the world around us and even our own existence.

The big gap that we see today among science and religion comes from the fact that science has evolved much faster and is today dynamic, open, and self-critic; qualities that religion is in much need of. Modern science is mainly characterized by its methodology. This distinguish science from other social constructions like arts. The method used by scientists is usually referred to as the "scientific method". It is hard (or even impossible) to distinguish science from non-scientific disciplines by subject-matter discussion alone, so it is assumed that any discipline making use of the scientific method is science and the rest is non-science.

Although it is clear that science is defined by the use of the scientific method it is not clear what exactly such method is. This is not clear for the simple fact that there is no such thing as a universal multipurpose scientific method. Formerly it was believed that scientific method was fixed and universal so it could and should be applied in every discipline of science in every context. Nowadays it is clear that this is not the case and that the scientific method is flexible and depends on the historical time-period and cultural context.

The scientific method refers to several stages in the scientific process, for example experiment design or hypotheses proposition. There is a methodology regarding sample collection and another for experiment performance, and there is also a methodology to validate results from experiments and a methodology to communicate those results. As these stages of science production differ from one science to another, it is simply obvious that the scientific method differs also between sciences. There are of course some general characteristics of the scientific method which are valid on every discipline, so it should be regarded as a multifaceted array of tools, techniques and procedures that govern scientific practices [2]. For example, as a result of the emergence of bioinformatics it was necessary to define new methodologies of experiment design and experiment validation which were coherent with this new field of study. I will comment on this again later.

Biology as we know it today was consolidated around 1750. One of the first definitions and the one that better states what we understand as biology today was given by Gottfried Reinhold Treviranus as presented in figure 1.1. In English the definition is usually translated as, "The objects of our investigation will be the various forms and manifestations of life, the conditions and laws under which held the state of life and the



Unsere Absicht ist, einen neuen Versuch der Art zu wagen, wobey wir keine Rücksicht auf die Anwendungen nehmen werden, die sich von den Resultaten unserer Untersuchungen in irgend einer Kunst oder andern Wissenschaft machen lassen. Die Gegenstände unserer Nachforschungen werden die verschiedenen Formen und Erscheinungen des Lebens seyn, die Bedingungen und Gesetze, unter welchen dieser Zustand statt findet, und die Ursachen, wodurch derselbe bewirkt wird. Die Wissenschaft, die sich mit diesen Gegenständen beschäftigt, werden wir mit dem Namen der Biologie oder Lebenslehre bezeichnen.

Figure 1.1: Definition of biology - The seminal definiton of biology by the German scientist Gottfried Reinhold Treviranus in 1802.

causes, whereby the same is effected. The science that deals with these things, we will designate by the name of biology or the science of life". It can be argued that Anatomy and Botany already existed since a long time before this definition was elaborated, but it is anyway this definition generally accepted as the founding act of modern biology.

At the beginning, biology was mostly composed of taxonomy works, later with the invention (or just improvement) of the microscope, scientists were capable of a deeper analysis which led to the formulation of the concept of cell. One hundred years after its birth, would occur the biology big revolution thanks to the appearance of "*On the origin of the species*" from Charles Darwin. This book fostered and popularized the central axiom of biology, which is still in use nowadays. Darwin's evolution theory introduced purely casual account explanations for biological phenomena, attaching to these phenomena a clear objective: survival. This theory gave biology a central pole to hang on to and became the necessary structure for approaching biological science to physical sciences.

Evolution theory is still the strongest and more general paradigm in biology. Thanks to this theory, biology moved from a purely phenomenological study to a theory and model generation form. As it should be in science, evolution theory is not free of critique and counter arguments. For example, there is the discussion around its statistical appearance, how much of the changes we see in nature correspond really to an evolutionary process and not because of the drift process. Drift is the process of adaptation of species, not because of evolutionary purposes but because of the persistent impact of factors such as low population size or exogenous natural occurrences, like volcano eruptions, forest fires or floods [2]. There is the discussion whether evolution and drift are independent processes or are in such form entangled that they cannot be separated. And there might even be other forces in action. It is for example proposed that living organisms seek diversity even in absence of evolutionary causes. Anyway, beyond any critique that can be made to the evolution theory, it is certainly a main protagonist in biology.

Biological taxonomy was always functional and remains so. In biology concepts are mostly explained according to their function. The heart is defined as a pumping machine that propels blood all around the body. Today, biological taxonomy remains largely the same in terms of explaining concepts by its purpose, but it has evolved into identifying smaller structures with much more specific functions.

There comes into play a second important biological revolution which is still in course, the molecular biology. By means of this revolution, biology is moving towards a better understanding of the underlying mechanisms of biological processes and with that it is also moving towards other forms of taxonomy beside the functional one.

Thanks to new technologies, there is an increasing capacity to observe biological phenomena in high detail. Starting with microscopes and moving to gene knockout techniques, fluorescence methods, and molecular biology; we count today with an unprecedented set of tools to the service of biology. This important development led to the appearance of fields of study within biology, for example neurobiology. This field was almost non existent until the microscope resolution became good enough so that Ramon y Cajal, Golgi and others were capable of visualizing the first neuronal structures. Before that only superficial experiments were done in humans, measuring external characteristics like size and weight. Also, some experiments in animals were performed. Causing injuries in the brain of rats and reporting the affected behavior. These were all very shallow experiments so we see that the great amount of neuroscientific knowledge we posses today is due to the advance in technology.

Biology is now a more structured science. This means a science which is moving towards models, mechanisms, and understanding of all system constituents. This movement towards a more structured science is happening thanks to the quantification of biological experimental results. As in every other science, evidence is the key stone to scientific knowledge, but there are different forms of evidence. It can be found in qualitative or a quantitative form. In biological sciences evidence used to be qualitative but thanks to the advance of technology we are generating an enormous amount of quantifiable data in present times. Such is the case for example of microscopy images. In former days it used to be good enough to present that in a single image obtained on a microscope some structure was present. It was not clear how much of that structure was there, on which concentration or exactly where it was localized. Today it is common to talk about quantitative microscopy, which means that these images are processed in order to extract numerical information about them.

The quantitative information is the first step towards modeling a system. In biology there are a lot of models and many more are being generated every day, but these are all highly specific models. We lack of general models that apply to more than one biological system. In order to achieve these, it is necessary to assemble the information found in several systems and see if they share common characteristics. The integration and processing of so much information can of course only be done by computational means.

Bioinformatics was originally defined in 1970 as the study of information processes in biotic systems. This definition is today a little narrow since it was intended for exclusively genome sequencing data. Nowadays we could maybe say that Bioinformatics is a science (or field) which studies living organisms and the phenomena related to those organisms, just like biology, but using a computational methodological approach.

Bioinformatics acquired importance because of the appearance of gene sequences. When the genetic code was discovered, it was clear for biologists that a computational solution would be required to understand and analyze that great amount of data. The gene sequence analysis is still a big part of Bioinformatics but it is definitely not the only part or even the most important one. There are two main areas of Bioinformatics, the theoretical and the practical. The practical area refers to those cases where Bioinformatics is used as a tool at the service of biology, for example in order to interpret, organize, or classify data produced in biological research. The theoretical area is not so well known but it is by no means less important and it acquires more relevance as it grows. In this area, the central idea is to generate models for biological systems and try to generate a structure for biological knowledge. These two areas are often mixed together and it is hard to set a fixed border between them, but this is also, of course, not necessary. The important thing is to notice that Bioinformatics is more than a tool of biology. It is a discipline with a research agenda of its own. Bioinformatics is today applied on several different fields: genetic sequences analysis, image analysis, DNA sequences analysis, cell signaling simulation, metabolic networks simulation, protein-protein interaction simulations, molecular biology simulations, and biology standardization.

The use of computers for biological data processing or for assisted analysis poses no problem in terms of validity of the results obtained since the computer is just a mere tool that speeds up the process and makes it more precise. The same is not true when it comes to pure computer simulations, where data is not simply being processed but it is also being generated in a computer. Computer simulations as research experiments were first done in the 1950's in the context of nuclear weapon research and climate prediction. Since then, computational experiments have been expanding to every other science and presently they can be found even in social sciences. In biology the use is so common that the term *in silico* experiment has been coined and it is commonly used. These computational experiments constitute an important part of Bioinformatics.

Computational experiments consist in using a computer to solve a set of equations that represent a biological system. These equations are normally differential equations for which the starting point is known. Based on that starting point a solution is calculated for time point one, then time point two, and so on. The set of values changing through time are the results obtained from the simulation. This kind of experiments are used in biology to model a wide range of systems and types of systems, from signaling networks to ecological models, passing through metabolic networks and protein-protein interaction networks.

As mentioned before, the scientific method is not universal but particular to each science. In the case of Bioinformatics we find that computational experiments are a common practice so they form part of the scientific method of this field. These experiments follow their own methodology and have some particular characteristics. For example, one characteristic is the discussion of whether these computational experiments can be regarded as normal experiments, and if its results can be accepted as scientific evidence. Bioinformatic experiments are carried out because of the impossibility of doing a traditional experiment or observational study, normally due to practical or ethical reasons, or simply because of the impossibility of carrying out such experiment. So, simulations are a replacement for traditional experiments and as such this replacement is bound to follow the same principles of the traditional experiments. There are three conditions that a simulation must follow. It should be theory guided, its parameters and equations should be coherent with current accepted theory. It has to be clear for the researcher that the simulation is biased, by means of parameter choosing, algorithm programming, numerical solution method, *ad hoc* assumptions, and so on. And finally, it is important to know that the simulation results should be compared to existent data that validates the results. Of course, not all of the results can be compared because this was in the first place the reason to do the simulation, the impossibility to obtain the real data.

Computer simulation, similar to other techniques in science, is self-vindicating. Their own success is the strongest argument in favor of their reliability. Nevertheless, simulations should always be confronted with verification and validation procedures. Verification can be seen as questioning whether we are using the right equations and validation would then be questioning whether those equations were correctly solved. The verification part is solved as long as the simulation is designed according to scientific theory and what is known from the system under study. And the validation part is solved by comparing some results of the model with existing real observed data. *In silico* experiments are of course in principle different from traditional experiments but that does not mean less valid. Results from traditional experiments are somehow more general, broader and a lot of information can be extracted out of them. Simulations are more specific, they are designed to solve a particular question or a small set of questions and nothing else. In most of the cases no other inferences can be made from their results.

Not using bioinformatic tools or not performing computer experiments is simply not an option in current biological research. There is a full dependency of biology on the calculation capacity of computers. It is probably impossible to find a study today where some form of computation is not involved (maybe the simplest form is the statistical analysis of the results). Our aim as bioinformaticians is to foster all the fields of bioinformatics and to make sure that the computational tools we use issue trustworthy reliable results. Biology can profit from this as well as our understanding of living organisms.

Neurobiology is precisely a great example of a field which could not further develop without the support of Bioinformatics. The nervous system of living organisms, even small ones like the *C. elegans*, is so complex that there is no other alternative to study it beside computational tools. The field of neurobiology and neuroscience in general have experienced a substantial revolution since the second half of the 19th century when the first neurons were visualized on the microscope. At that time, Golgi and Ramon y Cajal found the neuron and devised that this was the basic functional unit in the brain, setting the bases of current neurobiology and making themselves worth of winning the Nobel prize in medicine in 1906. Today we have a much deeper knowledge of the neurons, we understand the different forms of synapses in them, we understand many processes, such as apoptosis, plasticity, migration, and neurogenesis among others. But we also do not understand many things, like the relation among all these processes, the function of the different synapses, and their interplay. We know that connectivity plays an important role in the brain, it is the architecture of the mind, but it is not the mind itself. We still struggle to understand how is the color red conceived in the brain. We know how some parts of the brain are involved with certain tasks, for example that the basal ganglia and the parietal lobe are involved in time perception but we do not know how the abstract concept of time is generated in the brain. In this project I present four studies that illustrate the use of bioinformatic tools in neurobiology applications. Some of these uses are directly related to neuroscience, while others are more general but are nevertheless highly relevant for neurobiology as well.

This doctoral thesis is composed of four chapters beside this introduction. Chapters two and three refer to the use of bioinformatics as a tool for analyzing biological data. Chapter four presents the use of bioinformatics to create *in silico* models and simulate biological systems. And finally Chapter five is a theoretical contribution to bioinformatics.

The first two chapters (2 and 3) deploy the analysis of biological images using computational tools. They present present two projects on this same field. A first project consists in analyzing calcium imaging videos and automatically detecting neuronal activity in those videos. The second project is a segmentation tool, where we try to segment neuronal nuclei in electron microscopy images using deep learning algorithms. Imaging is a growing field in biology and these two projects are great examples of the importance of bioinformatics in order to deal with this great amount of visual data produced in biological research.

The biological system simulation section (chapter 4) presents a project where we studied the dynamics of the Tumor Necrosis Factor (TNF) system. TNF is a ligand which can be bounded to two different receptors. Some characteristics are known about the system but most of its functioning remains unknown. Through computational experiments we elucidate important characteristics of the system which can guide future biological research. This TNF system is of mayor importance in neurobiology since it regulates the inflammatory response in the brain, which is a very important protection mechanism but it can also represent a high risk in case of chronic inflammation.

Finally the theoretical contribution is a position stand on the discussion whether nature is capable of solving NP complete problems. NP stands for Non deterministic Polynomial time. This means that these problems are most difficult and presumably cannot be efficiently solved. We use protein folding, a well-known NP problem, as study case. We present the argument that nature solves a simplified version of the protein folding problem and this simplified version is just a P problem (Polynomial time problem, efficiently solvable). Our claim is that nature solves the problem sub-optimally under a simple pragmatic approach. The problem of protein folding is interesting to us from several perspectives. First of all protein misfolding is the cause of many important diseases in the brain, like Alzheimer's, Parkinson's, neurodegenerative disease, and other prion diseases. Understanding the process of protein folding is the first step towards creating treatments and designing drugs which can heal and stop those diseases. The second reason why protein folding is of such interest to us is because it is a very interesting mathematical problem. It is regarded as an impossible problem to be efficiently solved, so this activity of exploring how nature deals with such kind of problems can be very prolific.

This is clearly a non-conventional doctoral thesis because normally a single project is presented, not three or four. The are two main reasons for this particularity. First, because it was necessary to have two parallel projects, a risky one and a safer one. I believe a good scientist must be able to keep a balance between risk and innovation of the project. It is the aim of a doctoral thesis to achieve conclusive results, but those cannot be guaranteed since the doctoral thesis constitutes a research activity. Nevertheless, expert researchers like Professor Thomas Dandekar have an accurate sense to guess how reliable a project is in terms of conclusive results. In this case, Professor Dandekar wisely advised to tackle both projects in parallel and see how they developed. The safe project was the first section, as we could be sure that the use of bioinformatics for analyzing the images would be of much help, while the risky project was the theoretical contribution where we cannot provide solid proof of our claim but only insights into our perspective.

The second reason is that a beautiful characteristic of bioinformatics is its collaborative nature. It is a field that essentially nurtures itself from other fields like biology and computer science. In the spirit of the collaborative essence of Bioinformatics, I participated in several collaboration projects and one of them is the TNF project in association with Professor Harald Wajant. This project is included in the doctoral thesis due to the great results obtained in it and because it illustrates the field of bioinformatics for biological systems simulation.

I am happy the way my doctoral thesis evolved. There are of course still many open questions regarding each of the projects, but I consider the thesis to be a valuable input in general to bioinformatics and in particular to neurobiology in each of the areas presented. The rest of the thesis is organized as mentioned before, one chapter dedicated to each project. As methods are here bioinformatic methods, their development and application is a major and first part of the results and hence presented here together with the data analysis and other results. Conclusions and discussion sections are included in each chapter and a final general discussion is presented in the end.

2

Neuronal calcium activity assisted detection

2.1 Introduction

Calcium signaling is a vital mechanism in neurons. It is fundamental for several processes such as neuronal development, synapses, and plasticity. The calcium influx into the cell body occurs from the outside through the cell membrane by means of different mechanisms, calcium channels or voltage gated channels. This influx is responsible for the depolarization of the neuron which originates a synaptic train [3, 4, 5, 6, 7]. There is also important calcium trafficking occurring within the neuron. Calcium is stored in internal compartments such as the endoplasmic reticulum and opportunely released to maintain the required concentration in the cell body. The multiple mechanisms and multiple functions that calcium signaling involves build up to make of neuronal calcium signaling a highly complex system, difficult to understand and model. There are some attempts to detect and analyze calcium signaling but this has proven to be a hard task due to the following aspects. First, the diversity of the calcium peaks: they can be exponentially shaped and tall as a synaptic peak, but can also be small and have a random shape. Second, the high functional diversity of the proteins involved in calcium signaling. And third, calcium signaling can occur in an unpredictable spatial-temporal location and on an independent way.

Calcium imaging techniques are well established and have been proved to constitute a reliable neuronal model *in vivo* and *in vitro* [8, 9]. Trying to manually analyze the results of calcium imaging results too expensive in cost and time, and it is furthermore highly unreliable. In order to conduct a complete analysis of the calcium dynamics in a group of neurons or even in a single neuron, it is necessary to have an automated system capable of reliably identify the calcium activity and mark it in space and time in order to further analyze it. In this regard important efforts have been done from the scientific community [10, 11, 12, 13, 14, 15, 16]. These tools are quite powerful and the most recent ones include a quite complete pipeline that goes from activity identification to neuron segmentation [13]. Most of these tools require a user defined ROI (Region of Interest), which causes a bias in the results and can be a huge work load. Also most of these approaches depend on a spike-shape fitting model in order to identify calcium activity peaks. The problem with this scheme is that it discards all non-spike like activity which we claim is also of high relevance and a key part of calcium signaling dynamics.

The spiking activity in neurons is seemly well understood and accounts for very important process in the brain but the local, independent, and smaller non-spike activity is also very important but much less understood. This local calcium activity is involved in activity-dependent axon growth, network wiring, neurotransmitter specification, neurite extension, growth cone dynamics, and synaptic scaling [17, 18, 19, 20, 21]. This form of calcium activity is essentially different to spiking activity, it is non-regular in shape, and it is highly independent in location and time. These characteristics make it hard to identify.

In collaboration with the Neurobiologie Klinikum in the University of Wuerzburg we developed a tool with the purpose of assisting researchers in the detection and analysis of local calcium activity. The proposed tool deals with the mayor problems of local calcium activity detection. Due to the unpredictable location of the local calcium activity, the tool was designed to simply explore the complete field of view of calcium imaging videos. And in order to be able to identify real calcium activity peaks which might be close to noise level and which are not spike shaped, we decided to use a Wavelet transform guided detection algorithm. This algorithm was successfully applied to x-ray traces [22] and to mass spectrometer signals [23], both applications have similar characteristics to calcium local activity since all kind of shapes can be expected and the real activity peaks can be easily confused with noise peaks. Through the wavelet transformation a CWT (Continuous Wavelet Transform) of each signal is constructed and this CWT is further explored in order to find the ridges and then use these to guide the search of activity peaks within the signal.

The algorithm pipeline is basically as follows: read the calcium imaging videos and extract the calcium activity traces. Transform the traces into the wavelet space and find the activity peaks. Two extra features are available to the user: the calculation of a point of general tendency change in the signal, and the estimation of the variance of the signal. Finally, the processed signals are presented to the user with the activity locations in time and space.

We conducted several experiments to verify the correct functioning of the tool and the accurate detection of local calcium activity. Even though the spike calcium activity is not specifically targeted, it is also correctly detected and analyzed by our tool. The tool is easy to use and to understand, which enables the researcher to better interpret the obtained results. We expect that this tool will reinforce research on local calcium activity and complement existing research on neuron spiking activity.

In the rest of the chapter the tool is presented in detail and also the conducted experiments in order to validate it.

2.2 Methods and Results

2.2.1 Biological methods

The biological methods include all experiments performed in the wet lab, that means, everything that was done until the point of recording the calcium imaging videos (including the recording itself), are presented in detail in the paper we published on this project. Here I reproduce this part of the paper with the consent of all the authors.

Primary hippocampal neurons

All experimental procedures were approved by the animal welfare committee of the

University of Würzburg, in accordance with European Union guidelines. Hippocampal neurons were prepared from CD1 mice of either sex. Hippocampi of new born mice were collected in Hank's buffered saline solution (HBSS). Trypsin (Worthington) was added to a final concentration of 0.1% and the tissue was incubated for 15 min at 37°C. The protease digestion was stopped with 0.1% Trypsin inhibitor (Sigma). After four steps of trituration in Neurobasal/B27 medium (Life Technologies), cells were plated on poly-L-lysine-coated glass coverslips in Neurobasal, $1 \times B27$, 0.5% penicillin/streptomycin, 1% Glutamax, and $1 \times N2$ supplement (all Life Technologies) and cultured at 37°C under an atmosphere of 5% CO_2. Calcium imaging experiments were performed after indicated days in vitro (DIV).

Primary motoneurons

Primary motoneurons were prepared from spinal cord [24]. The lumbar spinal cord of mouse embryos was dissected at embryonic day 13 or 14. Motoneurons were enriched by affinity-panning with antibodies against the p75NTR receptor, and plated at a density of 1,000 - 2,000 cells on 10mm glass coverslips coated with polyornithine and laminin-1. Motoneurons were grown in Neurobasal/B27 medium (Life Technologies), 2% horse serum, $10nM\beta - mercaptoethanol$, and $1 \times GlutaMax$. The neurotrophic factors BDNF and CNTF were added at a concentration of 5ng/ml. One day after motoneuron isolation 40% of the medium was replaced. Calcium imaging was performed at DIV 3.

Calcium imaging

Calcium indicator dye loading and Ca2+ imaging was performed in artificial cerebral spinal fluid (ACSF). For motoneuron imaging ACSF contained (in mM): 127 NaCl, 3 KCl, 2.5 NaH2PO4, 2 CaCl2, 1 MgCl2, 23 NaHCO3 and 25 D-glucose, bubbled with 95% O2/5% CO2. Hippocampal neurons were imaged under continuous perfusion with (in mM): 135 NaCl, 6 KCl, 1 MgCl2, 2 CaCl2, 5.5 D-glucose; 10 HEPES). For chemical LTP stimulation the following buffer composition was used: 128 NaCl, 13 KCl, 3 CaCl2, 5.5 D-glucose; 10 HEPES, 0.1 glycine (in mM). Calcium-free ACSF contained 0.1 mM EGTA. The calcium indicator Oregon Green 488 BAPTA-1, AM (OGB; Invitrogen) was prepared as 5mM stock solution in 20% Pluronic F-127 / DMSO (Invitrogen). For dye loading, $0.5\mu l$ of the OGB/Pluronic mixture was mixed in $500\mu l$ of ACSF and neurons were labeled for 15 minutes at 37°C and 5% CO2. Changes in OGB/calcium-fluorescence were monitored with the help of an upright microscope (BXWI, Olympus, objective: Olympus 40x LUMPlanFI/IR, 0.8 W), in a heated imaging chamber (Luigs & Neumann). Imaging was performed under continuous perfusion with ACSF. Images (8-bit) were captured at indicated speed in a streaming approach, with a Rolera-XR camera (Qimaging) and StreamPix 4 software (Norpix) under continuous illumination with a 470nm LED light source (Visitron Systems). Fluorescence filters with the following parameters were used: excitation: $482 \pm 35nm$; dichroic filter 506nm, emission filter $536 \pm 40nm$.

2.2.2 Informatic Methods

Neuronal Activity Detection Tool

The tool was programmed in the Bio7 environment `http://bio7.org/`. Bio7 is an open source software, designed initially for ecological modeling but with the strong capability of bridging R [25] and imageJ [26]. Bridging these two programs is not as simply as one could expect, there are differences in the Java version which make it pretty hard. But as hard as it may be, it is also useful. ImageJ is also an open source tool with an ever increasing number of libraries and capabilities. It is probably the most used software for image processing among the life science research community. While learning to work with ImageJ, I was most surprised of all the features it offers and at zero cost. There is an almost unlimited source of filters to process and denoise images. The use I made of ImageJ in this project is absolutely basic; simply loading the images, increasing the image contrast, and finally extracting the pixel traces. The reason why it was necessary to include ImageJ for such a simple task is that this is not possible in R (as amazing as it sounds). R is a very powerful software, also ubiquitous in the scientific world, but it is a statistics-oriented software. It is true that because of its open source character and open to community contributions R has grown tremendously and it is nowadays used for many other applications different from its original purpose. But in terms of image processing it is still far away from a software like ImageJ. Only opening a video in R turns out to be quite a challenge and even if one manages to overcome this problem, most probably the program will run out of memory due to its deficient way in handling of video files.

One could also ask why not doing the complete analysis in ImageJ. Well, because then the story is the other way around. Operating matrices and calculating basic mathematical functions like the wavelet transform are seemly simple things in R, whereas in ImageJ such calculations require high programming skills and complex data conversions. So in summary it was much more effective to load the image into ImageJ, extract the traces, and leave the rest to R.

There are four user defined parameters. These are:
1) Window size (WS), the size in pixels of each square region that will be analyzed. This means, the size of each square on the grid.
2) Signal Average Threshold (SAT), a threshold value for the average of the intensity trace. If an intensity trace average is below this value then it will be discarded and no peaks will be searched on it. This discards regions that belong to the background, where no neuron is depicted.
3) Signal to Noise Ratio (SNR), the standard definition from signal processing. This is also a threshold value for how much has to be the power ratio between the estimated noise and the peak candidate to be accepted as a real peak.
4) Variance Area sliding window size (VA^{30}), the size of the sliding window use to calculate the variance and the average of the signal to plot the variance area.

Wavelet Transform

The wavelet transform is a well known analysis, very useful because it transforms the signals in frequency and time. This means that it makes it possible to analyze a signal's frequency changes through time. A brief explanation is presented here (for a more complete one the reader can refer to [27]). The wavelet transform on its simplest description is a convolution between two signals, one is the signal being analyzed and the other one is the *mother wavelet*. The idea is that the mother wavelet has such characteristics that when it is convoluted with the other signal it enhances the characteristics of the signal under analysis. In our case, for example, such characteristics are the activity peaks we want to find. Define a mother wavelet ψ as any function with finite energy such that equation Eq. 2.1 is satisfied:

$$\int_{-\infty}^{\infty} \frac{\mid \Psi(w) \mid}{\mid w \mid} dw < \infty \qquad \text{(Eq. 2.1)}$$

For a well behaved function, that means a function nicely localized in time and space, it is also sufficient to state the constraint in equation Eq. 2.2:

$$\Psi(0) = \int_{-\infty}^{\infty} \Psi(t) dt = 0 \qquad \text{(Eq. 2.2)}$$

There are many examples of mother wavelet functions, that means, functions that satisfy the conditions stated. Some are: the haar wavelet, the mexican hat wavelet and, the four Daubechis wavelets. In our algorithm we will use the mexican hat wavelet, also known as the second derivative gaussian. The mother wavelet is presented in figure 2.1. As it can be seen, this mother wavelet enhances the peaks while diminishing the surrounding noise. It is also very important for our application to note that the wavelet transform with the mexican hat mother wavelet allows to have an almost constant phase transformation, which is a necessary requisite to be able to locate the activity peaks in time.

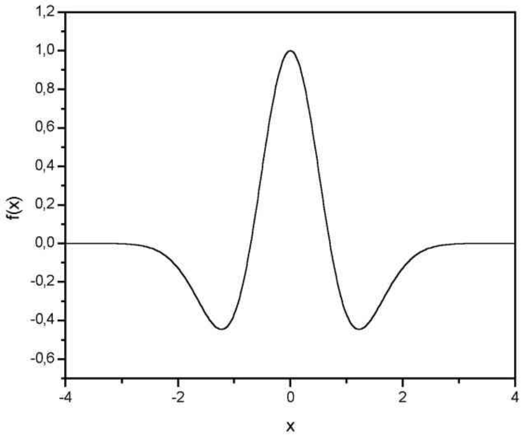


Figure 2.1: Mexican hat mother wavelet - The second derivative gaussian is an ideal mother wavelet for peak finding. It enhances the peak while diminishing the surrounding noise.

Since the wavelet analysis is a multiresolution analysis in time and frequency, it requires a set of displaced and scaled versions of the mother wavelet. This set of functions is defined in equation Eq. 2.3:

$$\psi_{\tau,s}(t) = \frac{1}{\sqrt{\mid s \mid}}\psi\left(\frac{t-\tau}{s}\right) \quad s,\tau \in \Re \quad s \neq 0 \tag{Eq. 2.3}$$

In equation Eq. 2.3 s is a parameter that modifies the scale of the mother wavelet so it detects peaks of different lengths. And τ is a translation parameter that allows to position the mother wavelet in different places with respect to the calcium trace signal.

Now, given that a calcium trace in time $x(t)$ is a finite signal composed of values between 0 and 255 then we know that it is a finite energy signal. We have that the wavelet transform of the calcium trace $x(t)$ with respect to the mother wavelet ψ is defined in equation Eq. 2.4 and is denoted $X(\tau, s)$. As it can be seen, the transformation performs a mapping from a one-dimensional data to a two-dimensional surface which depends on s and τ.

$$X(\tau, s) = \int_{-\infty}^{infty} x(t)\frac{1}{\sqrt{\mid s \mid}}\psi^*\left(\frac{t-\tau}{s}\right) \quad dt \tag{Eq. 2.4}$$

The wavelet transformation of the original calcium trace is a characterization of the time and frequency components of the signal in terms of the variables s and τ. In figure 2.2 it can be observed how the signal activity peaks are enhanced in the 2D surface of the continuous wavelet transform.

In figure 2.2 it is also possible to see the ridges of the wavelet transform as the dotted yellow lines that seem to follow the peaks of the signal. The way these ridges are formed will be explained next.

Peak Identification

The algorithm follows the wavelet transformation as a guideline in order to identify the activity peaks. The wavelet transform is used in two ways. On one hand, the ridges of the wavelet transformations are used and on the other hand, the signal noise is estimated from the transformation. I will first explain the process to form the ridges

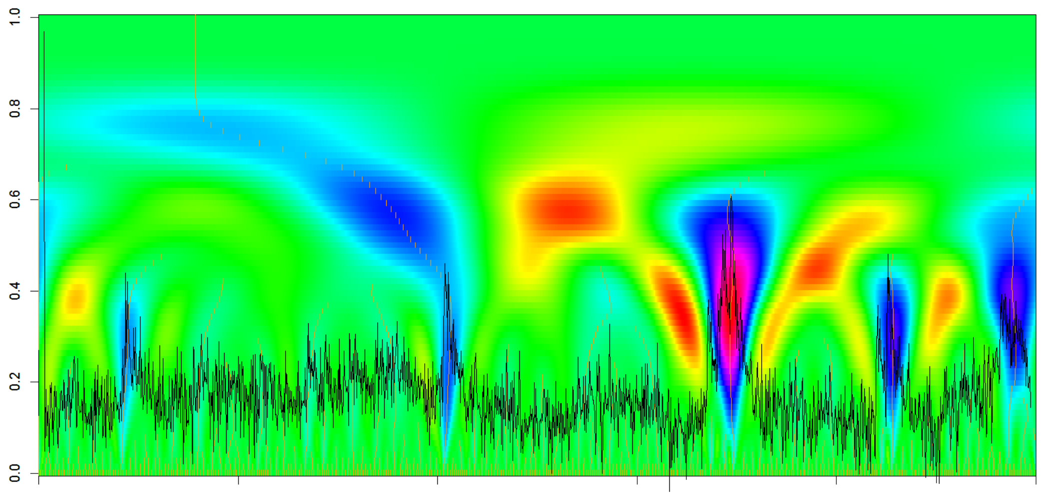


Figure 2.2: Example of a signal and its wavelet transform - The calcium trace is the black line in the front and the wavelet transform of the signal is plotted as background. It can be observed how the transform characterizes the desired activity peaks on the signal.

and how these are used as stencils for the peak location. Later I will explain how the noise is estimated from the wavelet transform.

The ridges are formed by following the local maximum in the transformation space (the colorful background in figure 2.2). Starting at the largest scale of the transform, the algorithm finds the local maximum by simply using a sliding window. Then, beginning at the largest scale, the ridges are formed by linking local maximum points of successive scales. Each of the local maximum points on the biggest scale is a new ridge and the algorithm finds the closest local maximum point on the next scale and adds it to the correspondent ridge. After ridges started on the highest scale are formed, new ridges are defined starting at the local maximum point of the second largest scale which were not yet assigned to any ridge. Then the ridge conformation procedure is repeated. At the end, all local maximum points are assigned to some ridge. In some cases the ridges are long and in other cases they are short. This can be seen in the yellow dotted lines in figure 2.2. It is easy to observe that the ridges follow the activity peaks on the signal but exactly how they are used to identify the peaks is explained next. Each ridge is taken as a candidate to contain a peak and three simple criteria are applied to define whether the peak it contains is a real activity peak or simply noise. First, the local maximum point on the highest scale on the ridge should be within a certain interval. This ensures that the peak is not too wide and not too fast. The second criteria is that the peak contained in the ridge has a signal to noise ratio (SNR) higher than a certain user defined value. This criteria of course depends on the noise estimation which will be explained later. The third criteria to define whether the ridge contains a peak is the length of the ridge. It should be at least longer than a certain threshold. This last criteria discards lots of small local maximums which are just noise. The noise is estimated as the 95% quantile of the CWT coefficients at the smallest scale. This accounts for all the small peaks and small variations observed in the signal. The algorithm is based on the work presented in [23] and was implemented using the R package [28].

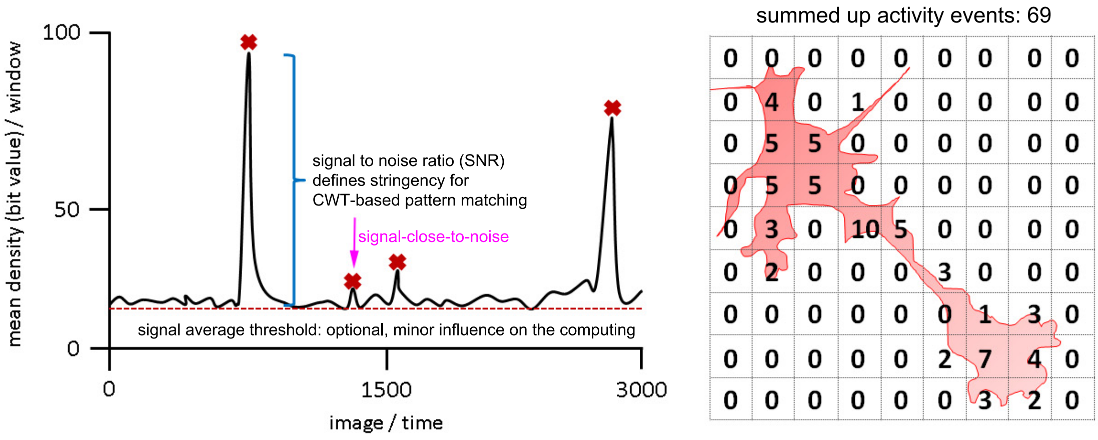
Extra features

Once peaks are identified and if the user decides to use these features, the variance of the signal is estimated and a general change in tendency point can be found. The variance is simply calculated as the second moment of the signal using a sliding window of a size defined by the user and the point of change in the general tendency is estimated by finding the point that divides the signal into the two most different parts, different in terms of mean and variance. This difference is probed with a chi square test.

2.2.3 Results

The algorithm works in the following manner. The field of view of the video is divided according to a grid of a user defined size in pixels. Each square on the grid is processed as a single region of activity. This represents no loss of resolution since the minimum size of the grid can be one pixel. The pixels within each square of the grid are averaged and the intensity trace of each of them is extracted. At the end of this stage we have a set of signals. Each of them corresponds to an intensity trace of one square in the grid. On the right side of figure 2.3 the grid conformation is presented.

The concepts of SNR and estimated noise are also presented in figure 2.3 and were previously explained in the informatics methods section. In this figure the three main


100
50
0
mean density (bit value) / window
signal to noise ratio (SNR)
defines stringency for
CWT-based pattern matching
signal-close-to-noise
signal average threshold: optional, minor influence on the computing
0
1500
3000
image / time
summed up activity events: 69

0	0	0	0	0	0	0	0	0
0	4	0	1	0	0	0	0	0
0	5	5	0	0	0	0	0	0
0	5	5	0	0	0	0	0	0
0	3	0	10	5	0	0	0	0
0	2	0	0	0	3	0	0	0
0	0	0	0	0	0	1	3	0
0	0	0	0	0	2	7	4	0
0	0	0	0	0	0	3	2	0

Figure 2.3: Algorithm basic features - The basic features for the peak detection are presented on the left. The cartoon shows an example signal and gives an idea of what the SNR and the noise level are. On the right, also a cartoon is presented to clarify the concept of the grid used to analyze the video.

values defined by the user are presented. There is the SNR already mentioned, the signal average threshold, and the window size. The window size is simply the size in pixels of a square in the grid. The signal average threshold is a value to discard signals that correspond to the background pixels. There are pixels where no neuron is present but nevertheless contain intensity variations product of the noise. Such intensity variations can be confused with real activity. So the easiest is to discard such regions from the beginning and not even process the correspondent traces.

Once the traces are extracted, they are stored in a matrix and analyzed in R. Finally the results are presented on the graphical user interface in Bio7 and also in a PDF file stored in the results folder. The complete pipeline of the software is presented in figure 2.4.

In figure 2.5 the user interface of the tool is presented. A detailed explanation of how to use the tool is presented in the supplemental material at the end of the thesis. That is a user manual we wrote for the tool.

The generated PDF file is composed of a first page where the spatial location of the detected activity is depicted. A frame of the video is presented and red circles mark the points where calcium activity was detected. The bigger the circle, the more the

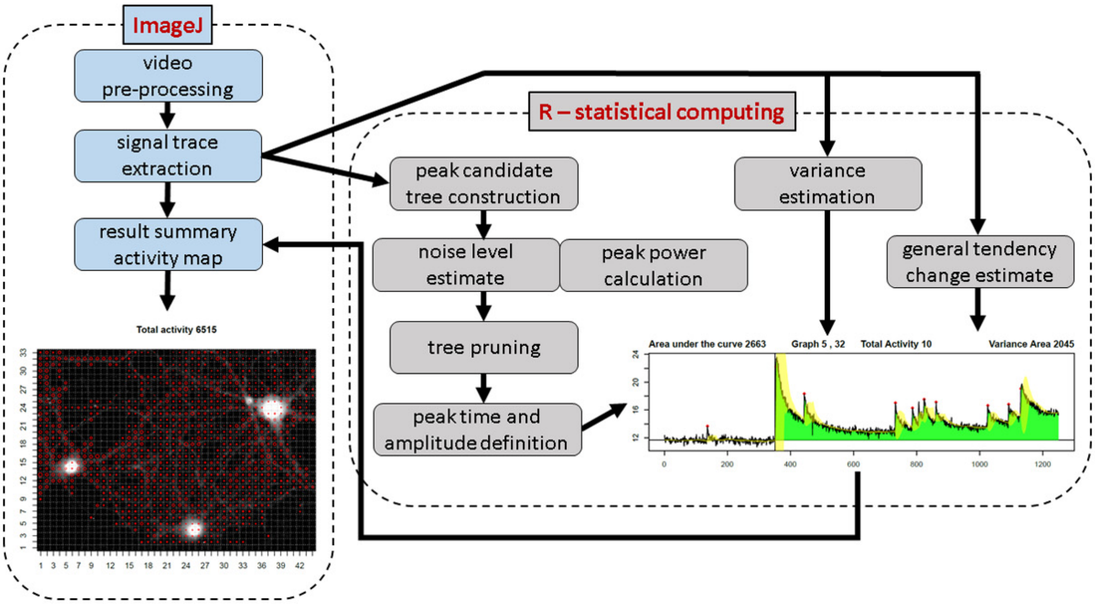
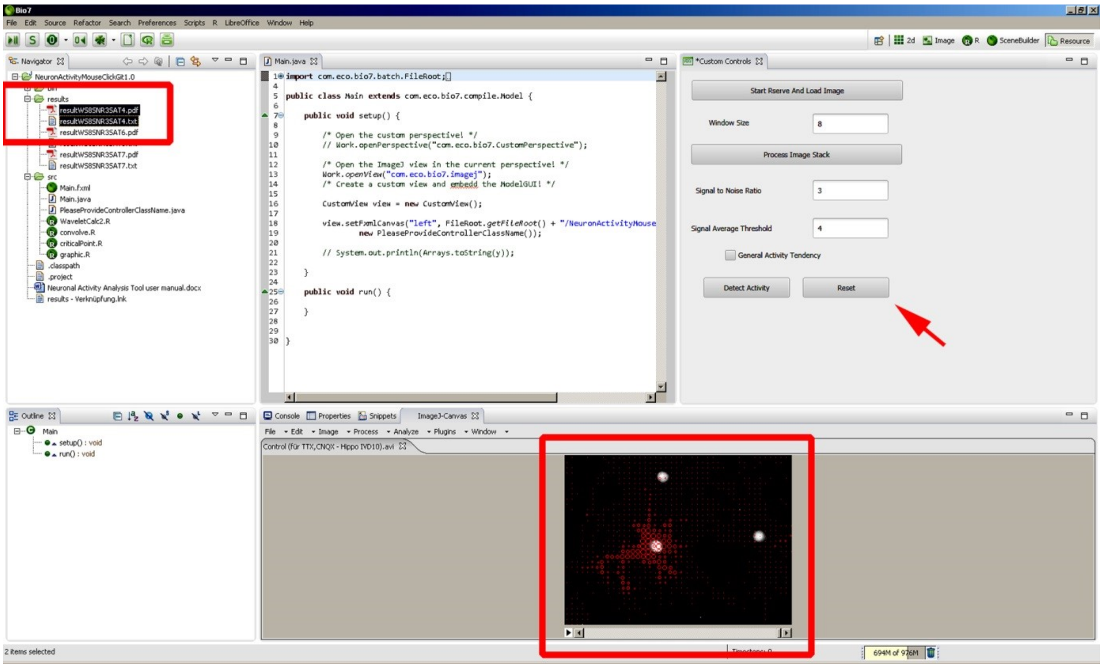


Figure 2.4: Pipeline of the tool - The complete pipeline of the tool is presented, marking which parts were programmed in ImageJ and which ones in R. Both environments, imageJ and R, are embedded in Bio7.


Bio7
File Edit Source Refactor Search Preferences Scripts R LibreOffice Window Help
2d Image R SceneBuilder Resource
Navigator
NeuronActivityMouseClickGit1.0
results
resultWS8SNR3SAT4.pdf
resultWS8SNR3SAT4.txt
resultWS8SNR3SAT6.pdf
resultWS8SNR3SAT7.pdf
resultWS8SNR3SAT7.txt
src
Main.fxml
Main.java
PleaseProvideControllerClassName.java
WaveletCalc2.R
convolve.R
criticalPoint.R
graphic.R
.classpath
.project
Neuronal Activity Analysis Tool user manual.docx
results - Verknüpfung.lnk
Outline
Main
setup() : void
run() : void
Main.java
*Custom Controls
Start Rserve And Load Image
Process Image Stack
Detect Activity
Reset
Console Properties Snippets ImageJ-Canvas
File Edit Image Process Analyze Plugins Window
Control (für TTX,CNQX - Hippo IVD10).avi
2 items selected

```
import com.eco.bio7.batch.FileRoot;

public class Main extends com.eco.bio7.compile.Model {

    public void setup() {

        /* Open the custom perspective! */
        // Work.openPerspective("com.eco.bio7.CustomPerspective");

        /* Open the ImageJ view in the current perspective! */
        Work.openView("com.eco.bio7.imagej");
        /* Create a custom view and embedd the ModelGUI! */

        CustomView view = new CustomView();

        view.setFxmlCanvas("left", FileRoot.getFileRoot() + "/NeuronActivityMouse
                new PleaseProvideControllerClassName());

        // System.out.println(Arrays.toString(y));

    }

    public void run() {

    }

}
```

Window Size 8

Signal to Noise Ratio 3

Signal Average Threshold 4

☐ General Activity Tendency

Figure 2.5: User interface of the tool - So looks the user interface of the tool in the Bio7 environment. A full explanation of how to use the tool can be found in the supplemental material.

events detected on that point. The rest of the PDF document presents the intensity traces of each of the regions where activity was detected. The regions are organized using an X-Y coordinate system so any red circle on the initial page can be related to its intensity trace by searching the correspondent coordinate pair. An example can be found in the user manual in the supplemental material.

Now we present the result of several tests applied to the tool. Motoneurons and also hipocampal neurons were used for the experiments. In the following experiments, the tool characteristics were studied to define how the user tuned parameters affect the results. It was also studied how the tool can be used to detect local spontaneous activity. The figures also give some insight into the functioning of the algorithm. The results presented here are reproduced from the manuscript submitted to the journal Plos Computational Biology. Even though the figures and legends are the same, the discussion is done here with a different approach.

The figure 2.6 shows the calcium activity assessment of a spontaneously active motoneuron. It also depicts the influence of the user input parameters, the signal average threshold, and the signal to noise ratio. The panel *a*) of the figure presents the principle of calcium activity event detection. Activity event identification is shown for a representative grid window (per se representing a ROI) on a single motoneuron. This motoneuron at DIV 3 shows global calcium transients and its activity shifts from a low activity state to a high activity state. Calcium imaging was performed at 2.5Hz and 2600 images (frames; x-axis) were acquired. The grey trace shows the raw mean intensity values of a representative grid window. After extraction of the image signal in a grid window, all local maxima of the intensity signal are identified at several scales (y-axis). These sets of points are shown by the vertical blue lines and are organized in branches forming a tree. The belonging to a branch is determined by the ridges of the wavelet transform. These ridges are shown as brighter regions (white to yellow) in the heat-map to indicate higher transformation values and regions of higher activity. This process ends in a tree where each branch contains local maxima points which belong to the same ridge of the wavelet transform. The tree branches are pruned to select the remaining extreme points which are marked as activity events and here shown as blue dots.

