

Artificial intelligence meets diabetes at the micro-scale: forecasting complications and quantitatively mining insulin granules motions

Candidate: **Ermanno Cordelli**

Supervisor: **Prof. Paolo Soda**

A Thesis submitted in partial fulfillment of the requirements for the
degree of Doctor of Philosophy in Bioengineering and Bioscience

Computer Science and Bioinformatics Laboratory
Department of Engineering
Università Campus Bio-Medico di Roma

XXXI cycle

Academic Year 2015-2016

December 13th, 2019

Ermanno Cordelli

Tesi di dottorato in Bioingegneria e bioscienze, di Ermanno Cordelli,
discussa presso l'Università Campus Bio-Medico di Roma in data 13/12/2019.
La disseminazione e la riproduzione di questo documento sono consentite per scopi di didattica e ricerca,
a condizione che ne venga citata la fonte.

*Ad ignotum
per notum.*

Ringraziamenti

Quando ho iniziato questo periodo di dottorato avevo molti dubbi sul mondo della ricerca. Adesso i dubbi sono rimasti, ma il percorso per provare a levarli mi ha permesso di formarmi, buona parte dal punto di vista professionale, una parte ancora migliore sul lato umano. Per questo sono orgoglioso di poter ringraziare per iscritto le più vicine e da tempo presenti persone del Laboratorio che mi hanno aiutato a crescere.

Vorrei ringraziare il Professor Giulio Iannello per la professionalità, la competenza e l'efficienza che ha sempre mostrato, ma specialmente per la capacità di mettere davanti ad ogni cosa ciò che è meglio per la coscienza piuttosto che per la funzionalità.

Il Professor Luca Vollero è sempre stato una garanzia di onestà sia intellettuale che caratteriale, oltre che una delle persone più appassionate e preparate che abbia avuto il piacere di conoscere.

Nonostante Ludovica mi abbia accompagnato solo per i primi tempi di questa esperienza, oltre ad essere stata la persona assieme alla quale sono entrato come dottorando, con lei ho condiviso molte delle situazioni più importanti di questi anni, che hanno rafforzato la mia stima in lei e permesso di conoscerla come la persona vincente che è.

Un grazie particolare va a Mario, il quale si è sempre comportato da amico, collega e fratello maggiore allo stesso tempo, condividendo in modo sincero e leale paure e speranze di questo percorso e confrontandosi con me su tutte le decisioni più importanti.

Ringrazio il Professor Pierangelo Afferni per avermi insegnato molto della sua cultura ed esperienza in ambito scientifico e a volte anche di vita.

Per poter ringraziare Roberto dovrei prima raccontare tutti i momenti in cui è stato fondamentale e ancora fondamentale per il laboratorio e per me, ma visto che si è dimostrato presente sempre, sia come amico che come maestro e spesso mettendo in secondo piano i suoi stessi interessi, non posso fare altro che dirgli un gigantesco “a buon rendere”.

Un grandissimo grazie è poi per Rosa, la mia “compagna” di dottorato, è la persona con cui ho condiviso e costruito più esperienze sul campo in questi anni, vivendo insieme idee e difficoltà incontrate e con la quale sono molte volte nel mio piccolo cresciuto.

Nonostante fin qui l'ordine alfabetico, se è vero che i nomi più importanti vanno alla fine, non posso che ringraziare ora il Professor Paolo Soda, che oltre ad essere il mio supervisore è da quando lo conosco un affidabile, coerente e protettivo punto di riferimento.

Delle molte altre persone importanti presenti e passate del laboratorio ho il piacere di ringraziare Alessio, Andrea, Beatrice, Domiziana, Federico, Luca, Natascha e Sara, i quali spero di continuare a conoscere per continuare a rinnovare il già attuale rapporto di sincera stima.

Un grazie a parte è inoltre per il Professor Giuseppe Maulucci, per mezzo del quale ho iniziato e portato avanti gli studi oggetto di questo elaborato ed a Flavio e Gabriele, per la loro sempre presente solidarietà.

In ultimo, un grazie immenso ai miei genitori, costante sfondo di questo periodo felice.

Contents

1	Introduction	8
1.1	Type 1 and type 2 diabetes mellitus	9
1.2	Personalised medicine and microscopy advanced techniques	10
2	T1DM disease first exploratory investigation	12
2.1	Chapter overview	12
2.2	Introduction	12
2.2.1	Red blood cell	12
2.2.2	Peripheral mononuclear cell	13
2.2.3	Functional fluidity of cellular membranes	14
2.3	Dataset	15
2.4	Methods	17
2.4.1	Image classification	17
2.4.2	Feature computation	18
2.4.3	Principal Component Analysis (PCA)	21
2.4.4	Reliability computation of the base classifier	22
2.5	Patient classification	23
2.6	Results	23
2.7	Conclusions	24
3	T1DM disease second exploratory investigation	25
3.1	Chapter overview	25
3.2	Introduction	26
3.3	Dataset	27
3.4	Methods	28
3.4.1	Image classification	30
3.4.2	Feature computation	32
3.4.3	Principal Component Analysis (PCA)	36
3.4.4	Classification reliability	36
3.4.5	Patient classification	37

3.4.6	Combiner	37
3.5	Experimental setup	38
3.6	Results	39
3.7	Discussion	41
3.8	Conclusion	44
4	Insulin granules tracking	45
4.1	Chapter overview	45
4.2	Introduction	46
4.3	Materials	49
4.3.1	Cell cultures and glucose stimulation	49
4.3.2	Images Acquisition	49
4.3.3	Dataset	50
4.4	Methods	50
4.5	Automatic tracking	50
5	Visual4DTracker: a software for manual tracking	53
5.1	Background	54
5.2	Computational methods	55
5.3	Software description	57
5.4	Volume visualization	59
5.5	Handling, proofreading and evaluating the traces	60
5.6	Step by step software guide	61
5.7	Installation	62
5.8	Loading of 4D stack and traces	62
5.9	Volume handling	64
5.10	Slice visualization	66
5.11	Traces handling	67
5.12	Traces definition	69
5.13	Saving 4D traces	71
5.14	Performance evaluation	71
5.15	Other utilities	73
6	T1DM - Tracks features extraction and dynamics clustering analysis	74
6.1	Features extraction	74
6.2	Exploratory clustering analysis	77
6.3	Results and Discussion	79

6.4 Fully quantitative clustering analysis	81
6.5 Internal clustering evaluation indexes	83
6.6 Results and Discussion	86
6.7 Conclusion	90
7 Conclusions	91
Bibliography	93
Appendix A	103
Appendix B	105

1 Introduction

In computer science, roughly speaking, Artificial intelligence (AI) is the simulation of human intelligence processes by machines in contrast to the natural intelligence displayed by humans. Colloquially, the term artificial intelligence is often used to describe machines (or computers) that mimic *cognitive* functions that humans associate with the human mind, such as *learning* and *problem solving* [1]. Machine Learning (ML) is a subset of AI that gives computers the ability to learn from data. In recent decades it has overwhelmed many aspects of our lives including the healthcare of patients.

The principal health-oriented applications of ML can be summarized as follows [2]:

- *Supporting in Medical Imaging*: machine learning and deep learning have been one of the most remarkable breakthroughs fostering the development of Decision Support Systems supporting healthcare specialists. For example, Radiomics is an emerging translational field of research aiming to extract mineable high-dimensional data from clinical images [3].
- *Personalized Medicine*: the promise of personalized medicine is a world in which everyone's health recommendations and disease treatments are tailored based on their medical history, genetic lineage, past conditions, diet, stress levels, and more;
- *Scaled Up / Crowdsourced Medical Data Collection*: there is a great deal of focus on pooling data from various mobile devices in order to aggregate and make sense of more live health data;
- *Drug Discovery*: while much of the healthcare industry is a morass of laws and criss-crossing incentives of various stakeholders, drug discovery stands out as a relatively straightforward economic value for machine learning healthcare application creators;
- *Robotic Surgery*: this device allows surgeons to manipulate dextrous robotic limbs in order to perform surgeries with fine detail and in tight spaces (and with less tremors) than would be possible by the human hand alone;

- *Improving Performance*: from a more general point of view, machine learning may be implemented to track worker performance or stress levels on the job, as well as for seeking positive improvements in at-risk groups (not just relieving symptoms or healing after setbacks).