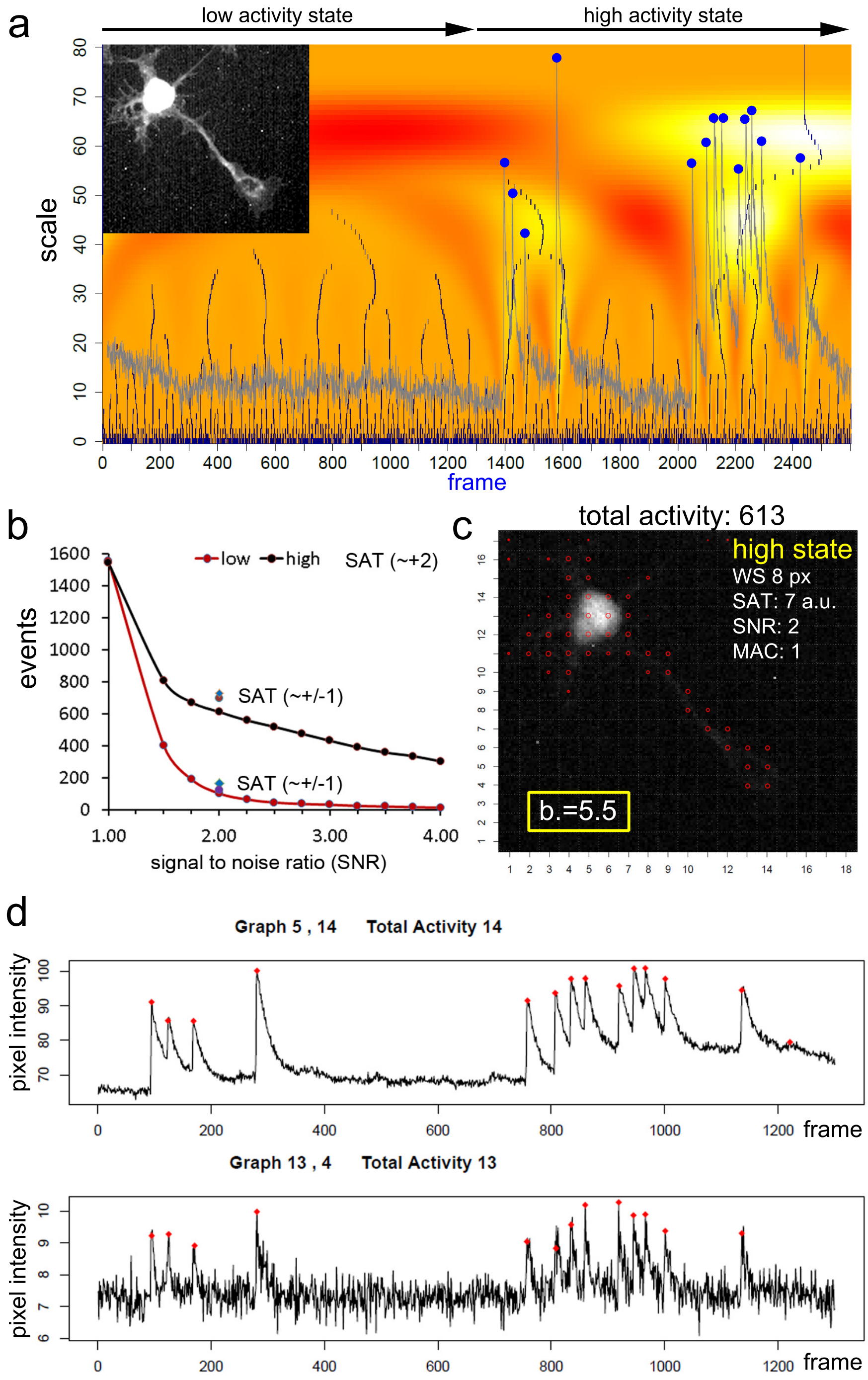


Figure 2.6: User parameters influence in spontaneous calcium activity detection in motoneurons - a) Presents an example intensity trace, its wavelet transform in the background and a frame of the video from which the trace was extracted. b) Essays of several user conformations. c) Space location of the detected activity and total number of events found. d) Two example traces in presented as they appear in the result PDF file. A more detailed explanation of the figure is presented in the text.

Panel *b*) of figure 2.6 presents the effect of the tuning parameters on calcium activity events detection. The total number of computed activity events (y-axis) is depicted in relation to changes in the user-defined signal-to-noise ratio (SNR). Two activity stages of the motoneurons are compared: the low activity state (panel *a*), frame 1 - 1300) and the high activity state (panel *a*), frame 1301 - 2600). Discrimination of the high activity state and the low activity state is very effective over a broad range of SNR values from 1.5 to 4. The signal average threshold was set to an intensity value of 6. A conservative SAT value was selected (7) and modified at a SNR of 2 (blue square, SAT = 5 ; purple circle, SAT = 6). Changes in SAT do not affect the robust discrimination between the high activity state and the low activity state.

Also in figure 2.6, in panel *c*), we find the data documentation and the spatial x,y summary. The image shows the distribution and number of calcium activity events raised by a spontaneously active motoneuron. This image is automatically generated by the program. The user-dependent tuning parameters for this analysis are given. The image field (142 x 130 pixels) was automatically split in a grid of 8 x 8 pixel (WS 8 px). SAT was selected to be 7 (see above), and SNR was selected as 2. Red circles indicate areas with calcium events. The smaller the diameter, the less activity is found in the corresponding grid window. All detected activity events are summed up to offer the value "total activity". Finally in panel *d*) of this same figure, two examples of signal traces are shown. The individual traces represent changes in fluorescence in one grid window. The tool automatically generates traces (black line) representing a grid window showing the raw bit values (y-axis) over the frame number (x-axis). Calcium activity events detected by the tool are labeled with a red square. The upper panel describes the graph in grid 5/14 (x/y-axis) in the somatic region of the motoneuron. Here, raw bit values ranged from about 65 to 110. In the lower panel a region from the growth cone of the motoneuron was analyzed (grid 13/4; x/y-axis). Here, raw mean bit values in the grid range from 6 to 10. Note the robust detection of global activity despite an almost 10-fold difference in the mean intensity values in the corresponding grid window.

A confocal image of synchronously spiking glia-derived neurons is presented in figure 2.7. The data was earlier acquired and published in [29] and now it was analyzed with the help of the tool. Cells were loaded with calcium indicator OGB1 to label glial cells

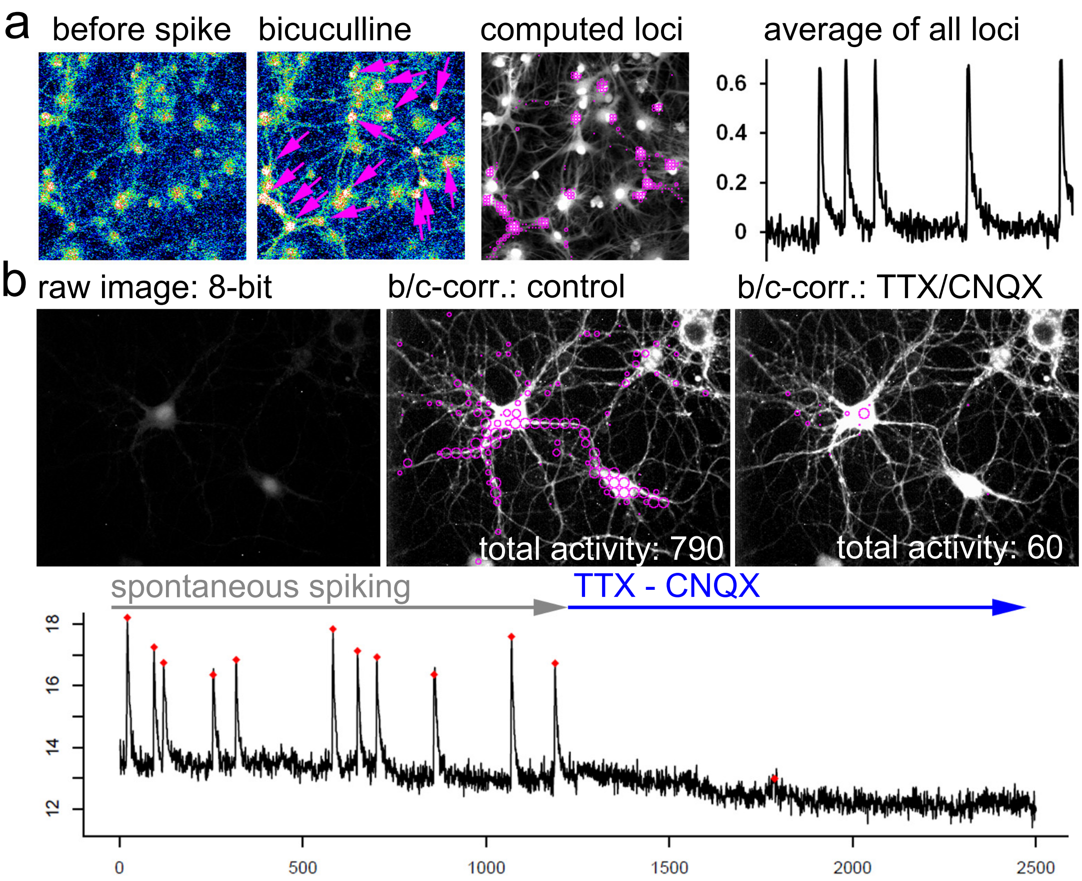


Figure 2.7: Calcium spike detection under stringent computing conditions - a) comparison of the confocal microscopy view of the motoneuron culture and the computed activity by the tool after the analysis of the video. All neurons in the imaged are marked with the magenta arrows. b) spike train and the times when the control condition is present and when the spike blocking compound is applied.

and neurons. Spontaneous activity was induced by inhibition of GABAergic signaling with bicuculline to induce the spiking behavior in the neural network. All neurons in the image are marked with the magenta arrows. On the left part of the panel *a*) in figure 2.7 the average spiking trace of the computed loci is shown. Imaging was performed under low-light conditions. The raw image represents this imaging situation. Loci of computed activity events are shown on a brightness-contrast corrected image of the neurons. Under control conditions, spontaneous spiking is observed on the somata, but also in the periphery of the neurons. Spike blockade (TTX, CNQX) correlates with a reduced number of computed activity events. Finally, in the lower part of panel *b*) of figure 2.7 the raw intensity values are plotted against the frame number. The calcium spikes are efficiently blocked by TTX and CNQX.

In Figure 2.8 the average intensity of cultured Hippocampal neurons loaded with a fluorescent calcium indicator is shown. The control neurons are at a low activity state and the cLTP are neurons in a global activity state after being treated with a chemical LTP induction solution. Calcium imaging was performed for 55 seconds with a speed of 20 Hz. By setting the user parameters to WS = 8, SNR = 2.5, SAT = 11 and MAC (Minimum activity count)= 2 a total activity of 2135 events for the control condition is obtained. Regions of high activity are indicated by circles with a larger diameter and these are regions where synchronous activity occurs. Under cLTP treatment, neurons increase their total activity number to 4785 events. Also, in figure 2.8 in panel *c*) some example traces are shown, for the control condition as well as for the high activity state condition. Under the normal condition the detected activity is localized and spontaneous as in the high state condition there is a strong correlation between distant parts of the culture. Calcium spikes in loci of synchronous activity. Neuronal somata do not exhibit a spiking behavior. Calcium spikes are identified in the periphery in the indicated grid windows. Yellow squares point to grid windows which are shown on the right with the corresponding signal trace and the activity marks on the traces. The blue arrows point to other grid windows showing the same synchronous activity pattern. Out-of-synchronicity events are also detected (red arrow on the right). Some obvious calcium spikes were overseen by the computation due to the stringency parameters used for this analysis (purple arrows).

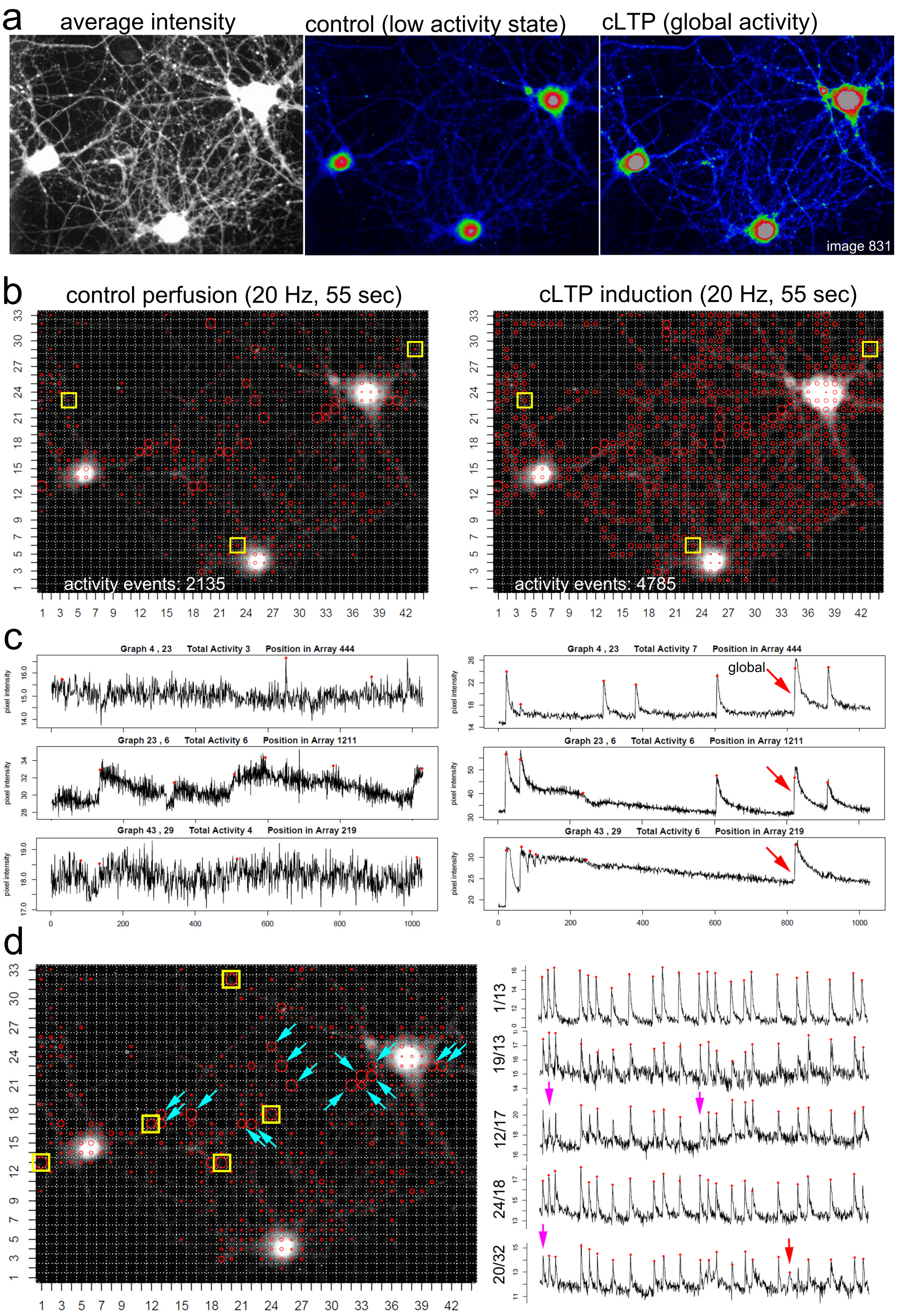

a
average intensity
control (low activity state)
cLTP (global activity)
image 831
b
control perfusion (20 Hz, 55 sec)
cLTP induction (20 Hz, 55 sec)
activity events: 2135
activity events: 4785
c
Graph 4 , 23 Total Activity 3 Position in Array 444
Graph 23 , 6 Total Activity 6 Position in Array 1211
Graph 43 , 29 Total Activity 4 Position in Array 219
Graph 4 , 23 Total Activity 7 Position in Array 444
Graph 23 , 6 Total Activity 6 Position in Array 1211
Graph 43 , 29 Total Activity 6 Position in Array 219
global
pixel intensity
d
1/13
19/13
12/17
24/18
20/32

Figure 2.8: Activity profile of cultured hippocampal neurons before and after activity induction - a) Field of view of a calcium imaging video of cultured hippocampal neurons under two different conditions, low activity state in the middle and high activity state on the right. b) Same two conditions from panel *a*) after the tool analysis. c) Some example traces from the loci marked in panel *b*). d) detailed analysis of the loci with high activity during the low activity state.

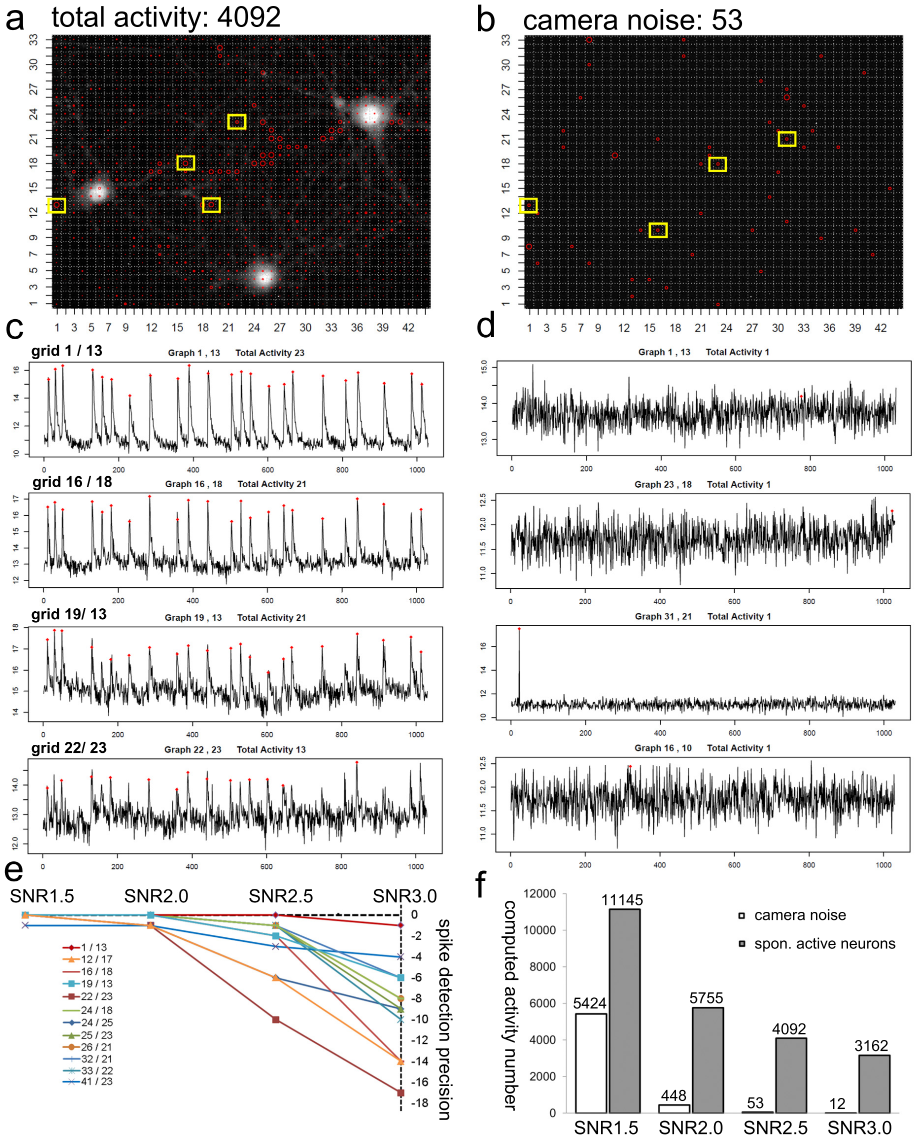


Figure 2.9: Comparison of spontaneous active hippocampal neurons and a pure noise video - a) Field of view of a calcium imaging video of cultured hippocampal neurons. b) Field of view of the pure noise video. c) Some example traces from the hippocampal culture in panel *a*). d) Sample traces from the noise video in panel *b*). e) Curves of the detection precision as the SNR parameter is changed. f) Statistics of the total number of events detected for different SNR values.

In figure 2.9 a comparison between two videos is presented. The first video is a calcium imaging of cultured hippocampal neurons. Since the second video is a recording of an empty plate with only the buffer we could study the setup noise. This figure shows the field of view of the two videos where it can be already seen that the pure noise video presents a seemly high number of activity points. But looking more attentively into it we see that all these points are marked with only a single activity peak detected on each of them. This is shown later on the same figure in panel *d*). Finally, in panels *e*) and *f*) from the same figure, the statistics comparing the results of detected activity in the two cases are depicted. It can be seen that for low SNR values the false positive detection increases. This illustrates how critical is the tuning of this parameter.

Figure 2.10 presents a case of a control culture of hippocampal neurons in normal conditions compared to a cultured exposed to a spike blocking substance. The cocktail used to block activity and excitatory neurotransmission is tetrodotoxin, CNQX, and APV (D-2-amino-5-phosphonovalerate), an inhibitor of NMDA receptors. In the figure it is appreciated that the blocking cocktail severely reduces the calcium activity and that the tool properly detects the activity in the control condition and ignores most of the noise under the blocking condition. The figure also illustrates how some activity remains despite the blocking cocktail and how the tool correctly marks these activity peaks. Most of the activity found during the blockade condition is local, non-spike like activity which corresponds both to noise and real remaining calcium activity.

It is natural to wonder whether the small peaks detected by the tool correspond to real calcium activity or they are just misinterpreted noise. In order to shed some light into this question we performed another blocking condition experiment. We hypothesized that under blocking conditions the calcium influx should be reduced as well as all calcium influx in general and not only the one seen during spike peaks. If this is true then the variance of the signal should be diminished, so we used the variance calculation feature of the tool to check this hypothesis. The results are presented in figure 2.11. We again imaged hippocampal neurons (DIV10; 10.000 images, 20Hz) and compared neuronal activity under control conditions and after acute activity blockade with TTX and CNQX. We saw that the calcium spiking was blocked and using the variance calculation we could also see that it was also much lower during the blocking condition. The yellow shadow seen on the intensity traces is an area equal to two times

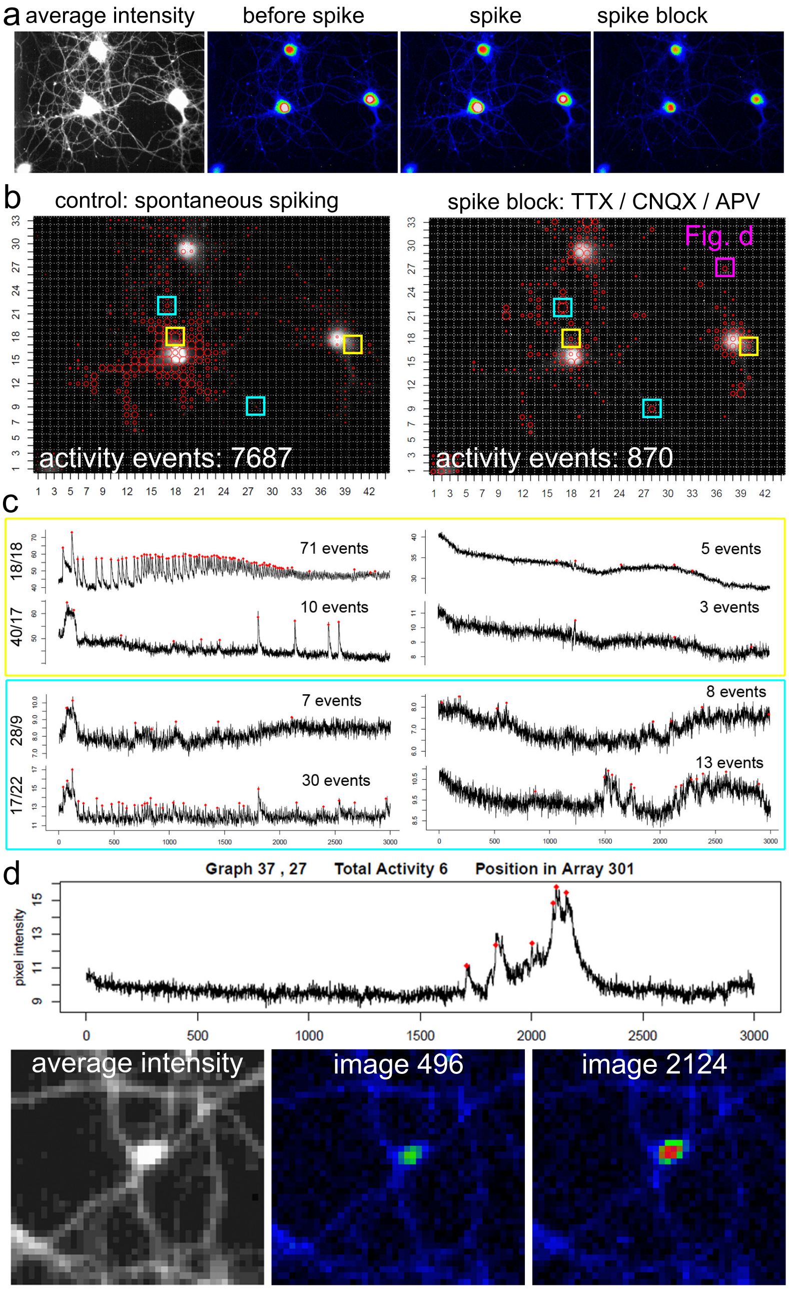


Figure 2.10: Activity profile of hippocampal neurons after calcium spike blockade - a) Confocal microscopy images of the recorded videos. b) Total results of the control culture and the blocking condition. c) Intensity traces from the marked regions in panel b. d) Intensity trace and confocal images of the region marked in purple in the panel b right picture.

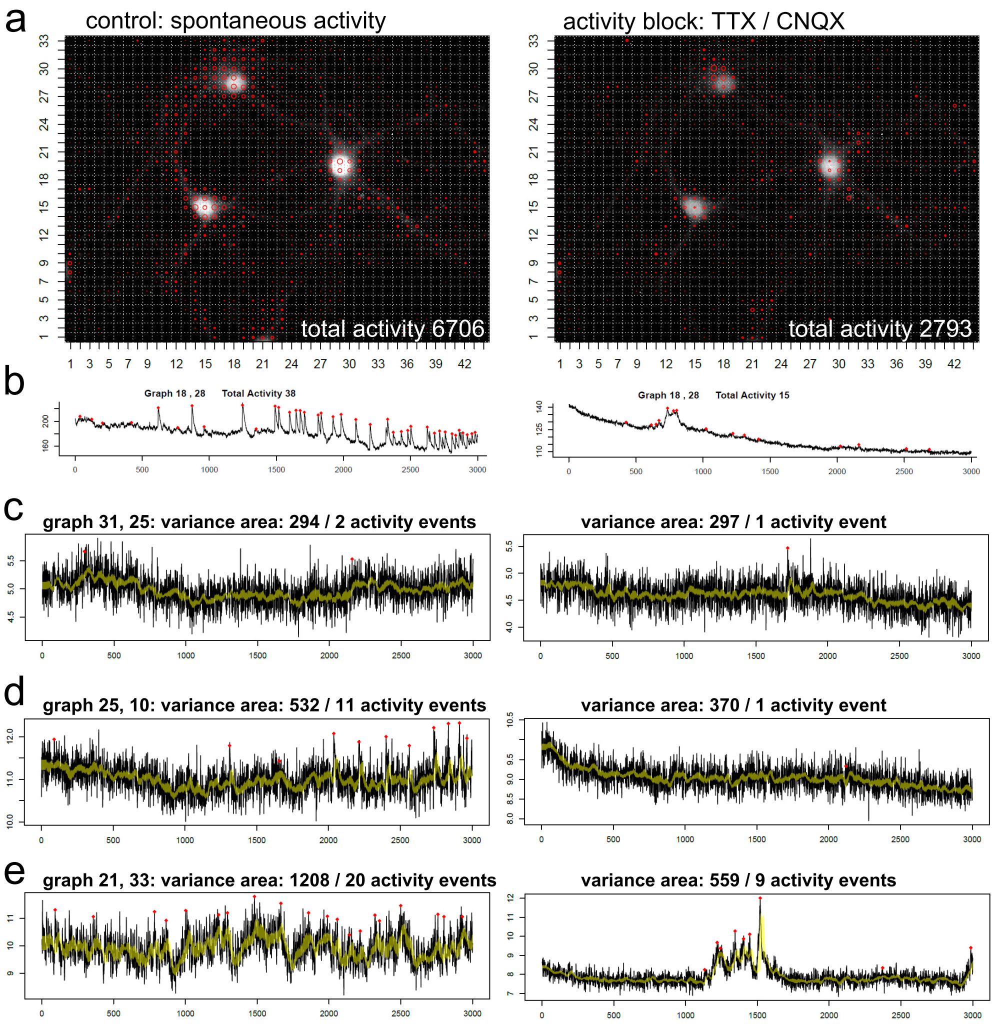


Figure 2.11: Parallel computing of activity events and variance area - a) Frames of the spontaneous activity video (left) and the blocked condition (right). b) Traces examples of each condition, control (left) and blocked (right). c), d), e) Intensity traces plot with the estimated variance of each trace. The variance is calculated using a sliding window of size 30 and shown as the yellow area. Each panel shows a different case. In panel *c*) The variance is almost the same in both conditions and also the detected events. In panel *d*) the variance remains close but the detected events drops. And finally, in panel *e*) both variance and detected events drop.

the variance, simply plot around the signal average. It can be seen that the variance area reduction is significant in axons and dendrite areas and not so in background areas and directly in the neuron cell body. In some areas the variance was significantly reduced to less than 50% which is a high reduction.

Despite the blocking conditions imposed to cultured neurons, some activity remains and it is detected by the tool 2.12. Zooming in on corresponding loci shows that growth cone-like structures (Fig. 8, trace 1-3), neuritic elements (trace 4), or varicosity-like structures (trace 5) form these local activity hotspots. Computed activity events exhibit either a spike-like shape (trace 1, 2), or reflect a phase of increased fluctuations (e.g. trace 4: images 1000 - 2000), or a sudden jump in the activity state (trace 5: image 2300 - 3000). By looking at the data it is possible to verify that the detected activity is not a misinterpretation of noise by the tool but real activity that remains despite the blocking cocktail. We see different kinds of remaining activity like, spikes, some general increase in the base line, and some non-spike like activity. This non-spike like activity is further studied in figure 2.13.

In order to study the problem of false positive recognition we decided to perform an essay of calcium activity blocking and posterior rescue of the activity to see if the tool correctly distinguished the different states. The results are presented in figure 2.13. To perform this experiment, we imaged calcium fluxes in resting hippocampal neurons and acquired 6600 images at a frequency of 10 Hz. After one minute under steady-state conditions, neuronal activity was blocked with a high amount of TTX (500 nM), CNQX, and APV (each 20 uM). Then, extracellular calcium was withdrawn for more than three minutes, before extracellular calcium was re-added to stimulate neuronal SOCE. Computational analysis was then performed under medial stringency conditions (SNR 2.5, MAC2). We found that the non-spike like activity detected was clearly reduced during the blocking stage, as well as the computed signal variance area. As the blocking condition is terminated and the calcium activity is rescued, the variance area and the non-spike like activity detection go back to normality. This clearly shows that the tool correctly identifies real non-spike like calcium activity, which by no means corresponds to noise in the measurements.

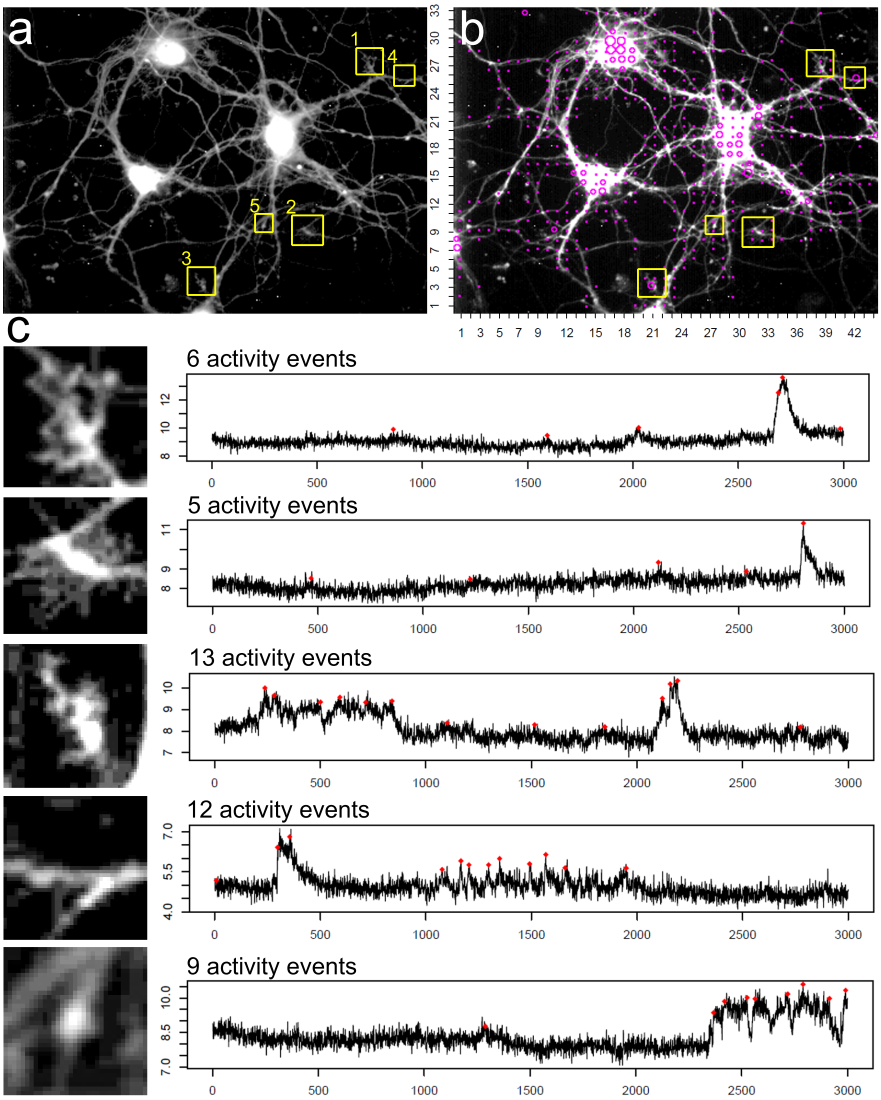


Figure 2.12: Structures with high rates of local activity after calcium spike blockade - a) Average intensity image of hippocampal neurons loaded with the calcium indicator. Grid windows with high rates of local activity are shown in yellow. b) Activity map. c) Five regions of interest are indicated. The upper three represent growth cone-like structures, the lower two traces represent hotspots on neurites. Note the variability and the diverse character of the calcium signal patterns detected by the computational approach.

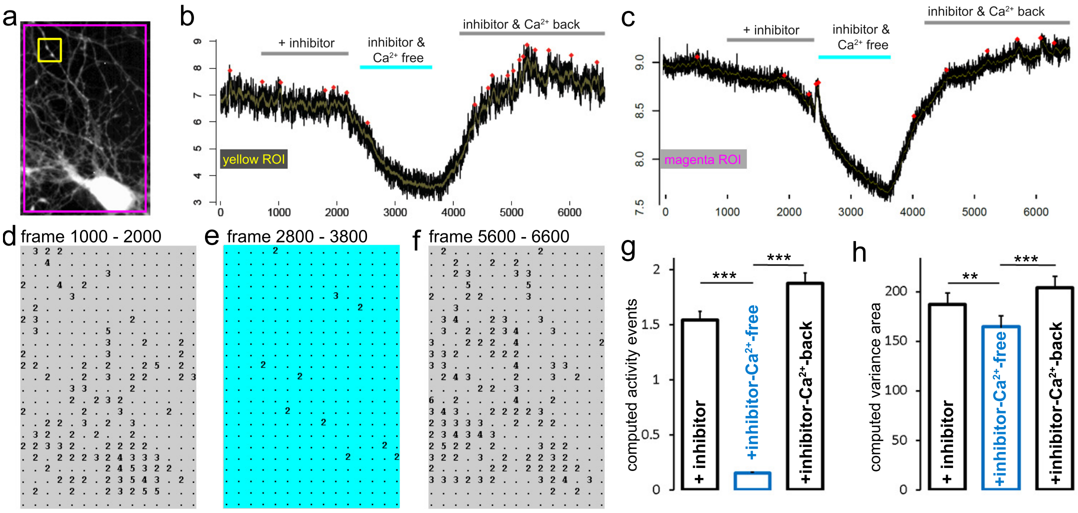


Figure 2.13: Computing of signals close to the noise level - a) Typical hippocampal neuron, loaded with calcium indicator. Two ROIs are indicated. b) and c) Calcium traces representing the yellow and magenta ROI in panel a. Removal of extracellular calcium causes a decline in the calcium indicator signal. This correlates with a reduced number of computed activity events. d), e) and f) Number of computed activity events are shown on the x,y-grid. Under calcium-free conditions (cyan), a low amount of activity events was found in the signal trace. In presence of extracellular calcium, more activity events are computed by the algorithm. Regions of activity are found in the soma, but also on distal neurites. g) and h) Summary graphs for computed activity events and variance area.

2.3 Discussion

In this project we developed and exhaustively tested a tool for automatic calcium detection which is designed to assist researchers to solve neurobiological questions. The tool is basically an algorithm for peak detection but it also offers other useful features and it is embedded in the Bio7 environment which is familiar to researchers in biology. The tool is fully open source which means that advance users can adjust the code to their particular needs in case they consider it necessary. For more basic users it is also possible to take advantage of the imageJ and R compartments in Bio7.

The peak detection algorithm is based on the wavelet transform, using it to build a tree of peak candidates and to estimate the noise present in the signals. Then the tree is pruned according to criteria of noise and relevance of the candidate leaving at the end only real calcium activity peaks. This method ensures that not only spike like peaks are detected but also smaller non regular shaped peaks.

As mentioned before, the tool was thoroughly tested in order to fully understand its functioning, its capabilities, and weaknesses. There are of course other tools available for calcium image analysis but none of them is open source and none of them is capable of detecting non-spike like calcium activity. Most of them also rely on user defined ROI's and do not explore the complete field of view of the videos. It is also fair to say that some of them are more complex tools than ours. They not only identify the calcium activity but also try to segment the neurons in the video. In summary, we believe in the utility and user friendly design of our tool as a strong argument to encourage the use of it, but we are aware of the existence and high capabilities of other tools available [10, 11, 13, 15, 30, 31].

Influence of the user defined parameters

An algorithm that understands the researcher's concerns and is capable of setting a value for the user defined parameters of our tool is not yet realistic. Different research projects have different interests so they will probably focus on different aspects of the calcium videos. This level of flexibility is not yet reached by automatic algorithms, which is why the most logical option is to leave some parameters to be defined by the user. Even more if the tool is used for some other applications outside neuroscience.

In our tool we created a total of 4 user defined parameters. These parameters provide the flexibility that the tool requires to be used in several applications, but they are still intuitive enough so the user can set their values in a coherent way, not guessing. The window size should be set according to the size in pixels of the phenomenon to be studied. The SAT should be set according to the average background level, so an unnecessary amount of computation is done. The SNR is chosen according to the level of noise present and to the expected size of the activity peaks. This is for sure the value that requires more trials in order to find a satisfying point. This value shows a simple direct relation with the result so it is simple to try different values and get a feeling of how it affects the outcome. Finally, there is the variance sliding window size which is not critical and simply depends on the capture rate of the video.A good thumb rule is 1% from the total number of frames.

Signal fluctuations and activity detection

The calcium activity fluctuations can be classified in three separate groups. The first group being the well studied neural spike activity, with a clear shape and easy to detect. The second group is the non-spike like activity, which corresponds to local spontaneous activity and is way less understood. Finally, the last group would be the signal fluctuations which not even classify as peaks but are clearly present and most probably have an effect in the general dynamics of calcium activity in neurons. The brain maintains such a delicate balance in calcium dynamics that it would be really surprising to see that such changes in variability as the ones observed in figure 2.11 are meaningless in a physiological sense. As seen in figure 2.13, the number of detected activity and the signal variability are strongly correlated, which is a clear indicator of the relevance of these two phenomena in neurons.

ROI definition versus a grid of ROIs

The most recent advanced tools for calcium imaging videos analysis are capable not only of detecting the calcium spike activity but also of segmenting the neurons in the video visual field. This is achieved by assuming that the activity matrix is a product of two matrices. One matrix corresponds to the time print of the calcium activity and the second one to spatial print. By doing matrix factorization they estimate each of these matrices and based on each one they do the peak detection on one side, and the neuron segmentation on the other. There is a high risk in these approach since the correct segmentation of the neurons depends on the spike activity detected. That means that in cases of non-spike like activity the tool would not be able to correctly segment the neurons since it is tuned to discard this type of activity. So, these other tools work satisfactorily well when it comes to spiking activity, but as mentioned before, there is a lot of non-spike like relevant activity which cannot be regarded as noise because it is clear that it is biologically relevant. We believe that analyzing the complete field of view is not a big loss, since most of it is normally populated by neurons or other structures of interest (e.g. glia cells). Moreover, we are able to discard a big portion of background simply by setting the correct value of the SAT parameter. With this method we are able to detect the activity reliably without depending on a correct segmentation.

One mayor drawback from not having the neurons segmented is that we are not able to do network analysis. Since we do not separate the neurons as independent entities, then we cannot do an analysis of how they interact with each other. This might not be a big problem in videos of cultured images, since the network organization of these cultures might not have much to say biologically, but as *in vivo* imaging is getting better, it is certain that the point will come when it will be really important to have access to this information.

Practical considerations

The presented tool is completely open source and it is embedded in an also open source environment. This is a very important feature in terms of flexibility of the tool and its utility for different research areas. The tuning procedure of the parameters is pretty simple so it should be seemly a fast task to find the correct parameters for a determined application. The installation of the tool is also pretty straightforward since the Bio7 environment already includes R and imageJ. Hence, after the installation of Bio7 the user just needs to download the tool from the Github repository and import it to his Bio7 workspace as it is explained in the instruction manual.

In order to tune the parameters we recommend starting with low values for the SAT and SNR and increase them slowly until finding the expected performance. On the contrary, for the window size parameter it might be a good idea to start from a big number and slowly reduce it. What a big number means is of course dependent on the application and the set up of the experiments. The algorithm performance and computational complexity is seemly low, for example using a standard desktop computer, 3,000 images (348 x 260 pixel) are computed for one minute to process the image stack, and for another three minutes to complete the wavelet transform and the generation of the documentation PDF. One thousand images are computed in less than two minutes.

Typical biological questions for use of the activity detection tool

The local non-spike like activity that our tool is capable of detecting is a crucial part of the physiological processes occurring in the brain. The following information is mostly taken from our manuscript presenting the tool in the Plos Computational Biology journal. It is known that spontaneous, local, and small calcium signals are involved in several biological functions [18, 19, 32, 33, 34]. It is most likely that these signals result from the parallel contribution of different biological mechanisms [3, 4, 6, 35]. It is not clear exactly which molecules are involved in these processes but they are certainly good targets for potential protective and functionally restorative treatments in psychiatric and neurological disorders. For example, the treatment of motoneuron diseases offers a research environment to try our calcium activity detection tools. Because motoneruons show cell-autonomous spontaneous calcium transients, which appear in an unpredictable spatiotemporal on-off, and high versus low frequency pattern [20, 36, 37, 38, 39]. A clear example of an interesting mechanism of local and spontaneous activity is the NaV1.9 channel [19, 40]. This is a subthreshold active ion channel capable of triggering the local excitability patterns in motoneurons. According to experiments in mouse models, the NaV1.9 channel is disturbed in cases of spinal muscular atrophy [19, 40]. This is critical since that disease is the most common cause of infant mortality [41]. Such mechanisms like this channel can only be studied with the help of an unbiased detection, capable of identifying non-spike like local activity. So as we mentioned in the manuscript ”Screening-like approaches on the basis of calcium imaging and automated excitability analysis with bioinformatics may offer new information on the role of these genetic factors in motoneurons and patient-derived induced neurons.” Beside this specific mechanism, there are many other mechanisms, some of them probably still unknown, which are excitability factors of the neural networks in the brain. These factors maintain the brain oscillatory activity [42, 43], so the study and understanding of these factors is a key component of neurobiology research.

It is also true that other local acting signaling factors like neuropeptides, neurotrophins or the contribution of subthreshold voltage changes are not well understood in terms of its role in synaptic activity. It is known that local tuning and scaling of excitability by means of calcium dependent pathways is an important agent in the process of synaptic development [33] and since the neural excitability is also affected by homeostatic calcium fluxes at rest, then these mechanisms that trigger and modulate the calcium internal dynamics must be studied and understood [44, 45, 46].

3

Deep Learning for biological image analysis

3.1 Introduction

Advanced electron microscopy techniques gain more importance every year. This development is accompanied by an ever increasing amount of image data. It is not only a matter of having more images but also of having more detailed images. Current electron microscopes are capable of imaging with resolution structures as small as the neuron nuclei and other organelles. The analysis of this data becomes more and more of a bottleneck in research. Automated image analysis will be the only way to deal with this issue effectively. Luckily there are also recent advances in machine learning algorithms allowing computers to consistently identify complex patterns in images. One of such advanced algorithms is the Deep Neural Network proposed by Geoff Hinton [47]. In this project we applied this algorithm to the difficult task of segmenting the neurons' nuclei in electron microscopy images of the *c. elegans* dauer larva.

C. elegans is a well know and used model in biological research. Since around 1963 it has been studied specially for neuroscience and molecular and developmental biology. This worm is about 1mm long and its robustness and transparent skin make it ideal for research. It was the first multicellular organism with a fully sequenced genome and it was the first organism with a fully known connectome. The connectome of the *c.*

elegans was first published in 1986 by White et al [48]. In that occasion, the labelling of the worm was done by hand and the quality of the images were not nearly as good as they are today. The *c. elegans* connectome has been corrected up to a certain point but there is still a lot of doubt of how accurate is this connectome actually is. The human error on such kind of tasks can be quite high and it is also true that a small error on a connectome, (for example, to mark a non-existent synapsis connecting a pair of neurons) can lead to a much different network.

In order to make the connectome more reliable it is necessary to replicate the annotation on several worms. It is also true that the connectome might be different between different stages of the worm, so several annotations should be done on several examples of each stage. The original annotation of the worm took about three years of manual labor. So if we intend to annotate several different worms it is not possible to keep working at this pace. That is exactly why there is interest in creating an automatic or at least semiautomatic annotation system that can drastically speed up this task.

Deep learning algorithms are a particular configuration of the old machine learning algorithm: artificial neuronal networks. This configuration is inspired in the human brain visual cortex, which causes a massive improvement of the deep learning algorithms with respect to any other machine learning algorithm in visual pattern recognition tasks. In the last ten years the area of machine learning have experienced an important development due to the appearance of deep learning techniques. These techniques have been especially powerful in tasks of pattern recognition on image data sets. Which is exactly what is needed to label the thousands of electron microscopy images of the *c. elegans*. With this in mind we are working on developing an algorithm capable of finding the neuron nuclei on the images. This nuclei has a very particular shape which we hope can be learned by the algorithm and then found in new data sets. We are using as base code the scripts of Geoffrey Hinton, a professor from the University of Toronto who is one of the precursors on this topic. These scripts are in Matlab.

3.2 Methods and Results

Electron microscopy imaging

The dataset shown here consists of over 6000 individual images that were stitched together to form 201 layers, each layer representing one 70 nm longitudinal section through a dauer larva head region. The worm was high pressure frozen and freeze substituted to obtain near-native tissue preservation [49]. Since many aspects of dauer neuroanatomy remain unknown, high-throughput data analysis of automatically identified neurons would be an invaluable tool for researchers. This data set was generated on the lab of Professor Christian Stiegloer at the University of Wuerzburg. Also with the help of experts from Stiegloer laboratory some example images were labeled. These images are used as training data set for the deep learning algorithm.

Deep Neural Networks

This algorithm resembles the classical neuronal networks with the difference that each layer is formed by a Restricted Boltzman Machine and pretrained independently with the objective of representing the input data. This results in a codification of the image (dimension reduction) which is then exploited in the classification task. The architecture of the network used in this project is presented in figure 3.1. The restricted Boltzman machines are three layer neuronal networks where there are no connections within each layer and the neurons between layers are fully connected.