Within this scenario, I have focused most of my efforts on ML and the main topics I have dealt with during my PhD concern the development of AI-based methods for diabetes and for oncology. In the field of diabetes I studied diabetes mellitus in general and its complications with particular focus on β -cells' membrane, insulin physiology and insulin granules' lifecycle in physiological environment as well as in glucose solutions. In the field of oncology I gave contributions to radiomics, dealing with oncologic images and investigating on local advanced rectal cancer, non-small cells lung cancer, prostate cancer and acoustic neuroma, a tumor of the auditory vestibular nerve. In this thesis I present the results of my research on AI applications for diabetes, while at the end I also report the abstracts of my other works.

1.1 Type 1 and type 2 diabetes mellitus

Diabetes is a chronic, metabolic disease characterized by elevated levels of blood glucose (or blood sugar), which leads over time to serious damage to the heart, blood vessels, eyes, kidneys, and nerves [4].

In 2011 and 2012, the annual incidence of diagnosed diabetes in youth was estimated at 17,900 with type 1 diabetes and 5,300 with type 2 diabetes. In 2014 the prevalence of diabetic adults was 8.5% and was estimated that over 1.6 million of deaths were directly caused from this disease, whereas for the age group 0-19 years, the IDF Diabetes Atlas 2017 provides a global number of 1,106,500 people with type 1 diabetes with an annual incidence of 132,600 newly diagnosed cases.

In a healthy subject the insulin hormone is regularly synthesized in the pancreas, specifically into β -cells, and stored inside them in order to be rapidly released into the blood when the glucose level increases. Then the hormone can reach several body districts through the capillaries and in those specific districts stimulate the absorption of exceeding glucose, thus lowering the global level of sugar. A diabetic subject is not capable to maintain a proper insulin production or guarantee its effectiveness and therefore he/she usually presents abnormal levels of sugar in the blood.

It is possible to identify three different types of diabetes:

- Type 1 Diabetes Mellitus (T1DM): also known as juvenile diabetes, it is characterized by a lack of insulin production in the pancreas and results from autoimmune

mediated destruction of the beta cells of the pancreas. Its symptoms are usually frequent urination, increased thirst, increased hunger, and weight loss.

- Type 2 Diabetes Mellitus (T2DM): it is also known as adult-onset diabetes, this is a form in which the target organs show a resistance in absorption of the produced insulin and its secretion is disordered. The caused symptoms are similar to T1DM.
- Gestational Diabetes: in this form women without diabetes present an increase of glucose level during their pregnancy period, but it is usually not a permanent condition.

Subjects suffering from diabetes conduct a limited life, must follow a constant diet and require regular medical check-ups. Moreover, in both advanced T1DM and T2DM diseases there is an higher risk to develop severe complications *e.g.* cardiovascular disease, nerve damage, kidney damage and eye damage, and in the worst situation death. The incidence of various complications of T1DM and T2DM were: retinopathy 77.4 vs. 78.0/1000 person years, nephropathy, 62.0 vs. 58.8, neuropathy 7.8 vs. 13.9 and ischemic heart disease 1.2 vs. 5.4 [5].

1.2 Personalised medicine and microscopy advanced techniques

Unfortunately no cure exists for T1DM and the best way to tackle T2DM is prevention. Therefore, the usual disease treatment consists in delivering variable amounts of insulin using continuous monitoring and insulin injection systems.

In order to track the evolution of glucose levels inside blood vessels and regulate the quantity of administered insulin, a frequent or often continuous monitoring is required. This leads to the necessity to have a self blood monitoring glucose measurer (SBG) or a continue glucose monitoring system (CGM).

In both cases the control of sugar level during the day/night cycles presents always a moderate degree on uncertainty and the obtained measures do not always meet the level of analytical accuracy currently required for regulatory clearance [6]. In fact, in SBGs all the measures depend on their acquisition frequency and are therefore prone to humane error, instead the approach with a CGM brings the drawback to perform a glucose levels' measure in the subcutaneous tissue of the patient, which may differ from the actual glucose level in blood thus often needing to be integrated with a glucose trend by the patient, who is asked to use the finger pricker 5-6 times per day. Moreover, such

devices typically exploit proprietary algorithms, which are related to specific hardware components, which makes them expensive and non-affordable to a large audience.

It worth also notice that in both cases there is no personalization of the therapy, nor prediction of the general tendency of the disease over the patient life. On this ground some effort could be moved toward the direction of a tailored approach, in order to obtain a more punctual and reliable prediction of the disease in diabetic subjects.

Personalised medicine has been attracting research efforts since it can play an increasingly important role in disease diagnosis, treatment, prognosis and therapeutics. Its importance in clinical management is demonstrated by the recent introduction into routine clinical practice of various individualised therapies with increased efficacy and/or reduced toxicity [7].

In this scenario the technological advance has thrown the basis to gather the required information for a tailored treatment and a special position is occupied by confocal microscopy. It is a powerful tool that creates sharp images of a specimen that would otherwise appear blurred when viewed under a conventional microscope. Therefore, it has become an invaluable tool for imaging thin optical sections in living and fixed specimens ranging in thickness up to 100 micrometers [8].

In addition, fluorochromes matched to laser excitation premises to investigate specific molecules, reducing the acquisition noise produced by the environmental light scattering.

The technological progress observed in last years has led to the collection of a huge amount of heterogeneous data, making unfeasible their analysis without advanced approaches. Therefore, in order to cope with the described scenario, adequate instruments of analysis are required. All recent applications in data science that treat a large amount of data involve the development and optimization of computer-based approaches most of them leveraging on machine learning algorithms.

2 T1DM disease first exploratory investigation

2.1 Chapter overview

Investigation of membrane fluidity by two photon fluorescence microscopy opens up a new and important area of translational research, being a useful and sensitive method for disease monitoring and treatment. On this ground, we investigate if biomembranes in human red blood cells (RBC) and peripheral mononuclear cells (PMC) could be used as markers for type 1 diabetes mellitus (T1DM) diagnosis, leading to the development of a method for monitoring T1DM progression that nowadays is lacking, as clinical exams cannot pursue this task with enough reliability. To this aim, we present a set of features computed from PMC and RBC images that are given to a multi-experts system leveraging on multi-spectral information for positive/negative classifications. The experiments are carried out on a dataset of 800 blood cell images belonging to 18 subjects adopting the leave-one-person-out approach.

2.2 Introduction

In order to provide a better comprehension of the motivations that drove us to focus on the RBCs and PMCs, will follow a brief description of the cells' structure and functionalities.

2.2.1 Red blood cell

Red blood cells are the most common type of blood cell and the vertebrate's principal means of carrying oxygen (O_2) and carbon dioxide (CO_2) through the circulatory system.

Red blood cells are deformable, flexible and are able to adhere to other cells. In these terms their membrane plays many roles in in facts, this and these functions are highly dependent on the membrane composition. More specifically, the red blood cell



Figure 2.1: Illustration of the biconcave disk shape of red blood cells.

membrane is composed of 3 layers: the glycocalyx on the exterior, which is rich in carbohydrates, the lipid bilayer, which contains many transmembrane proteins and the membrane skeleton, a structural network of proteins located on the inner surface of the lipid bilayer, that is mainly constituted of actin and spectrin. The membrane mass is in half composed of proteins, whereas the other half are lipids, namely phospholipids and cholesterol [9].

The composition of the lipid bilayer is important as it defines many physical properties such as membrane permeability and fluidity. The RBCs' function, indeed, is to take up O_2 in the lungs and release it into tissues, as well as bring CO_2 from tissues to lungs, while squeezing through the body's capillaries. For this purpose, the cytoplasm of the cells is rich in hemoglobin, an iron-containing biomolecule that can bind oxygen and is responsible for the red color of the cells and the blood and the typical biconcave disc shape of their membrane provides a surface area to volume ratio that is optimal for gas exchange (as shown in Fig. 2.1). However, due to their blood flow path, all cells are exposed to high amounts of shear force; in fact, in order to withstand such stress, especially experienced during capillaries circulation, the specialized cytoskeleton is deformable enough to provides them the required mechanical stability and flexibility [10].

2.2.2 Peripheral mononuclear cell

Peripheral mononuclear (blood) cells are a diverse mixture of highly specialized immune cells that originate from hematopoietic stem cells that reside in the bone marrow and are carried in the blood flow through the body. The PMCs encompass a heterogeneous cell population as T cells, B cells, NK cells (types of lymphocytes), dendritic cells and monocytes and all of them play key roles in keeping our bodies healthy defending the

body against viral, bacterial, and parasitic infection, as well as destroying tumor cells and foreign substances.