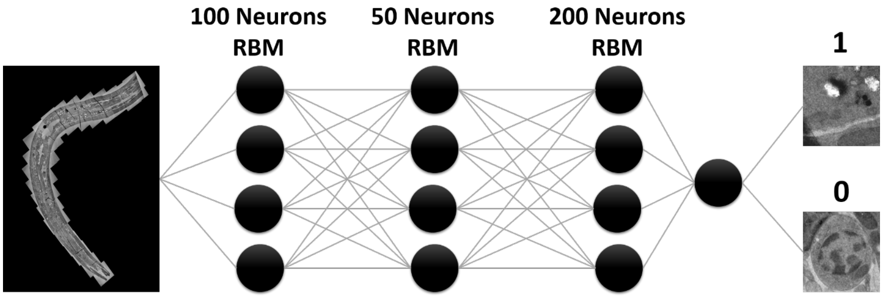


Figure 3.1: Deep Learning architecture used - A three stacked auto-encoder structure was used, based on the model used by Yann LeCun to solve the MNIST data set.

Since the invention of neuronal networks, it was proved that a neuronal network was capable of learning any function as long as an infinite number of neurons were allowed on each layer. This proof created big expectation with respect to the power of machine learning algorithms. Many even imagined that the invention of artificial intelligence was just a matter of a few years. Later, as neuronal networks were applied in several different fields, the big disappointment came. As much as neuronal networks were capable of solving some problems, they were far worst than people had imagined. They had a certain success in data analysis problems, for example in the stock market, or modeling of industrial processes. All the success cases have in common that a good mathematical description of the problem is possible. The problem occurs when that mathematical description is not good enough, as is the cases of visual patterns. For us humans, it is very easy to identify visual patterns, but it is extremely complex to express those patterns in mathematical terms which is the kind of information that can be fed into a neuronal network. Such mathematical descriptions of the things are known as features in machine learning. To overcome this difficulty in the image processing domain, the computer science community came up with the idea of new descriptors which were calculated directly on the images and dependent on the context surrounding each part of the image. A great example of such descriptors are the Histogram of Gradients, which was quite successful, being used in many applications and winning some visual recognition challenges. Later came the appearance of non-feature based learning. These type of algorithms give the input direct to the algorithm instead of giving a set of mathematical descriptors (features). This idea turned out to be a great solution to the visual pattern recognition problem.

The greatest representative of such algorithms is Convolutional Neural Networks. This algorithm takes as input the original images and passes those images through a set of adaptive filters and pooling layers in order to discover the set of features that characterize an image and allow to classify it. For a long time the reason why traditional Neural Networks failed on visual pattern recognition tasks remained unknown, until finally in 1991, thanks to the work of Josef Hochreiter under the supervision of Dr. Jürgen Schmidhuber, it was clear that the failure of Neural Networks was due to a problem called "vanishing gradients". Neural Networks use backpropagation as learning method. Backpropagation, as its name implies it, consists in calculating the difference between the predicted result and the expected result, and updating the values on the network according to such difference. The way the update is done is by means of gradient decent methods. The gradient of the cost function that represents the difference between expected and obtained result is calculated and used to update the weights on the network. The problem is that the gradients become smaller and smaller as the network get deeper until the updating is no longer of any use. This was nicely presented on the diploma thesis from Hochreiter together with a possible solution to the problem. The use of rectified linear units as activation function which will be explained later. However, better solutions to the problem appeared later.

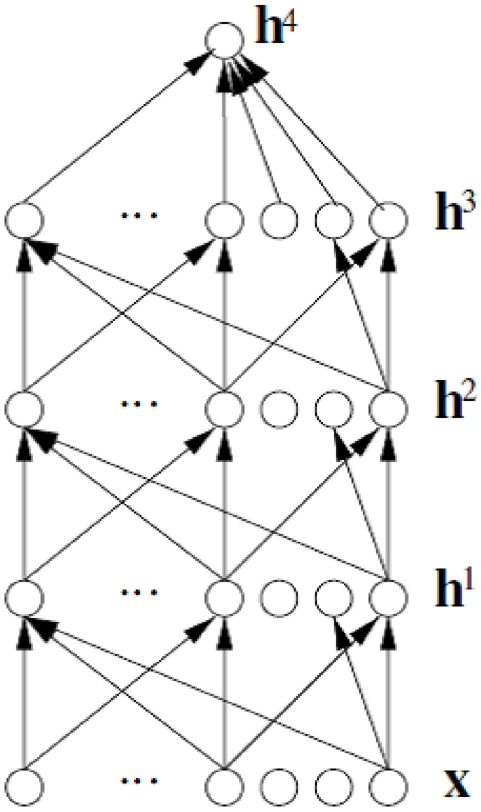
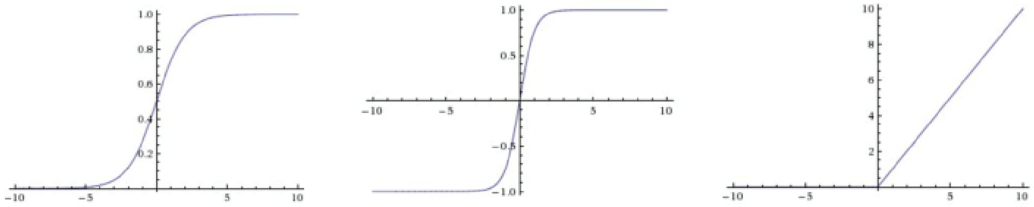


Figure 3.2: Structure of a multilayer neural network - The multilayer structure is presented and shows how the inputs are propagated along the layers

Multilayer Neural Networks

Multilayer neural networks can be mathematically presented in diverse forms. A common set of equations to present it is the one I use next. The layer K computes an output vector hk using the output of the previous layer hk-1, starting with the original input X=h0. The output on each layer is computed through an activation function. These functions are non-linear saturated integrators of the inputs that the neuron receives. There are several possible activation functions but three of the most common ones are the ones presented in figure 3.3.



Sigmoid tanh ReLU

Figure 3.3: Three common activation functions - The three activation functions presented are, sigmoid, hyperbolic tangential, and rectified linear unit. These are the most commonly used activation functions (Image taken from internet).

Let us supposed we use the tanh (hyperbolic tangential) function. Then the activation function would be:

$$h^k = tanh\left(b^k + W^k h^{k-1}\right) \qquad \text{(Eq. 3.1)}$$

where bk is a vector of offsets, one for each neuron on the layer. And Wk is a matrix of weights, one for each input on each neuron on the layer. The output layer, called hl, is used to compute a prediction and using as a reference the expected output a loss function is defined as L(hl,y). On a neural network used for classification it is necessary to have a different activation function on the output layer. Normally, the softmax function is used. This functions allows to differentiate a predicted class from another. The softmax activation function looks like this:

$$h_i^l = \frac{e^{b_i^l + W_i^l h^{l-1}}}{\sum_j e^{b_j^l + W_j^l h^{l-1}}} \qquad \text{(Eq. 3.2)}$$

where Wli is the ith row of Wl. The output hli is always positive and it is normalized such that addition over the whole set of outputs equals 1. If we suppose that the data is structured and that the input set X is representative of the class Y, then the output hl can be seen as an estimation of P(h=y/x). A common choice for the loss function is the log-likelihood function. Then we have:

$$\beta_w = \frac{2\gamma}{\omega v} \tag{Eq. 3.3}$$

The objective during the training stage is to minimize the expected value of this loss function. In order to accomplish that, we use gradient decent techniques and the backpropagation algorithm.

Backpropagation

The backpropagation algorithm was proposed in 1962 by Stuart Dreyfus and since then it has been used as the training procedure for neural networks. The idea behind the algorithm is very simple but the implementation and the mathematics required are not always that simple. The algorithm consists of three steps:

- Feed forward
- Feed back
- Weight update

The feed forward step is simply the computation of the activation function using the weighted inputs. It of course starts at the input layer of the neural network and moves towards the output layer. The feed back step consists in calculating the error of the estimation on the output layer and then propagating that error backwards towards the input layer. Neuronal networks are gradient descent learning algorithms. That means that the training process is done by means of gradient decent. So the feed back step of the algorithm propagates the derivative of the cost function with respect to the parameters of each neuron on the layer. Using the chain rule, the partial derivative of the loss function can be found at any neuron u of the network:

$$\frac{\partial L}{\partial u} = \sum_i \frac{\partial L}{\partial v_i}\frac{\partial v_i}{\partial u} \tag{Eq. 3.4}$$

Now, for the case of $tanh$ in the hidden layers we have that the activation function of the neuron i in the layer k is:

$$h_{ki} = tanh\left(b_{ki} + \sum_j W_{kij} h_{k-1,j}\right) \quad \text{(Eq. 3.5)}$$

As mentioned before, in the case of a probabilistic classifier, the output layer uses a softmax activation function:

$$p = h_L = softmax\left(b_L + W_L h_{L-1}\right) \quad \text{(Eq. 3.6)}$$

Again, as mentioned before, the loss function is defined as $L = -log(p_y)$, where y is the expected class. That means that training the classifier equals to maximizing $p_y = P(Y = y \mid x)$.

For clarity we will use $a_k = b_k + W_k h_{k-1}$ and we have to keep in mind two important derivatives.

$$\frac{\partial(-log(p_y))}{\partial a_{L,i}} = p_i - 1_{y=i} \quad \text{(Eq. 3.7)}$$

$$\frac{\partial\, tanh(u)}{\partial u} = \left(1 - tanh(u)^2\right) \quad \text{(Eq. 3.8)}$$

Now we apply backpropagation starting from the output node:

$$\frac{\partial L}{\partial L} = 1 \quad \text{(Eq. 3.9)}$$

and then, we have that the gradient for the output layer units with softmax activation function is:

$$\frac{\partial L}{\partial a_{L,i}} = \frac{\partial L}{\partial L}\frac{\partial L}{\partial a_{L,i}} = p_i - 1_{y=i} \quad \text{(Eq. 3.10)}$$

Now, for the rest of the layers we iterate using the following equations. The gradient with respect to the biases is:

$$\frac{\partial L}{\partial b_{k,i}} = \frac{\partial L}{\partial a_{k,i}} \frac{\partial a_{k,i}}{\partial b_{k,i}} = \frac{\partial L}{\partial a_{k,i}} \tag{Eq. 3.11}$$

and the gradient with respect to the weights is:

$$\frac{\partial L}{\partial W_{k,i,j}} = \frac{\partial L}{\partial a_{k,i}} \frac{\partial a_{k,i}}{\partial W_{k,i,j}} = \frac{\partial L}{\partial a_{k,i}} h_{k-1,j} \tag{Eq. 3.12}$$

and so, the gradient is propagated into the previous layer by:

$$\frac{\partial L}{\partial h_{k-1,j}} = \sum_i \frac{\partial L}{\partial a_{k,i}} \frac{\partial a_{k,i}}{\partial h_{k-1,j}} = \sum_i \frac{\partial L}{\partial a_{k,i}} W_{k,i,j} \tag{Eq. 3.13}$$

$$\frac{\partial L}{\partial a_{k-1,j}} = \frac{\partial L}{\partial h_{k-1,j}} \frac{\partial h_{k-1,j}}{\partial a_{k-1,j}} = \frac{\partial L}{\partial h_{k-1,j}} \left(1 - h_{k-1,j}^2\right) \tag{Eq. 3.14}$$

Following the previous equations, the gradients are calculated for every neuron in every layer and then using those results the weights are updated. This accounts for one step in the training process, which has to be repeated a good number of times using a representative training set. Such training set must be so big and with such quality that it represents as good as possible the complete space of the data.

Stacked auto-encoders

Convolutional Networks deal with the vanishing gradients problem mainly in two forms. The first one is by using a different activation function as mentioned before, for example, the rectified linear unit. But the most common and powerful method is to pre-train the network in layers and then do a fine tuning of the whole network. This technique of pre-training is commonly used and the sections of the network which are pre-trained can be of different types. Two very common types are the auto-encoders and the restricted boltzman machines. The architecture I used on this project was a Convolutional Network based on Auto-encoders. The autoencoders are a special type of neural networks with only a depth of two, one hidden layer and one output layer. An example is shown in figure 3.4.

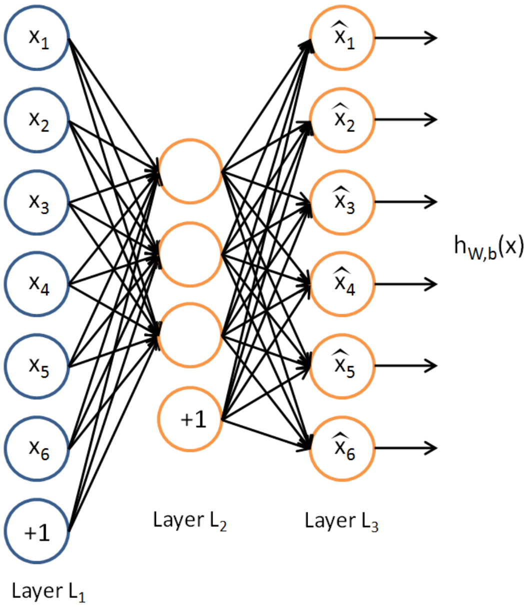


Figure 3.4: Auto-encoder structure - The auto-encoder is a three layer neural network with one hidden layer and one output layer (Graphic took from internet).

The auto-encoders are trained under a very simple principle. The network is simply supposed to learn the identity function. So the network is forced to simply represent the input as close as possible on the output layer. In order to achieve this the Kulback Leiber divergence is used to compare the input distribution to output distribution, once the difference is computed the backpropagation step is performed based on the gradient calculation of such difference. Since this is a shallow architecture, the vanishing gradients problem can be ignored and the auto-encoder is satisfactory trained.

Through experimentation it has been established that the layers of auto-encoding perform a filtering of the image where the relevant features of it are extracted. These layers of auto-encoders are mixed with layers of pooling. Pooling layers are layers where the image is simply decimated, sometimes by averaging pixel values, sometimes by simply taking the maximum value in the region. The pooling layers are necessary because thanks to these, the algorithm learns a representation of not only the local features of an image, but also of more general features as of hierarchy of characteristics, which is necessary to understand or classify the image. A much more detailed explanation of convolutional neural networks and other deep learning architectures can be found in [92]. A final stacked auto-encoder architecture looks like the one presented in figure 3.5.

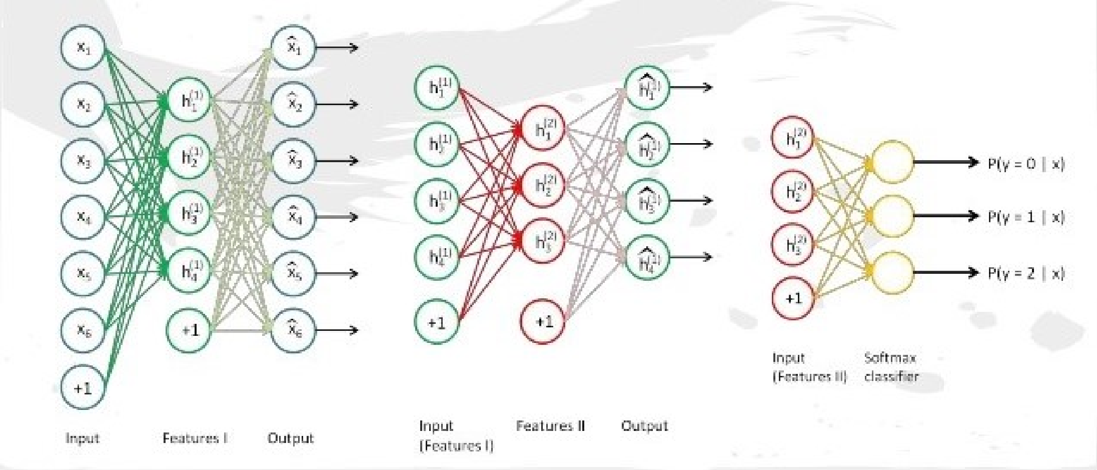


Figure 3.5: Stacked auto-encoder structure - The accumulation of several auto-encoders enables a coding of the input which extract the relevant features and allows the images to be classified (Graphic took from internet).

This architecture was one of the first approaches to deep learning. Today there are other proposals. We started this project using the stacked autoencoder architecture, from which we learned a lot and obtained satisfactory results that are presented later in this chapter. But it is also planned to try new deep learning algorithms which are more memory efficient. Our tests were made using a three layer stacked autoencoder. That means that each of those three layers is an autoencoder by itself. The autoencoder construction is a two layer neuronal network which is trained so that the output is simply a replica of the input data. So in the final architecture we had a 100, 50, 200 network. The first encoder had 100 hidden units; the second one 50; and the last one 200. At the end of the network there is a final layer with only two neurons which serve as a softmax classifier to distinguish among the two classes of interest. This network was trained for 10 thousand epochs. That means, each example in the training data set was seen 10 thousand times by the network.

Results

The algorithm I used in this project belongs to the supervised learning class. Supervised learning means that these algorithms need a set of training data. These are example objects whose belonging class is known. Also known as labeled data. The labeled data is presented to the classifier during the training stage, where the parameters of the algorithm are adjusted to learn the classification task. The deep learning algorithms are known for its capacity to take raw data and make predictions out of it. This is a very handy characteristic but that does not mean that sometimes some pre-processing of the data does not help. In order to explore that possibility I created two separate data sets. One of them is the absolute original data set. The second is a processed version of the images. The processing consists in, noise removal using a gaussian filter, and then a thresholding of the image to take advantage of the fact that the neuron nuclei is darker than its surrounding. See figure 3.6.

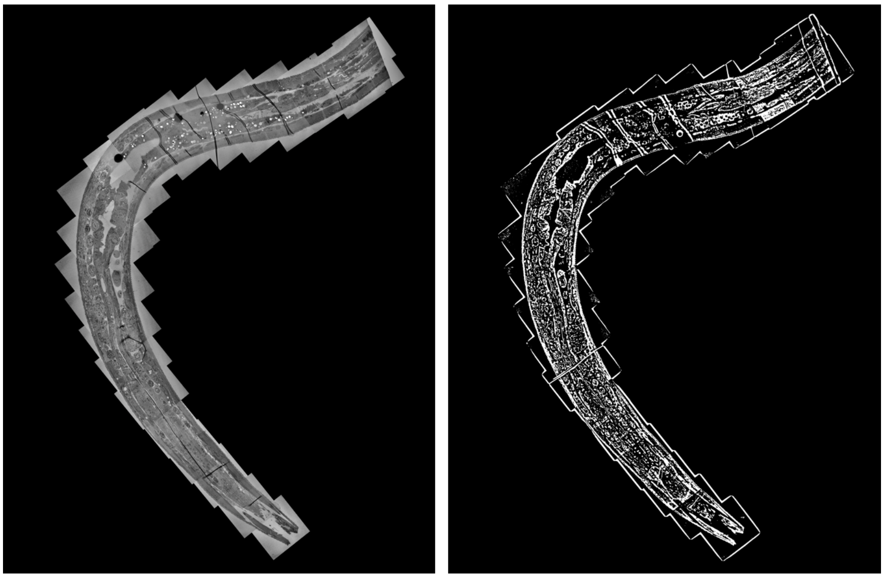


Figure 3.6: Image example on its original and its processed versions - The processed image is denoised using a gaussian filter and then thresholded to enhance the neuron nuclei.

In the figure 3.7, a closer view is presented of an original image and its preprocessed version. In each of the images, two examples of the nuclei that we want to find are marked with the yellow shadows.

Due to the form on which the algorithm operates, the image cannot be used completely. Instead, it has to be split in patches. This is due to the fact that we are using and algorithm design for classification and not for segmentation. Taking into account the maximun size of the nuclei under the resolution of the images, patches of 101 pixels were defined. On the training stage the patches are manually selected, creating two

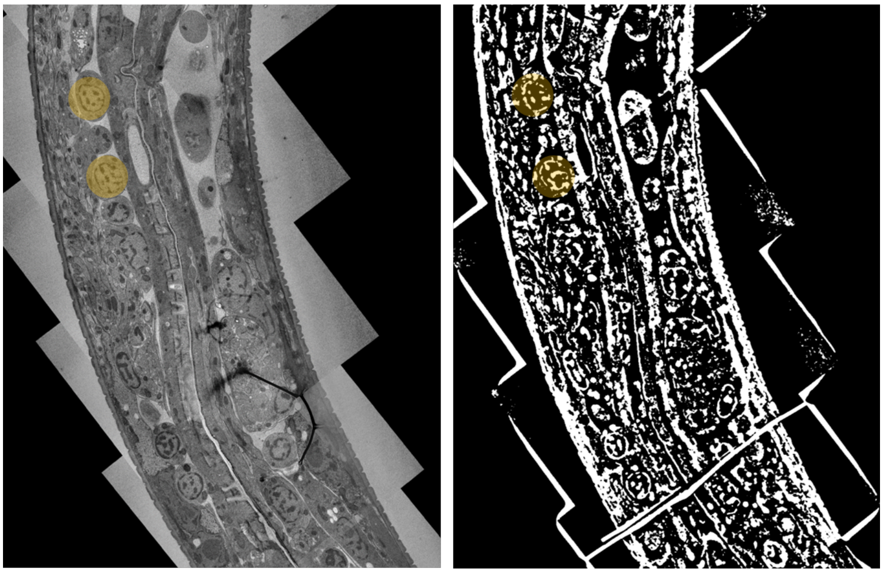


Figure 3.7: Close view to an image example on its original and its processed versions - Two examples of neuron nuclei are marked with yellow circles. The nuclei has a very characteristic shape that is however extremely difficult to describe.

separate groups. The first group is formed by patches that contain a neuron nuclei, and the second group are patches presenting a variety of parts of the image where no neuron nuclei is depicted. The first part should be conformed by a set of patches such that they represent as good as possible all the forms and sizes the nuclei might have. And the second set, should represent all other parts of the image that do not correspond to neuron nuclei. Every part in the image which is not a neuron nuclei is regarded as background.

The patches obtained were transformed in order to enrich the training data set. Making it more representative of the underlying structure of the data. The patches were rotated and flipped. In the end, for the training process of the original images, there was a total of 10872 neuron nuclei patches and 7204 background patches. And for the testing set, there was a total of 14786 nuclei patches and 11258 background patches. In the case of the filtered images the total amounts of the training set were, 20868 nuclei patches and 12180 background patches. And the total amounts of the testing set were of 19318 nuclei patches and 11542 background patches. The learning curves obtained on the original and filtered data sets are presented in figures 3.10 and 3.11. The X axis presents the number of epocs as the training process advances, and the Y axis is simply

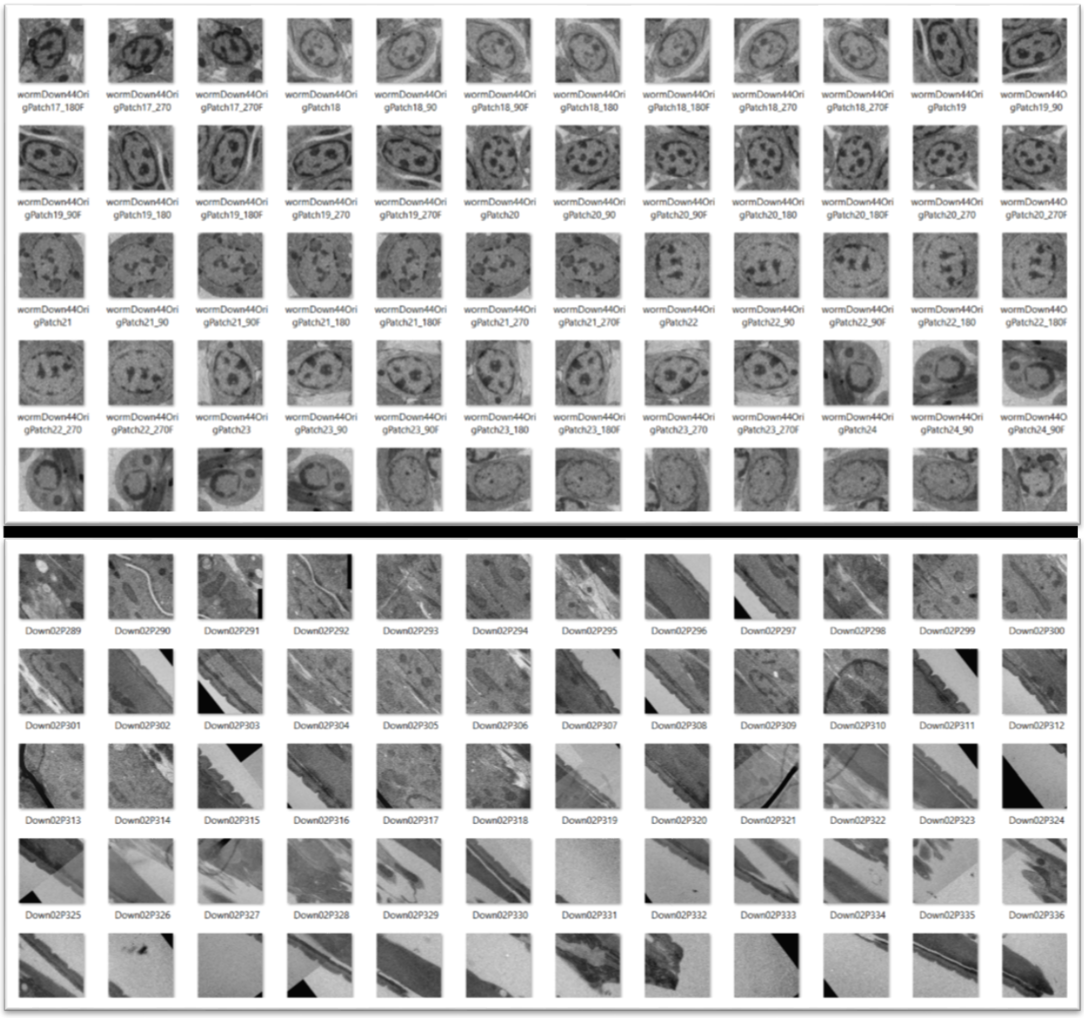

wormDown44Ori gPatch17_180F
wormDown44Ori gPatch17_270
wormDown44Ori gPatch17_270F
wormDown44Ori gPatch18
wormDown44Ori gPatch18_90
wormDown44Ori gPatch18_90F
wormDown44Ori gPatch18_180
wormDown44Ori gPatch18_180F
wormDown44Ori gPatch18_270
wormDown44Ori gPatch18_270F
wormDown44Ori gPatch19
wormDown44O gPatch19_90
wormDown44Ori gPatch19_90F
wormDown44Ori gPatch19_180
wormDown44Ori gPatch19_180F
wormDown44Ori gPatch19_270
wormDown44Ori gPatch19_270F
wormDown44Ori gPatch20
wormDown44Ori gPatch20_90
wormDown44Ori gPatch20_90F
wormDown44Ori gPatch20_180
wormDown44Ori gPatch20_180F
wormDown44Ori gPatch20_270
wormDown44O gPatch20_270F
wormDown44Ori gPatch21
wormDown44Ori gPatch21_90
wormDown44Ori gPatch21_90F
wormDown44Ori gPatch21_180
wormDown44Ori gPatch21_180F
wormDown44Ori gPatch21_270
wormDown44Ori gPatch21_270F
wormDown44Ori gPatch22
wormDown44Ori gPatch22_90
wormDown44Ori gPatch22_90F
wormDown44Ori gPatch22_180
wormDown44O gPatch22_180F
wormDown44Ori gPatch22_270
wormDown44Ori gPatch22_270F
wormDown44Ori gPatch23
wormDown44Ori gPatch23_90
wormDown44Ori gPatch23_90F
wormDown44Ori gPatch23_180
wormDown44Ori gPatch23_180F
wormDown44Ori gPatch23_270
wormDown44Ori gPatch23_270F
wormDown44Ori gPatch24
wormDown44Ori gPatch24_90
wormDown44O gPatch24_90F
Down02P289
Down02P290
Down02P291
Down02P292
Down02P293
Down02P294
Down02P295
Down02P296
Down02P297
Down02P298
Down02P299
Down02P300
Down02P301
Down02P302
Down02P303
Down02P304
Down02P305
Down02P306
Down02P307
Down02P308
Down02P309
Down02P310
Down02P311
Down02P312
Down02P313
Down02P314
Down02P315
Down02P316
Down02P317
Down02P318
Down02P319
Down02P320
Down02P321
Down02P322
Down02P323
Down02P324
Down02P325
Down02P326
Down02P327
Down02P328
Down02P329
Down02P330
Down02P331
Down02P332
Down02P333
Down02P334
Down02P335
Down02P336

Figure 3.8: Some patch examples of the original images - The upper part of the figure contains patches of neuron nuclei and the lower part contains background patches.

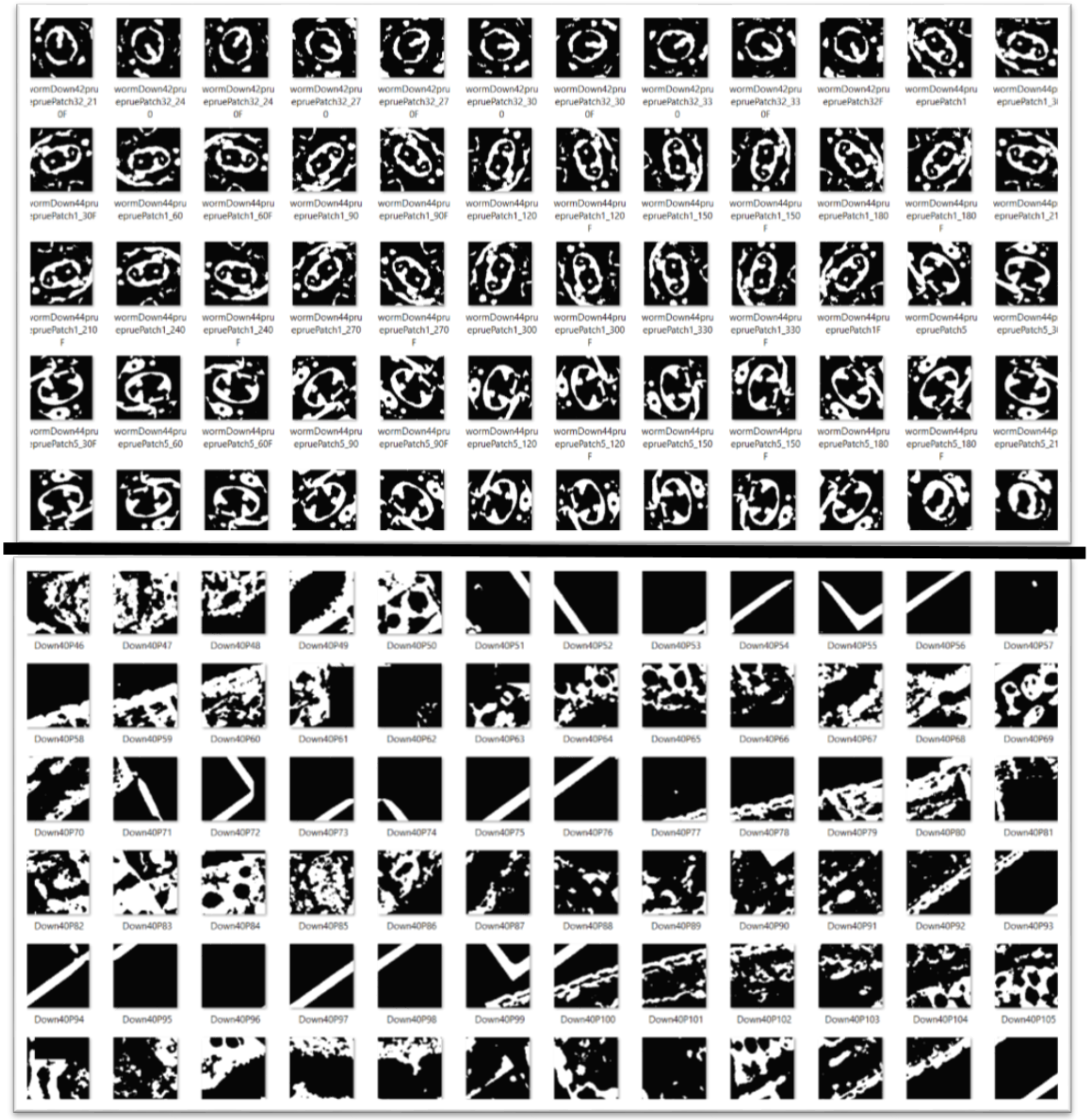

vormDown42pru :pruePatch32_21 0F
wormDown42pru epruePatch32_24 0
wormDown42pru epruePatch32_24 0F
wormDown42pru epruePatch32_27 0
wormDown42pru epruePatch32_27 0F
wormDown42pru epruePatch32_30 0
wormDown42pru epruePatch32_30 0F
wormDown42pru epruePatch32_33 0
wormDown42pru epruePatch32_33 0F
wormDown42pru epruePatch32F
wormDown44pru epruePatch1
wormDown44p epruePatch1_3
vormDown44pru :pruePatch1_30F
wormDown44pru epruePatch1_60
wormDown44pru epruePatch1_60F
wormDown44pru epruePatch1_90
wormDown44pru epruePatch1_90F
wormDown44pru epruePatch1_120
wormDown44pru epruePatch1_120 F
wormDown44pru epruePatch1_150
wormDown44pru epruePatch1_150 F
wormDown44pru epruePatch1_180
wormDown44pru epruePatch1_180 F
wormDown44p epruePatch1_21
vormDown44pru :pruePatch1_210 F
wormDown44pru epruePatch1_240
wormDown44pru epruePatch1_240 F
wormDown44pru epruePatch1_270
wormDown44pru epruePatch1_270 F
wormDown44pru epruePatch1_300
wormDown44pru epruePatch1_300 F
wormDown44pru epruePatch1_330
wormDown44pru epruePatch1_330 F
wormDown44pru epruePatch1F
wormDown44pru epruePatch5
wormDown44p epruePatch5_3
vormDown44pru :pruePatch5_30F
wormDown44pru epruePatch5_60
wormDown44pru epruePatch5_60F
wormDown44pru epruePatch5_90
wormDown44pru epruePatch5_90F
wormDown44pru epruePatch5_120
wormDown44pru epruePatch5_120 F
wormDown44pru epruePatch5_150
wormDown44pru epruePatch5_150 F
wormDown44pru epruePatch5_180
wormDown44pru epruePatch5_180 F
wormDown44p epruePatch5_21
Down40P46
Down40P47
Down40P48
Down40P49
Down40P50
Down40P51
Down40P52
Down40P53
Down40P54
Down40P55
Down40P56
Down40P57
Down40P58
Down40P59
Down40P60
Down40P61
Down40P62
Down40P63
Down40P64
Down40P65
Down40P66
Down40P67
Down40P68
Down40P69
Down40P70
Down40P71
Down40P72
Down40P73
Down40P74
Down40P75
Down40P76
Down40P77
Down40P78
Down40P79
Down40P80
Down40P81
Down40P82
Down40P83
Down40P84
Down40P85
Down40P86
Down40P87
Down40P88
Down40P89
Down40P90
Down40P91
Down40P92
Down40P93
Down40P94
Down40P95
Down40P96
Down40P97
Down40P98
Down40P99
Down40P100
Down40P101
Down40P102
Down40P103
Down40P104
Down40P105

Figure 3.9: Some patch examples of the filtered images - On the upper part of the figure are shown neuron nuclei patches and on the lower part background patches.

the error percentage.

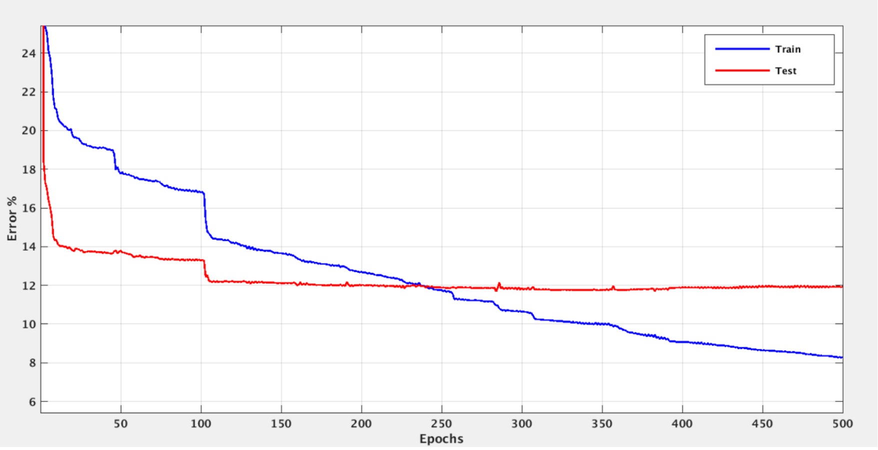


Figure 3.10: Learning curves on the original data set - The red line presents the testing learning curve and the blue line presents the training learning curve. Both nicely decay as the training process advances.

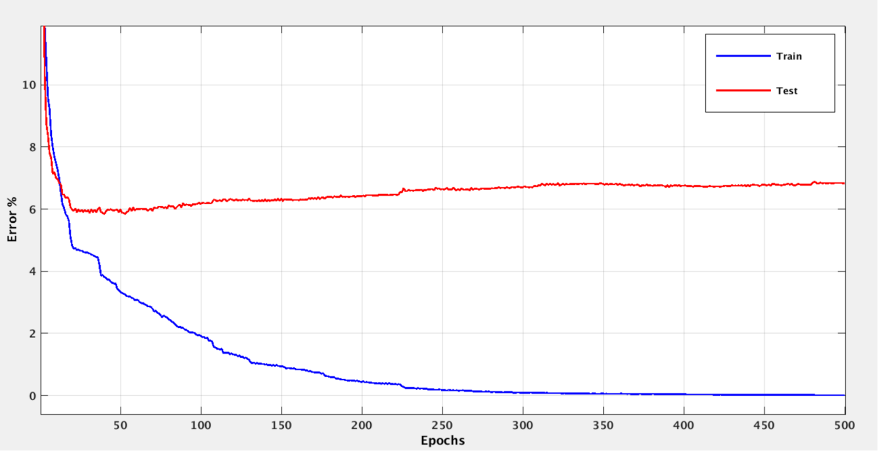


Figure 3.11: Learning curves on the filtered data set - The red line presents the testing learning curve and the blue line presents the training learning curve. The final error achieved is lower as in the original data set.

In order to do the final segmentation of the neuron nuclei on new images which were not used for training and which are not split in patches it is necessary to implement a sliding window procedure along the complete original image. The sliding window moves along the complete image and on each position of the window the algorithm does a prediction whether there a neuron nuclei is depicted or not.

3.3 Discussion

Deep learning algorithms are extremely capable in tasks of pattern recognition. This can be explained due to the structure of the algorithm, resembling the human visual cortex. These algorithms have been used in many applications. In some cases they even reach above-human performance. As I got to know about the existence of this algorithm I knew it could be of most use in biology were results are quite often visually qualified and images are considered final results. As mentioned before, imaging is a growing field in biology and the amount of information produced every day in a laboratory is huge. Luckily a new tendency in machine learning algorithms is also growing. These algorithms are capable of understanding images better than any other algorithm before.

The spectrum of images which are produced in biological research is enormous but I would dare to say that for most of them deep learning algorithms are a great analysis tool and that for all of them a computational analysis is (or at least should be) required. There are two main reasons why a computational analysis is recommended for biological images. In some cases, it is necessary due to the huge amount of data that is produced, so huge that it overwhelms human capacity. The second reason is the reproducibility of the results. If images are computationally analyzed, this same analysis can be applied to other results and thus have an objective comparison. If, instead of a computational analysis a human analysis is used, the results are much less objective and the claims based on the obtain images loose power.

In the problem stated here, we see both scenarios. The amount of images is too big to be processed by a person and the classification of pixels as neurons or background requires of an objective rule. The old connectome of the *c. elegans* is known to contain several mistakes, like synaptic junctions which were marked but are not really present. Deep learning algorithms are without any doubt a great choice to solve this problem, but solving the problem itself is much more complicated than simply choosing the right algorithm.

The results obtained in this stage of the project, although satisfactory and promising, are not good enough to consider the problem solved. In order to have a real automatic or assisted segmentation it is necessary to severely improve the false positive rate of

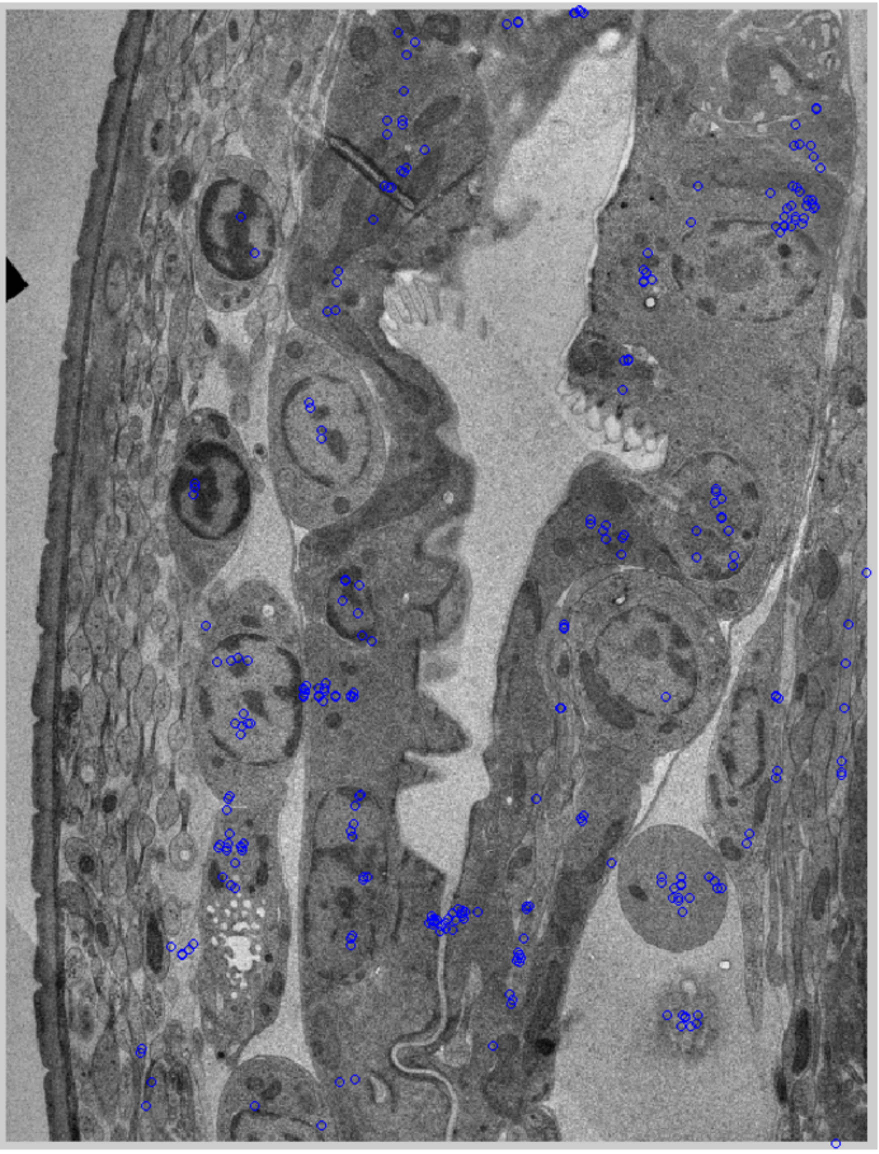


Figure 3.12: Result of an image tested for segmentation - The final process of segmentation was applied to an image to evaluate the algorithm. The blue points mark the place where the algorithm thinks there might be a neuron.

the algorithm. Even though the error curves look nice (reaching levels as low as 6 percent), when the real application of the algorithm is done and a complete image is scanned in search of neuron nuclei, the error percentage of 6 percent traduces into a huge amount of false positives. This is due to the fact that passing a sliding window of 101 pixels wide throughout an image of around 3000 pixels generates about 150000 patches to classify, so the expected number of patches in error are about 8000. To manually correct this error results too expensive, making the algorithm nonviable.

It is also a big concern that the error is constituted more by false positives than false negatives. That means that the algorithm mostly correctly identifies the neuron nuclei as such but it also, more often as we would like, attributes a neuron nuclei to parts of the image where there is no neuron. It is nevertheless very important to remark that the algorithm is capable of understanding the complex pattern of the neuron nuclei. Without including any feature the algorithm is capable of learning the most important visual features of the neuron nuclei and the background in order to classify them correctly.

We used the current results to apply for a post-doctoral grant that allow us to continue developing this project. Since it was granted to us, we have a plan for a second phase of the project. In this second phase we will move from Matlab to python due to the extra flexibility and better memory performance of this environment. For this second phase of the project we also count with a better data set. The new data set has a better contrast and a higher resolution which we believe will improve the performance of the algorithm. We also want to move from an image wise classification scheme to a pixel wise classification. This means that to each pixel a probability of belonging to a neuron nuclei is assigned. So in the end we have a probability map of the same size of the image where each pixel is a number between 0 and 1 indicating how probable it is that the correspondent pixel is part of a neuron nuclei. To obtain this probability maps other algorithms have been proposed. We will try those algorithms, in particular one called Elektronn. We are aware that the implementation of this type of algorithms is never "plug and play" but it will be probably a good starting point.

One more aspect that we need to consider in this second phase of the project is a post processing stage. It is common that results from deep learning algorithms need a little tuning and this is probably also the case for us. In the end we hope that we will get a nicer curve, not only ending on a lower error percentage but also with a smother monotonic behavior.

4

Biological systems simulations on inflammation

4.1 Introduction

The term model is not an easy one to define but I will try to give the reader an idea of what I mean by it. A model is a representation in an abstract and idealized way. We need models to represent systems of higher complexity. There are several types of models, such as analogical, scale, and mathematical. Our interest focuses on mathematical models because they are much more general than the other type of models. Because of that, its use is much more spread, at least in natural sciences. Models are driven both by theory and phenomenology. Models serve as a bridge between theory and experimental data. They are a melting point were the current accepted theories about a system and the experimental data available for such system congregate. We create models because of the impossibility to test our hypothesis in the real system. This impossibility can be absolute, for example while studying the big bang; or it can be circumstantial. For example, we do not test certain hypothesis in humans. In any case, it is true that without modeling science cannot advance.

The discussion whether models constitute or not a real advance in science is an empty discussion because models are absolutely necessary. A mathematical model is not necessarily a computational model, but given the technological advances from the last 60 years, computational models have overtaken almost the complete spectrum of mathematical models in life science. A computational model is established based on what is known about the biological system (theory), on quantitative data obtained from experimental essays, and on some plausible assumptions. There is not a unique way to define a model. Computational biology models are inherited from other disciplines like physics and engineering. For example, a widely used model nowadays in biology and which was inherited from engineering is the ODE (Ordinary Differential Equations). This is a very useful model and has been successfully applied to several biological systems [50]. Despite being such a useful technique, the ODE's have also an important disadvantage. When modeling a biological system using ODE's, it is necessary to list all the possible configurations of the molecules involved and to define explicitly the equations that model each of those states. Given the size of a common biological signaling network and the dense populated space of conformations of each of the molecules involved, this is a mayor problem since it is too difficult to cover the full range of possibilities defining each of the necessary equations.