The predominant use PBMCs in research involve areas relating to immunology, infectious disease, hematological malignancies, vaccine development, transplant therapy, personalized medicine and toxicology [11].

In mammals, cell membranes consist of a double layer of phospholipids, into which fatty acids are incorporated interspersed with cholesterol. The cells maintain a specific trans-membrane phospholipid distribution, with phosphatidylcholine and sphingomyelin (lipid compounds) contained on the outer side of plasma membranes, and phosphatidylethanolamine, phosphatidylserine and phosphatidylinositol (phospholipids) on the inner leaflet. Loss of membrane phospholipid asymmetry is associated with many physiological and pathological phenomena and changes in the membrane phospholipid composition can affect phagocytosis, T-lymphocyte signaling and antigen presentation. Phagocytosis is an important mechanism in many cells for the elimination of microorganisms or foreign particles, and membrane fluidity plays an important role in this process [12].

2.2.3 Functional fluidity of cellular membranes

Organisms modify the order, or fluidity, of their cellular membranes in response to changes in their physiochemical environment by adjusting the lipid composition of their membranes. Fluidity is a broad term adopted to monitor the membrane state, which depends on the membrane structure, curvature, microviscosity as well as its phase, the lipid shape, packing and composition. The maintenance of plasma membrane fluidity is a prerequisite for function, viability, growth and reproduction of cells [13]. Membrane fluidity was shown to have a decisive role in the efficiency of ligand binding, in the outcome of direct cell to cell contacts and in the modulation of the activity of membrane enzymes. Increased release of free radicals and reactive oxygen species affects membrane fluidity, cellular Ca^{2+} homeostasis, induces lipid peroxidation and finally cell death [14–16]; moreover, membrane fluidity changes have been reported in cancer and in neurodegenerative and metabolic diseases [17–19]. In this context, the assessment of membrane fluidity in cells isolated from human blood, as the red blood cells (RBC) and the peripheral mononuclear cells (PMC) opens up a new and important area of translational research for treatment and diagnosis of disease: it is now becoming clear that blood cells membranes can sense metabolic stress in patients and serve as biomarkers of metabolic dysfunction in human pathologies. Membrane fluidity alterations was indeed found in RBC and PMC as strongly interrelated with the severity of neurological outcome in mul-

tiple sclerosis [20,21]. However, these papers employed tools needing an expensive and time consuming preparation, and the detection techniques are particularly slow, making their use senseless for translational applications. Investigation of membrane fluidity by Laurdan two photon fluorescence microscopy and other functional imaging techniques is proved to be a useful and sensitive method to monitor and establish the mechanisms by which cellular functions are affected [18, 22, 23]. Laurdan, a fluorescent dye highly sensitive to the presence and mobility of water molecules within the membrane bilayer, yields information on membrane fluidity by a shift in its emission spectrum [24,25], and has the potential to monitor disease progression with important advantages such as velocity, high specificity, resolution and fine subcellular targeting. However, determination of personalized disease progression and therapeutic effectiveness asks that the experts look at a large amount of data: this procedure, further to be time consuming, is prone to error and utterly impractical due to huge amount of available images collected by the microscopes, also because the useful information should be embedded in the raw images. The promising outcomes of both such translational studies and the application of computational algorithms to the problem of diabetes monitoring [26] have motivated us to investigate if images of RBC and PMC contain information to establish differences between healthy and pathologic subjects with the potential to monitor type 1 diabetes mellitus (T1DM) progression. This is especially important for this disease, since traditional diagnostic methods (glycated haemoglobin, glucose-based tests and measurements based to the detection of specific antibodies) are not enough sensitive, or are based on multiple analysis that need a correlation with other clinical evidences [27]: a diagnostic system, giving the outcome from a single and very sensitive analysis, it would be essential for improving healthcare system. In this paper we therefore present a multi-experts system (MES) that allows distinguishing between patients and healthy subjects which makes use of a set of texture features computed over the images.

2.3 Dataset

Blood samples were obtained after consent from 8 controls and 10 patients. Upon collection, the RBC and PMC will be isolated from blood by density gradient centrifugation, counted, seeded in an uncoated two well dish (RPMI-1640 5% FCS) and labeled with Laurdan, 2-dimethylamino(6-lauroyl) naphthalene (Laurdan, Molecular Probes, Inc., Eugene, OR, USA), a fluorescent dye highly sensitive to the presence and mobility of water molecules within the membrane bilayer, yielding information on membrane fluidity by a shift in its emission spectrum [24]. Laurdan exhibits an emission spectral shift

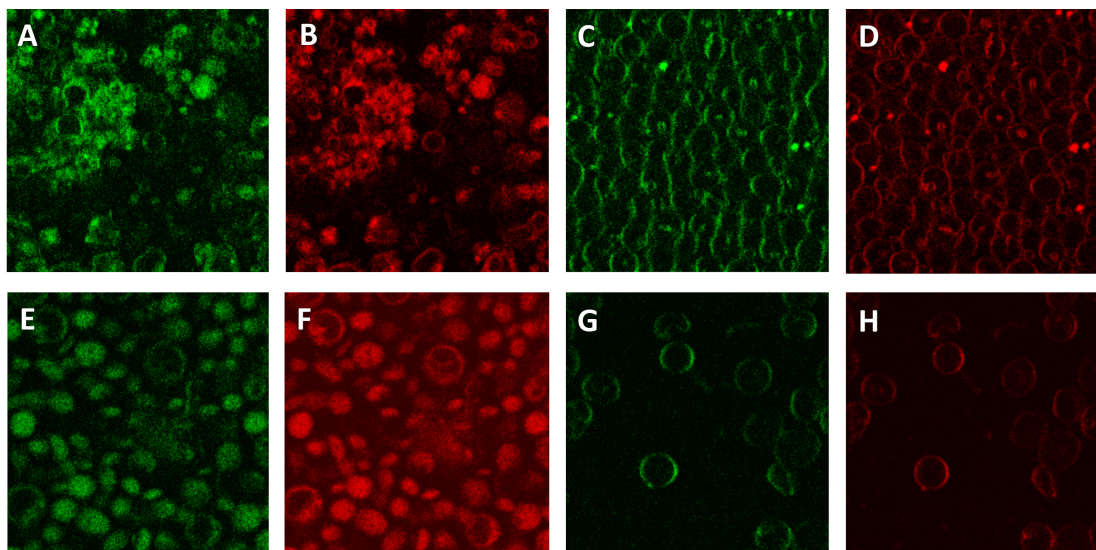


Figure 2.2: Examples of the images. Panels A, B, E, and F: PMC (green and red channels). Panels C, D, G and H: RBC (green and red channels). First row shows images acquired from a healthy subject, whereas second row shows the images of T1DM patient.

depending on the lipid phase state, i.e. bluish in ordered, gel phases and greenish in disordered, liquid-crystalline phases. Laurdan distributes equally between lipid phases and does not associate preferentially with specific fatty acids or phospholipid headgroups. 10 μl of Laurdan stock solution was added per milliliter of cell culture medium. Laurdan intensity images were obtained with an inverted confocal microscope (DMIRE2, Leica Microsystems, Germany) using a 63x oil immersion objective (NA 1.4) under excitation at 800 nm with a mode-locked Titanium-Sapphire laser (Chameleon, Coherent, Santa Clara, CA). The voxel size is $230 \times 230 \times 2500 \mu\text{m}^3$. Internal photon multiplier tubes collected at a 400 Hz scan speed five images per well at different z-axis positions; the images are stored using 8 bit and they are saved as uncompressed tif. Image resolution is 512×512 . In order to detect the spectral shift, Laurdan intensity images were recorded simultaneously with emission in the range of $400 \div 460$ nm (ch01-Green) and $470 \div 530$ nm (ch02-Red) and imaging was performed at 37°C . At the end of the acquisition, we collected 800 images in total, equally divided between the RBC and PMC. Figure 2.2 shows some example of the images.