Having this obstacle in mind, the computational biology community developed an alternative form of biological system modeling. It is called *ruled based modeling*, and as its name indicates it works by defining the set of rules that govern the interactions of the molecules in the biological system. Rule-based models are often solved with the use of ordinary differential equations but this does not mean that they are the same. Since the definition of the model itself is different, the rule-based models are essentially different from ODE's but they make use of the same mathematical tools in order to compute the model behavior. The rule-based model approach is particularly designed to represent systems where structured objects interact via component parts in a modular way [50]. Such kind of systems are precisely a cell signaling system. Within the rule-based models there are several standards. In this project we decided to use BNGL [51], a well known tool in the computational biology community. In order to illustrate the use of models in biology, I present an example system: the TNF ligand and its two binding partners, the receptor 1 and receptor 2. The chapter focuses on the modeling considerations more than on the biological details of the model, but of course the relevance of the model can only be appreciated under the light of the biological system. The relevant biological information is mostly obtained from a manuscript in preparation.

4.1.1 TNF system

Citokines are small proteins which are often involved in cell signaling processes. The TNF (Tumor Necrosis Factor) superfamily is a citokine and it can be observed in the cells in two forms, one is as weakly bounded trimer transmembranic protein and the second is a soluble trimeric molecule [52]. The TNFSF (Tumor Necrosis Factor SuperFamily) is a group of ligands which activate a group of receptors known as the TNFRSF (Tumor Necrosis Factor Receptor SuperFamily). The two most important receptors of the TNFRSP group are the TNF receptor 1 (TNFR1) and the TNF receptor 2 (TNFR2). TNFR1 and TNFR2 binding site to the ligand are the grooves formed between the protomers of the TNF trimer [53]. In the work from Mukai et al. [54] it was shown that the TNF-TNFR2 complex is formed by a single trimeric TNF ligand molecule which binds to three TNFR2 molecules. Similar results were found for TNFR1 but with respect to another ligand from the TNFSP [55]. Given this evidence, a model was proposed concerning ligand-mediated trimerization of the TNFRSP. Nevertheless, current research has been producing an increasing amount of evidence that higher order clustering of TNF-TNFR complexes is necessary to efficiently pass on the cell signaling cascade. There is evidence that the oligomerization state of ligands and receptors is a determinant factor for the signaling process [53].

A particular factor of the TNFRSF is a cysteine rich domain, from which there can be one to six copies in the extracellular part of the receptors [56]. This cysteine rich domain is part of a homophilic protein-protein interaction domain known as the Pre-Ligand Assembly Domain (PLAD) [57, 58]. The interactions in this domain, named PLAD-PLAD interactions mediate the formation of receptor homogeneous polymers, but also some inter-class receptor polymerization can occur [59].

Inflammation is a most important process in the biological response to harmful stimulus. In cases of pathogens, infections, damaged cells or irritants, the inflammation process is triggered. This process involves quite a number of players. Among them are inmune cells, blood vessels, and molecular mediators. The TNF ligand has been identified as a key protein in the signaling cascade of the inflammation process. Inflammation is vital to stop infections or to initiate tissue regeneration, that means that a too low level of inflammation can result harmful. But chronic inflammation (the excess of inflammation) is also very dangerous. It can cause a number of diseases among which the most relevant might be cancer. It is also known that inflammation in the brain is related to highly problematic diseases like Alzheimer's, Parkinson's, and depression. This delicate balance of the inflammation process obliges the organism to keep a strict regularization of it, which is why it turns out to be so relevant to study the TNF signaling cascade.

The stoichiometry of PLAD-mediated receptor oligomerization is still a matter of discussion and little is known about the affinity and kinetics of PLAD-PLAD interactions (see figure 4.1). After studying the existing models of the TNF signaling cascade [60, 61] we constructed a model which focuses on three important aspects of the TNF dynamics. First, it represents the dimerization of the receptors 1 and 2 in absence of the ligand. Second, the binding of the soluble TNF molecule to the receptor dimmers; and third, the secondary clustering of the signaling-incompetent TNF-TNFR1/2 complexes formed to become signaling competent ligand-receptor clusters.

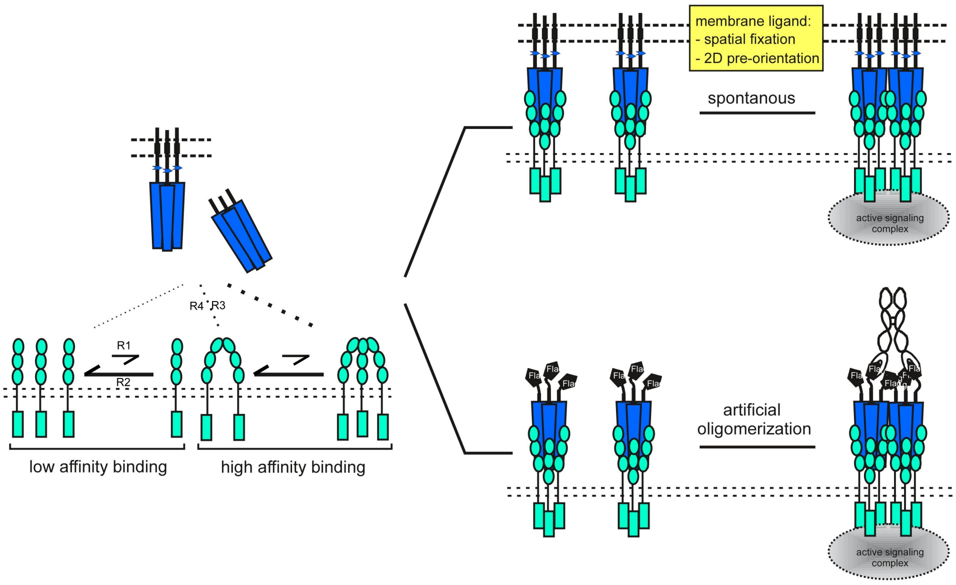


Figure 4.1: Schematic TNFR1 pre-assembly and cluster formation after ligand binding - Membraneous TNFR1 monomers are capable of self-aggregation through their PLAD domain, driven by the kinetic parameters konPLAD and koffPLAD. Upon ligand (TNF)-binding, this forms a multimeric recepter-ligand network, where one TNF molecule is surrounded by three receptor molecules.

4.2 Methods and Results

We had the fortune to collaborate in this project with Professor Harald Wajant. He is an expert in TNF and has been working in his lab for a long time performing experiments to elucidate the dynamics of the signaling cascade initialized by the TNF ligand. Using the results of his research and his expertise to do coherent assumptions we were able to create a simple, realistic, and informative model.

4.2.1 TNF wetlab experiments

The Tumor Necrosis Factor is a powerful cytokine which is a main player on the systemic inflammation pathway in human cells. Its important role in this pathway makes it a common point of study for diseases such as Alzheimer, cancer, depression, psoriasis, and inflammatory bowel disease. Because of this the TNF protein and the receptors it interacts with have been thoroughly studied and a considerable amount of information is known about them. On studies like [57, 58] it has been shown that both TNFR1 and TNFR2 form homogenic multimers. This means that the receptors in absence of the ligand are capable of establishing bounds among them, forming molecule multimers. A further study [62] showed that the TNFR1 and TNFR2 affinity for the ligand depends on the formation of the receptor dimmers, much more strongly in the case of receptor 2. This was shown by mutating the binding sites of the TNF ligand. On the TNF ligand trimmer molecule there are three binding sites for the receptors. They obstructed first only one of those sites and the affinity for both receptors remain quite similar. Then they obstructed two of the binding sites and the affinity of the receptor 1 remained still similar, but the affinity of receptor 2 was strongly diminished. This shows that the TNFR1 interacts with the ligand just as actively in the dimmer or the monomer state but as for the TNFR2, the monomer conformation has a much weaker affinity for the ligand compared to the dimmer state.

Coherently, in the experiments conducted in the lab of Professor Harald Wajant here at the University of Wuerzburg, it was observed that the extracellular domain of the monomer conformation of TNFR1 is more prone to bind the ligand as the same conformation of the TNFR2. Also, it was shown that after exciting the formation of dimmers of the TNFR2 the affinity with the TNF ligand is significantly increased. This can be observed in figure 4.2.

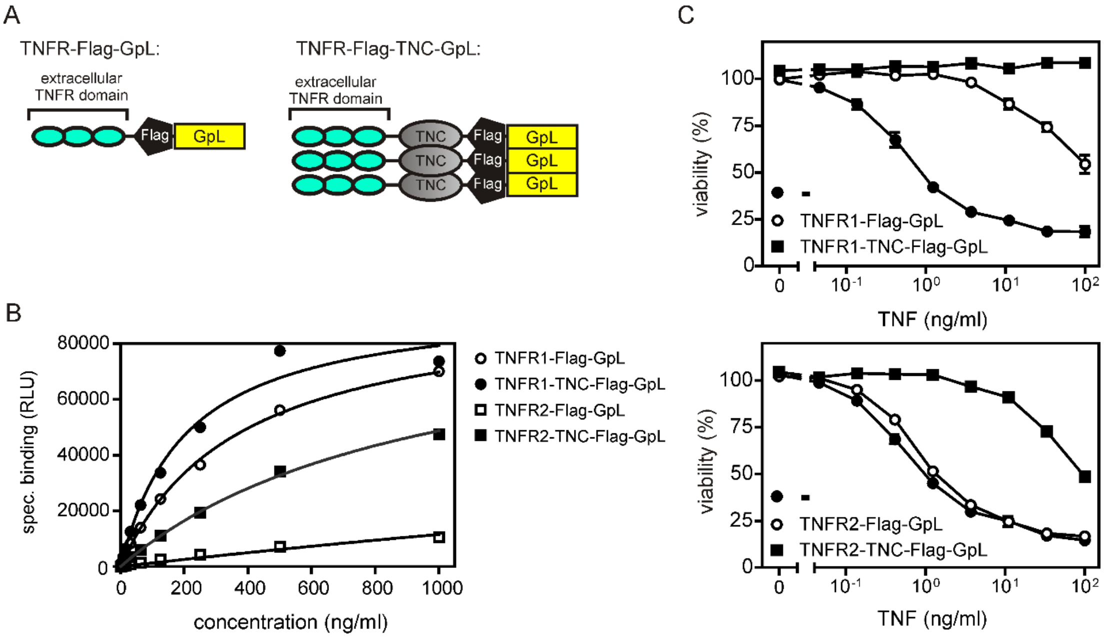


Figure 4.2: Experimental observations of TNFR1 and TNFR2 affinities for TNF - (A) Domain architecture of soluble fusion proteins of the TNFR1 and TNFR2 ectodomains with the luciferase from Gaussia princeps. TNFR = ectodomain of TNFR1 or TNFR2, GpL = Gaussia princeps luciferase, TNC = Tenascin-C trimerization domain. (B) CHO cells and CHO transfectants expressing membrane TNF were incubated at 37°C for two hours with the indicated concentrations of the various TNFR-GpL fusion proteins. After extensive washing cell associated GpL activity was measured to determine total (CHO-memTNF) and non-specific binding (CHO). Shown are the specific binding values obtained by subtraction of unspecific binding values from the corresponding total binding values. Specific binding data was analyzed by non-linear regression with the constriction that all data sets share the same number of binding sites (C) TNF (10 ng/ml) was preincubated with the indicated concentrations of the various GpL fusion proteins and was then used to trigger cell death in CHX sensitized L929.

4.2.2 The model

In order to present the model it is necessary first to introduce the nomenclature I used and some concepts from stoichiometry.

Molecules involved

$R1_m$	Receptor 1 monomer
$R1_d$	Receptor 1 dimmer
$R1_t$	Receptor 1 trimmer
T	TNF ligand
$R2_m$	Receptor 2 monomer
$R2_d$	Receptor 2 dimmer
$R2_t$	Receptor 2 trimmer

Some definitions

This is probably too basic for somebody with a minor knowledge of chemistry, but since this project is intended to be read by people of all sorts of specialties, I will even present definitions of equilibrium and diffusion constants. Nevertheless, I limited myself to present just the definitions. For the reader that wishes to know where these definitions come from, I invite him to read as (I did at the beginning of this project) the text from Olson [63] or any other academic text.

The equilibrium rate is defined as the point of stability were the ratio of the reactants and the resultants is not changing anymore. Suppose we have reactants A and B, then the equilibrium binding constant (KB) is defined as shown in equation Eq. 4.1. Observe that the units of the reactants are moles $\{M\}$. The equilibrium diffusion constant (D) is also defined in equation Eq. 4.1.

$$\begin{aligned} KB &= \frac{[AB]}{[A][B]} \qquad \{M\} = \frac{\{M\}\{M\}}{\{M\}} \\ KD &= \frac{[A][B]}{[AB]} \qquad \{M^{-1}\} = \frac{\{M\}}{\{M\}\{M\}} \end{aligned} \tag{Eq. 4.1}$$

using the definition of the diffusion constant it is now possible to define the association and dissociation rates, which are actually the ones that must be included in the model. Equation Eq. 4.2 presents the definition.

$$KD\{M\} = \frac{K_{off}\left\{\frac{1}{s}\right\}}{K_{on}\left\{\frac{1}{Ms}\right\}} \tag{Eq. 4.2}$$

Due to the fact that rule-based modeling is compartmental, an important parameter to be defined in the creation of a model is the volume. In our case we have a cell diameter (d) of $10\mu m$. And we also know that the length of the extracellular domain of the TNF receptors is about $0.015\mu m$ which we call r_R. The volume of the shell where the binding occurs is defined in equation Eq. 4.3. Note that the factor of 1000 is included in order to express the volume in liters and not in cubic centimeters.

$$V = \frac{4}{3}\pi(r^3 - r_R^3)/1000 \qquad \text{(Eq. 4.3)}$$

Constant definitions

We define the constants and make clear to which reaction they correspond.

Receptor 1

Diffusion constant of TNFR1 PLAD-PLAD interaction and its rate constants

$$KD_{pR1_m} = \frac{[R1_m][R1_m]}{[R1_d]} = \frac{k_{off\_pR1_m}}{k_{on\_pR1_m}}$$

Diffusion constant of TNF trimmer interaction with the TNFR1 monomer

$$KD_{TR1_m} = \frac{[R1_m][T]}{[TR1_m]} = \frac{k_{off\_TR1_m}}{k_{on\_TR1_m}}$$

Diffusion constant of TNF trimmer interaction with the TNFR1 dimmer

$$KD_{TR1_d} = \frac{[R1_d][T]}{[TR1_d]} = \frac{k_{off\_TR1_d}}{k_{on\_TR1_d}}$$

Receptor 2

Diffusion constant of TNFR2 PLAD-PLAD interaction and its rate constants

$$KD_{pR2_m} = \frac{[R2_m][R2_m]}{[R2_d]} = \frac{k_{off\_pR2_m}}{k_{on\_pR2_m}}$$

Diffusion constant of TNF trimmer interaction with the TNFR2 dimmer

$$KD_{TR2_d} = \frac{[R2_d][T]}{[TR2_d]} = \frac{k_{off\_TR2_d}}{k_{on\_TR2_d}}$$

As it can be seen in the constants table above, there are three constants defined for receptor 1 and only two for receptor 2. This is because receptor 1 have an extra mechanism of interaction with the ligand that receptor 2 does not posses, as it was explained in the TNF wetlab experiments section. Simply, receptor 2 monomer does not have enough affinity for the ligand, so this form of interaction was not included in the model. This will be further explained in later as I present the reactions that were modeled. The set of reactions modeled for receptor 1 are presented in the figure 4.3. Similarly, the set of reactions for the receptor 2 are shown in figure 4.4.

$$3[R1_m] + [T] \overset{KD_{pR1_m}}{\leftrightharpoons} [R1_d] + [T] + [R1_m] \overset{KD_{TR1_d}}{\leftrightharpoons} [R1_dT] + [R1_m] \overset{KD_{TR1_m}}{\leftrightharpoons} [R1_tT]$$

$$3[R1_m] + [T] \overset{KD_{TR1_m}}{\leftrightharpoons} 2[R1_m] + [R1_mT] \overset{KD_{TR1_m}}{\leftrightharpoons} [R1_dT] + [R1_m] \overset{KD_{TR1_m}}{\leftrightharpoons} [R1_tT]$$

$$3[R1_m] + [T] \overset{KD_{TR1_m}}{\leftrightharpoons} 2[R1_m] + [R1_mT] \overset{KD_{pR1_m}}{\leftrightharpoons} [R1_d] + [R1_mT] \overset{KD_{TR1_d}}{\leftrightharpoons} [R1_tT]$$

$$3[R1_tT] \overset{KD_{pR1_m}}{\leftrightharpoons} [R1_tT]_2 + [R1_tT] \overset{KD_{pR1_m}}{\leftrightharpoons} [R1_tT]_3$$

Figure 4.3: Set of reactions modeled for the TNFR1- Four reactions were modeled for the case of the receptor 1. The first reaction corresponds to the case where the receptor first interacts with other receptors to form dimmers and only then binds to the ligand. The second and third reactions are similar. In these two, receptor 1 interacts with the ligand before forming dimmers. Later it can happen that the ligand keeps recruiting receptor 1 monomers (as is the case in the second reaction), or that the ligand recruits a preformed dimmer (as in the third reaction). All these three reactions have the same resultant; the ligand-receptor 1 complex. The last reaction presented is the clusterization of the complex formed on the previous reactions.

We implemented all the before presented reactions in Rule Bender [64]. The contact map obtained is presented in figure 4.5. It can be seen that the set of edges between the receptors and the ligand molecule are identical. The difference is that the receptor 1 counts with two extra routes to form the complex, but these extra routes cannot be observed on the contact map since they correspond to the same bounds.

In order to simulate the model results, some parameters needed to be fixed. The set of values used for the parameters are presented in table 4.1. These values were used on all simulations unless it is otherwise stated in the result plot. These are only the values of the constants. Beside these, there were other values that needed to be set. The number of receptor 1 molecules was always 1000, the number of receptor 2 molecules was always 10000. The concentration of the TNF ligand was always 1.4×10^{-10}, and the volume of the shell where the reaction occurs was always as presented above.

$$3[R2_m] + [T] \overset{KD_{pR2_m}}{\leftrightharpoons} [R2_d] + [T] + [R2_m] \overset{KD_{TR2_d}}{\leftrightharpoons} [R2_dT] + [R2_m] \overset{KD_{TR2_d}}{\leftrightharpoons} [R2_tT]$$

$$3[R2_tT] \overset{KD_{pR2_m}}{\leftrightharpoons} [R2_tT]_2 + [R2_tT] \overset{KD_{pR2_m}}{\leftrightharpoons} [R2_tT]_3$$

Figure 4.4: Set of reactions modeled for the TNFR2- In the case of receptor 2, only two reactions were modeled. In this case the formation of the receptor 2 dimmer is always necessary for the ligand binding process to occur. This is modeled on the first equation, the second equation models the clusterization process.

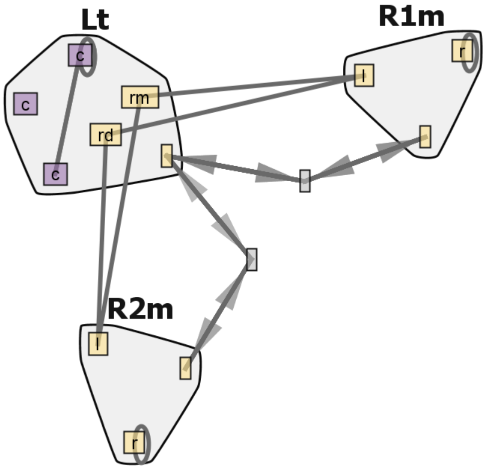


Figure 4.5: Contact Map of the TNF model - The bounds occurring in the TNF binding model are observed. The receptors self-bound is seen on the loops on the PLAD domain and the transformation of the molecules into a binding plausible conformation can also be seen. The ligand (l) contact in the receptors bound to the ligand and the receptor contact (r) on the receptors is used to form the dimmers. We can see the ligand contact that bounds to receptor monomers (rm) and the ligand contact that bounds to dimmers (rd). The three (c) contacts of the ligand are use to keep track of the number of receptors attached to the ligand molecule. Then, they are also used to form higher order clusters of the complexes.

Receptor 1	
$KD_{pR1_m} = 1 \times 10^{-6} \{M\}$	$k_{on\_pR1_m} 1 \times 10^{9} \left\{\frac{1}{Ms}\right\}$
$KD_{TR1_m} = KD_{TR1_d} = KD_{TR1_t} = 2 \times 10^{-11} \{M\}$	$k_{on\_TR1_d} = 2 \times 10^{7} \left\{\frac{1}{Ms}\right\}$
Receptor 2	
$KD_{pR2_m} = 10 * KD_{pR1_m} = 10 \times 10^{-6} \{M\}$	$k_{on\_pR2_m} = 1 \times 10^{9} \left\{\frac{1}{Ms}\right\}$
$KD_{TR2_d} = KD_{TR2_t} = 4.2 \times 10^{-10} \{M\}$	$k_{on\_TR2_d} = k_{on\_TR2_t} = 2.5 \times 10^{7} \left\{\frac{1}{Ms}\right\}$

Table 4.1: Summary of constants used in the model

4.2.3 Results

The results are divided in five parts. Our first interest was to define a suitable value for the activation rates of the PLAD-PLAD interaction domain. This value is not experimentally known so we created figure 4.6 to try to define a value which would be coherent with the dynamics of the system. The second step was to study the dimerization of the receptors depending on the value of the diffusion constant. This result is presented in figure 4.7.

The next two parts of the results present the influence of the diffusion constants in the high order cluster formation of the receptor-ligand complex. First we study the influence of the TNF ligand concentration. The results correspond to a set of three figures. The first shows the behavior of receptor 1 (figure 4.8), and the other two correspond to receptor 2 (figures 4.9 and 4.10).

In figures 4.11 and 4.12, we studied the dependency of the multimer formation on the diffusion constants of receptors 1 and 2 respectively.

The final part presents the dependency of the system dynamics on the population of the receptors. As both receptors bind to the same ligand, the total amount of each of them is a relevant factor in the competition for the available ligand. Figure 4.13 presents four plots for four different values of the receptor 1 and a constant population of receptor 2 of 10000 molecules. Similarly, in figure 4.14 four different values of receptor 2 population are used for a stable value of 1000 molecules of receptor 1.

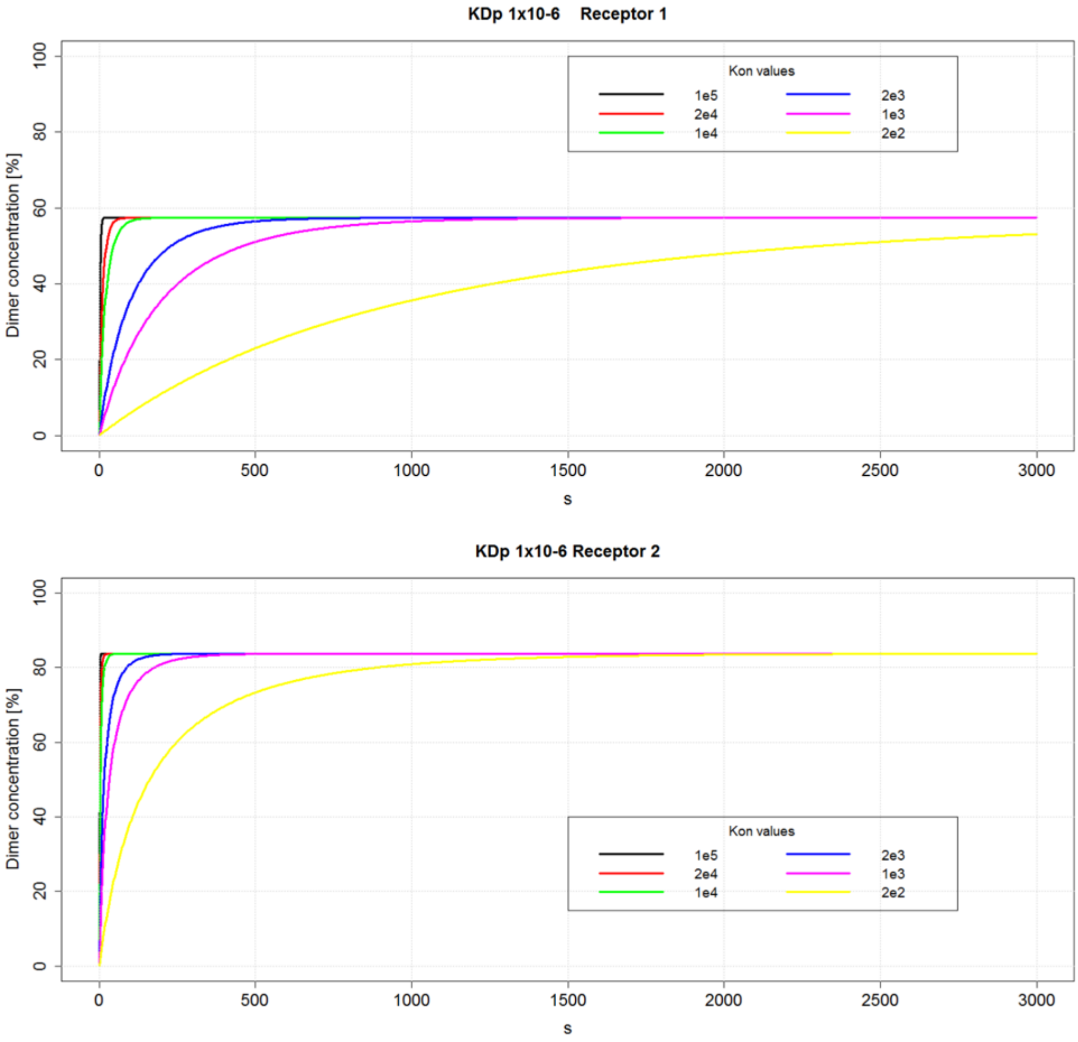


Figure 4.6: Percentage of dimmers in time for several values of the activation constant - Curves for the estimation of the $k_{on\_pR1_m}$ (rate on constant for the plad-plad domain). Receptor 1 population is 1000 molecules and Receptor 2, 10000 molecules. The curves show how the rate value cannot be too high, since we know that the reaction take some time. On the other hand, the rate can also not be that low, since the reaction is completed within a few seconds.

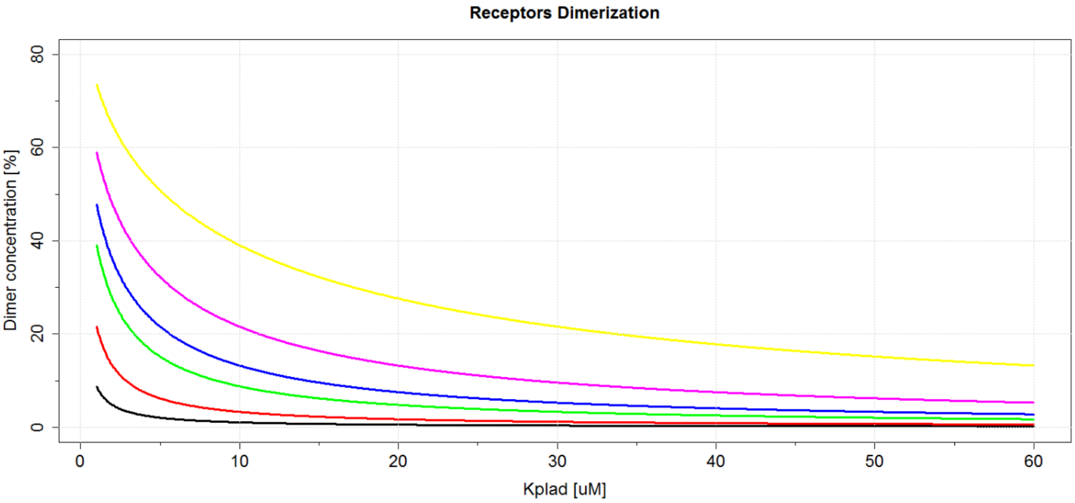
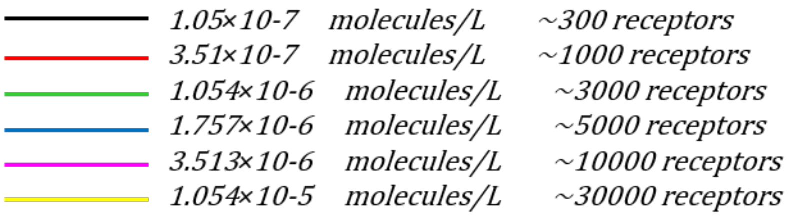


Figure 4.7: Percentage of dimmers with respect to the diffusion constant - Several populations of receptors are depicted as their dimmer concentration changes with respect to the diffusion constant. The respective values of receptors populations for each color are presented next.



⬛	1.05×10-7	molecules/L	~300 receptors
🟥	3.51×10-7	molecules/L	~1000 receptors
🟩	1.054×10-6	molecules/L	~3000 receptors
🟦	1.757×10-6	molecules/L	~5000 receptors
🟪	3.513×10-6	molecules/L	~10000 receptors
🟨	1.054×10-5	molecules/L	~30000 receptors

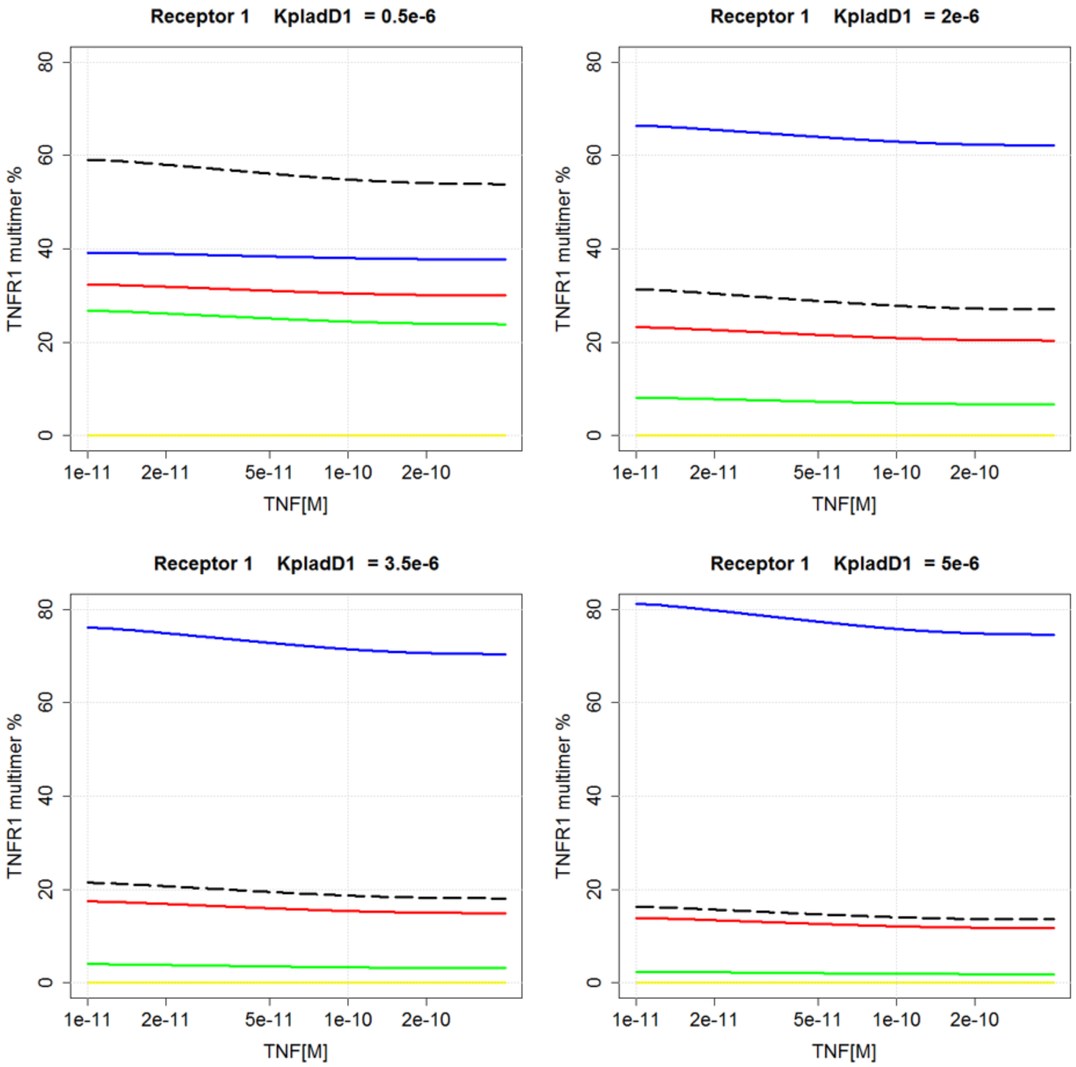


Figure 4.8: Receptor 1 multimmer formation as function of TNF concentration - Several curves are depicted presenting different multimers formed by TNFR1. Dimmers (yellow), trimers (also mentioned before as TNF-TNFR2 complex, blue), hexamers (the bound of two complexes, red) and nonamers (the bound of three complexes, green) are shown. The hexamers and nonamers are the most active structures in the TNF system dynamics. To emphasize the contribution of these, a dashed line shows the sum of them.

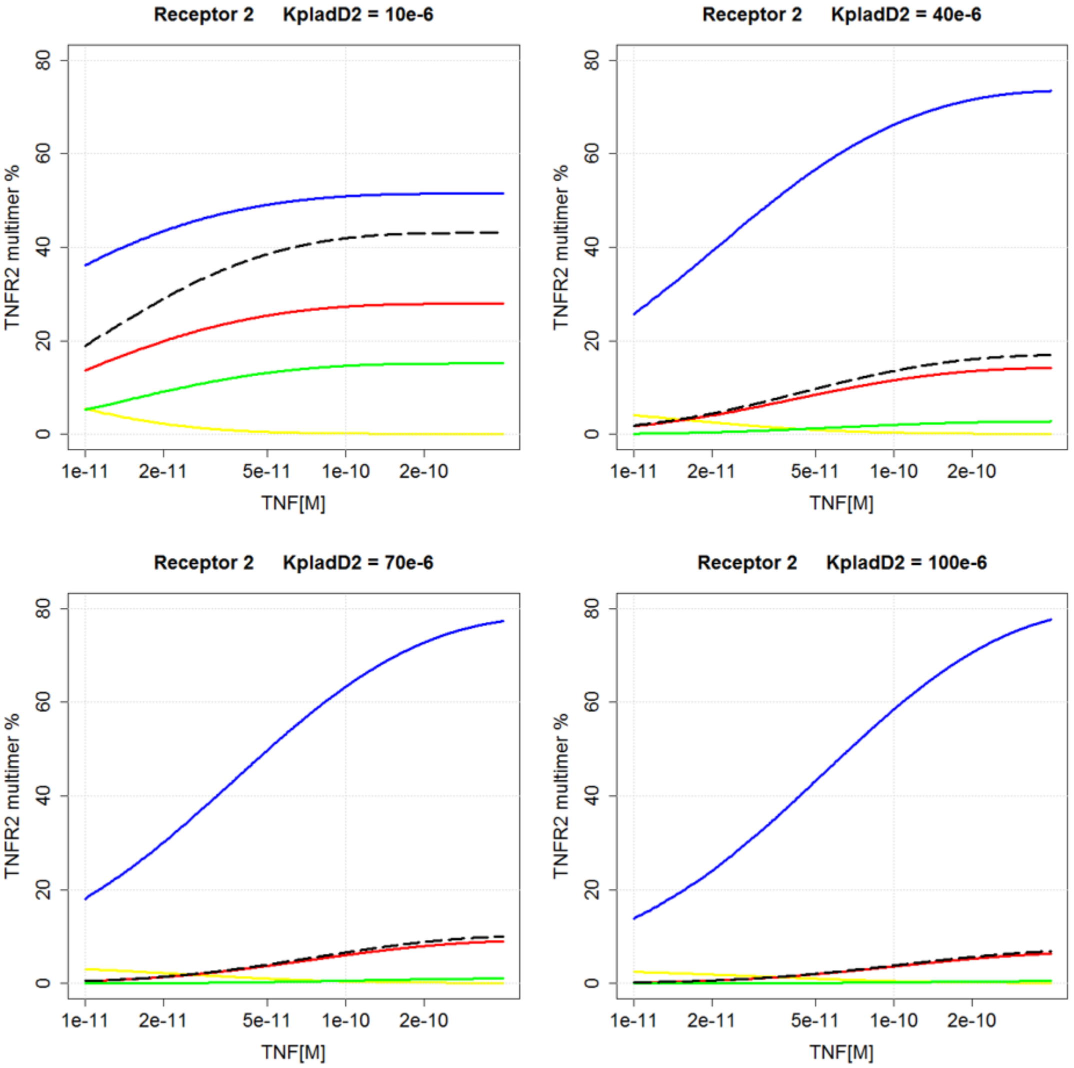


Figure 4.9: Receptor 2 multimmer formation as function of TNF concentration - Several curves are depicted presenting different multimers formed by TNFR2. For these set of plots the ratio between the diffusion constant of receptor 1 and receptor 2 was set to 20. Dimmers (yellow), trimers (also mentioned before as TNF-TNFR2 complex, blue), hexamers (the bound of two complexes, red) and nonamers (the bound of three complexes, green) are shown. The hexamers and nonamers are the most active structures in the TNF system dynamics. To help visualize the contribution of these, a dashed line shows the sum of them

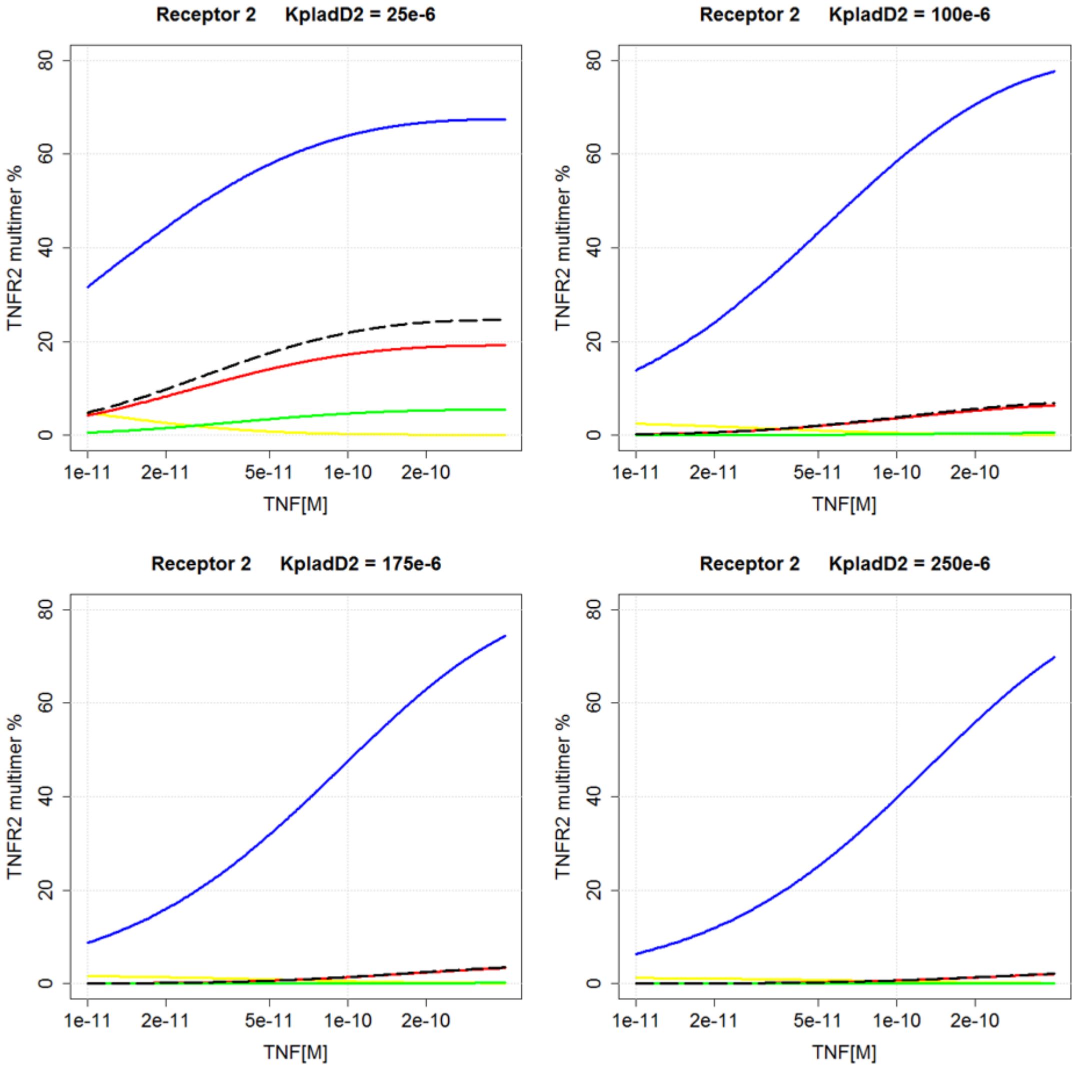


Figure 4.10: Receptor 2 multimmer formation as function of TNF concentration - Several curves are depicted presenting different multimers formed by TNFR2. For these set of plots the ratio between the diffusion constant of receptor 1 and receptor 2 was set to 50. Dimmers (yellow), trimers (also mentioned before as TNF-TNFR2 complex, blue), hexamers (the bound of two complexes, red) and nonamers (the bound of three complexes, green) are shown. The hexamers and nonamers are the most active structures in the TNF system dynamics. The dashed line represent the sum of these two multimers.

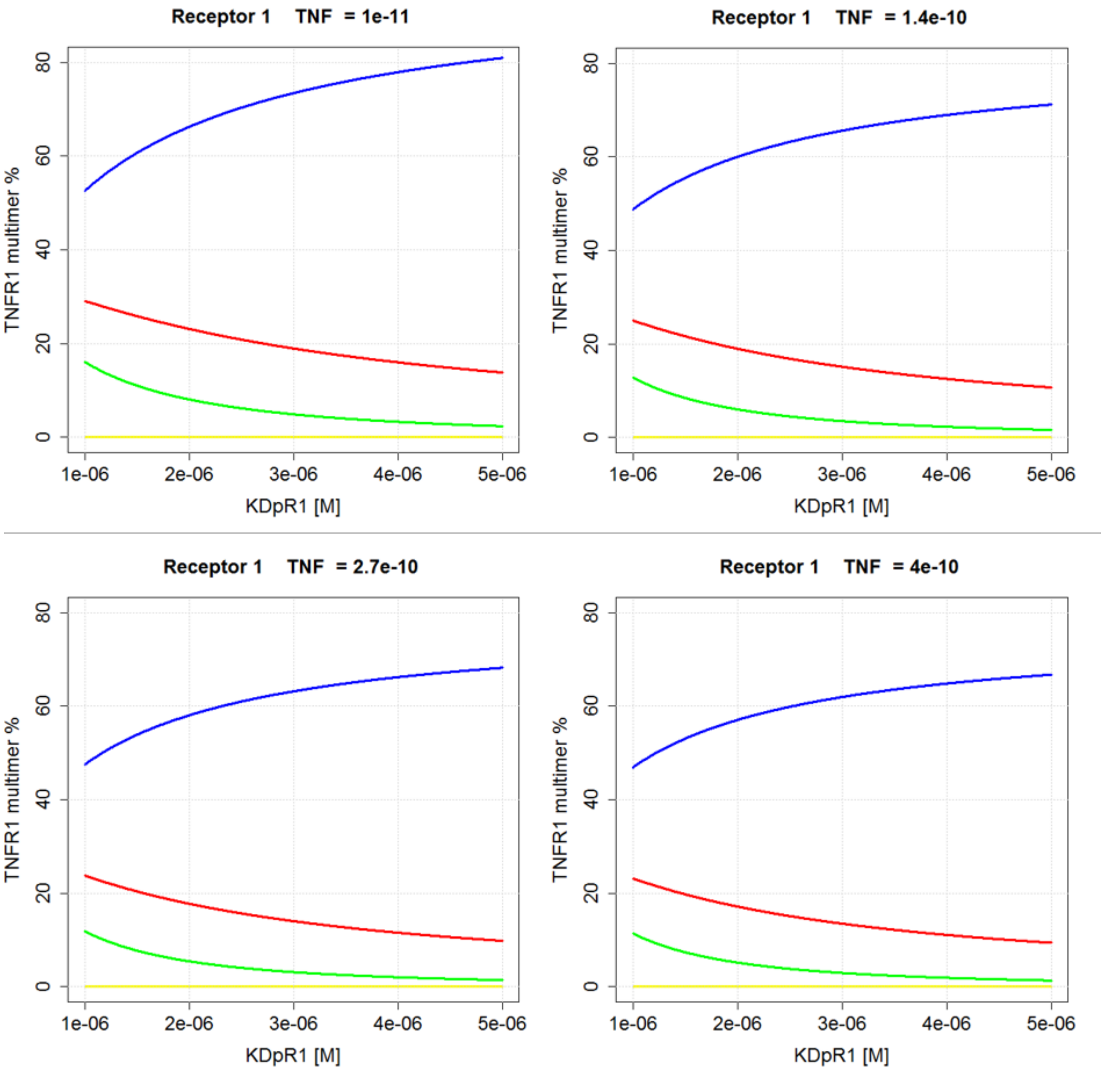


Figure 4.11: Receptor 1 multimmer formation as function of KD_{pR1_m} - Several curves are depicted presenting different multimers formed by TNFR1. Dimmers (yellow), trimers (also mentioned before as TNF-TNFR2 complex, blue), hexamers (the bound of two complexes, red) and nonamers (the bound of three complexes, green) are shown. Four plots are presented, each of them corresponds to a different value of the TNF ligand concentration.

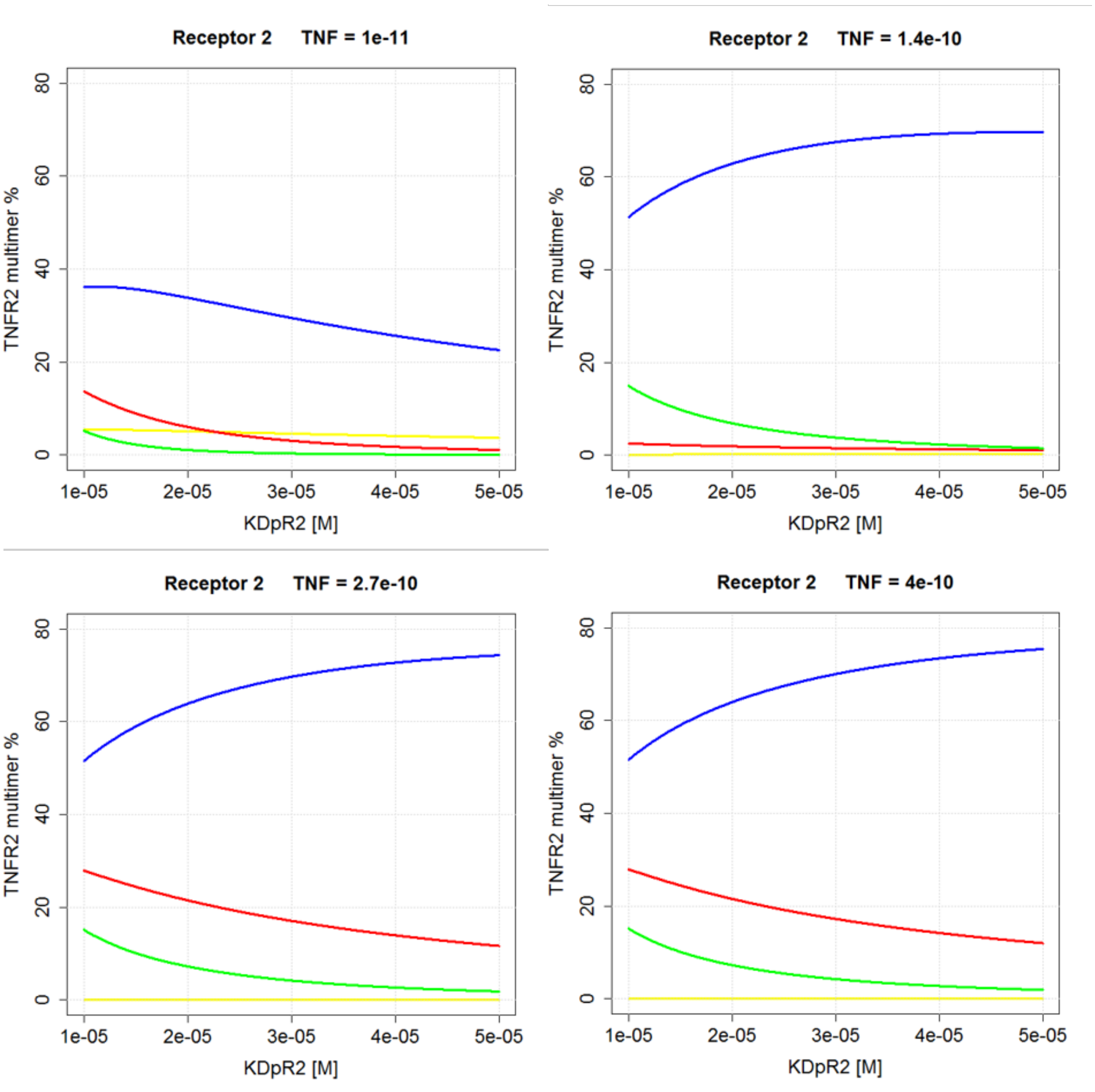


Figure 4.12: Receptor 2 multimmer formation as function of KD_{pR2_m} - Several curves are depicted presenting different multimers formed by TNFR2. Dimmers (yellow), trimers (also mentioned before as TNF-TNFR2 complex, blue), hexamers (the bound of two complexes, red) and nonamers (the bound of three complexes, green) are shown. Four plots are presented, each of them corresponds to a different value of the TNF ligand concentration. In this case the ratio of the receptor 1 diffusion constant to the receptor 2 diffusion constant was of only 10.

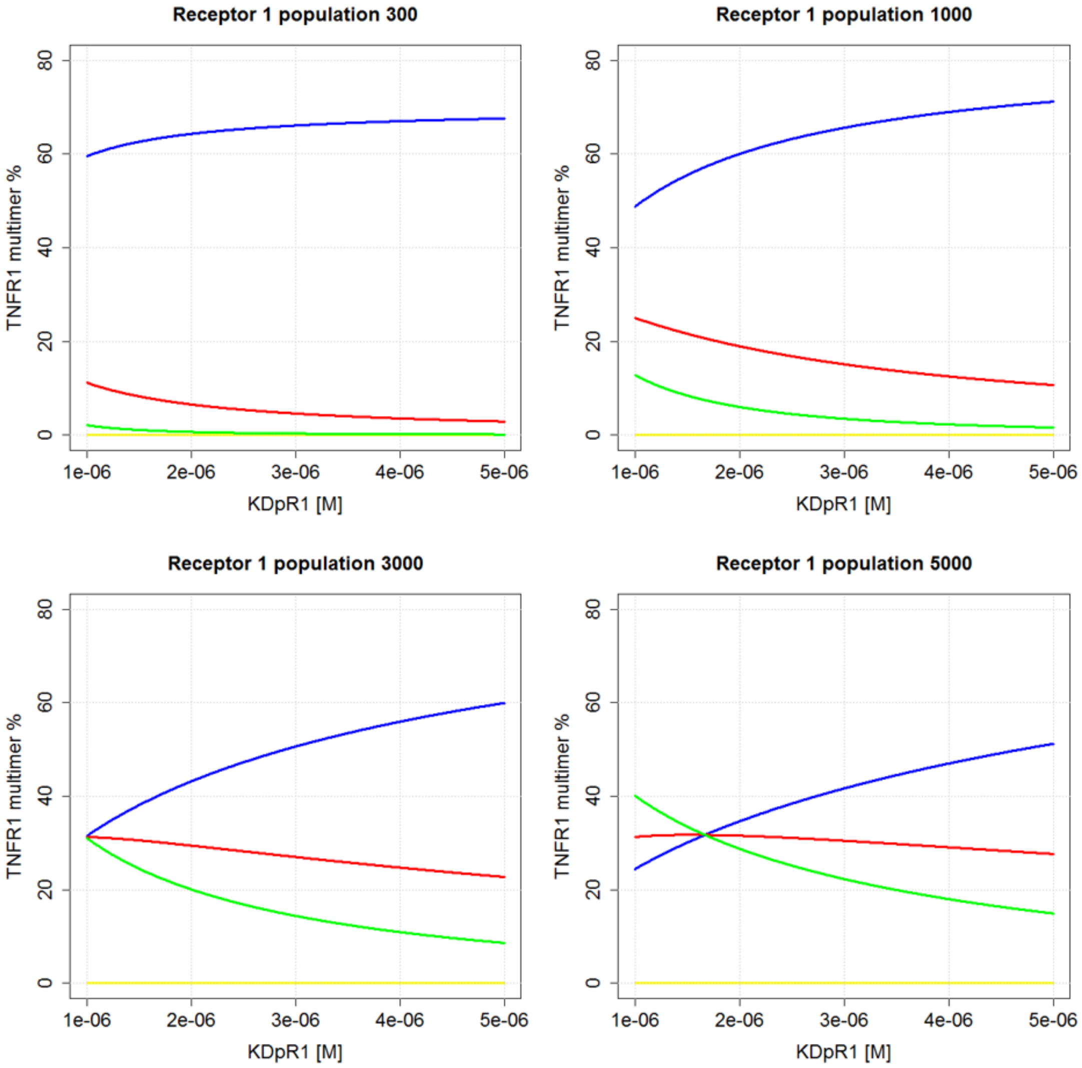


Figure 4.13: Receptor 1 multimmer formation as function of KD_{pR1_m} for several values of receptor 1 population - Several curves are depicted presenting different multimers formed by TNFR1. Dimmers (yellow), trimers (also mentioned before as TNF-TNFR1 complex, blue), hexamers (the bound of two complexes, red) and nonamers (the bound of three complexes, green) are shown. Four plots are presented, each of them corresponds to a different value of the TNF ligand concentration. Each of the four plots correspond to a certain population value of the receptor 1, the value is shown in the title of each plot.

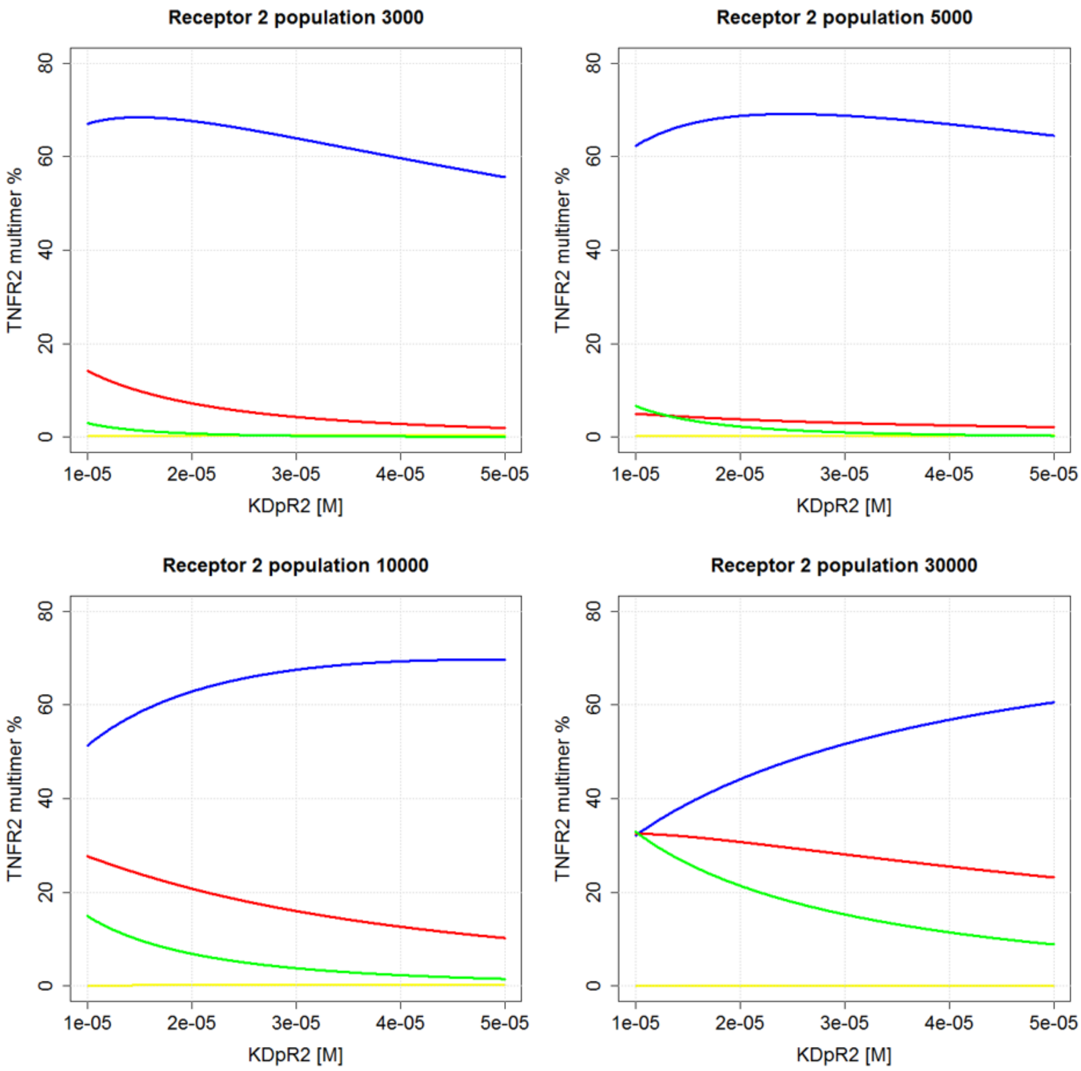


Figure 4.14: Receptor 2 multimmer formation as function of KD_{pR2_m} for several values of receptor 2 population - Several curves are depicted presenting different multimers formed by TNFR2. Dimmers (yellow), trimers (also mentioned before as TNF-TNFR2 complex, blue), hexamers (the bound of two complexes, red) and nonamers (the bound of three complexes, green) are shown. Four plots are presented, each of them corresponds to a different value of the TNF ligand concentration. Each of the four plots correspond to a certain population value of the receptor 2, the value is shown in the title of each plot.

4.3 Discussion

This section is divided in two parts. First I will refer to the TNF system itself and will later discuss about modeling of biological systems in general.

4.3.1 TNF model

We consider that the model we implemented representing the TNF ligand and its receptors 1 and 2 dynamics is precise, simple, and flexible enough to allow some exploration of the system. This means it is an informative model. Next, I will discuss the results obtained from the model itself as well as considerations about the model design.

The BioNetGen language is a very simple one, learning it is a mostly straight forward process. In my case it took a little bit longer since I also had to learn in parallel about stoichiometry. Because I had some experience with programming languages and also because I have worked before with differential equations. It was easy to understand the philosophy of the language and to visualize how the TNF model could be represented with it. Learning about chemical reactions and its constants was a little bit harder. I had to learn from scratch about reactants and resultants, had to understand what the binding equilibrium constant means; how is it measured, and what role does it play in the reaction. The same for the diffusion constant. Then there was also the problem that I had to learn the association and dissociation rate constants, which control the speed of the reaction. Once I understood the basic concepts, I had to learn about dynamics in the TNF system. This was done very fast since Professor Wajant knows the system in great detail and explained it to me quite easily.

The first attempts to program the model were unsuccessful because BioNetGen has a very specific characteristic which could be easily overseen. In BioNetGen the rate constants have to be converted from general rate constants to local rate constants. Due to the fact that it is a compartmental language, BioNetGen needs to change the reaction rates from Moles per second to reactions per second. This conversion is done by multiplying the Avogadro number times the volume of the compartment where the reaction takes place. Once the model was correctly programmed and we could verify that it contained no mistakes, we began extracting information about the dynamics of the TNF system. This information was further processed and analyzed.

Following the same order of the results presented above, the first aspect we studied from the model was the rate constant of the PLAD-PLAD interaction. This rate constant (as mentioned before) is not experimentally known so we decided to infer it from the model by searching for a suitable speed in the dimmer formation of the receptors. The curves presented in figure 4.6 clearly show how the speed of the reaction changes as the rate constant is changed. We saw that a rate of 200 reactions per second is way too low (yellow curve in the plot). The TNF ligand binding is a process that occurs in about three minutes, so we know that the dimerization process has to be much faster. The fastest rate that we tried was 1×10^5 reactions per second and we consider this is a more appropriate value for this constant. We finally decided to increase the factor even more and ended up with a constant rate of 1×10^9 reactions per second. Since it is not at all clear whether the activation rates of receptor 1 and receptor 2 differ, we decided to use the same rate for both reactions. This might not be quite the case in the real system but it is a coherent assumption and it guarantees that no rate difference will affect the results. It is simply one less degree of freedom in the model. From this experiment it was nice to verify that the model realistically represents the concepts of diffusion constant and rate constant. The diffusion constant controls the equilibrium point of the reaction. It changes the percentage of reactants and resultant as the reaction reaches a stable point. And on the other hand, the rate constant does not affect this equilibrium point but determines how fast the curve reaches such equilibrium point.

Then we analyzed how the dimmerization of the receptors changed according to the diffusion constant of the PLAD domain for several population numbers. The result is presented in figure 4.7. In this graph we make no distinction between receptor 1 and 2 because the dimmerization process of the two receptors is exactly the same. The only difference is the diffusion constant and the population number of each receptor. This graph shows the results for several population numbers, and along a broad spectrum of values of the diffusion constant. In this graph we see that a high population increases the dimmerization process. It is known that receptor 2 population is bigger than that of receptor 1. Nevertheless, a similar concentration of receptor 1 and 2 dimmers is found. This means that the higher affinity of receptor 1 to form multimers has to compensate for the lower population factor. From this result it is clear that the diffusion constant of the receptor 2 has to be significantly bigger than that of receptor 1 (big diffusion constant means lower affinity). We estimate that a factor of 50 would be appropriate. It is not experimentally known what is the difussion constant of the receptor 2, only that of receptor 1 is known (about $1\mu M$). The fact that the diffusion constant of receptor 2 can be inferred from the model shows how informative the model is. Knowing an approximate value of the receptor 2 diffusion constant makes it easier to design experiments that allow researchers to find out the true value of this constant.

The next point that we wanted to evaluate was the dependency of the cluster formation on the TNF ligand concentration. The results presented in figures 4.8, 4.9, and 4.10 give some important insights. The first thing to notice is that the behavior of the receptor 1 complex formation and clustering is more or less stable with respect to the TNF concentration. The lines only change slightly but it is nevertheless quite surprising to see that there is a decreasing tendency. We had expected that, as the TNF concentration gets higher then the formation of higher order multimers of the TNF-TNFR1 complex increases as well. The decrease tendency we observed results quite surprising but it can be understand when we see the results from TNFR2. In the graphs corresponding to TNFR2 we see that as the ligand concentration gets higher the cluster formation is also increased significantly. We see that for high values of the PLAD domain diffusion constant this effect is not so strong on the high order cluster structures but it remains always strong on the TNF-TNFR2 complex formation (blue line on the graphs). So in summary we see that the complex formation of the ligand and the receptor 2 is strongly increased as the ligand concentration is augmented. It is probably this strong increase in the TNF-TNFR2 complex which causes the decline that we see in the receptor 1 curves. Here we see how the two receptors compete for the ligand available. Our model was programmed in such a way that a replenish mechanism of the TNF ligand was included. The concentration of the ligand is maintained along the simulation. Nevertheless, the receptor still compete for the ligand available and we see here how the receptor 2 benefits from a higher concentration of the lingand, while receptor 1 is capable of forming high order structures also when the ligand concentration is low.

Now, the influence of the diffusion constant in the PLAD domain on the formation of high order structures is discussed. The dependency on the diffusion constant of the receptors 1 and 2 clusterization results as expected. As the diffusion constant gets higher, which is equivalent to lower affinity, the formation of higher order structures (hexamers and nonammers) decreases. In the case of the TNF-TNFR 1 and 2 complexes, these structures not only do not decay with the increase of the diffusion constant but they actually increase. This is in a certain way surprising, specially for the receptor 2 where the formation of dimmers is necessary to bind to the ligand. Receptor 2 presents also an unexpected behavior when the ligand concentration is too low. We see that if the TNF concentration is low then as the diffusion constant is increased the complex formation decreases as we expected. We can infer from these results that a high concentration of the TNF ligand compensates for a low affinity in the PLAD-PLAD interaction so the complex TNF-TNFR2 is formed anyway.

The last set of experiments we made was to check how the receptor population would interfere with the clusters formation. We created the graphs from figures 4.13 and 4.14 for this purpose. In the case of receptor 1 we see a quite direct relation between the number of receptors and the percentage of nonamers and hexamers formed. When the receptor population is low most of the receptors remain in the bounded trimmer conformation, but as the population increases the receptors actively move into more complex structures. The tendency of the clusters to decay with the increase of the diffusion constant is maintained.

Now, concerning receptor 2 we see an interesting behavior. When the population number is low then the percentage of complexes (bounded trimmers) decreases with the diffusion constant increase. But for higher populations this behavior is inverted and the percentage of complexes augments with the diffusion constant, which is the same behavior observed for the receptor 1. It is very important to notice how relevant the concentration of the receptors in the membrane is. When it comes to stoichiometry concentrations of substances play a mayor role since they directly affect the binding constant of the reactions. It is interesting to see that receptor 2 compensates for its low affinity for the ligand with a higher concentration than that of receptor 1. We see that for populations of receptor 2 from 10000 and above the behavior with respect to the diffusion constant resembles that of receptor 1.

Many insights about the TNF system dynamics were inferred from our *in sillico* model. Further work is required to experimentally verify such insights and once they are confirmed, include them in the model.

4.3.2 Models in Biology

There is not much left to say about biological modeling. I already mentioned the most relevant aspects like the validity of the models, their importance, and their inherent bias which cannot be ignored. Although, some discussion on the modeling that was inspired by the realization of this project might of interest for the reader.

The process of creating this model was a long one. As mentioned before, it was necessary for me to learn from scratch almost every aspect of it: the biological system, the programming language and the basic concepts of stoichiometry. The first lesson that comes to my mind is that there is no way to create such a model alone. I think a good biological model has to be the product of the work of several people which can contribute from their particular area of knowledge. In this project I had the help of Dr. Gaby Wangosch and Proffesors Dandekar and Wajant. From Gaby I learnt the basics of rule-based modeling; while with Proffesor Dandekar we discussed and conceived the model itself; and from Proffesor Wajant I learned about stoichiometry and about the TNF biological system. With the help of Proffesor Wajant the model was thoroughly tested and the *in sillico* experiments were designed.

Another important conclusion is that biological systems resemble each other. Nature is a smart planner and uses the resources it has more than once. This means that a signaling cascade resembles another one. They are of course not exactly the same, but they might have more in similar than it is originally believed. For example, a process like ubiquitination is found everywhere in living organisms. This is a very important lesson because the researcher needs to know that he can find help from other investigators even though if they work on different biological systems. There is an increasing interest in the bioinformatics community to create a general framework for modeling biological systems. Some attempts have been already made and published. BioNetGen is one them. It is true that we still lack of information to completely understand the way biological networks interact, but it is also true that we already know a lot and that probably the only possible way to acquire the knowledge that remains outside from our comprehension is by creating biological models and then refining them with new experiments design according to the results of such models.

In physics, a general model is much closer, they already have a first strong candidate known as the standard model. This is a framework that intends to categorize all the known particles in quantum physics. Whether the model is correct or not is still under debate, but what is absolutely clear is that thanks to this model physics has experimented a great development and experimental physicists find in the standard model a guiding route for their research.

Exactly that is what the bioinformatics community is searching. We want a model that resembles the way nature design living organisms. It is a compartmental design and it has a set of shared principles, which might be a lot but still limited so we should be able to modeled them. It is common to read in biological research that a pathway in the brain follows the exact same route as a pathway in the liver for example. This is no coincidence. It is just that nature has developed a set of building blocks and it makes use of these same building blocks across a whole organism, and even more across different organisms and species. Many of those building blocks are already known and there exist already models of them, but we still need to discover the rest of them. And most importantly, we have to understand the relationship between these building blocks; how they interact with each other, why ones are used in certain occasions, and why others in different circumstances.

The never-ending advance of technology is of much help in the creation of the biology standard model. The amount of information needed for such model, cannot be computed nor stored by a human. We are lucky to have computers of great capacity to assist us in such tasks. Nevertheless, computational capacity is not the only restricting factor in this enterprise. It is necessary that researchers strongly collaborate and realize the importance of a communal model over personal achievements. Luckily, science is moving towards open access and I hope in the near future, more and more research collaborations will be seen.

5

Protein folding and theoretical implications from Bioinformatics

5.1 Introduction

Chaitin is a well known mathematician who proved that in mathematics exists an infinite number of statements which can not be proved nor disproved from the existing set of axioms. He discussed this in an article from the Scientific American magazine in 2006 and before that in standard scientific publications [65]. Similarly before him, Gödel presented his incompleteness theorem. Chaitin claims that in sight of this situation it is necessary that mathematics moves towards a more experimental area. He says that in order to keep progressing, mathematics has to start using experimental data and not constrain itself to hard proves.

Mathematics is by far the hardest of the hard sciences. It is the only field where all things have to be proved analytically and there is no room for experimental data. Only recently some voices are being raised claiming for a experimental branch in mathematics. Biology arguably resides on the opposite side of the spectrum, with tons and tons of experimental data and a limited number of hard proved theories. It is of course clear to me that an absolute model of biology cannot exist. It is not possible to create a model that represents each aspect of living organisms. Nevertheless, it is possible to discover some guiding principles and establish models using them, for example the evolution theory we discussed before. Just as mathematics is in need of experimental data in order to keep progressing, biology is in need of theoretical models to represent the huge amount of data available, and to guide the search for more data.

It is also true that from the theoretical study of biological models many other areas of science will benefit. It is very common today to talk about nature inspired designs and other similar terms. The fact that nature has been improving its mechanisms since such a long time, makes it a really good idea to look at it for inspiration. The Deep Learning algorithm presented before is a clear example of this. But this is not the first algorithm to be inspired by biological systems, before deep learning there were the famous genetic algorithms and before those ones probably many others.

We decided that it was important for us to come up with some theoretical contributions and not just use Bioinformatics as a mere tool. We realized that protein folding was an ideal scenario to work on. It is a problem of high relevance from the biological point of view, and from the theoretical point of view as well. Protein folding is of extreme importance since it is known that a lot of diseases correspond to an accumulation of misfolded proteins, for example prion diseases, Alzheimer's disease, and Parkinson's disease. It is curious that the term *amyloid*, which is how this cumulus of misfolded proteins are normally referred to, was coined by Rudolf Virchow, who was professor of pathology here at the University of Würzburg in 1849 (presently the Graduate School of Life science operates in a building named after this scientist). These amyloids are of special importance for neuroscientists due to its appearance in the neurological pathologies named before, among many others.

From the mathematical point of view, protein folding turns out to be also of high interest since it is related to one of the most popular open questions in mathematics, the NP vs P discussion. P problems are those which can be solved in polynomial time by a deterministic turing machine and the NP problems are those which might not be possible to solve in polynomial time using a deterministic turing machine, but a solution can be checked in polynomial time and they can also be solved in polynomial time but by a non-deterministic turing machine. Solving a problem in polynomial time means that the time to find a solution for the problem scales polynomially with the size of the problem. The deterministic turing machine is the standard computer that we use everyday and the non-deterministic turing machine is a theoretical construction of a machine which correctly guesses the next step on the solving process. The big discussion in mathematics is whether the P class of problems is inherently different from the NP class, or if they are actually the same and we just need to keep searching for polynomial time solutions to those problems we haven't been able to solve. Protein folding has been proven to be an NP problem. So, the fact that nature folds proteins so fast is of course quite interesting. Being protein folding an NP problem means that it takes an exponential amount of time to predict the structure of a protein. For a protein of regular size, lets say 150 aminoacids, it should take millions and millions of years to find the correspondent structure. But, somehow nature folds thousands of proteins in our organism every day, how can this be? What does nature know about NP problems that we don't? These are the questions that motivated our research in this project. Lets explain in more detail the protein folding problem as used here.

The protein folding problem is the task to predict the tertiary structure or native state of a protein just based on its primary structure or aminoacid sequence [66]. There is enough evidence to support the theory that the native state conformation is dictated completely by the aminoacid sequence, so it should be possible to predict the native folding solely from the primary sequence [66]. For this purpose several models have been proposed [67]. Some are detailed models which pretend to simulate the complete set of interactions and forces playing a role during the folding process [67]. On the other hand, there are the simplistic models which discard many factors and focus on one or two alone [66, 68]. The most common ones focus on hydrophobic interactions and/or covalent bridges such as the 2D lattice model. We proposed an algorithm that can be used to solve the Protein Folding problem on any of these instances and started by proving it on the 2D lattice model.

The 2D lattice model for Protein Folding was introduced by Dill K.A. in 1985 in the paper "Theory for the folding and stability of globular proteins." Later, protein folding was proven to be NP-hard by B. Berger and T. Leighton (1998) in their "Protein folding in the hydrophobic-hydrophilic (HP) Model is NP-complete" article. This model is entirely guided by hydrophobic forces which account for packing the hydrophobic side chains in a central core and surround them by polar aminoacids. The algorithm we propose solves the problem via an optimization problem. We maximize the number of contacts established among hydrophobic aminoacids following the constraints imposed by the 2D lattice model. As the 2D lattice model is known to be NP-hard [Berger and Leitova 1998], our problem is only a relaxed version of the problem so that it can be solved and from the solution it delivers, it could be necessary to remove some contacts in order to state the definite solution. However, it is shown to serve as a good heuristic to approximate protein structures in 2D and 3D lattice models and in constrained-boxed modeling in general. We also present a discussion on the NP-hardness of protein folding and the ability of nature to deliver in short time (seconds) well folded protein structures.

5.2 Results and Methods

There are no biological methods in this project since we performed no wet lab experiments, neither directly or through a collaborator. We did, nevertheless, read a lot of literature from other groups working on protein folding [69, 70, 71, 72] and we discussed the topic with experts. These included Doctor Hannes Neuweiler here at the University of Wuerzburg and Professor Roman Jerala, head of the National Institute of Chemistry in Slovenia, as well as others.

The result of this project is composed of two parts. The first part is a theory of how protein folding occurs in nature. The second part is an algorithm that solves the protein folding problem in a simple form based on the assumption of the first part.

Our theory regarding how nature is capable of folding proteins in just some seconds despite being an NP-hard problem is that nature does not solve the whole problem. It solves a limited version (in mathematics we would call it a relaxed version) of the problem. Protein folding is a molecular process where bounds are formed between aminoacids in such a way that the protein, originally unstructured, acquires an structure which is in most cases vital for its function. Being a molecular process, protein folding is governed by laws of physics, being the free energy law the most relevant in play. Although one must not forget that other important forces are also in play like the electromagnetic and mechanic forces. The hydrophilic and hydrophobic forces play also an important role. It is known that free energy is a very important aspect in the process and it is common to use this driving force to describe the problem. In figure 5.1 the free energy landscape of protein folding is shown. This landscape is known as the funnel. The surface we present doesn't look much like a funnel because the term funnel was used and is still used by those who think that the energy landscape of a protein folding process has a funnel-like shape and that the native conformation of the protein is the lowest energy state in the surface. As it can be seen in the figure, we have a different idea. This is the most important result of this project. We hypothesized that the energy landscape of the protein folding process is a highly non-convex surface with plenty of local minimums and that the native conformation of the protein is not necessarily (and not usually) the lowest energy state. Just as it should be expected from such a complex process and from an NP problem.

Nature manages to take the protein from the unfolded state to tertiary state conformation by carefully setting the starting conditions of the process. Just the same as humans, nature is incapable of quickly solving NP problems. It takes an easy way out of the problem. The organism makes sure that the starting conditions of the process are identical every time and so goes to the same local minimum every time. Knowing the initial conditions, protein folding becomes a simple convex problem which nature and also us are capable of solving efficiently. The initial conditions of protein folding are given by the physical conditions at the moment the protein comes out of the ribosome. Aspects like, temperature, pressure, electromagnetic forces, mechanical forces, pH, and concentrations of glucose and calcium. Nature is a gifted handcraft worker capable of maintaining a delicate balance which ensures that the protein folding process starts always from the same initial point. And so, it goes to the same nearest local minimum which is the point of the native conformation of the protein. Of course errors occur. One can have fiber for example and then the internal temperature is affected. When that is the case, protein misfolding takes place. There are mechanisms in the body which take care of decomposing the misfolded proteins but when the amount of misfolded proteins is more that what the organism is capable of processing and the amyloids start to form, then diseases appear.

Now we formally state the contact placement method used to solve the protein folding problem on the 2D lattice model. In order to make clear the proposed optimization problem we present an example along the way.

Given an aminoacid chain x of length n, defined as:

$$x \in \Re^n \mid x_i = \begin{cases} 1, & \text{if } \textit{amino } i \text{ is hydrophobic} \\ 0, & \text{if } \textit{amino } i \text{ is polar} \end{cases} \qquad i = 1, ..., n$$

We define a matrix $X \in \Re^{n \times n}$, such that $X_{ij} = 1$ means that aminoacid i and aminoacid j establish a contact on the folded structure and $X_{ij} = 0$ otherwise. *Contact* means that a pair of aminoacids occupy adjacent positions on the 2D lattice. For example, in Figure 5.3 aminos 1 and 6 establish a contact. It is important to state that a contact cannot occur among consecutive aminoacids on the original chain, and contacts can only occur among hydrophobic aminoacids. It is also necessary that matrix X reflects the structure of 2D lattice constrictions.

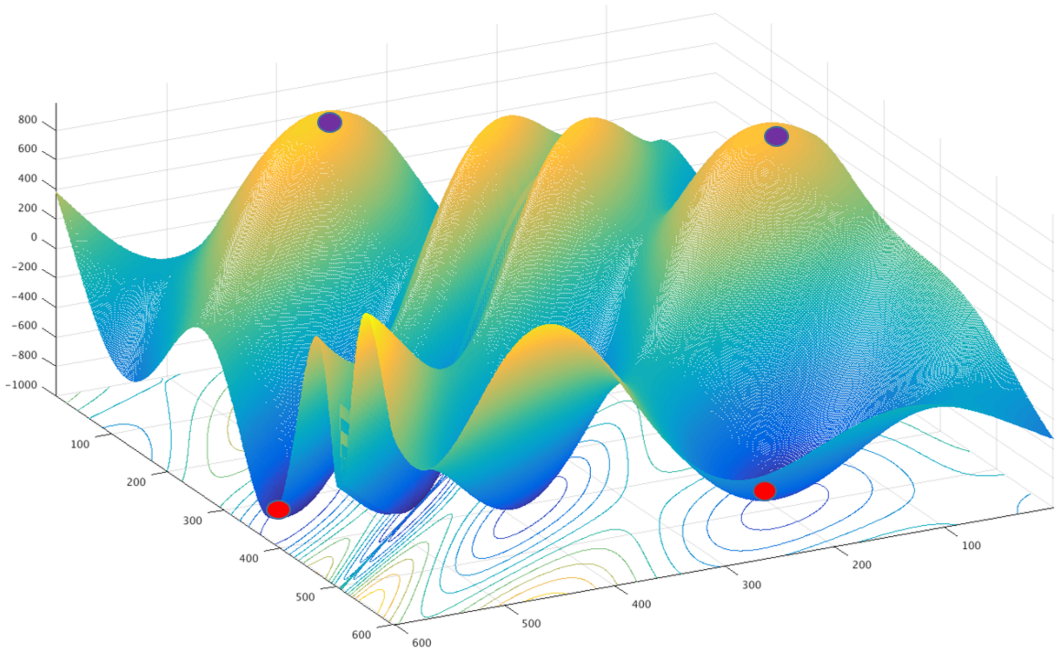


Figure 5.1: Proposed "funnel" of the protein folding process. We present a limited version of the real funnel, here only three dimensions are depicted being the z-axis the free energy and the x-axis and y-axis two of the before mentioned characteristics in the cell interior, for example temperature and ph. The real funnel is a high dimensional surface but the principle is the same. All the physical conditions in the cell define a starting point in the folding process and the final conformation of the protein is given by the closest local minimum to this starting point. The purple points represent two possible starting points and the red points are the local minimum correspondent to each of those starting points.

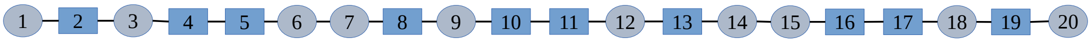


Figure 5.2: Amino acid chain of 20 units. The circles represent hydrophobic contacts and the squares polar contacts. The numeration is to facilitate further analysis.

Hence, for example, having the unfolded aminoacid chain from figure **??** the corresponding vector x would be defined as $x_{example} = [1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1]$. Having a folding like the one presented in figure 5.2, the correspondent matrix X would be as presented in figure 5.4.

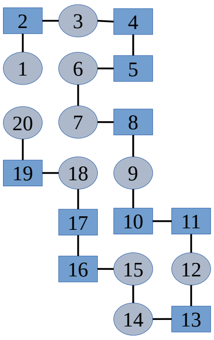


Figure 5.3: Possible folding state of the aminoacid chain- A possible conformation of the final structure of the chain presented in figure 5.3 is shown. As mentioned before the circles are the hydrophobic contacts and these are the ones expected to form bounds or to be nearby in the interior of the protein.

This statement of the problem is universal and can be applied to any model of protein folding by doing the proper adjustments to the problem constrictions. The problem turns specific according to the restrictions we impose and the way the objective function is stated. We consider that the best point to start is the 2D lattice model. Because the results can be interpreted directly and the data available can be used almost straight forward. Furthermore, we consider that this is the hardest model to apply the algorithm to (probably equally hard to a 3D lattice folding problem). In the other models the statement of the problem might be easier. The inconvenient is that those other models require more detailed information about the aminoacid chain, the interactions between aminoacids, and the kinetic restrictions of the model.

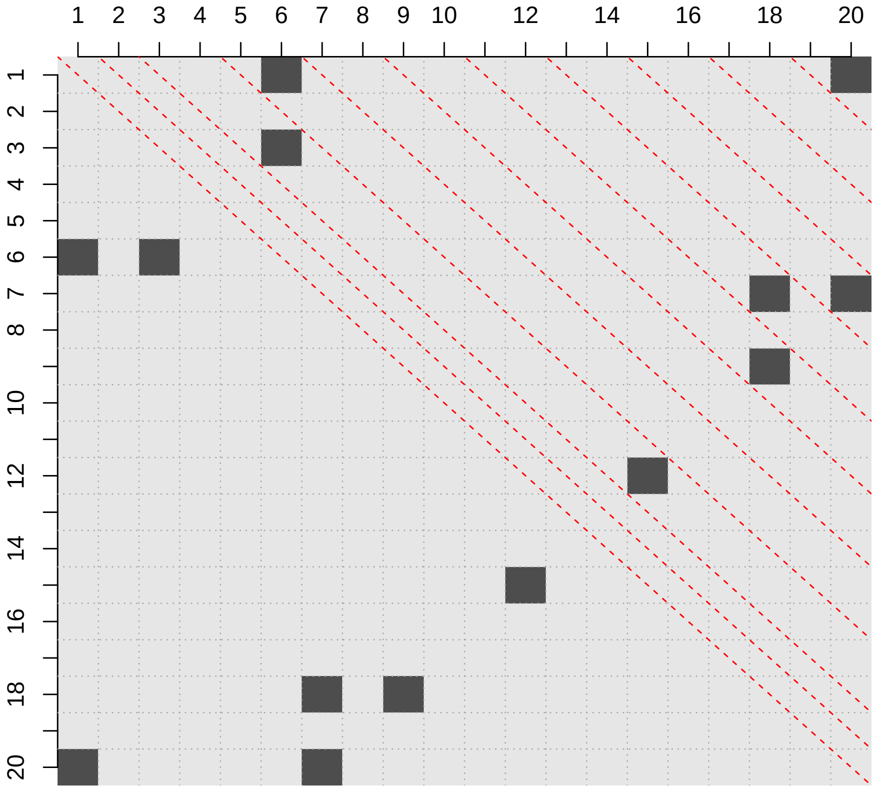


Figure 5.4: Contact matrix for the folding presented on Figure 5.3- The matrix is 20 by 20, because 20 is the number of aminoacids in the example. A dark square in the graphical matrix represents a value of 1 in the numerical matrix. This number 1 means that a contact is established between the aminoacids in the row and column of the entry. For example, the entry 1-20 is marked with a dark square because aminoacid 1 and aminoacid 20 form a bound in the folded protein.

5.2.1 Contact placement problem for 2D lattice

We present the mathematical formulation of the problem as a linear program. This is a maximization problem with X as the main variable.

$$
\begin{aligned}
\max \quad & trace(CX) \\
\text{s.t.} \quad & trace(A1 \cdot X) = \mathbf{0} \in \Re^{n\times n} \\
& Trace(A2 \cdot X) = \mathbf{0} \in \Re^{n\times n} \\
& A3 \cdot X \leq B1 \\
& A4_m * X \leq \mathbf{3} \in \Re^{n\times n} \quad m = 2, 3, 5, 7, 9, ... \\
& Trace(A5'_{ss} * X) \leq 1 \\
& Trace(A6'_{rc} * X) \leq 2 \\
& Trace(A7'_{dd} * X) \leq 1 \\
& X_ij = X_ji \quad \forall ij \\
& X_ij \in [0, 1] \quad \forall ij
\end{aligned}
\qquad \text{(Eq. 5.1)}
$$

-Define the logical NOT function as:

$$Not(a) \rightarrow b \triangleq \{a_i = 0 \rightarrow b_i = 1, a_i = 1 \rightarrow b_i = 0\}$$

where the vector a is a logic vector. Applying this function to our aminoacid chain example results in:

$$Not(x_{example}) = [0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0]$$

-This NOT function can also be used in other arrays. For example, in 2 dimensional matrices.

-The operator $*$ is a 2D convolution defined as:

$$Result_{x,y} = A * X = \sum_{i=-a}^{a} \sum_{j=-a}^{a} A_{i,j} X_{x,-i,y-j}$$

-The operator $\leq$ is used here as an element wise comparison.

-Define $AllContacts = x' \cdot x$ as the matrix with the maximum number of contacts possible if there were no space restrictions.

-A short-step pair in a matrix is defined as the existence of a couple of entries with a

relative position of (-3,-1) or (-1,-3) of one with respect to the other. The set of all short-step pairs in $A2$ is called ss.
-An inverse subdiagonal is defined as the subdiagonal in a matrix going from the left down corner to the right up corner.
-A rhomboid upper-corner is a set of three contacts where we define an upper contact which lays on a higher row than the other two and this upper contact lays on the same subdiagonal with one of the other two contacts and on the same inverse subdiagonal with the other.
-The set of all upper rhomboid corners is called rc.
-The distance on the diagonal is the number of entries in the matrix between a pair of contacts.
-According to the spatial restrictions a certain distance is relevant for each subdiagonal. In subdiagonal 4 the distance is one (one entry in between contacts), in subdiagonal 6 the distance is 2. In subdiagonal 8 the distance is 4, and from there on it grows adding 2 entries for the next possible subdiagonal.
-The set of all subdiagonals and its respective diagonal-distance constraint is called dd.

The matrices from the restrictions are defined as:

$$C = AllContacts \in \Re^{n\times n}$$

$$A1 = Not(AllContacts) \in \Re^{n\times n}$$

$$A2 = AllContacts\ \&SubDiag(A2,m) = 0,\ m = 4,6,8,10,...$$

$$A3 = \mathbf{1} \in \Re^{1\times n},\ B1 = [3,2,2,...,2,2,3] \in \Re^{1\times n}$$

$$A4_m = \mathbf{1} \in \Re^{3\times m}\ m = 3,5,...$$

$$A5 = Set\ of\ matrices,\ one\ for\ each\ short-step\ pair\ in\ ss$$

$$A6 = Set\ of\ matrices,\ one\ for\ each\ rhomboid\ upper-corner\ in\ rc$$

$$A7 = Set\ of\ matrices,\ one\ for\ each\ diagonal-distance\ constrain\ in\ dd$$

Analysis of each restriction:
$trace(A1 \cdot X) = \mathbf{0}$.
Links can only be established among hidrophobic aminoacids. $A1_{ij}$ is 0 iff aminoacids i and j from the chain are hydrophobic. This restriction prevents polar aminoacids from forming contacts.

$A2 = AllContacts\ \&\ Diag(A2) = 0\ \&\ SubDiag(A2,m) = 0,\ m = 2,3,5,...(n-1)$.
The links occur between hydrophobic contacts of different parity and an aminoacid

cannot link itself, its immediate neighbor or the neighbor after that. This restriction is shown in figure 5.4 with the red dotted diagonal lines. No contact can be formed on the positions these lines pass by.

$A3 = \mathbf{1} \in \Re^{1 \times n}$.
The sum per column of X is equal or less than 2 for the interior columns, and 3 for the extreme ones.

$A4 = \mathbf{1} \in \Re^{3 \times m} m = \{3, 4, 5, ..., n\}$.
Four aminoacids can establish at most three contacts between them. In the matrix of figure 5.4 we present an example. The zero on the position 3,20 could not become a one. It means no contact can be establish between aminoacids 3 and 20 because of the contacts 1-20, 1-7 and 3-7. Otherwise stated, a pair of aminoacids cannot have two contacts in common.

$Trace(A5'_{ss} * X) \leq 1$.
No short-step pair can occur because of spatial constraints.

$Trace(A6'_{rc} * X) \leq 2$.
No rhomboid upper-corner can occur because of spatial constraints.

$Trace(A7'_{dd} * X) \leq 1$.
On each subdiagonal, a pair of contacts cannot be closer than the correspondent minimum diagonal-distance.

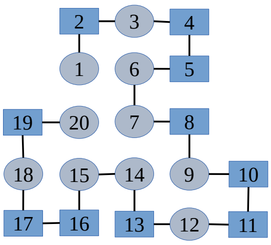


Figure 5.5: Optimal folding state of the amino acid chain from figure 5.2. This is the optimal folding of the toy example presented. It can be seen that the hydrophobic aminoacids are nicely packed in the interior of the protein and the hydrophilic ones more or less remain on the surface.

To end up the example we present the optimal fold of this chain in Figure 5.5. And the correspondent contact matrix in Figure 5.6.

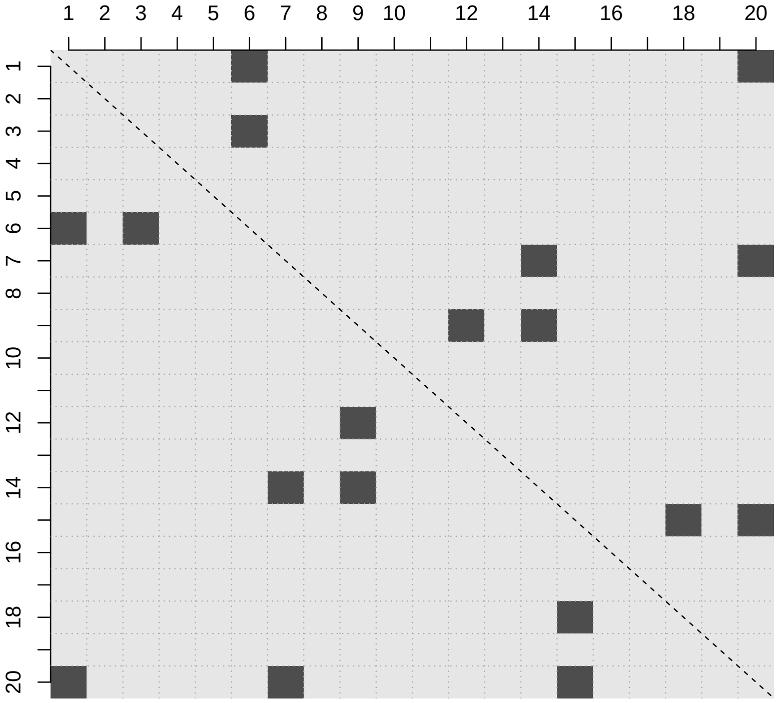


Figure 5.6: Contact matrix for the optimal folding. This is how the contact matrix of the optimal folding of this example looks like. We see that a total of 9 bounds are formed.

By counting the number of ones above the diagonal it is possible to see that the number of contacts established is 9. This is the result we would expect from the optimization problem. This is only a toy example, but it helps to exemplify that given the initial conditions, the native conformation of a protein can be easily found. For example, with a convex optimization problem as we showed. In our statement of the problem the initial conditions can be included as cost biases. This means that instead of having a uniform distributed cost matrix we can make some positions more desirable. The point is to be able to map the physical conditions in the cell to a correct conformation of the cost matrix. This is no easy task and we still have a lot to learn and to experiment before we can do it. Nevertheless, we can start playing with the cost matrix. There are many proteins for which the native structure is known. Or we can also try using homologous models since we know that many proteins have a similar native conformation. As a matter of fact the most successful protein folding prediction tools from the last years are based on this principle.

5.3 Discussion

According to our hypothesis the main player in protein folding is the starting point of the folding process. The result of the folding process is dictated by the amino acid code, but this code follows always the same steps hence, the point where it ends up depends on the point where it starts. It is like an origami process where the person doing the folding is not an expert so he is just following the steps from a tutorial. Because he simply repeats a set of steps, the result depends more on the starting conditions than on the steps themselves. In the same way, nature has designed an amino acid code which follows the laws of physics always in the same manner. So the living organisms just need to take care that the initial conditions correspond to the desired result.

If we manage to understand this process and model it, we will be able to simulate the protein folding process in nature under several conditions, for several proteins, and even for different amino acid codes. The idea of creating artificial proteins is not new, but it has not been too successful. That is because, we cannot correctly predict how they will fold in the in-vivo situation. Once we are capable of simulating the starting conditions inside the body, we should be able to predict the right fold.