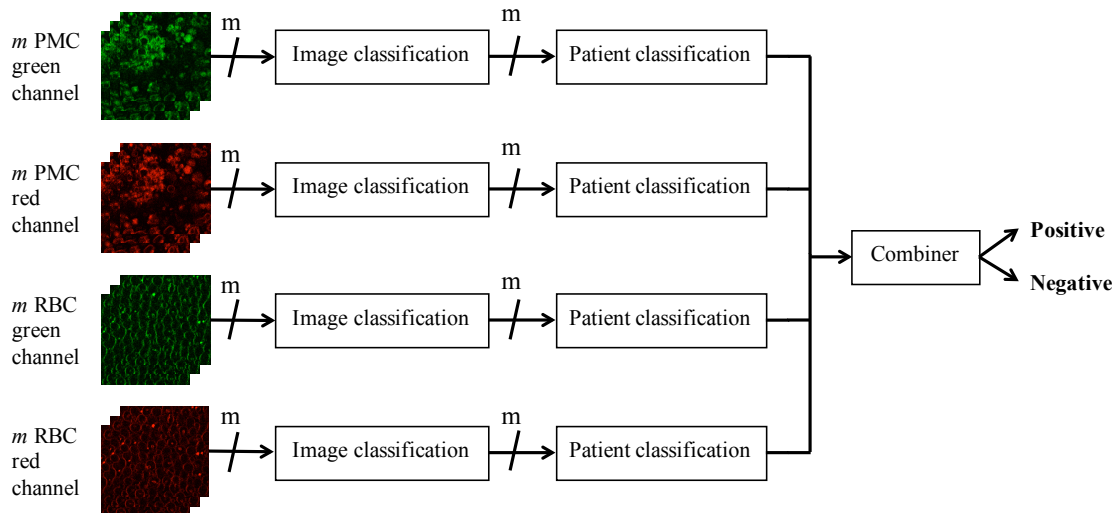


Figure 2.3: Schematic representation of the multi-spectral pipeline used to classify the data.

2.4 Methods

We are interested in discovering if the space-resolved spectral shift determination, obtained on the PMC and RBC membranes by acquiring simultaneously the green and the red channel, allows distinguishing between T1DM patients and healthy subjects. As reported in section 2.3, we have available four kinds of images that we employ in a multi-spectral fashion to get information on the patient, as depicted in Figure 2.3. Indeed, the figure shows that for each image type we first classify the whole set of images in the stack (*image classification* block) and then, the labels assigned to the images of the stack are combined to assign a label to the patient (*patient classification* block). The decisions provided by each of the four pipelines are aggregated in the *combiner* block so that multi-spectral information is exploited. It is worth noting that this final classification stage realizes a multi-expert system (MES) as it combines the classification outputs provided by four classification systems.

Next subsections now describes each of the block represented in Figure 2.3. Indeed, section 2.4.1 describes the image classification block detailing the texture image features we considered as well as the classifier employed, whereas section 2.5 details how we compute the patient classification.

2.4.1 Image classification

This block operates in two modes, namely training (learning) and testing (classification), which are depicted in Figure 2.4.

In the training mode, we have the training images $\{I_{tr}^i\}_{i=1}^m$ and the corresponding binary labels $\{y_{tr}^i\}_{i=1}^m$, where y can be 0 or 1. The *feature computation* block computes for each I_{tr}^i the set of features \mathbf{x}_{tr}^i that will be presented hereinafter (section 2.4.2); then, the *principal component analysis* (PCA) block finds the principal components (section 2.4.3). Finally, a binary classifier learns how to partition the transformed feature space using training data.

In the classification mode, given a test image I_{te} or a collection of images $\{I_{te}^i\}_{i=1}^n$, we first compute the features \mathbf{x}_{te}^i which are then projected into the new feature space given by the PCA. The classifier then assigns to each I_{te}^i a label $O(I_{te}^i)$, which can be 0 or 1, and a reliability measure $\phi(I_{te}^i)$. This latter quantity, that we will detail in section 2.4.4, ranges in $[0; 1]$ where 0 represents an unreliable classification, whereas 1 characterizes very reliable classification. It is worth noting that it has demonstrated its usefulness in different fields of pattern classification, such as the application of different fusion rules [28] and the assessment of the quality of the outputs provided by the classifier [29, 30].

Next subsections will focus on the three main steps of the image classification block: first, in subsection 2.4.2 we will describe the features we considered; second, subsection 2.4.3 introduces the PCA, whereas subsection 2.4.4 describes how to compute the reliability of the classifier.