Advances are needed in different areas. On one hand, we have to do wet lab experiments to understand which are the parameters that determine the initial conditions. We know that temperature, pH, and free space are some of them, but there are probably many others. On the other hand, we have to keep creating better models that map the amino acid code to the folded protein under certain conditions. In this regard we have a long way to go. We want to test the algorithm on real protein sequences, starting with globular proteins, and relate our work to the work of Professor Baldi from the University of California, Irvine [68, 73, 74].

We have already conducted some tests in a small globular protein, the crambine. This protein is only about 50 aminoacids long and its tertiary structure is known with high resolution. We could verify that if we run the algorithm without including any further information the result is acceptable, but still far from the optimum. Whereas we include some information in the cost matrix knowing the contacts that should be close in the native conformation, then the result significantly resembles the real contact matrix of the protein. This result gives us the confidence that our claim about the protein folding process in nature is correct and that we can apply this concept not only to our algorithm but in general to any protein folding prediction tool available.

The work from Professor Baldi is of most interest to us because he also seeks to predict the contact map of a protein given only its aminoacid chain. In the lab of Professor Baldi the most recent method to do the prediction is using deep learning algorithms. They train the neuronal networks to learn the patterns of the contact maps which correspond to a certain aminoacid chain. I believe that our claim about the starting point of the process could be nicely included in the research of Professor Baldi. There is a strong tendency to include bias content into neuronal networks in order to improve their performance. We believe that protein folding is an exceptional opportunity to do such thing. If we manage to understand the physical conditions inside the cell at the moment of folding the proteins, then we can try to include certain bias in the artificial neuronal network of Professor Baldi and see if the performance is improved.

Even though this project is far from being complete and we were only able to test it on toy examples, it is a very interesting idea which I would be happy to pursuit further and see what else can we learn from it. The mental exercise of imagining what might be going on inside the cell is a fascinating experiment and it is thrilling to discuss and evaluate our hypothesis awaiting to find out whether its true or not. Through this project we were able to explore all kinds of theories. From the continuum hypothesis to the NP vs P discussion, passing through the Kolmogorov complexity, the halting problem, the Gödel incompleteness theorem, the Chaitin omega number, and the maxcut algorithm from Goemans-Williamson. This learning process was incredibly satisfactory and will always remain with me.

To conclude on the topic of theoretical contributions, it is fair to say that these might be the hardest projects and the most uncertain ones. There are no guarantees of success in the field of theory. Although it can be argued that in the experimental field there are also no guarantees, at least when one performs experimental research there is data as concrete results that testifies the work done. As in theoretical research, all work might turn out to be a waist of time and at the end it can happen that no tangible contribution is made. It is, nevertheless, a most satisfactory enterprise. And if it turns out well, then it is probable that the impact of the research will be great and prolonged. We see that many mathematical conjectures which were presented a long time ago are still in use, and might be even more relevant today as when they first appear. As I mentioned before, biology is in much need of a theoretical framework so each scientist willing to contribute in this direction is most welcome.

6

General discussion

Because I already discussed and concluded about each of the projects presented on the respective chapter, then there is not much left to say. Nevertheless, I present some conclusions regarding the role of Bioinformatics in research in general and specifically in the area of neurobiology. Beside from that, even though I am not an expert in the area, I want to make some comments about the PhD system, its relevance, and future perspectives. These comments are inspired from my personal experience as PhD candidate in the United states first and then here in Germany. They are not scientific facts, but my humble opinion.

As I mentioned before science is our only hope for survival. It is true that some of the biggest challenges that we face today were originated by technology, like climate change, antibiotic resistance or monoculture farming. It is however also true that the only way to solve those problems is technology, and technology can only be obtained by scientific research. Some people claim for a retreat from modern life back to the preindustrial times, where most of the population are farmers, there is very moderate global contact, and only local products are consumed. This is a highly unrealistic scenario. We live in a society of knowledge and expecting the world to move back to the eighteen century makes no sense. If we wait patiently for this to happen we will passively witness the destruction of society either by means of global warming, an epidemy, or hunger. Only through technology we will acquire the means to stop bacterial resistance, develop sustainable farming, and reverse global warming. These are some broad challenges but there are some others which are more specific. Within the fields of neurobiology and neuroscience there are some well known challenges.

In these areas is necessary to point research towards four fronts. The first is to organize the data available. We have to pair data from different species and from different origins. The second front is toward development of software and hardware capable of modeling the complex systems in the brain. The third is to give a practical use to the knowledge we have. Diseases have to be understood and classified. Appropriate treatments must be designed and drug discovery for those diseases must be fostered. Finally, the fourth front is the development of brain-inspired software and hardware. Deep Learning is an important advance in algorithms inspired by the brain, specifically by the visual cortex, but there is still a long way to go before we have a computer as powerful and efficient as the brain.

The first front of work, the organization of data available, is of vital importance. Only by having a structured form of saving data it will be possible to establish fluent collaborations between researchers. If there is something we have learned from engineering is that standardization plays a key role in development. This is why it is so important to organize and classify the data available and the data we will be producing. If we are able to unify formats, then the information available will be readable for everyone. We also need to understand the relation between biological mechanisms in the brains of different species. A lot of research is done in other species (due to the impossibility of doing many experiments in humans), which is why we need to clearly understand the relation of those mechanisms in other species with the human brain which is our real interest.

In the second front, there is a lot of ongoing research. For example, the biological modeling research that we presented in chapter 4 about the TNF receptor. Better programming languages are being designed with the capacity to model high-complex systems like the human brain. These modeling tools need to be able to represent more than just the connectivity in the brain. We are aware that the complex regulatory systems of electrical and chemical synapses as well as the spontaneous local brain activity play determinant roles in the brain functioning. Therefore, we need to create models that also contain such information and not simply represent neurons as ON/OFF switches. In terms of hardware, big advances have been made in recent years and even more important advances might come in the near future. We count now with GPU's (Graphical Processing Units) which are capable of computing in parallel a characteristic very useful for algorithms like convolution networks. This parallel computing characteristic might present a super development in the near future if the quantum computers become feasible. Feasible in terms of availability to all researchers. Quantum computers would also bring the advantage of opening the door to new algorithm development, algorithms which would not work in normal computers. From our discussion in chapter 5 we learned that not even quantum computers will be capable of solving NP-hard problems but they will definitely be capable of solving more NP problems, and even speed up the solution of some P problems.

Now we come to the third research front, translational research in brain diseases. Citing a paper from Henry Markram [75], the director of the Blue Brain Project. I want to emphasis the importance of translational research in order to bring the knowledge we have into action against the diseases that affect the human brain. It is a fact that one out of three people have suffered, currently suffer or will suffer of a brain related disease at some point in his life. European health care system spends around 800 billion euros per year in brain disease treatments. This accounts for a 25 % of the direct cost of the health care system. In response to these problems neuroscience research has been experiencing a boom since the last 10 to 15 years. The number of publications in the area and the amount of research groups have been constantly raising during this period. The problem is that this blooming has not been translated into clinical research. The elevated cost of neurobiology research discourages companies from developing new drugs. Most of the cost is due to the failure in the third clinical stage of the drugs in development. Our shallow understanding of the brain doesn't allow us to find drug targets which are specific enough. When the clinical trials are done the drugs often severely affect other areas and functions of the brain so they have to be rejected. This constant failure has caused that we keep using drugs that were developed years ago, in the 90's, 80's, and in some cases even 50's. And even worse is that these drugs which are being used are of course no better than the ones that are being developed now and which fail the clinical trials. Just that these old drugs were approved in a time when regulation was not so strict and so they remain in use still today. The truth is that many of those drugs are harmful. All of them have important side effects in the brain and almost none of them are curative medicines. Brain diseases are often misdiagnosed, for example in the work of Beach et al. [76] it was shown that 20% of Alzheimer's disease patients were misdiagnosed. This occurs because we do not have reliable biological markers for brain diseases. The diagnose is based mostly in behavioral and cognitive symptoms, not like for diseases like cancer or vascular problems. The reason why we do not have such biological markers is because we do not understand the signaling cascades in the brain. We have a broad knowledge of how things work in the brain, but we lack details and drug candidates are in the details. This lack of markers also means that brain diseases are often diagnosed too late when the damage to the brain is irreversible. Because of all these reasons, it is so important that neurobiology and neuroscience research are fostered and specifically directed to the discovery of new drugs and treatments. It is also true that clinical research is the biggest research field because of the financial capacity of the pharmaceutical industry and the government funding for health issues. Pointing neurobiolgy research in this direction ensures also the funding of it.

Finally, the last challenge in neurobiology and neuroscience research is to be able to design bioinspired algorithms and hardware that propel other sciences and improve human lives. As mentioned before, the field of algorithm development presents some good examples with evolutionary genetic algorithms and deep learning. In the particular case of deep learning, there is still much we can learn from the brain. For example, it would be nice to be able to include the attention feature of the brain into the algorithms. When we look at the world we see in high resolution only some five degrees from our about 130 degree field of view. This is because we have a field of attention and only what lies within this field is considered relevant and it is stored in the memory. The rest is scanned by the visual cortex but not further stored. Attempts to include this attention feature into deep learning algorithm have already been made and they show some improvement, but it is still highly heuristically done and not so robust.

Another important feature from the brain that could inspire an improvement of deep learning algorithms is the hierarchical time processing. The brain clearly organizes the events that it witnesses in a hierarchical structure, being able to reconstruct the sequence of events, which probably is a huge advantage at the time of doing predictions of the time to come. This time hierarchical feature is not present in deep learning algorithms. They have a structure which correctly represents spatial hierarchy, but the time dimension is not included. Another big difference between deep learning architectures and brain functioning is that the brain architecture is multipurpose. One are in the brain is capable of doing tasks originally assigned to other areas. It has been shown that people with vision impairment can use an electrode set in the tongue and "see" with it. Probably, the most amazing example is the frontal cortex itself. This part of the brain is capable of replacing almost any other area from the brain. With the frontal cortex we can do things as different as multiply numbers and play table tennis. Of course we are far far away form an algorithm that resembles the frontal cortex. The frontal cortex is the pinnacle of evolution and it should take some time before we can emulate such an amazing system.

I believe that each of the projects presented in this doctoral thesis is a substantial contribution to the bioinformatics field in general and to the application of bioinformatic tools in neurobiology. The calcium activity detection tool from chapter two allows the researcher to find and analyze local spontaneous activity in an unbiased form. This capacity to objectively detect spontaneous local activity will foster research in non-spike like neuronal activity and will change the paradigm of the spiking synapse as the only form of relevant neuronal interaction. The tool can also be used in other fields of research. In fact, another research group at the university of Wuerzburg is using it to describe and analyze the movement of Cilia cells in the lungs.

Then there is the neuron segmentation project in chapter 3, which is clearly a great advantage for researchers since labeling the electron microscopy images by hand is a futile and unreliable task. As machines take the place of humans in tasks like this, the results can be objectively evaluated and the researcher can focus on the creative task of interpreting results and developing theories and models. Just like the tool before this algorithm has the capacity to be exported to other tasks. It can be used to label other types of images, not just electron microscopy, and it can label other things, not just neurons. I believe that this project will allow me to keep working on image processing applications of all kinds.

In chapter 4, the TNF binding model was presented. This project is of great relevance because of the above mentioned importance of models in biology and specifically because of the key role that the TNF ligand plays in inflammation processes in the body. Inflammation is just as important as dangerous which is why it is determinant to understand it. Having a realistic informative model from it allows researchers to easily evaluate hypothesis and design wet lab experiments. The rule-based modeling used in this project is also universal and can be applied to all kinds of biological systems.

Finally, the protein folding model is a theoretical contribution which not only addresses the problem of protein folding itself from the mathematical and biological point of view, but it also seeks to inspire other scientists to do interdisciplinary science, to look for contributions from other disciplines in order to solve problems in their own area. Protein folding has been since a long time studied from a mathematical and a biological point of view and it has been a profitable enterprise for both areas. We wanted to participate in this dialogue because as bioinformaticians we are somehow a connecting bridge between these two sciences. Our view of the protein folding problem as a matter of setting the appropriate starting conditions puts this fundamental problem into a new framework. This view was inspired by non-convex optimization and differential equations. In both cases the initial conditions determine the final outcome of the problem. The truth is that this project is on a very early stage and most of the work is still missing. Nevertheless I am confident that the toy examples we used to examine the hypothesis serve as a proof of concept and that eventually we will be able to take this project to a practical application phase. Discarding the idea of nature solving NP-complete problems is very important for both biologists and mathematicians. Biologists could learn a lot once we understand nature mechanisms as a form of computation. And Mathematicians would be glad to hear that NP-complete problems cannot be efficiently solved even by nature after millions of years of working on them.

6.1 Conclusions

In conclusion my doctoral thesis presents the following contributions:

- We designed and developed a tool to assist researchers in the detection and analysis of calcium activity recordings of neuron cultures[1].
- With the help of this tool, we controverted the paradigm of the calcium spiking as the unique relevant calcium event in the brain. We focused the attention in spontaneous local calcium events.
- We created a script which is capable of automatically segmenting neuron nuclei from electron microscopy images without any input from the user beside the raw images. The deep learning approach used reaches an error bellow 8%.
- On a second phase of the neuron nuclei segmentation project, we will improve the performance of the algorithm and create a post-processing stage which allows us to fully mark the neurons from a whole organism.
- We created an *in silico* model of the TNF receptor system which enables researchers to study inflammatory processes in biological cells [2].

---

[1] First author paper, currently in revision for PLOS Comp Biol

[2] First author manuscript. In collaboration with Prof. Wajant

- Our study of the inflammatory process in the cell could be highly relevant for the study of several diseases, cancer for example, among others.
- We proposed a novel view regarding the protein folding process in living organisms. We hypothesized that the aminoacid code carries instructions for a simple convex optimization process and that the actual delicate point of the process is the initial state of the whole system (coil conformation and intracellular environment)[1].
- With the study of the starting conformation of the protein folding process, new insights into amyloid related diseases are possible. If we understand better the system conformations that lead to misfolded proteins and their posterior aggregation, then it will be easier to create treatments and drugs which help to avoid such conformations.

6.2 Discussion remarks on the PhD thesis system

I am very content with my doctoral thesis. I had the opportunity to be funded by the German government through the Excellence Initiative program and I believe they have spent their money well. My doctoral thesis not only contributes to science in general, but it also directly contributes to other research groups here in the University of Wuerzburg as in other parts of Germany. It has been a fulfilling experience to do this PhD and I have nothing to regret about it. Nevertheless, along the way I have learned a lot about the world of PhD studies and I consider that the whole system needs to be reevaluated. This is not just my opinion, there are plenty of experts which have been arguing for a change for already some time now. Some of the critics and concerns of researchers were summarized in the Nature journal, volume 472 number 7343. In this edition the PhD system situation was discussed with different experts like Raymond Gosling and Mark Taylor.

The problem is paradoxical. We need the PhD candidates because of their valuable contribution to research but so many candidates results in too many people holding a PhD title. We have a system which is driven by the supply of research funding, but not by the job market demand. PhD candidates are low-paid high-qualified workers, which is great for science but not so great for the candidates themselves. PhD candidates

---

[1] First author paper, to be submitted to PLOS Comp Biol - Protein Folding

under the current situation are really only trained for one job, the academic path. The problem is that the academic market is saturated. Thirty years ago, 55% of the doctorates ended working on a full time permanent position in academia, whereas by 2012 only 15% could achieve that [77]. Germany is one of the few countries doing things right. In an article called ”The PhD Factory” from the above named issue of Nature, it is explained that contrary to other countries, Germany has maintained the number of PhD positions and by implementing the system of graduate schools they have managed to diversify the career options for their doctorates. They also implemented the transferable skills courses which help to prepare students for other fields in industry beside academia. I hope for the sake of science that the PhD system can remain as fruitful as it is, but I also hope for that the system can be improved for the benefit of the PhD candidates to come.

Bibliography

[1] Yoshihiro Matsuoka, Yves Vigouroux, Major M. Goodman, Jesus Sanchez G., Edward Buckler, and John Doebley. **A single domestication for maize shown by multilocus microsatellite genotyping**. *Proceedings of the National Academy of Sciences*, **99**(9):6080–6084, 2002. 1

[2] S. Psillos and M. Curd. *The Routledge Companion to Philosophy of Science*. Routledge, 2008. 2, 4

[3] D. E. Clapham. **Calcium signaling**. *Cell*, **131**(6):1047–58, 2007. 13, 41

[4] M. J. Berridge, M. D. Bootman, and H. L. Roderick. **Calcium signalling: dynamics, homeostasis and remodelling**. *Nat Rev Mol Cell Biol*, **4**(7):517–29, 2003. 13, 41

[5] E. Neher and T. Sakaba. **Multiple roles of calcium ions in the regulation of neurotransmitter release**. *Neuron*, **59**(6):861–72, 2008. 13

[6] J. Soboloff, B. S. Rothberg, M. Madesh, and D. L. Gill. **STIM proteins: dynamic calcium signal transducers**. *Nat Rev Mol Cell Biol*, **13**(9):549–65, 2012. 13, 41

[7] D. De Stefani, R. Rizzuto, and T. Pozzan. **Enjoy the Trip: Calcium in Mitochondria Back and Forth**. *Annu Rev Biochem*, **85**:161–92, 2016. 13

[8] C. Grienberger and A. Konnerth. **Imaging calcium in neurons**. *Neuron*, **73**(5):862–85, 2012. 14

[9] R. Y. Tsien. **A non-disruptive technique for loading calcium buffers and indicators into cells**. *Nature*, **290**(5806):527–8, 1981. 14

[10] E. A. Mukamel, A. Nimmerjahn, and M. J. Schnitzer. **Automated analysis of cellular signals from large-scale calcium imaging data**. *Neuron*, **63**(6):747–60, 2009. 14, 38

[11] T. P. Patel, K. Man, B. L. Firestein, and D. F. Meaney. **Automated quantification of neuronal networks and single-cell calcium dynamics using calcium imaging**. *J Neurosci Methods*, **243**:26–38, 2015. 14, 38

[12] E. A. Pnevmatikakis, K. Kelleher, R. Chen, P. Saggau, K. Josic, and L. Paninski. **Fast spatiotemporal smoothing of calcium measurements in dendritic trees**. *PLoS Comput Biol*, **8**(6):e1002569, 2012. 14

[13] E. A. Pnevmatikakis, D. Soudry, Y. Gao, T. A. Machado, J. Merel, D. Pfau, T. Reardon, Y. Mu, C. Lacefield, W. Yang, M. Ahrens, R. Bruno, T. M. Jessell, D. S. Peterka, R. Yuste, and L. Paninski. **Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data**. *Neuron*, **89**(2):285–99, 2016. 14, 38

[14] T. Sasaki, N. Takahashi, N. Matsuki, and Y. Ikegaya. **Fast and accurate detection of action potentials from somatic calcium fluctuations**. *J Neurophysiol*, **100**(3):1668–76, 2008. 14

[15] R. Janicek, M. Hotka, Jr. Zahradnikova, A., A. Zahradnikova, and I. Zahradnik. **Quantitative analysis of calcium spikes in noisy fluorescent background**. *PLoS One*, **8**(5):e64394, 2013. 14, 38

[16] W. Q. Malik, J. Schummers, M. Sur, and E. N. Brown. **Denoising two-photon calcium imaging data**. *PLoS One*, **6**(6):e20490, 2011. 14

[17] L. N. Borodinsky, C. M. Root, J. A. Cronin, S. B. Sann, X. Gu, and N. C. Spitzer. **Activity-dependent homeostatic specification of transmitter expression in embryonic neurons**. *Nature*, **429**(6991):523–30, 2004. 14

[18] N. C. Spitzer. **Electrical activity in early neuronal development**. *Nature*, **444**(7120):707–12, 2006. 14, 41

[19] N. Subramanian, A. Wetzel, B. Dombert, P. Yadav, S. Havlicek, S. Jablonka, M. A. Nassar, R. Blum, and M. Sendtner. **Role of Na(v)1.9 in activity-dependent axon growth in motoneurons**. *Hum Mol Genet*, **21**(16):3655–67, 2012. 14, 41

[20] A. Wetzel, S. Jablonka, and R. Blum. **Cell-autonomous axon growth of young motoneurons is triggered by a voltage-gated sodium channel**. *Channels (Austin)*, **7**(1):51–6, 2013. 14, 41

[21] C. Lohmann, A. Finski, and T. Bonhoeffer. **Local calcium transients regulate the spontaneous motility of dendritic filopodia**. *Nat Neurosci*, **8**(3):305–12, 2005. 14

[22] J. M. Gregoire, D. Dale, and R. B. van Dover. **A wavelet transform algorithm for peak detection and application to powder x-ray diffraction data**. *Rev Sci Instrum*, **82**(1):015105, 2011. 14

[23] P. Du, W. A. Kibbe, and S. M. Lin. **Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching**. *Bioinformatics*, **22**(17):2059–65, 2006. 14, 22

[24] S. Wiese, T. Herrmann, C. Drepper, S. Jablonka, N. Funk, A. Klausmeyer, M. L. Rogers, R. Rush, and M. Sendtner. **Isolation and enrichment of embryonic mouse motoneurons from the lumbar spinal cord of individual mouse embryos**. *Nat Protoc*, **5**(1):31–8, 2010. 16

[25] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0. 17

[26] Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, Jean-Yves Tinevez, Daniel James White, Volker Hartenstein, Kevin Eliceiri, Pavel Tomancak, and Albert Cardona. **Fiji: an open-source platform for biological-image analysis**. *Nat Meth*, **9**(7):676–682, 2012. 17

[27] Stphane Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 3rd edition, 2008. 18

[28] W. Constantine and D. Percival. *wmtsa: Wavelet Methods for Time Series Analysis*, 2016. R package version 2.0-2. 22

[29] R. Blum, C. Heinrich, R. Sanchez, A. Lepier, E. D. Gundelfinger, B. Berninger, and M. Gotz. **Neuronal network formation from reprogrammed early postnatal rat cortical glial cells**. *Cereb Cortex*, **21**(2):413–24, 2011. 27

[30] F. v Wegner, M. Both, and R. H. Fink. **Automated detection of elementary calcium release events using the a trous wavelet transform**. *Biophys J*, **90**(6):2151–63, 2006. 38

[31] K. L. Ellefsen, B. Settle, I. Parker, and I. F. Smith. **An algorithm for automated detection, localization and measurement of local calcium signals from camera-based imaging**. *Cell Calcium*, **56**(3):147–56, 2014. 38

[32] N. C. Spitzer, N. J. Lautermilch, R. D. Smith, and T. M. Gomez. **Coding of neuronal differentiation by calcium transients**. *Bioessays*, **22**(9):811–7, 2000. 41

[33] G. Turrigiano. **Homeostatic synaptic plasticity: local and global mechanisms for stabilizing neuronal function**. *Cold Spring Harb Perspect Biol*, **4**(1):a005736, 2012. 41, 42

[34] P. Valnegri, S. V. Puram, and A. Bonni. **Regulation of dendrite morphogenesis by extrinsic cues**. *Trends Neurosci*, **38**(7):439–47, 2015. 41

[35] G. J. Augustine, F. Santamaria, and K. Tanaka. **Local calcium signaling in neurons**. *Neuron*, **40**(2):331–46, 2003. 41

[36] M. Sendtner. **Therapy development in spinal muscular atrophy**. *Nat Neurosci*, **13**(7):795–9, 2010. 41

[37] A. E. Renton, A. Chio, and B. J. Traynor. **State of play in amyotrophic lateral sclerosis genetics**. *Nat Neurosci*, **17**(1):17–23, 2014. 41

[38] D. M. Iascone, C. E. Henderson, and J. C. Lee. **Spinal muscular atrophy: from tissue specificity to therapeutic strategies**. *F1000Prime Rep*, **7**:04, 2015. 41

[39] S. Jablonka, M. Beck, B. D. Lechner, C. Mayer, and M. Sendtner. **Defective Ca2+ channel clustering in axon terminals disturbs**

excitability in motoneurons in spinal muscular atrophy. *J Cell Biol*, **179**(1):139–49, 2007. 41

[40] E. Leipold, L. Liebmann, G. C. Korenke, T. Heinrich, S. Giesselmann, J. Baets, M. Ebbinghaus, R. O. Goral, T. Stodberg, J. C. Hennings, M. Bergmann, J. Altmuller, H. Thiele, A. Wetzel, P. Nurnberg, V. Timmerman, P. De Jonghe, R. Blum, H. G. Schaible, J. Weis, S. H. Heinemann, C. A. Hubner, and I. Kurth. **A de novo gain-of-function mutation in SCN11A causes loss of pain perception**. *Nat Genet*, **45**(11):1399–404, 2013. 41

[41] U. R. Monani. **Spinal muscular atrophy: a deficiency in a ubiquitous protein; a motor neuron-specific disease**. *Neuron*, **48**(6):885–96, 2005. 41

[42] G. Buzsaki and A. Draguhn. **Neuronal oscillations in cortical networks**. *Science*, **304**(5679):1926–9, 2004. 41

[43] A. Draguhn, M. Keller, and S. Reichinnek. **Coordinated network activity in the hippocampus**. *Front Neurol Neurosci*, **34**:26–35, 2014. 41

[44] J. Hartmann, R. M. Karl, R. P. Alexander, H. Adelsberger, M. S. Brill, C. Ruhlmann, A. Ansel, K. Sakimura, Y. Baba, T. Kurosaki, T. Misgeld, and A. Konnerth. **STIM1 controls neuronal Ca(2)(+) signaling, mGluR1-dependent synaptic transmission, and cerebellar motor behavior**. *Neuron*, **82**(3):635–44, 2014. 42

[45] J. Lalonde, G. Saia, and G. Gill. **Store-operated calcium entry promotes the degradation of the transcription factor Sp4 in resting neurons**. *Sci Signal*, **7**(328):ra51, 2014. 42

[46] S. Samtleben, B. Wachter, and R. Blum. **Store-operated calcium entry compensates fast ER calcium loss in resting hippocampal neurons**. *Cell Calcium*, **58**(2):147–59, 2015. 42

[47] Geofrey Hinton, Simon Osindero, and Yee-Whye Teh. **A fast learning algorithm for deep belief nets**. *Neural Computation*, 2006. 43

[48] J.G. White, E. Southgate, J. N. Thomson, and S. Brenner. **The structure of the nervous system of the nematode C. elegans**. *Philosophical transactions Royal Society London*, **314**:1–340, 1986. 44

[49] Philippe Rostaing, Robby M. Weimer, Erik M. Jorgensen, Antoine Triller, and Jean-Louis Bessereau. **Preservation of Immunoreactivity and Fine Structure of Adult C. elegans Tissues Using High-pressure Freezing**. *Journal of Histochemistry & Cytochemistry*, **52**(1):1–12, 2004. PMID: 14688212. 45

[50] Lily A Chylek, Leonard A Harris, James R Faeder, and William S Hlavacek. **Modeling for (physical) biologists: an introduction to the rule-based approach**. *Physical Biology*, **12**(4):045007, 2015. 64

[51] James R. Faeder, Michael L. Blinov, and William S. Hlavacek. *Rule-Based Modeling of Biochemical Systems with BioNetGen*, pages 113–167. Humana Press, Totowa, NJ, 2009. 64

[52] Richard M. Locksley, Nigel Killeen, and Michael J. Lenardo. **The TNF and TNF Receptor Superfamilies**. *Cell*, **104**(4):487–501, 2001. 65

[53] H. Wajant. **Principles of antibody-mediated TNF receptor activation**. *Cell Death Differ*, **22**(11):1727–41, 2015. 65

[54] Yohei Mukai, Teruya Nakamura, Mai Yoshikawa, Yasuo Yoshioka, Shin-ichi Tsunoda, Shinsaku Nakagawa, Yuriko Yamagata, and Yasuo Tsutsumi. **Solution of the Structure of the TNF-TNFR2 Complex**. *Science Signaling*, **3**(148):ra83–ra83, 2010. 65

[55] David W. Banner, Allan D'Arcy, Wolfgang Janes, Reiner Gentz, Hans-Joachim Schoenfeld, Clemens Broger, Hansruedi Loetscher, and Werner Lesslauer. **Crystal structure of the soluble human 55 kd TNF receptor-human TNFbeta; complex: Implications for TNF receptor activation**. *Cell*, **73**(3):431–445, 1993. 65

[56] James H. Naismith, Tracey Q. Devine, Barbara J. Brandhuber, and Stephen R. Sprang. **Crystallographic Evidence for Dimerization of Unliganded Tumor Necrosis Factor Receptor**. *Journal of Biological Chemistry*, **270**(22):13303–13307, 1995. 65

[57] Francis Ka-Ming Chan, Hyung J. Chun, Lixin Zheng, Richard M. Siegel, Kimmie L. Bui, and Michael J. Lenardo. **A Domain in TNF Receptors That Mediates Ligand-Independent Receptor Assembly and Signaling**. *Science*, **288**(5475):2351–2354, 2000. 65, 68

[58] Richard M. Siegel, John K. Frederiksen, David A. Zacharias, Francis Ka-Ming Chan, Michele Johnson, David Lynch, Roger Y. Tsien, and Michael J. Lenardo. **Fas Preassociation Required for Apoptosis Signaling and Dominant Inhibition by Pathogenic Mutations**. *Science*, **288**(5475):2354–2357, 2000. 65, 68

[59] Lauren Clancy, Karen Mruk, Kristina Archer, Melissa Woelfel, Juthathip Mongkolsapaya, Gavin Screaton, Michael J. Lenardo, and Francis Ka-Ming Chan. **Preligand assembly domain-mediated ligand-independent association between TRAIL receptor 4 (TR4) and TR2 regulates TRAIL-induced apoptosis**. *Proceedings of the National Academy of Sciences of the United States of America*, **102**(50):18099–18104, 2005. 65

[60] Franziska Fricke, Sebastian Malkusch, Gaby Wangorsch, Johannes F. Greiner, Barbara Kaltschmidt, Christian Kaltschmidt, Darius Widera, Thomas Dandekar, and Mike Heilemann. **Quantitative single-molecule localization microscopy combined with rule-based modeling reveals ligand-induced TNF-R1 reorganization toward higher-order oligomers**. *Histochemistry and Cell Biology*, **142**(1):91–101, Jul 2014. 66

[61] Christian Winkel, Simon Neumann, Christina Surulescu, and Peter Scheurich. **A minimal mathematical model for the initial molecular interactions of death receptor signalling**. *Mathematical Biosciences and Engineering*, **9**(3):663–683, 2012. 66

[62] Verena Boschert, Anja Krippner-Heidenreich, Marcus Branschadel, Jessica Tepperink, Andrew Aird, and Peter Scheurich. **Single chain TNF derivatives with individually mutated receptor binding sites reveal differential stoichiometry of ligand receptor complex formation for TNFR1 and TNFR2**. *Cellular Signalling*, **22**(7):1088 – 1096, 2010. 68

[63] Dr. Jhon A. Olson. *Chemical Reactions: Stoichiometry and Beyond*, **1**. Cognella Academic Publishing, 1 edition, 2011. 70

[64] Adam M. Smith, Wen Xu, Yao Sun, James R. Faeder, and G. Elisabeta Marai. **RuleBender: integrated modeling, simulation and visualization for rule-based intracellular biochemistry**. *BMC Bioinformatics*, **13**(Suppl 8):S3–S3, 2012. 72

[65] G.J. Chaitin. **A note on the number of N-bit strings with maximum complexity**. *Applied Mathematics and Computation*, **59**(1):97 – 100, 1993. 91

[66] K. A. Dill, S. Bromberg, K. Yue, K. M. Fiebig, D. P. Yee, P. D. Thomas, and H. S. Chan. **Principles of protein folding–a perspective from simple exact models**. *Protein Sci*, **4**(4):561–602, 1995. 93

[67] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. **Critical assessment of methods of protein structure prediction (CASP) round x**. *Proteins*, **82**(0 2):1–6, 2014. 93

[68] J. Cheng, A. N. Tegge, and P. Baldi. **Machine Learning Methods for Protein Structure Prediction**. *IEEE Reviews in Biomedical Engineering*, **1**:41–49, 2008. 93, 103

[69] S. Walter Englander and Leland Mayne. **The nature of protein folding pathways**. *Proceedings of the National Academy of Sciences*, **111**(45):15873–15880, 2014. 94

[70] David Baker. **A surprising simplicity to protein folding**. *Nature*, **405**(6782):39–42, 2000. 94

[71] F. Ulrich Hartl, Andreas Bracher, and Manajit Hayer-Hartl. **Molecular chaperones in protein folding and proteostasis**. *Nature*, **475**(7356):324–332, 2011. 94

[72] Igor Drobnak, Ajasja Ljubeti, Helena Gradiar, Toma Pisanski, and Roman Jerala. *Designed Protein Origami*, pages 7–27. Springer International Publishing, Cham, 2016. 94

[73] Pietro Di Lena, Ken Nagata, and Pierre Baldi. **Deep architectures for protein contact map prediction**. *Bioinformatics (Oxford, England)*, **28**(19):2449–2457, 2012. Bioinformatics. 103

[74] Taeho Jo, Jie Hou, Jesse Eickholt, and Jianlin Cheng. **Improving Protein Fold Recognition by Deep Learning Networks**. *Scientific Reports*, **5**:17573, 2015. 103

[75] Henry Markram. **Seven challenges for neuroscience**. *Functional Neurology*, **28**(3):145–151, 2013. 109

[76] Thomas G. Beach, Sarah E. Monsell, Leslie E. Phillips, and Walter Kukull. **Accuracy of the Clinical Diagnosis of Alzheimer Disease at National Institute on Aging Alzheimer Disease Centers**. *Journal of Neuropathology and Experimental Neurology*, **71**(4):266–273, 2012. 109

[77] **Fix the PhD**. *Nature*, **472**(7343):259–260, 2011. 114

[78] T. Andreska, S. Aufmkolk, M. Sauer, and R. Blum. **High abundance of BDNF within glutamatergic presynapses of cultured hippocampal neurons**. *Front Cell Neurosci*, **8**:107, 2014.

[79] W. Gobel and F. Helmchen. **In vivo calcium imaging of neural network function**. *Physiology (Bethesda)*, **22**:358–65, 2007.

[80] X. Gu and N. C. Spitzer. **Distinct aspects of neuronal differentiation encoded by frequency of spontaneous Ca2+ transients**. *Nature*, **375**(6534):784–7, 1995.

[81] G. E. Hardingham, F. J. Arnold, and H. Bading. **Nuclear calcium signaling controls CREB-mediated gene expression triggered by synaptic activity**. *Nat Neurosci*, **4**(3):261–7, 2001.

[82] M. Hilario, A. Kalousis, C. Pellegrini, and M. Muller. **Processing and classification of protein mass spectra**. *Mass Spectrom Rev*, **25**(3):409–49, 2006.

[83] R. Hooper, B. S. Rothberg, and J. Soboloff. **Neuronal STIMulation at Rest**. *Sci Signal*, **7**(335):pe18, 2014.

[84] R. Maruyama, K. Maeda, H. Moroda, I. Kato, M. Inoue, H. Miyakawa, and T. Aonishi. **Detecting cells using non-negative matrix factorization on calcium imaging data**. *Neural Netw*, **55**:11–9, 2014.

[85] V. Rahmati, K. Kirmse, D. Markovic, K. Holthoff, and S. J. Kiebel. **Inferring Neuronal Dynamics from Calcium Imaging Data Using Biophysical Models and Bayesian Inference**. *PLoS Comput Biol*, **12**(2):e1004736, 2016.

[86] M. Sasi, B. Vignoli, M. Canossa, and R. Blum. **Neurobiology of local and intercellular BDNF signaling**. *Pflugers Arch*, **469**(5-6):593–610, 2017.

[87] J. Schindelin, C. T. Rueden, M. C. Hiner, and K. W. Eliceiri. **The ImageJ ecosystem: An open platform for biomedical image analysis**. *Mol Reprod Dev*, **82**(7-8):518–29, 2015.

[88] G. G. Turrigiano, K. R. Leslie, N. S. Desai, L. C. Rutherford, and S. B. Nelson. **Activity-dependent scaling of quantal amplitude in neocortical neurons [see comments]**. *Nature*, **391**(6670):892–6, 1998.

[89] G. X. Wang and M. M. Poo. **Requirement of TRPC channels in netrin-1-induced chemotropic turning of nerve growth cones**. *Nature*, **434**(7035):898–904, 2005.

[90] J. Winnubst, J. E. Cheyne, D. Niculescu, and C. Lohmann. **Spontaneous Activity Drives Local Synaptic Plasticity In Vivo**. *Neuron*, **87**(2):399–410, 2015.

[91] J. T. Lock, I. Parker, and I. F. Smith. **A comparison of fluorescent Ca2 indicators for imaging local Ca2 signals in cultured cells**. *Cell Calcium*, **58**(6):638–48, 2015.

[92] Yoshua Bengio. **Learning Deep Architectures for AI**. *Found. Trends Mach. Learn.*, **2**(1):1–127, January 2009. 53

A

Calcium Activity Tool - User manual

Neuronal Activity Assisted Assessment (NA^3) tool.

User manual 2017.1

Written by Juan Prada[1] & and Robert Blum[2].

1 Department of Bioinformatics, University of Würzburg, Würzburg, Germany.

2 Institute of Clinical Neurobiology, University hospital, University of Würzburg, Würzburg, Germany.

---

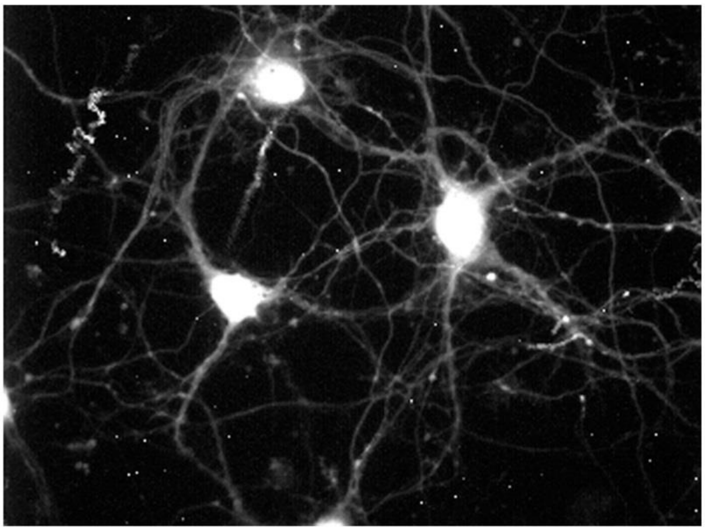
[1] 1

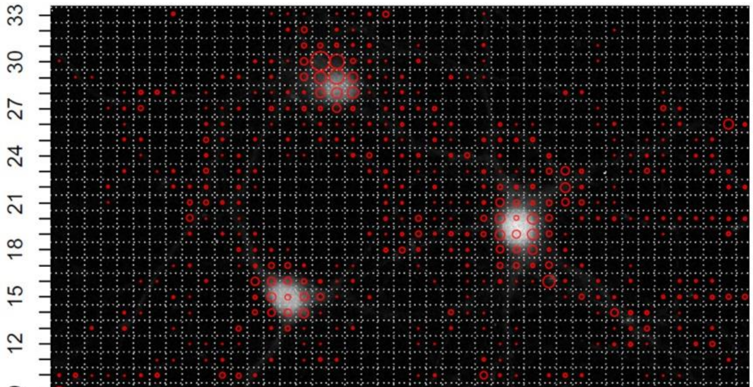
[2] 2

NA^3

Automated Calcium Signal Assessment

from x,y-t Calcium Imaging Data





General Comments

We created a bioinformatics tool to facilitate the unbiased assessment of calcium signals from x,y-t imaging raw data. The tool can be adapted for use with diverse imaging data sets in which changes in bit-values show a transient-like character.

The tool was created to support the identification of local calcium events and ‘signal-close-to-noise’ activity in neurites of neurons. However, the tool is also powerful to assess calcium spikes in neurons.

To compute calcium transients from raw image material the tool has been split in two stages. Signal extraction is computed on ImageJ and activity events are calculated on ‘R’ (https://www.r-project.org).

Both computations are embedded in the Bio7 environment, an open platform (http://bio7.org/).

To date, the software has been tested on Windows with the BIO7 2.4 for Windows 64 bits.

The system was also proved on a Linux 3.11.10-7. openSUSE 13.1 (Bottle)(x86_64) and a Mac OS X Yosemite 10.10.5.

The application was developed on a Windows X64 Intel core i-7 machine with 16 Gigabyte (GB) of RAM memory. For time series analysis, a Windows X64 Intel core i-5 machine with 4 Gigabyte of RAM memory is sufficient, but we recommend having more RAM (e.g. 8 – 16 GB).

The tool uses the signal-to-noise ratio after threshold correction to define the stringency of activity detection. Intensity signals are automatically calculated in the whole x,y-field of the image series. For this, a grid with defined pixel size separates the x,y-field in independently analyzed grid windows.

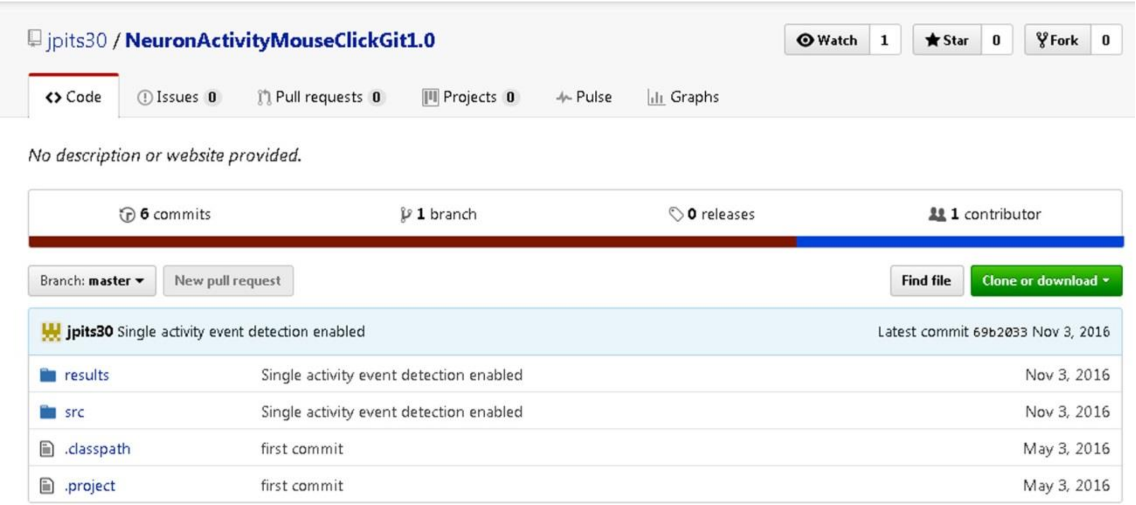
A dataset is created and shows loci of activity, individual traces of active loci, and calculates a virtual number of activity events.

I Installation

1) Download, unzip and open Bio7 following the instructions in the download section in the Bio7 website (http://bio7.org/). The 64bit version is recommended. Bio7 already includes ImageJ and R.

***NOTE:** R is already included in Bio7 for Windows. For Linux and Mac versions, R must be downloaded separately. Please follow the instructions on the Bio7 website.*

2) Download the Neuron Activity Analysis zip file from github (https://github.com/jpits30/Calcium-Activity-Tool-1.0). Unzip the file in some known location.

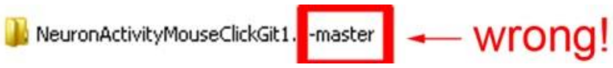


No description or website provided.

jpits30 Single activity event detection enabled — Latest commit 69b2033 Nov 3, 2016

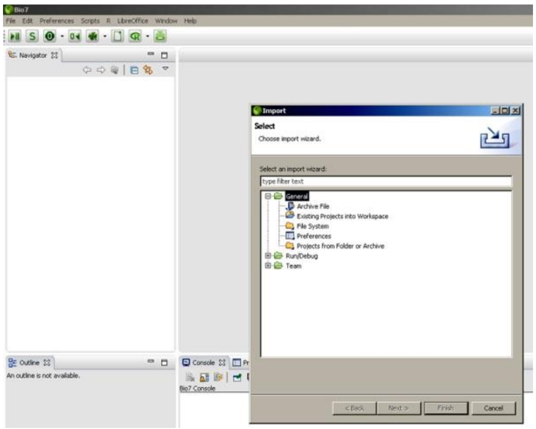
results	Single activity event detection enabled	Nov 3, 2016
src	Single activity event detection enabled	Nov 3, 2016
.classpath	first commit	May 3, 2016
.project	first commit	May 3, 2016

***NOTE:** The Zip software will probably change the name of the folder (in most cases adding "-master").*



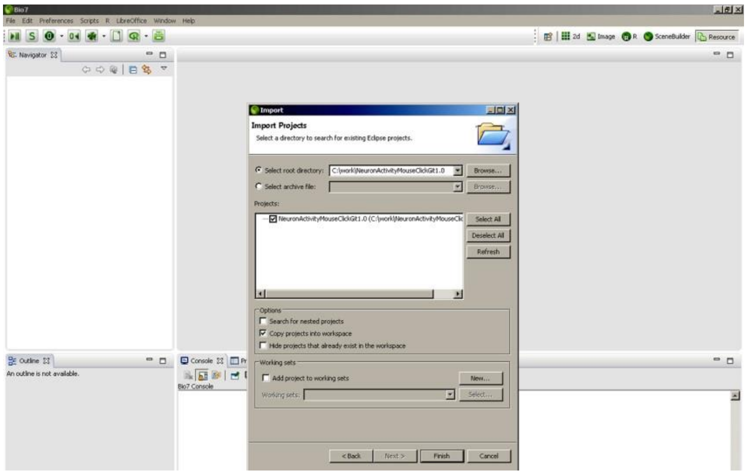
Make sure to rename the folder before moving forward. The name of the folder must be **neuronActivityMouseClickGit1.0**.

3) Open Bio7, go to the **File** menu and click **Import**.



4) Select **Existing Projects into Workspace** option then click **Next.**

5) Browse to the directory where you unzipped the file, select the **neuronActivityMouseClickGit1.0** folder (be careful because the ZIP software might have changed the name). Check the box **Copy projects into workspace** and click **Finish**.



Select a directory to search for existing Eclipse projects.

● Select root directory: C:\work\NeuronActivityMouseClickGit1.0

○ Select archive file:

Projects:

☑ NeuronActivityMouseClickGit1.0 (C:\work\NeuronActivityMouseClic

☐ Search for nested projects

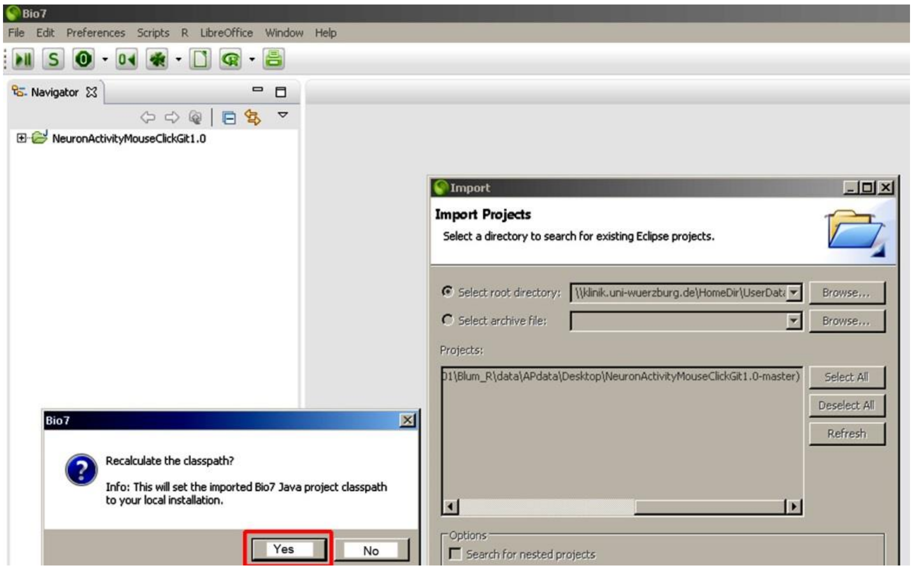
☑ Copy projects into workspace

☐ Hide projects that already exist in the workspace

☐ Add project to working sets

Working sets:

6) The program will ask if it should recalculate paths. Make sure you say **YES**.


Bio7
File Edit Preferences Scripts R LibreOffice Window Help
Navigator
NeuronActivityMouseClickGit1.0
Import
Browse...
Browse...
Select All
Deselect All
Refresh
Bio7
Yes
No

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: \\klinik.uni-wuerzburg.de\HomeDir\UserDat

○ Select archive file:

Projects:

01\Blum_R\data\APdata\Desktop\NeuronActivityMouseClickGit1.0-master)

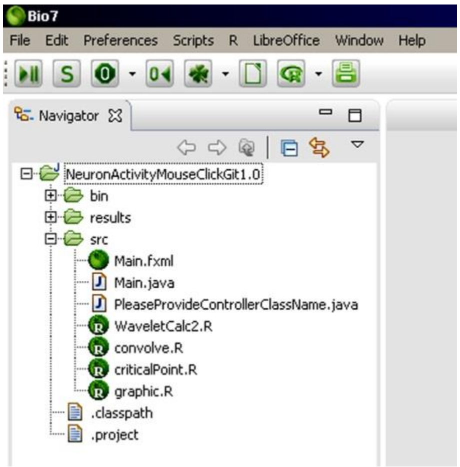
Options

☐ Search for nested projects

Recalculate the classpath?

Info: This will set the imported Bio7 Java project classpath to your local installation.

7) File structure when installation is completed.



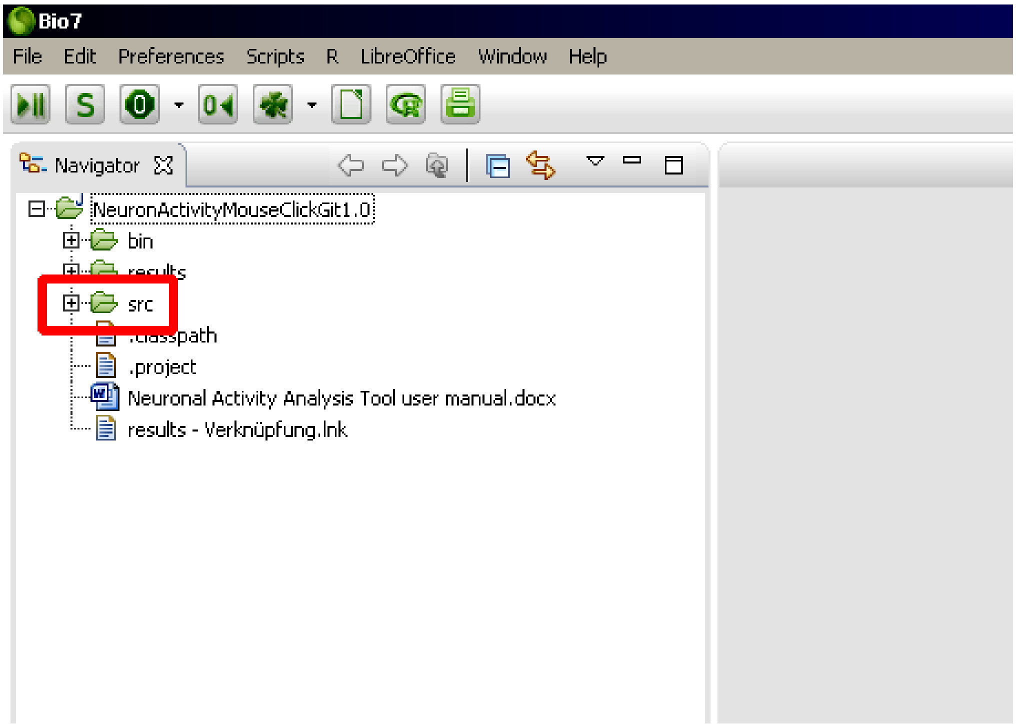
- NeuronActivityMouseClickGit1.0
  - bin
  - results
  - src
    - Main.fxml
    - Main.java
    - PleaseProvideControllerClassName.java
    - WaveletCalc2.R
    - convolve.R
    - criticalPoint.R
    - graphic.R
  - .classpath
  - .project

8) Continue to the use instructions.

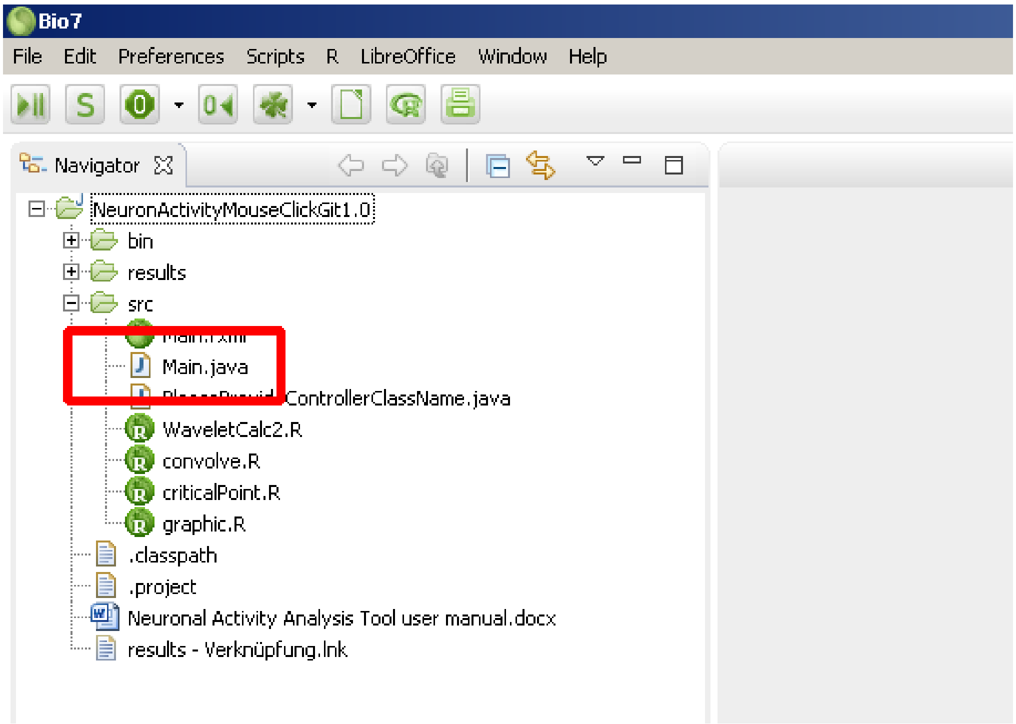
Note: When you are using the program for a first time, it may ask you to select a HTTPS CRAN mirror. Select one to complete the "R" installation in Bio7.

II Starting the program and NA^3

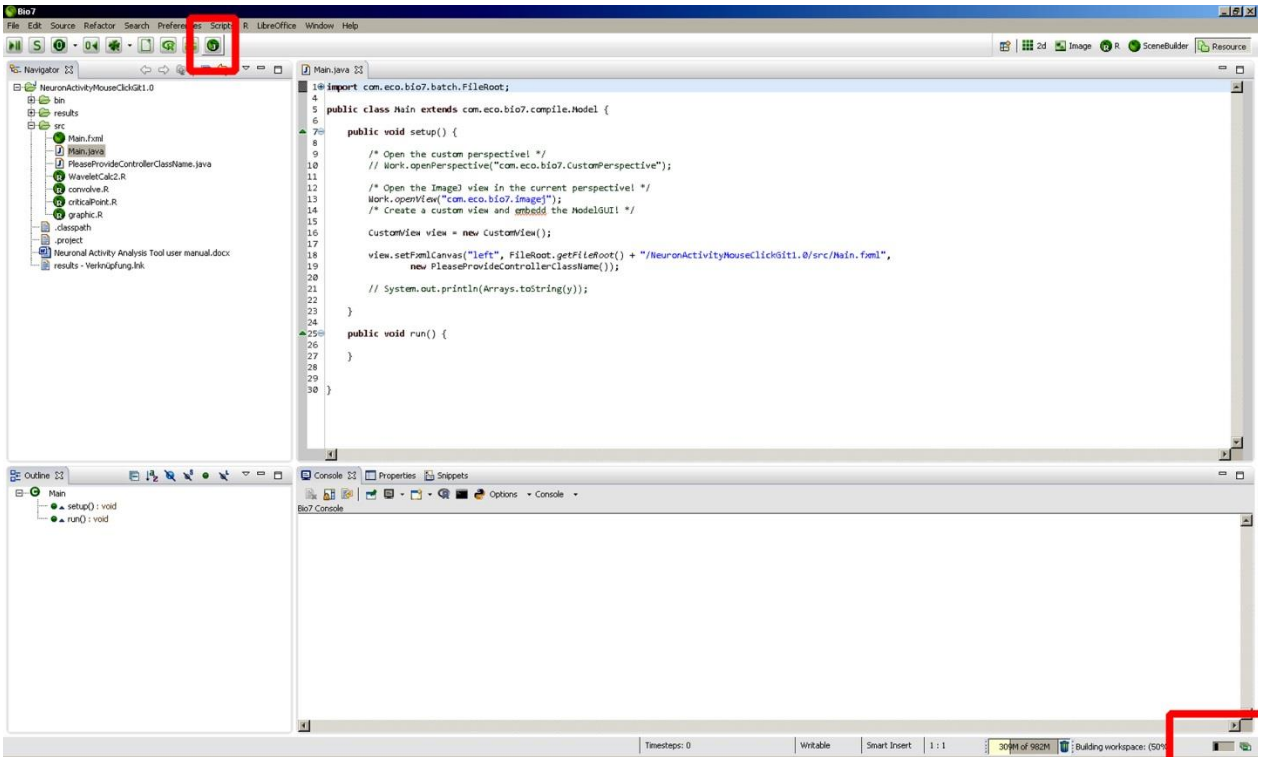
1) Open the program Bio7 and open the folder **src**.


Bio7
File Edit Preferences Scripts R LibreOffice Window Help
Navigator
NeuronActivityMouseClickGit1.0
bin
results
src
.classpath
.project
Neuronal Activity Analysis Tool user manual.docx
results - Verknüpfung.lnk

2) Open the **Main.java** file.

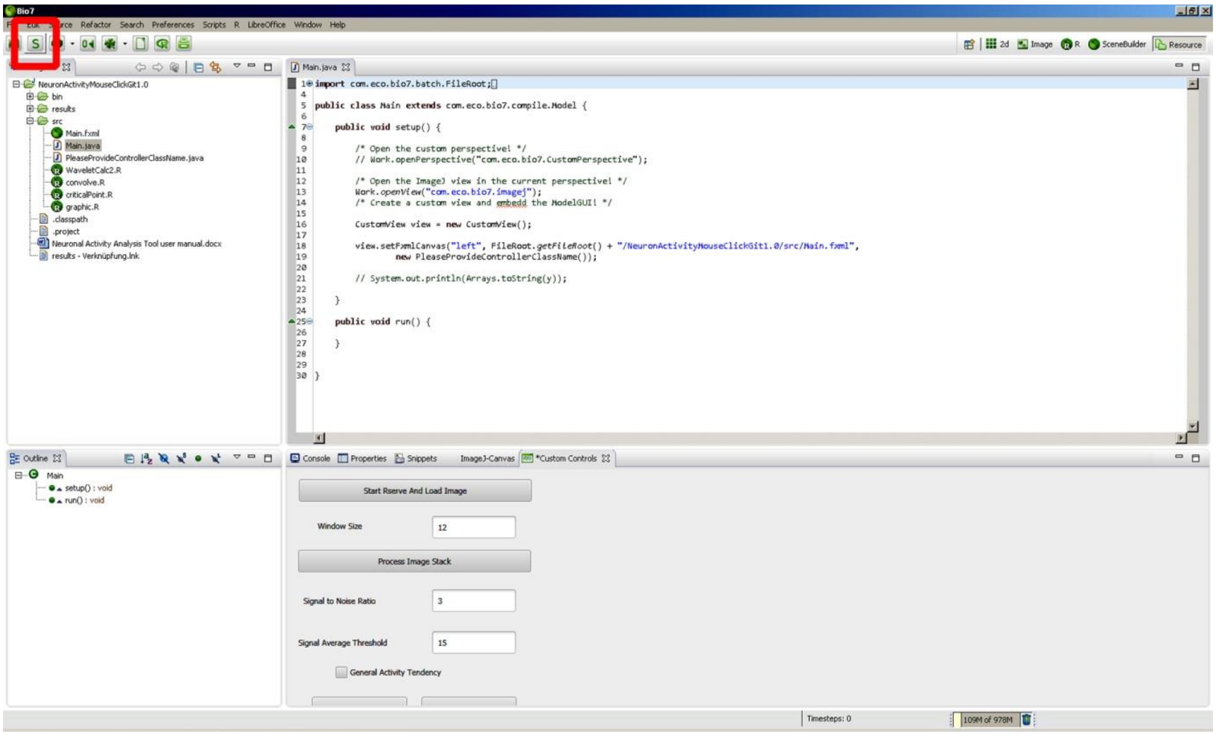

Bio7
File Edit Preferences Scripts R LibreOffice Window Help
Navigator
NeuronActivityMouseClickGit1.0
bin
results
src
Main.fxml
Main.java
ControllerClassName.java
WaveletCalc2.R
convolve.R
criticalPoint.R
graphic.R
.classpath
.project
Neuronal Activity Analysis Tool user manual.docx
results - Verknüpfung.lnk

3) With the file open, click on the **Java compiler** icon in the Bio7 tool bar. The program will be compiled. In the lower right corner of the Bio7 window a bar of progress is shown.


Bio7
File Edit Source Refactor Search Preferences Scripts R LibreOffice Window Help
2d Image R SceneBuilder Resource
Navigator
NeuronActivityMouseClickGit1.0
bin
results
src
Main.fxml
Main.java
PleaseProvideControllerClassName.java
WaveletCalc2.R
convolve.R
criticalPoint.R
graphic.R
.classpath
.project
Neuronal Activity Analysis Tool user manual.docx
results - Verknüpfung.lnk
Main.java
Outline
Main
setup() : void
run() : void
Console Properties Snippets
Options Console
Bio7 Console
Timesteps: 0
Writable
Smart Insert
1 : 1
309M of 982M
Building workspace: (50%

```java
import com.eco.bio7.batch.FileRoot;

public class Main extends com.eco.bio7.compile.Model {

    public void setup() {

        /* Open the custom perspective! */
        // Work.openPerspective("com.eco.bio7.CustomPerspective");

        /* Open the ImageJ view in the current perspective! */
        Work.openView("com.eco.bio7.imagej");
        /* Create a custom view and embedd the ModelGUI! */

        CustomView view = new CustomView();

        view.setFxmlCanvas("left", FileRoot.getFileRoot() + "/NeuronActivityMouseClickGit1.0/src/Main.fxml",
                new PleaseProvideControllerClassName());

        // System.out.println(Arrays.toString(y));

    }

    public void run() {

    }


}
```

4) Start the tool by a click on the **setup icon**.
The **Neuron Activity Analysis tool** menu will show up.


Bio7
File Edit Source Refactor Search Preferences Scripts R LibreOffice Window Help
2d Image R SceneBuilder Resource
NeuronActivityMouseClickGit1.0
bin
results
src
Main.fxml
Main.java
PleaseProvideControllerClassName.java
WaveletCalc2.R
convolve.R
criticalPoint.R
graphic.R
.classpath
.project
Neuronal Activity Analysis Tool user manual.docx
results - Verknüpfung.lnk
Main.java
Outline
Main
setup() : void
run() : void
Console Properties Snippets ImageJ-Canvas *Custom Controls
Start Rserve And Load Image
Process Image Stack
Timesteps: 0
109M of 978M

```java
import com.eco.bio7.batch.FileRoot;

public class Main extends com.eco.bio7.compile.Model {

    public void setup() {

        /* Open the custom perspective! */
        // Work.openPerspective("com.eco.bio7.CustomPerspective");

        /* Open the ImageJ view in the current perspective! */
        Work.openView("com.eco.bio7.imagej");
        /* Create a custom view and embedd the ModelGUI! */

        CustomView view = new CustomView();

        view.setFxmlCanvas("left", FileRoot.getFileRoot() + "/NeuronActivityMouseClickGit1.0/src/Main.fxml",
                new PleaseProvideControllerClassName());

        // System.out.println(Arrays.toString(y));

    }

    public void run() {

    }


}
```

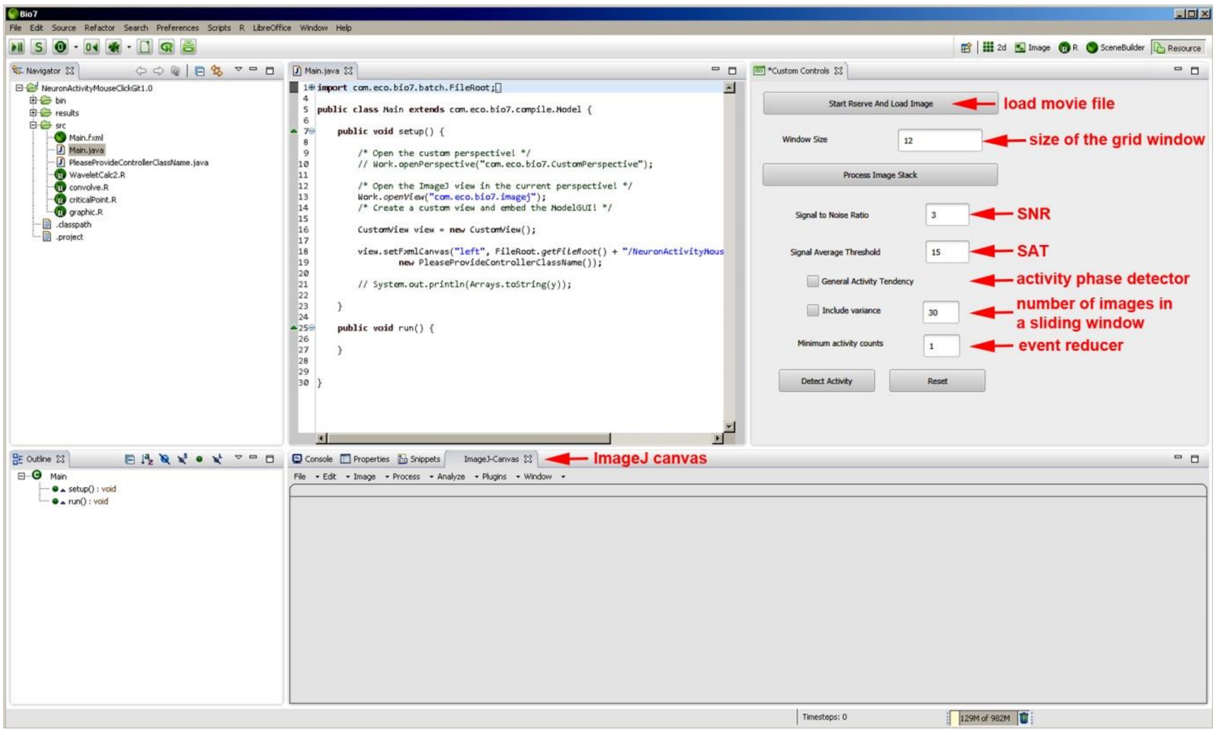
Window Size 12

Signal to Noise Ratio 3

Signal Average Threshold 15

☐ General Activity Tendency

5) For convenience, drag the window to the upper right corner. The activity analysis tool bar and the ImageJ implementation are ready for use.



```java
import com.eco.bio7.batch.FileRoot;

public class Main extends com.eco.bio7.compile.Model {

    public void setup() {

        /* Open the custom perspective! */
        // Work.openPerspective("com.eco.bio7.CustomPerspective");

        /* Open the ImageJ view in the current perspective! */
        Work.openView("com.eco.bio7.imagej");
        /* Create a custom view and embed the ModelGUI! */

        CustomView view = new CustomView();

        view.setFxmlCanvas("left", FileRoot.getFileRoot() + "/NeuronActivityMous
                new PleaseProvideControllerClassName());

        // System.out.println(Arrays.toString(y));

    }

    public void run() {

    }

}
```

Window Size: 12

Signal to Noise Ratio: 3

Signal Average Threshold: 15

☐ General Activity Tendency

☐ Include variance: 30

Minimum activity counts: 1

Short description of the toolbar:

Start Rserve And Load Image: import of the raw image material.

Window Size: defines the size of the grid window (12; means 12x12 pixel).

Process Image Stack: calculation of intensity signals.

Signal to Noise Ratio: definition of the stringency criterion SNR.

Signal Average Threshold: definition of the detection threshold.

General Activity Tendency: defines and calculates signal phases.

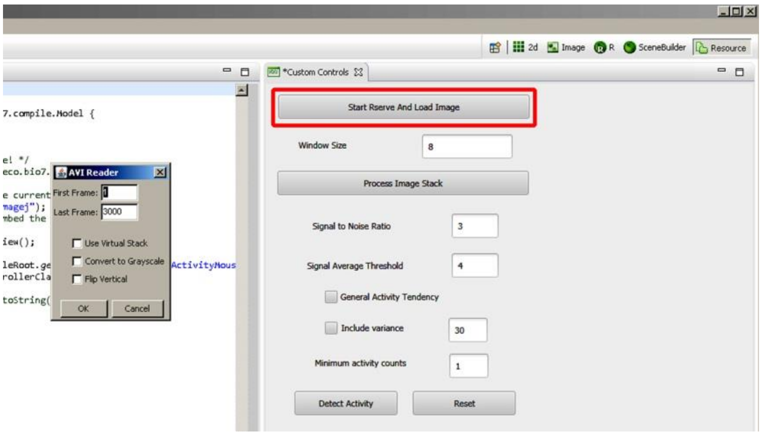
Include variance: labels and calculates fluctuations in the signal.

Minimum activity counts: Event reducer (The number “1” means that only those grid windows are displayed and counted, in which more than one activity event was found.

III Activity analysis

1) Click on **Start Rserve And Load Image** window, select an avi-file containing the uncompressed image material. The AVI Reader window appears and asks for the wished frame window (e.g. image 1 to 3000). Confirm with **OK.** The corresponding video appears in the ImageJ window.

NOTE: *The first time this is done, R needs to install some application packages. Simply click OK when asked.*



AVI Reader

First Frame: 1

Last Frame: 3000

☐ Use Virtual Stack

☐ Convert to Grayscale

☐ Flip Vertical

Window Size: 8

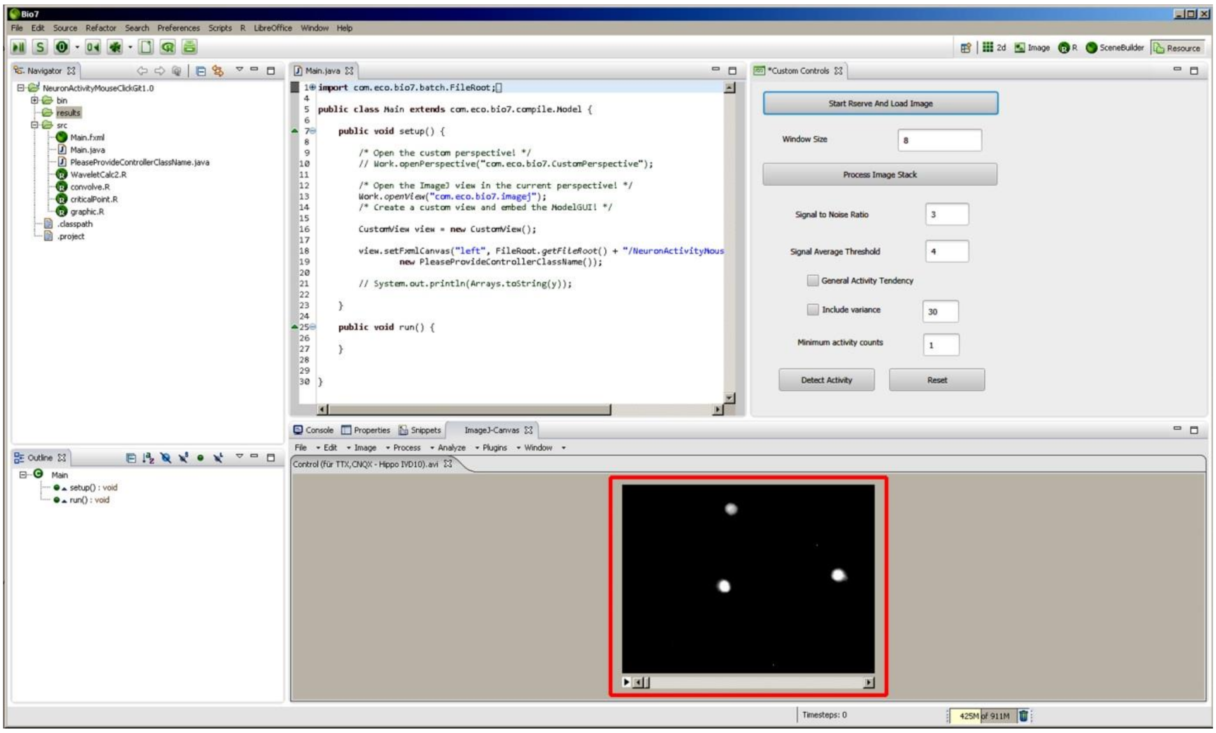
Signal to Noise Ratio: 3

Signal Average Threshold: 4

☐ General Activity Tendency

☐ Include variance: 30

Minimum activity counts: 1



Main.java

```
import com.eco.bio7.batch.FileRoot;

public class Main extends com.eco.bio7.compile.Model {

    public void setup() {

        /* Open the custom perspective! */
        // Work.openPerspective("com.eco.bio7.CustomPerspective");

        /* Open the ImageJ view in the current perspective! */
        Work.openView("com.eco.bio7.imagej");
        /* Create a custom view and embed the ModelGUI! */

        CustomView view = new CustomView();

        view.setFxmlCanvas("left", FileRoot.getFileRoot() + "/NeuronActivityMous
              new PleaseProvideControllerClassName());

        // System.out.println(Arrays.toString(y));

    }

    public void run() {

    }


}
```

Window Size: 8

Signal to Noise Ratio: 3

Signal Average Threshold: 4

☐ General Activity Tendency

☐ Include variance: 30

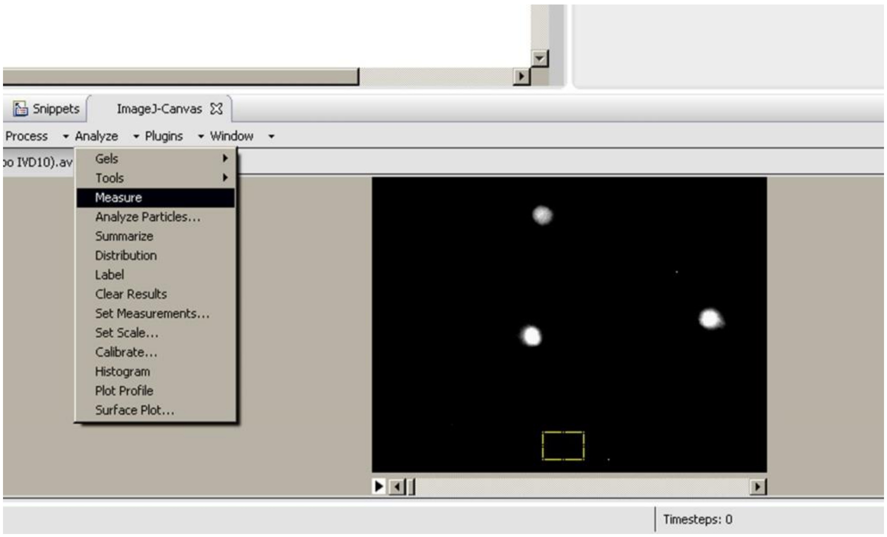
Minimum activity counts: 1

Control (für TTX,CNQX - Hippo IVD10).avi

Timesteps: 0

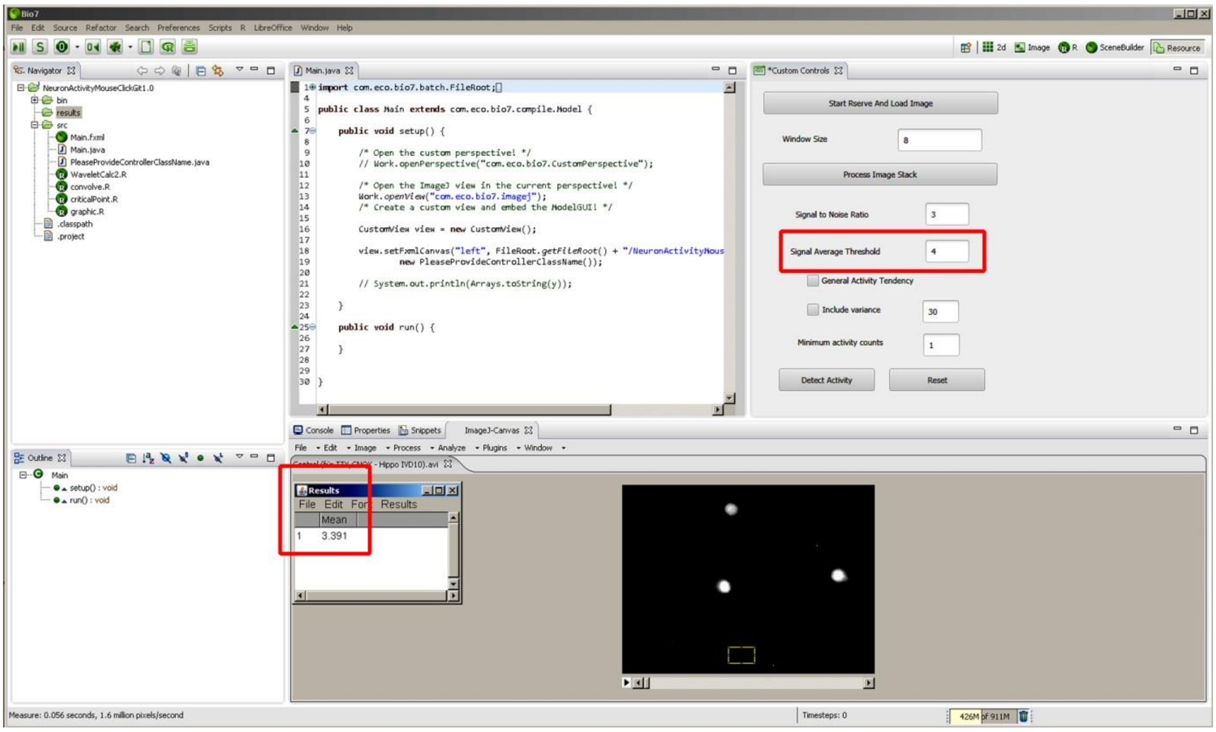
425M of 911M

2) Determine a threshold value. This can either been done in BIO7 or in any other image analysis program. In BIO7, use ImageJ to select a small ROI (yellow rectangle) either in the background or at any other reference locus in the x,y-field. Open the **Analyze** window of ImageJ. Click **Measure** to determine the mean grey value (mean intensity bit value) in the background ROI.



3) Set the **Signal Average Threshold** according to the rule: rounded threshold value ± 1 / 2.

For example: Here, a mean value of 3.391 was determined. This value was rounded up to 4. Therefore, a SAT of 4 is chosen for this analysis.


Bio7
File Edit Source Refactor Search Preferences Scripts R LibreOffice Window Help
2d Image R SceneBuilder Resource
Navigator
NeuronActivityMouseClickGt1.0
bin
results
src
Main.fxml
Main.java
PleaseProvideControllerClassName.java
WaveletCalc2.R
convolve.R
criticalPoint.R
graphic.R
.classpath
.project
Outline
Main
setup() : void
run() : void
Main.java
*Custom Controls
Start Rserve And Load Image
Process Image Stack
Detect Activity
Reset
Console Properties Snippets ImageJ-Canvas
File Edit Image Process Analyze Plugins Window
Hippo IVD10).avi
Results
File Edit Font Results
Timesteps: 0
426M of 911M

```
import com.eco.bio7.batch.FileRoot;

public class Main extends com.eco.bio7.compile.Model {

    public void setup() {

        /* Open the custom perspective! */
        // Work.openPerspective("com.eco.bio7.CustomPerspective");

        /* Open the ImageJ view in the current perspective! */
        Work.openView("com.eco.bio7.imagej");
        /* Create a custom view and embed the ModelGUI! */

        CustomView view = new CustomView();

        view.setFxmlCanvas("left", FileRoot.getFileRoot() + "/NeuronActivityMous
                new PleaseProvideControllerClassName());

        // System.out.println(Arrays.toString(y));

    }

    public void run() {

    }

}
```

Window Size: 8

Signal to Noise Ratio: 3

Signal Average Threshold: 4

☐ General Activity Tendency

☐ Include variance: 30

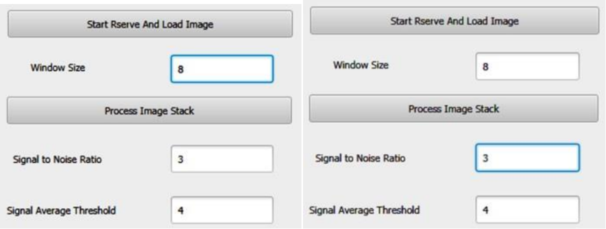
Minimum activity counts: 1

	Mean
1	3.391

Measure: 0.056 seconds, 1.6 million pixels/second

4) Select the size of the grid window according to the size of the neuronal structures. We routinely use a **Window Size** value of 12 – 8, meaning a grid of 12 x 12 or 8 x 8 pixel.

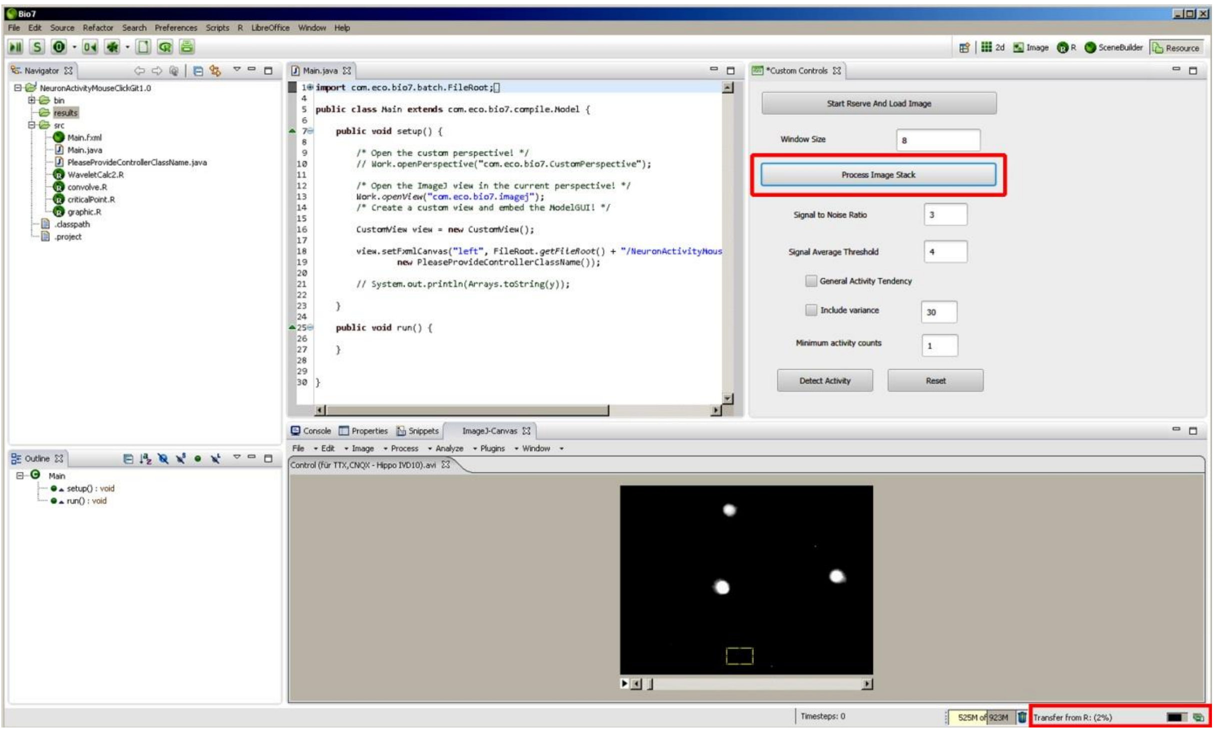
Select the **Signal to Noise Ratio** to tune the stringency of the tool. We recommend a SNR of 2.5 as default setting for an initial assessment. High SNR values of 3 – 4 will preferentially compute “larger” calcium transients, while SNR values of 1.5 – 2 will compute events which are close to the signal noise. Details are explained and discussed in the corresponding publication.



Window Size 8
Signal to Noise Ratio 3
Signal Average Threshold 4

Window Size 8
Signal to Noise Ratio 3
Signal Average Threshold 4

5) Click **Process Image Stack**. The signal intensity values are computed. The progress is seen in the right lower corner. Wait until this calculation is finished.

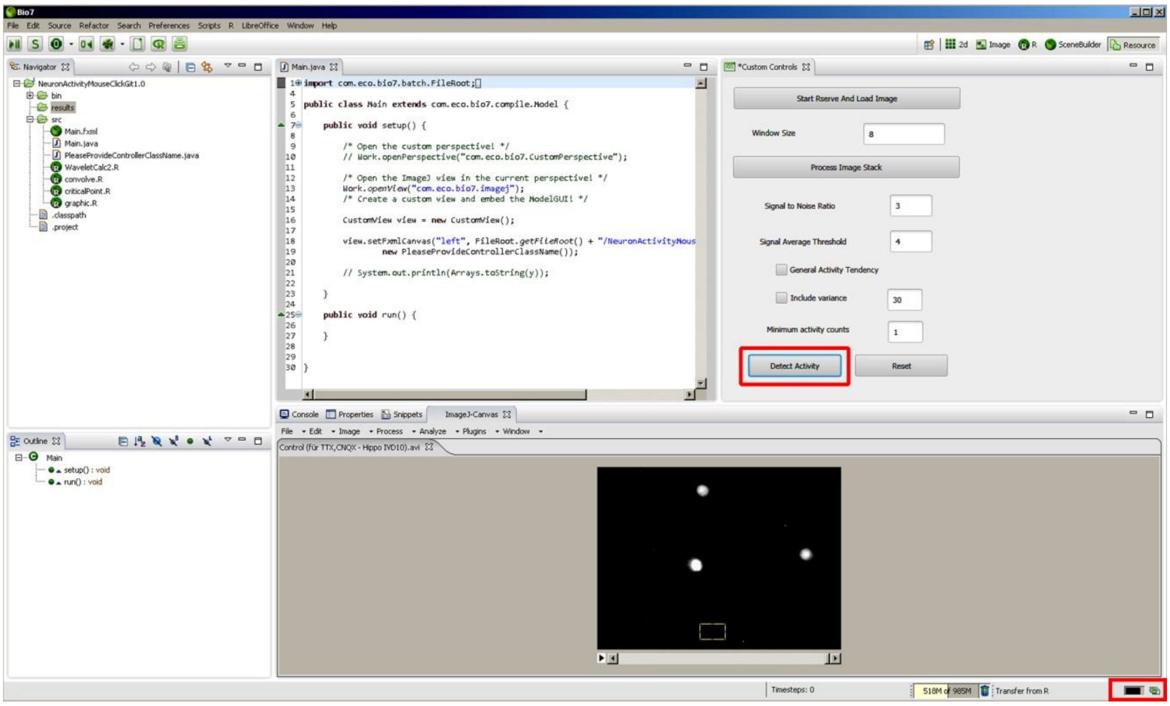

Bio7
File Edit Source Refactor Search Preferences Scripts R LibreOffice Window Help
2d Image R SceneBuilder Resource
Navigator
NeuronActivityMouseClickGit1.0
bin
results
src
Main.fxml
Main.java
PleaseProvideControllerClassName.java
WaveletCalc2.R
convolve.R
criticalPoint.R
graphic.R
.classpath
.project
Outline
Main
setup() : void
run() : void
Main.java
*Custom Controls
Start Rserve And Load Image
Process Image Stack
General Activity Tendency
Detect Activity
Reset
Console Properties Snippets ImageJ-Canvas
File Edit Image Process Analyze Plugins Window
Control (für TTX,CNQX - Hippo IVD10).avi

```
import com.eco.bio7.batch.FileRoot;

public class Main extends com.eco.bio7.compile.Model {

    public void setup() {

        /* Open the custom perspective! */
        // Work.openPerspective("com.eco.bio7.CustomPerspective");

        /* Open the ImageJ view in the current perspective! */
        Work.openView("com.eco.bio7.imagej");
        /* Create a custom view and embed the ModelGUI! */

        CustomView view = new CustomView();

        view.setFxmlCanvas("left", FileRoot.getFileRoot() + "/NeuronActivityMous
                new PleaseProvideControllerClassName());

        // System.out.println(Arrays.toString(y));

    }

    public void run() {

    }


}
```

Window Size 8
Signal to Noise Ratio 3
Signal Average Threshold 4
☐ General Activity Tendency
☐ Include variance 30
Minimum activity counts 1

Timesteps: 0
525M of 923M
Transfer from R: (2%)

6) Click **Detect Activity**. The calcium signal candidates are computed. The progress is seen in the right lower corner.



```
import com.eco.bio7.batch.FileRoot;

public class Main extends com.eco.bio7.compile.Model {

    public void setup() {

        /* Open the custom perspective! */
        // Work.openPerspective("com.eco.bio7.CustomPerspective");

        /* Open the ImageJ view in the current perspective! */
        Work.openView("com.eco.bio7.imagej");
        /* Create a custom view and embed the ModelGUI! */

        CustomView view = new CustomView();

        view.setFxmlCanvas("left", FileRoot.getFileRoot() + "/NeuronActivityMous
                new PleaseProvideControllerClassName());

        // System.out.println(Arrays.toString(y));

    }

    public void run() {

    }


}
```

Window Size: 8

Signal to Noise Ratio: 3

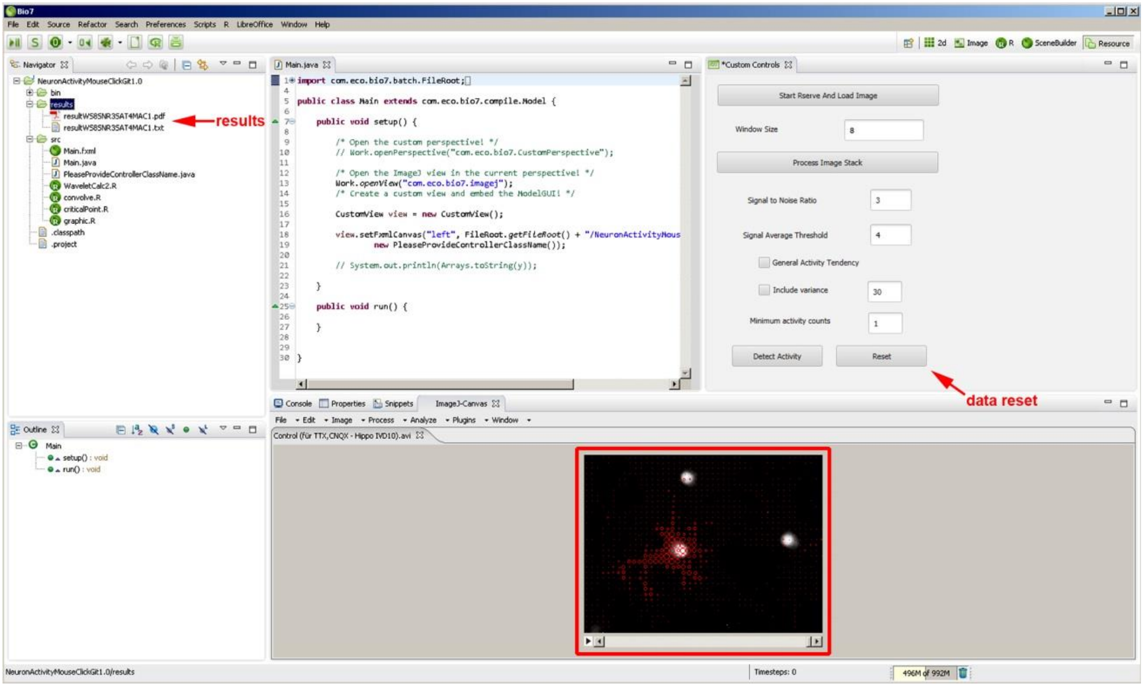
Signal Average Threshold: 4

☐ General Activity Tendency

☐ Include variance: 30

Minimum activity counts: 1

7) When the computation is finished, an image is formed showing the distribution of the computed activity events per grid window as red circles. An output pdf file and a text file are created and put to the results folder. The results are best opened from the file manager system. A mouse click on an individual grid window (**red circle**) opens an **Info!** window. Confirming the Info with **Ok** opens the signal trace in a new plot-window.



```
import com.eco.bio7.batch.FileRoot;

public class Main extends com.eco.bio7.compile.Model {

    public void setup() {

        /* Open the custom perspective! */
        // Work.openPerspective("com.eco.bio7.CustomPerspective");

        /* Open the ImageJ view in the current perspective! */
        Work.openView("com.eco.bio7.imagej");
        /* Create a custom view and embed the ModelGUI! */

        CustomView view = new CustomView();

        view.setFxmlCanvas("left", FileRoot.getFileRoot() + "/NeuronActivityMous
                new PleaseProvideControllerClassName());

        // System.out.println(Arrays.toString(y));

    }

    public void run() {

    }


}
```

Window Size: 8

Signal to Noise Ratio: 3

Signal Average Threshold: 4

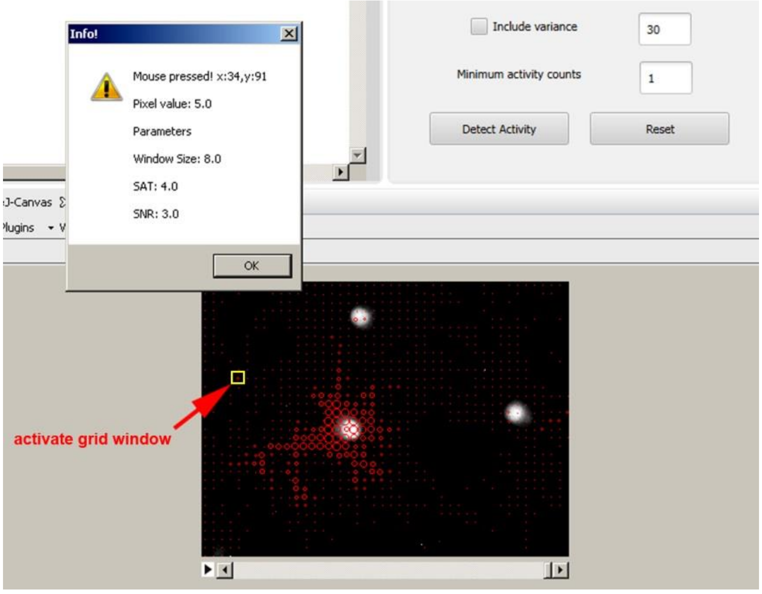
☐ General Activity Tendency

☐ Include variance: 30

Minimum activity counts: 1

Bio7
File Edit Source Refactor Search Preferences Scripts R LibreOffice Window Help
Navigator
NeuronActivityMouseClickGit1.0
bin
results
resultWS8SNR3SAT4MAC1.pdf
resultWS8SNR3SAT4MAC1.txt
results
src
Main.fxml
Main.java
PleaseProvideControllerClassName.java
WaveletCalc2.R
convolve.R
criticalPoint.R
graphic.R
.classpath
.project
Outline
Main
setup() : void
run() : void
Main.java
*Custom Controls
2d
Image
R
SceneBuilder
Resource
Start Rserve And Load Image
Process Image Stack
Detect Activity
Reset
data reset
Console
Properties
Snippets
ImageJ-Canvas
File Edit Image Process Analyze Plugins Window
Control (für TTX,CNQX - Hippo IVD10).avi
NeuronActivityMouseClickGit1.0/results
Timesteps: 0
496M of 992M

8) Activating a red circle opens an **Info!** window.



Mouse pressed! x:34,y:91

Pixel value: 5.0

Parameters

Window Size: 8.0

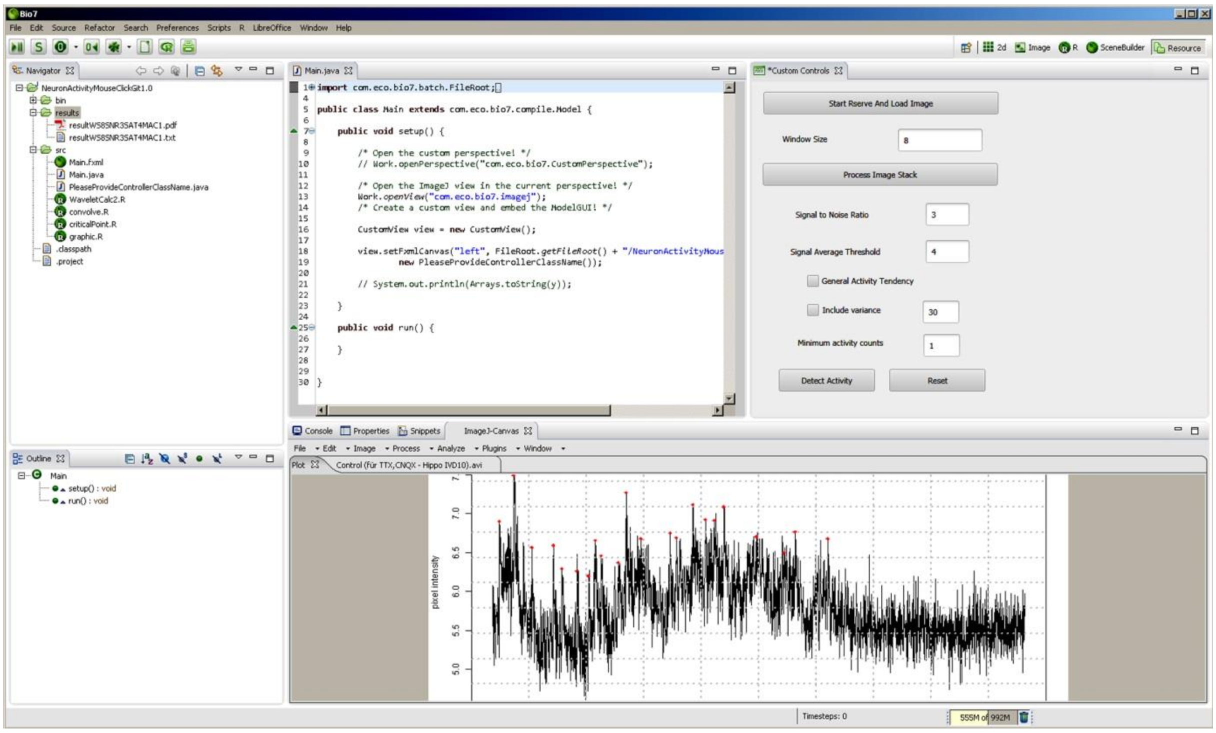
SAT: 4.0

SNR: 3.0

☐ Include variance: 30

Minimum activity counts: 1

9) The trace underlying this grid window opens.



Bio7
File Edit Source Refactor Search Preferences Scripts R LibreOffice Window Help
2d Image R SceneBuilder Resource
Navigator
NeuronActivityMouseClickGit1.0
bin
results
resultW5SSNR3SAT4MAC1.pdf
resultW5SSNR3SAT4MAC1.txt
src
Main.fxml
Main.java
PleaseProvideControllerClassName.java
WaveletCalc2.R
convolve.R
criticalPoint.R
graphic.R
.classpath
.project
Outline
Main
setup() : void
run() : void
Main.java
*Custom Controls
Start Rserve And Load Image
Process Image Stack
Detect Activity
Reset
Console Properties Snippets ImageJ-Canvas
File Edit Image Process Analyze Plugins Window
Plot
Control (für TTX,CNQX - Hippo IVD10).avi
pixel intensity
7.
7.0
6.5
6.0
5.5
5.0
Timesteps: 0
555M of 992M

```
import com.eco.bio7.batch.FileRoot;

public class Main extends com.eco.bio7.compile.Model {

    public void setup() {

        /* Open the custom perspective! */
        // Work.openPerspective("com.eco.bio7.CustomPerspective");

        /* Open the ImageJ view in the current perspective! */
        Work.openView("com.eco.bio7.imagej");
        /* Create a custom view and embed the ModelGUI! */

        CustomView view = new CustomView();

        view.setFxmlCanvas("left", FileRoot.getFileRoot() + "/NeuronActivityMous
                new PleaseProvideControllerClassName());

        // System.out.println(Arrays.toString(y));

    }

    public void run() {

    }

}
```

Window Size: 8

Signal to Noise Ratio: 3

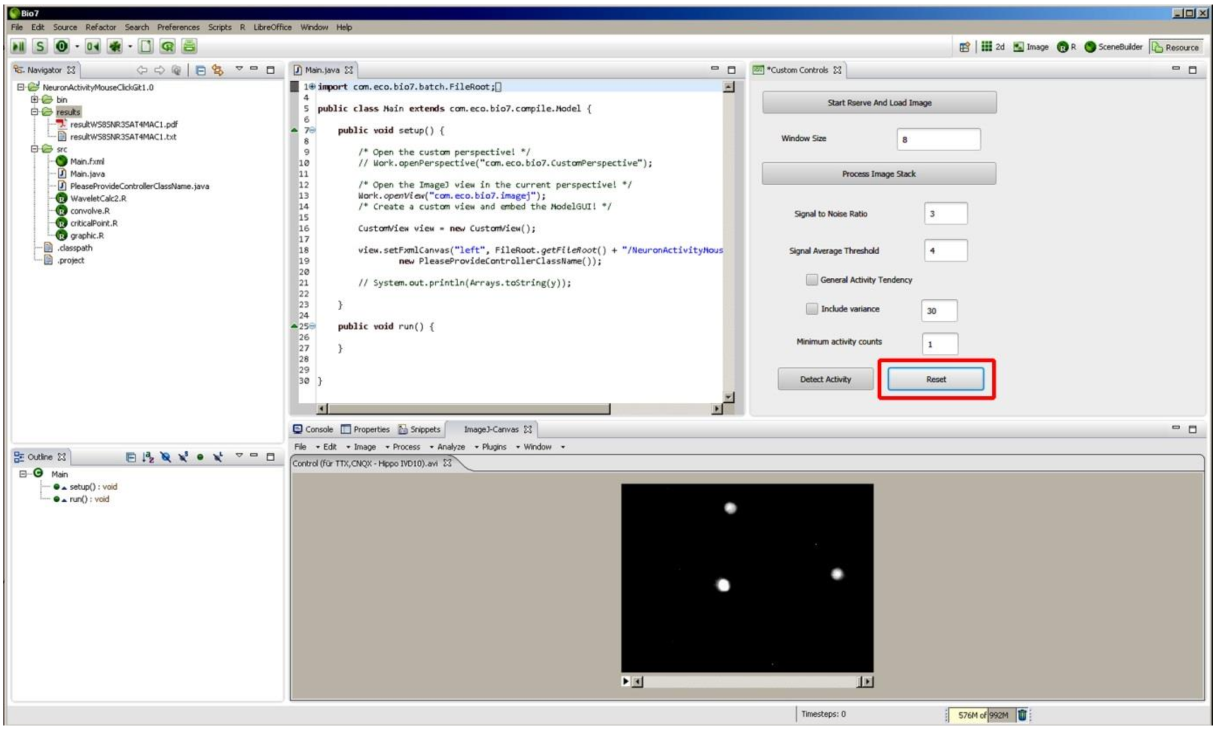
Signal Average Threshold: 4

☐ General Activity Tendency

☐ Include variance: 30

Minimum activity counts: 1

10) The **Reset** function must be used before new SAT or SNR settings can be computed.



Bio7
File Edit Source Refactor Search Preferences Scripts R LibreOffice Window Help
2d Image R SceneBuilder Resource
Navigator
NeuronActivityMouseClickGit1.0
bin
results
resultWS8SNR3SAT4MAC1.pdf
resultWS8SNR3SAT4MAC1.txt
src
Main.fxml
Main.java
PleaseProvideControllerClassName.java
WaveletCalc2.R
convolve.R
criticalPoint.R
graphic.R
.classpath
.project
Outline
Main
setup() : void
run() : void
Main.java
*Custom Controls
Start Rserve And Load Image
Process Image Stack
Detect Activity
Reset
Console Properties Snippets ImageJ-Canvas
File Edit Image Process Analyze Plugins Window
Control (für TTX,CNQX - Hippo IVD10).avi
Timesteps: 0
576M of 992M

```
import com.eco.bio7.batch.FileRoot;

public class Main extends com.eco.bio7.compile.Model {

    public void setup() {

        /* Open the custom perspective! */
        // Work.openPerspective("com.eco.bio7.CustomPerspective");

        /* Open the ImageJ view in the current perspective! */
        Work.openView("com.eco.bio7.imagej");
        /* Create a custom view and embed the ModelGUI! */

        CustomView view = new CustomView();

        view.setFxmlCanvas("left", FileRoot.getFileRoot() + "/NeuronActivityMous
                new PleaseProvideControllerClassName());

        // System.out.println(Arrays.toString(y));

    }

    public void run() {

    }

}
```

Window Size: 8

Signal to Noise Ratio: 3

Signal Average Threshold: 4

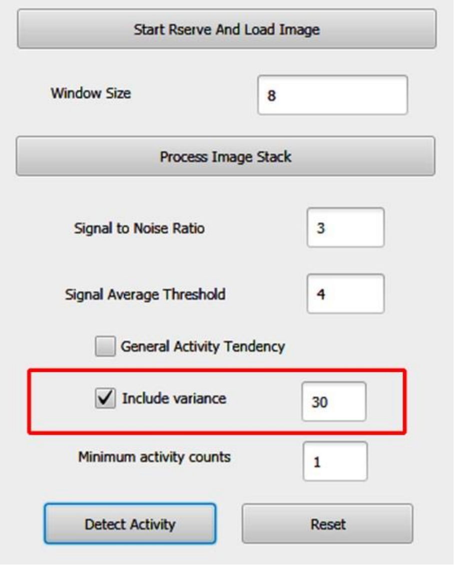
☐ General Activity Tendency

☐ Include variance: 30

Minimum activity counts: 1

IV Variance area analysis

1) To calculate the variance area, activate the **Include variance** tool. In the output data, the variance area will be calculated on base of a sliding window. The number of images which are used by this tool can be selected by the user (here 30 image frames).



Window Size: 8

Signal to Noise Ratio: 3

Signal Average Threshold: 4

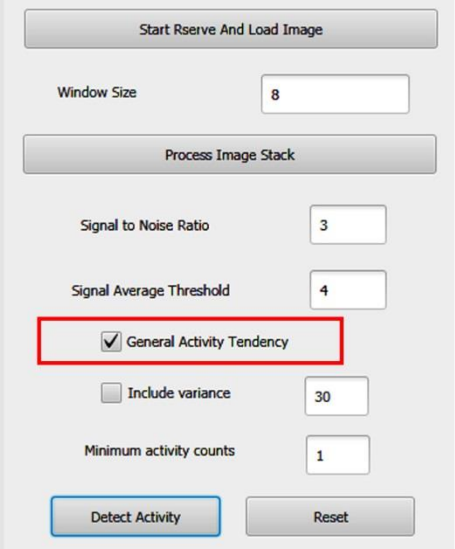
☐ General Activity Tendency

☑ Include variance: 30

Minimum activity counts: 1

V General Activity Tendency

1) To calculate and determine phases of long-lasting increase in calcium levels, activate the **General Activity Tendency** tool. This tool may be helpful for the identification of grid windows with a long-lasting increase of calcium levels after a stimulus or activity event, e.g. after stimulation of metabotropic receptors. The algorithm is based on a statistical test that defines a critical point of change in the signal intensity values. Details are given in the corresponding manuscript. The tool marks and calculates an area under the curve.



Window Size 8

Signal to Noise Ratio 3

Signal Average Threshold 4

☑ General Activity Tendency

☐ Include variance 30

Minimum activity counts 1

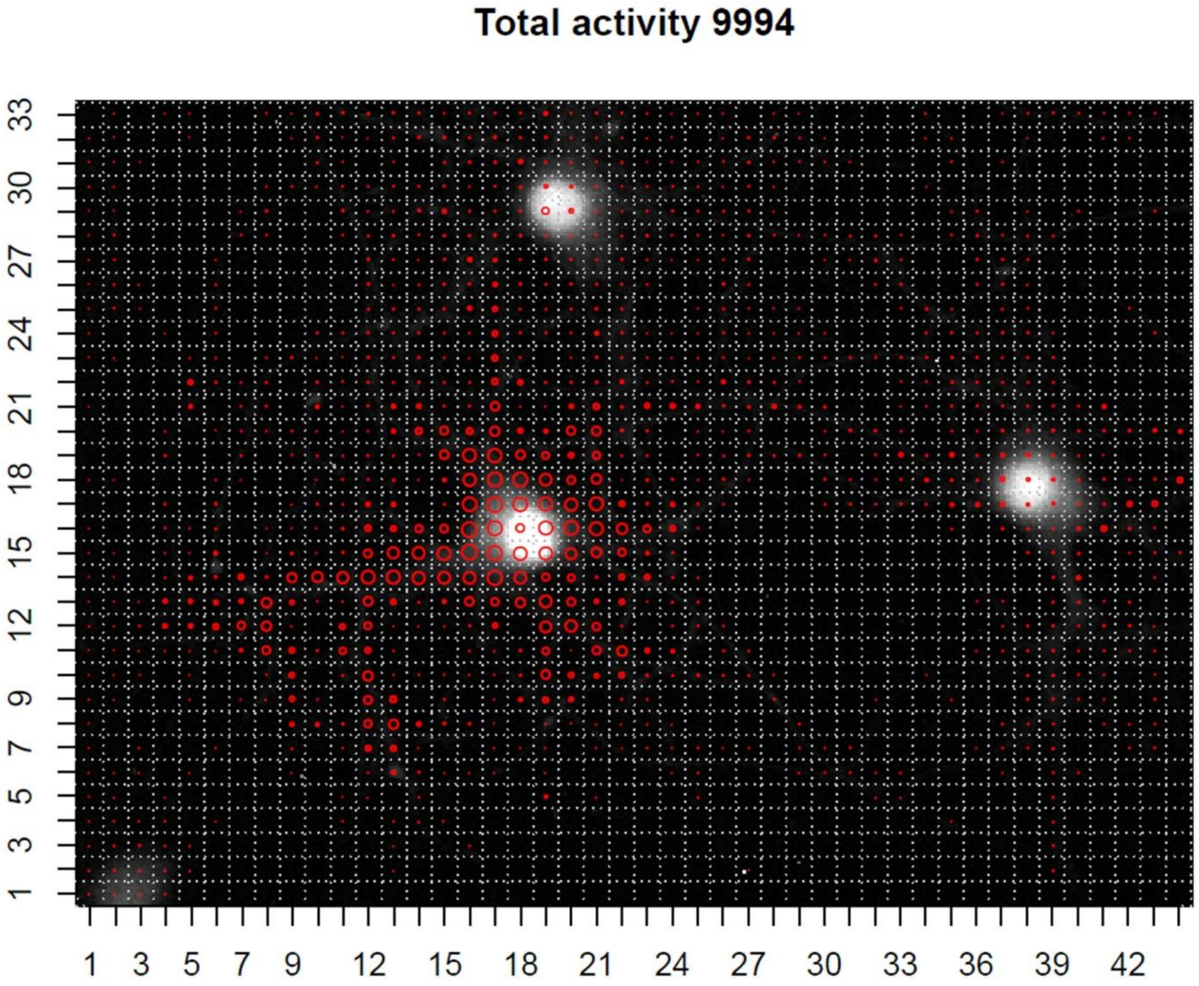
VI Data Output

The summary pdf shows on page 1 the first image of the time series, the grid, and loci of calcium activity indicated by red circles within the grid window. The diameter of the circles is the bigger the more activity the tool found. All activity events are summed up to give a virtual activity number, the 'total activity' value. This value represents all calcium events in the whole x,y-t images series to represent the activity state under a specific experimental condition.

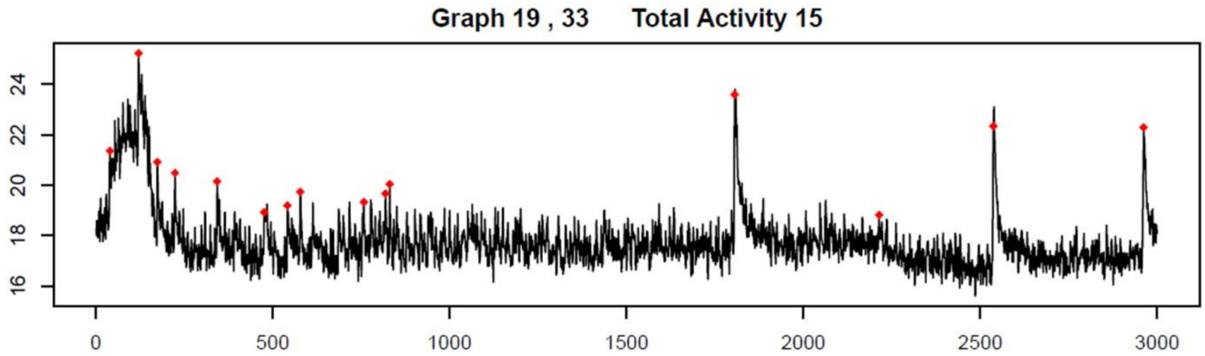
Furthermore, the resulting pdf document shows all traces in which a calcium activity event was found. Activity events are marked by a red dot and counted.

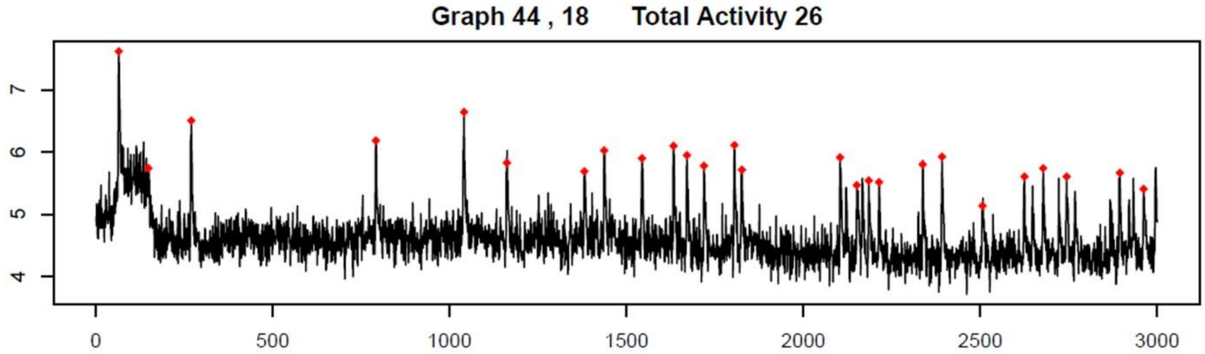
A txt-file is generated that shows the calculated numbers of activity per grid window in a x,y-table structure which is best opened with a typical editor program (e.g. Windows Editor).

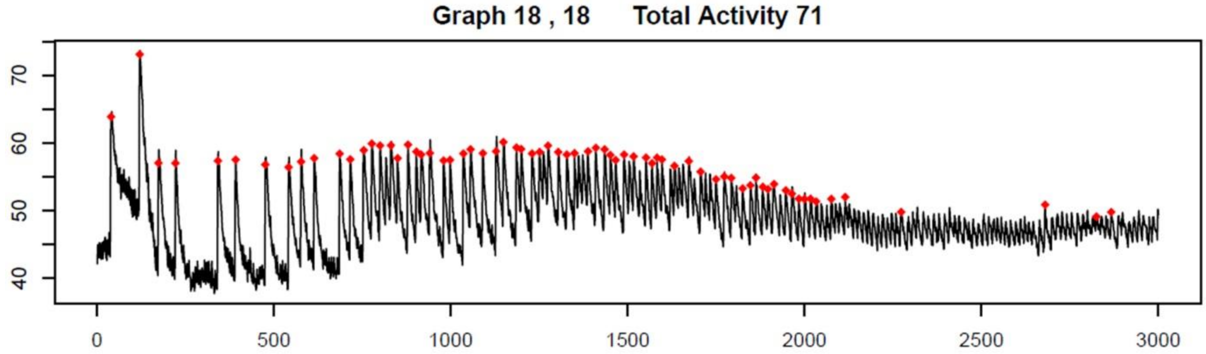
1) Activity map and the virtual activity number "Total activity"



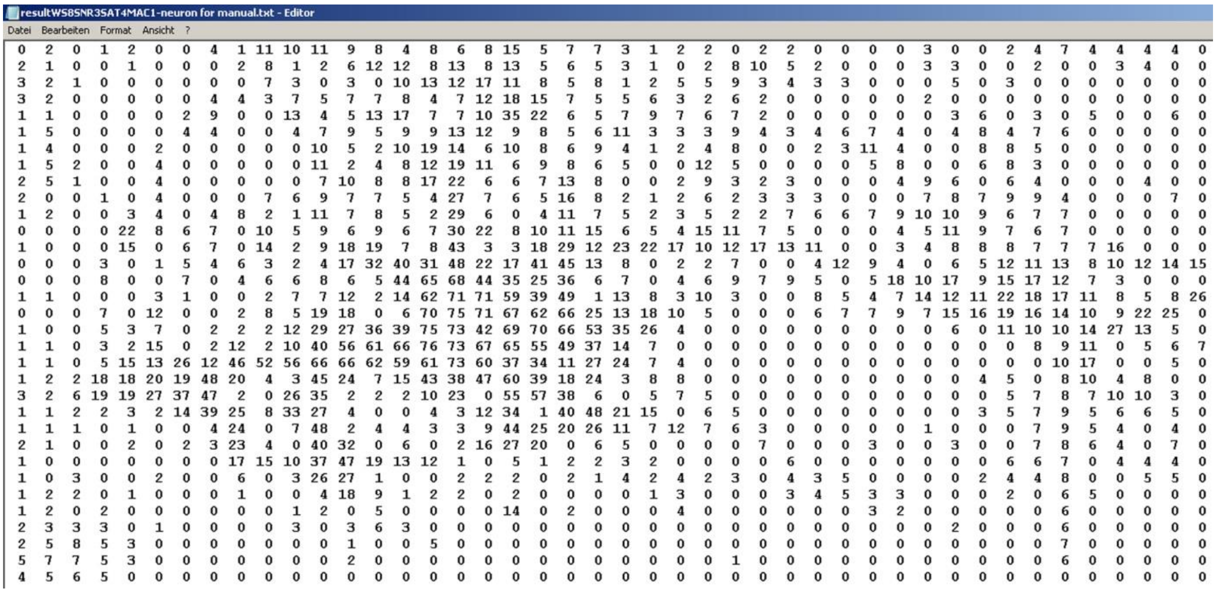
2) Signal raw traces (Graph) showing the mean grey value (pixel intensity) in a grid window (e.g. for the upper trace x,y = 19, 33). Red squares mark computed activity events which contribute to the total activity number. The upper graph shows 15 activity events.





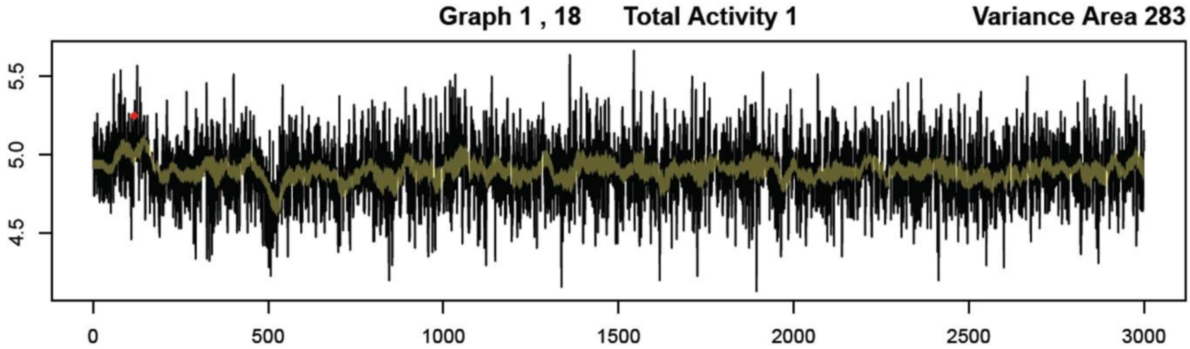


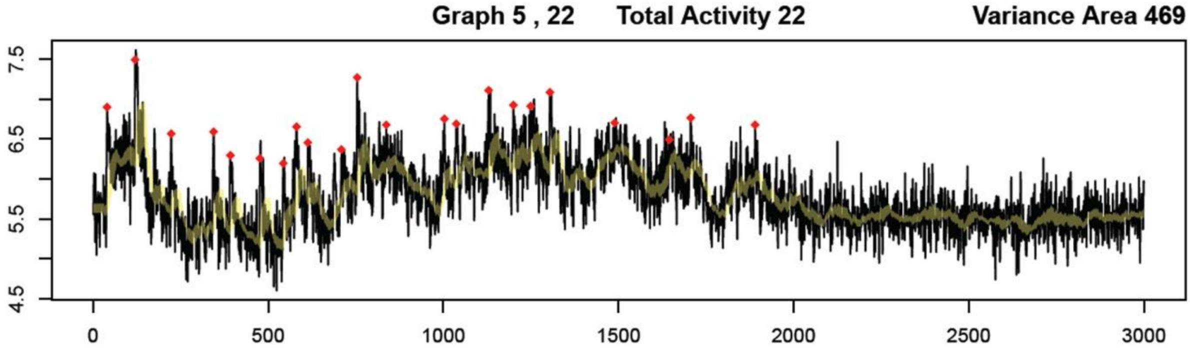
3) The Txt-file lists the grid-specific activity event numbers in a table-like format.

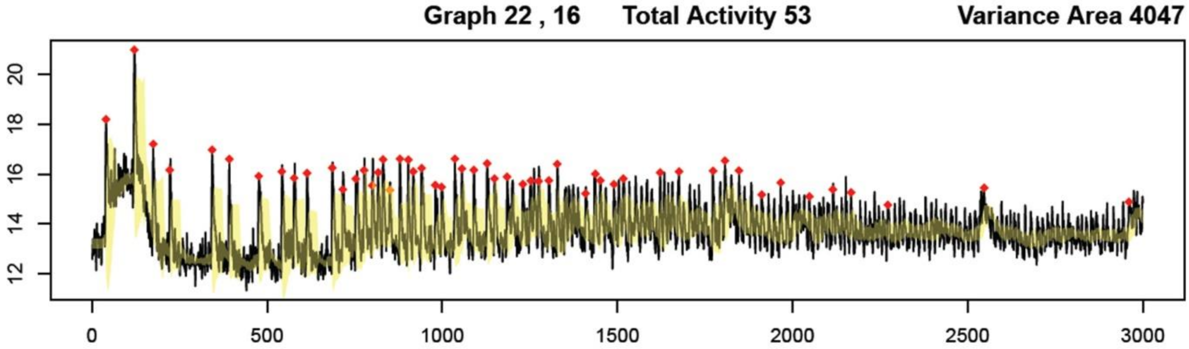

resultWS8SNR3SAT4MAC1-neuron for manual.txt - Editor
Datei Bearbeiten Format Ansicht ?

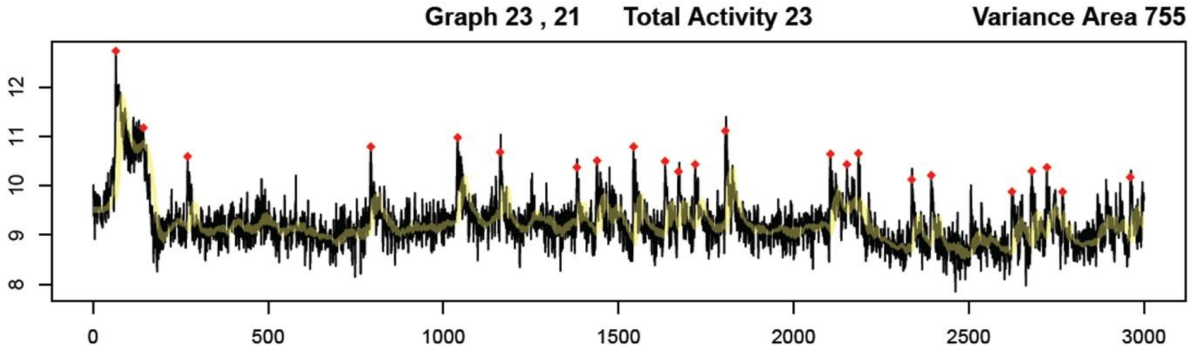
```
  0  2  0  1  2  0  0  4  1 11 10 11  9  8  4  8  6  8 15  5  7  7  3  1  2  2  0  2  2  0  0  0  0  3  0  0  2  4  7  4  4  4  4  0
  2  1  0  0  1  0  0  0  2  8  1  2  6 12 12  8 13  8 13  5  6  5  3  1  0  2  8 10  5  2  0  0  0  3  3  0  0  2  0  0  3  4  0  0
  3  2  1  0  0  0  0  0  0  7  3  0  3  0 10 13 12 17 11  8  5  8  1  2  5  5  9  3  4  3  3  0  0  0  5  0  3  0  0  0  0  0  0  0
  3  2  0  0  0  0  0  4  4  3  7  5  7  7  8  4  7 12 18 15  7  5  5  6  3  2  6  2  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0
  1  1  0  0  0  0  2  9  0  0 13  4  5 13 17  7  7 10 35 22  6  5  7  9  7  6  7  2  0  0  0  0  0  0  3  6  0  3  0  5  0  0  6  0
  1  5  0  0  0  0  4  4  0  0  4  7  9  5  9  9 13 12  9  8  5  6 11  3  3  3  9  4  3  4  6  7  4  0  4  8  4  7  6  0  0  0  0  0
  1  4  0  0  0  2  0  0  0  0  0 10  5  2 10 19 14  6 10  8  6  9  4  1  2  4  8  0  0  2  3 11  4  0  0  8  8  5  0  0  0  0  0  0
  1  5  2  0  0  4  0  0  0  0  0 11  2  4  8 12 19 11  6  9  8  6  5  0  0 12  5  0  0  0  0  5  8  0  0  6  8  3  0  0  0  0  0  0
  2  5  1  0  0  4  0  0  0  0  0  7 10  8  8 17 22  6  6  7 13  8  0  0  2  9  3  2  3  0  0  0  4  9  6  0  6  4  0  0  0  4  0  0
  2  0  0  1  0  4  0  0  0  7  6  9  7  7  5  4 27  7  6  5 16  8  2  1  2  6  2  3  3  3  0  0  0  7  8  7  9  9  4  0  0  0  7  0
  1  2  0  0  3  4  0  4  8  2  1 11  7  8  5  2 29  6  0  4 11  7  5  2  3  5  2  2  7  6  6  7  9 10 10  9  6  7  7  0  0  0  0  0
  0  0  0  0 22  8  6  7  0 10  5  9  6  9  6  7 30 22  8 10 11 15  6  5  4 15 11  7  5  0  0  0  4  5 11  9  7  6  7  0  0  0  0  0
  1  0  0  0 15  0  6  7  0 14  2  9 18 19  7  8 43  3  3 18 29 12 23 22 17 10 12 17 13 11  0  0  3  4  8  8  8  7  7  7 16  0  0  0
  0  0  0  3  0  1  5  4  6  3  2  4 17 32 40 31 48 22 17 41 45 13  8  0  2  2  7  0  0  4 12  9  4  0  6  5 12 11 13  8 10 12 14 15
  0  0  0  8  0  0  7  0  4  6  6  8  6  5 44 65 68 44 35 25 36  6  7  0  4  6  9  7  9  5  0  5 18 10 17  9 15 17 12  7  3  0  0  0
  1  1  0  0  0  3  1  0  0  2  7  7 12  2 14 62 71 71 59 39 49  1 13  8  3 10  3  0  0  8  5  4  7 14 12 11 22 18 17 11  8  5  8 26
  0  0  0  7  0 12  0  0  2  8  5 19 18  0  6 70 75 71 67 62 66 25 13 18 10  5  0  0  0  6  7  7  9  7 15 16 19 16 14 10  9 22 25  0
  1  0  0  5  3  7  0  2  2  2 12 29 27 36 39 75 73 42 69 70 66 53 35 26  4  0  0  0  0  0  0  0  0  0  6  0 11 10 10 14 27 13  5  0
  1  1  0  3  2 15  0  2 12  2 10 40 56 61 66 76 73 67 65 55 49 37 14  7  0  0  0  0  0  0  0  0  0  0  0  0  0  8  9 11  0  5  6  7
  1  1  0  5 15 13 26 12 46 52 56 66 66 62 59 61 73 60 37 34 11 27 24  7  4  0  0  0  0  0  0  0  0  0  0  0  0  0 10 17  0  0  5  0
  1  2  2 18 18 20 19 48 20  4  3 45 24  7 15 43 38 47 60 39 18 24  3  8  8  0  0  0  0  0  0  0  0  0  0  4  5  0  8 10  4  8  0  0
  3  2  6 19 19 27 37 47  2  0 26 35  2  2  2 10 23  0 55 57 38  6  0  5  7  5  0  0  0  0  0  0  0  0  0  0  5  7  8  7 10 10  3  0
  1  1  2  2  3  2 14 39 25  8 33 27  4  0  0  4  3 12 34  1 40 48 21 15  0  6  5  0  0  0  0  0  0  0  0  3  5  7  9  5  6  6  5  0
  1  1  1  0  1  0  0  4 24  0  7 48  2  4  4  3  3  9 44 25 20 26 11  7 12  7  6  3  0  0  0  0  0  1  0  0  0  7  9  5  4  0  4  0
  2  1  0  0  2  0  2  3 23  4  0 40 32  0  6  0  2 16 27 20  0  6  5  0  0  0  0  7  0  0  0  3  0  0  3  0  0  7  8  6  4  0  7  0
  1  0  0  0  0  0  0  0 17 15 10 37 47 19 13 12  1  0  5  1  2  2  3  2  0  0  0  0  6  0  0  0  0  0  0  0  6  6  7  0  4  4  4  0
  1  0  3  0  0  2  0  0  6  0  3 26 27  1  0  0  2  2  2  0  2  1  4  2  4  2  3  0  4  3  5  0  0  0  0  2  4  4  8  0  0  5  5  0
  1  2  2  0  1  0  0  0  1  0  0  4 18  9  1  2  2  0  2  0  0  0  0  1  3  0  0  0  3  4  5  3  3  0  0  0  2  0  6  5  0  0  0  0
  1  2  0  2  0  0  0  0  0  0  1  2  0  5  0  0  0  0 14  0  2  0  0  0  4  0  0  0  0  0  0  3  2  0  0  0  0  0  6  0  0  0  0  0
  2  3  3  3  0  1  0  0  0  0  3  0  3  6  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  6  0  0  0  0  0
  2  5  8  5  3  0  0  0  0  0  0  0  1  0  0  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  7  0  0  0  0  0
  5  7  7  5  3  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  6  0  0  0  0  0
  4  5  6  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

4) If selected, the variance area will be marked in the signal trace. The variance area value appears and is used to distinguish fluctuations in the imaging signal. The tool is useful to distinguish signal fluctuations, for instance, before and after blockade of a signaling event. The function of the tool is discussed in the corresponding manuscript.

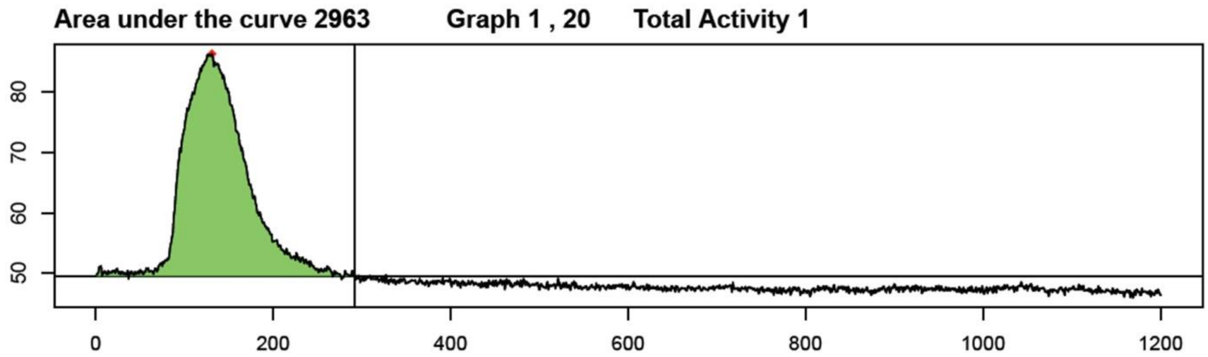


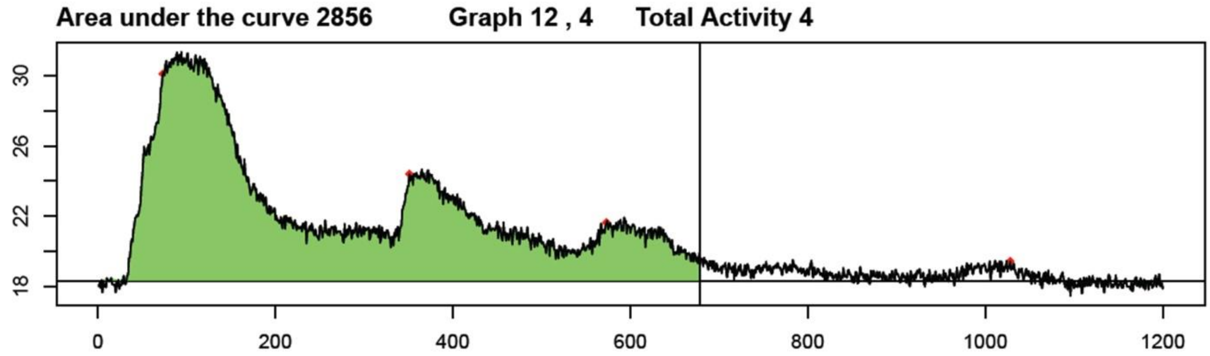






5) Phases of long-lasting activity are visualized by the General Activity Tendency tool. Please note, the activity detector is not well suited to analyze this waveform-like calcium transients. The example here shows a response of a monolayer of cultured astrocytes, stimulated with glutamate to activate metabotropic glutamate receptors.





Juan Pablo Prada Salcedo

Oeggstr 3 97070
Wuerzburg, Germany
(049)1522-311-6348
pradajuan25@gmail.com

Education

PhD. Graduate School of Life Science November 2017
Wuerzburg University, Wuerzburg Germany
Thesis topic: Calcium imaging processing and other topics in Bioinformatics
Advisor: Professor Thomas Dandekar

M.Sc. in Electric and Computer Engineering March 2012
Los Andes University, Bogota Colombia
Thesis topic: Active learning algorithm using Spectral Clustering
Advisor: Professor Fernando Lozano GPA: 4.88/5

B.S. in Electric Engineering September 2009
Pontificia Universidad Javeriana, Bogota Colombia
Thesis topic: Delta-Sigma modulation schemes study
Advisor: Professor Pedro Vizcaya GPA: 4.16/5

Technical Skills

Programming*:* C/C++, OpenCV, HTML, BioNetGen and Python
Simulator / Tools*:* Matlab, R, CellDesigner, RuleBender, Latex and Mevislab

---

Research Experience

Doctoral Student, *University of Wuerzburg, Wuerzburg, Germany* Feb. 2014 – Present

GSLS (Graduate School of Life Science) fellow. I participate in several projects involving different areas of bioinformatics. Most of this projects are collaborations with other laboratories within the university and also external institutions.

The main project is the analysis and automatic detection of neuronal activity in calcium imaging videos. I developed a tool which allows the Neurobiology department researchers to accurately identify the occurrence of ion exchange through calcium channels in neurons. This tool was programed in java and R.

On a second project we are modeling the interactions involving TNF (Tumor Necrosis Factor) ligand in presence of its two known receptors TNFR1 and TNFR2. This projects aims to create a model using BioNetGen language to simulate this biological system which is critical on every inflammation process found in the human body.

In collaboration with the Electron Microscopy department of the University of Wuerzburg and the A-Star institute in Singapore I work on the application of Deep Learning algorithms for the segmentation of neurons in C. elegans images obtained with an electron microscope.

Finally there is a theoretical project about "NP vs P" controversy. We study the way this controversy occurs in the nature. We present protein folding as a study case where nature is not solving an NP problem but a simple P problem and the way nature moves from a complex NP problem to a simple P problem is simply by fixing the initial conditions.

Research Assistant, *University of Delaware, Newark DE, United States* Mar. 2013 – Aug. 2013
Research and development in the area of Computer Vision under the supervision of Professor Kenneth Barner and Professor Jingyi Yu. I was a researcher on topics relative to the use of time-of-flight cameras as multiple camera systems registration and calibration, depth sensor image denoising, 3D model rendering and skeletonization and machine learning implementations for pose classification. Most of the work was developed in the Graphics and Imaging Laboratory from the University of Delaware.

Research Intern, *University of Delaware, Newark DE, United States* Jun. 2012 – Aug. 2012
Research and development in the area of Machine Learning under the supervision of Professor Kenneth Barner. I was a researcher in the topic of face recognition from 3D images. The aim of the project was to construct a data set of 3D images of faces. The data set is in process to be published and make it available to the academic community.

Research Assistant, *Los Andes University, Bogota, Colombia* Jan. 2012 – May. 2012
Research in the area of signal processing to design and implement a humidity measure system. The aim of the project was to study the use of non-linear circuits to enhance the performance of a commercial humidity sensor. Work on the Computer and Microelectronics Laboratory CMUA.

Research Assistant, *Los Andes University, Bogota, Colombia* Jul. 2011 – Dec. 2011
Research in the area of signal processing and machine learning to implement a non-supervised system for classifying stages of sleep. A spectral clustering strategy was used and several non-linear descriptors as feature extractors.

Other Experience

Teaching Assistant, *Los Andes University, Bogota, Colombia* Jan. 2010– Dec. 2010
Assistant teacher for the courses and the respective laboratories of: Circuits Analysis, Signal Processing, Introduction to the Engineering and Analogue Electronics.

Communications Engineer, *Global Crossing, Bogota, Colombia* Jan. 2009– Dec. 2009
Professional working for a telecommunication services provider to design and coordinate the installation of communication channels using several technologies such as Satellite, optic fiber, copper and radio.

Publications

Prada Juan, Sedano Nestor and Vizcaya Pedro. *Delta-Sigma High Resolution Analogue-Digital Converters Simulation.* In Proceedings of the XIII Simposio de Tratamiento de Senales, Imagenes y Vision Artificial (STSIVA 2008) Bucaramanga, Colombia. (Published in Spanish)

Prada Juan and Lozano Fernando.*Fourier Spectral Clustering.*In Proceedings of the XVI Simposio de Tratamiento de Senales, Imagenes y Vision Artificial (STSIVA 2011) Cali, Colombia. (Published in Spanish)

Prada Juan, Valderrama Mario and Lozano Fernando. Unsupervised sleep stage classification system.In Proceedings of the VI Seminario Internacional de Ingenieria Biomedica. Bogota, Colombia. (Published in Spanish)

Lopez Juan, **Prada Juan**, Alvarado-Rojas Catalina, Navarrete Miguel, Le Van Quyen Michel and Valderrama Mario. Identification of pre-ictal states based on an EEG-ECG multi-feature clustering approach. 6th International Workshop on Seisure Prediction, San Diego CA, United States. 2013. (Poster)

Prada Juan, Sasi Manju, Jablonka Sibylle, Dandekar Thomas and Blum Robert. An open source tool for automatic spatiotemporal assessment of calcium transients and local 'signal-close-to-noise' activity in calcium imaging data. (On submission to PLOS Computational Biology). 2017.

Honors and awards

Outstanding Scholars Award, *University of Delaware, Newark DE, USA* July 2013

Awarded by the Electrical and Computer Engineering department to prospective first year students showing exceptional promise in research.

Cum Laude Degree, *Los Andes University, Bogota, Colombia* March 2013

Degree awarded to the people with a GPA above the ninety percent of the historical accumulative of the Electric Engineering Department and with a Thesis project graded with the highest possible grade.

Ranked first on the M.Sc. class, *Los Andes University, Bogota, Colombia* March 2013

Honor awarded for being the graduate with the highest GPA on its class

Scholarship for outstanding student, *Los Andes University, Bogota, Colombia* January 2010

Scholarship awarded to students on the top ten of the official graduate examination of the Colombian government to the undergraduate students.

Other interests

I am a competitive triathlete, with the run as my strongest segment; I have participated in many races of all kinds and even won a couple of them. I am deeply interested in sports, besides from triathlon I play tennis, racquet ball, table tennis, soccer and chess. I am also very much interested in literature; I am huge fan of Jorge Luis Borges and Marcel Proust among many other writers. My favorite book today is Rayuela by Julio Cortazar, tomorrow might be other. I believe in being happy and help others to be happy as well, the way I see it one thing cannot happen without the other.

____

Juan Pablo Prada Salcedo