2.4.2 Feature computation

This section presents the overall set of descriptors computed in the training phase of the image classification block (Figure 2.3 and Figure 2.4). In the literature there exist several image descriptors applied to medical image analysis that are derived from either morphological or textural analysis [31–34]. Indeed, the integration of information of different nature provides a substantial improvement in the classification accuracy.

The visual inspection of the images suggests us to adopt a global approach computing the features from the whole image. The alternative consists in computing features after a segmentation step, but we withdrew this approach for the following reasons. First, the image intensity largely varies within the same image and between different images. Second, most of the information is located on the cell membrane and, hence, a miss-segmentation of cell border may lose potential useful information. Third, the images are affected not only by noise introducing variation in background illumination, but also by artifacts as revealed by the bright dots randomly diffused in the images.

The texture features we consider can be divided into the following families:

Statistical features These descriptors are based on first and second-order measures, as it has been shown that humans are sensitive to them [35].

The former features are the statistical moments up to the fourth order of the first-order image histogram, i.e., the mean, the standard deviation, the skewness, and the kurtosis. Furthermore, the picture of grey-level distribution is also grasped by the histogram width, the image energy, the entropy, the value of the histogram absolute maximum and the corresponding grey-level value, the energy around such maximum, the number of relative maxima in the histogram and their energy. The interested reader can find their formal definitions in [36].

Second-order features are derived from the the gray level co-occurrence matrix (GLCM) of each image: this quantity represents the distribution of co-occurring values between neighboring pixels according to different displacements [37]. Its statistics correlate well with the image structure [38]. Given a coordinate vector $\mathbf{r} = (x, y)$ in the image I , GLCM represents the distribution of probability of occurrence of a pair of gray-level values separated by a given displacement vector $\Delta\mathbf{r}_\theta$, i.e., $p(I(\mathbf{r}), I(\mathbf{r} + \Delta\mathbf{r}_\theta))$. Note also that $\Delta\mathbf{r}_\theta$ is defined as $(d \cos \theta, d \sin \theta)$, where d is an interval between pixel pair. We compute a set of texture features from GLCM, namely its autocorrelation, covariance, inertia, absolute sum of GLCM occurrences, inverse difference moment, energy and entropy. Their mathematical formulation is omitted here for space reason, while the interested reader may refer to [38].

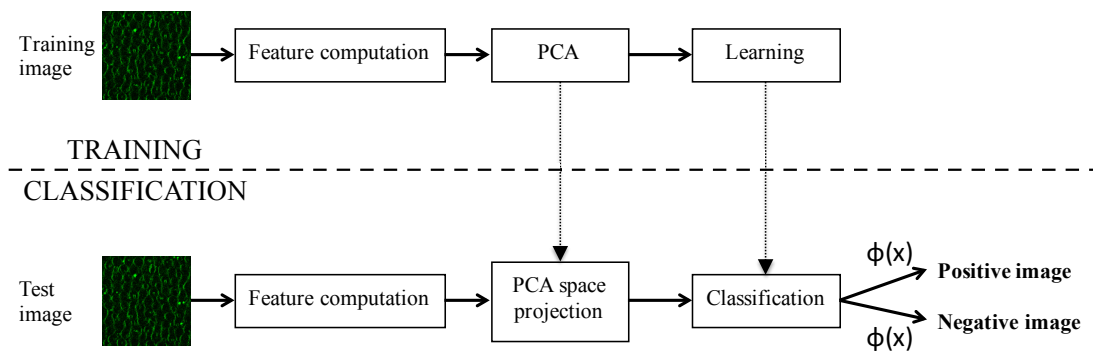


Figure 2.4: Details on the model used to classify each image (image classification block).

Rotation invariant co-occurrence local binary patterns Local Binary Pattern (LBP) is a grayscale texture descriptor that assigns to each pixel of the image a label obtained comparing it with its neighborhood matrix. Different types of LBP operators can be computed varying both the pixel arrangement and the number of neighbors, respectively. The standard LBP operator works in a 3×3 neighborhood square matrix [39] as follows.

Defining as g_c the value of the pixel with coordinates $\mathbf{r}_c = (x_c, y_c)$, and with g_p the value of its p -th neighbor at displacement \mathbf{d} , the LBP operator is defined as:

$$LBP(\mathbf{r}_c) = \sum_{p=0}^{P-1} 2^p \text{sgn}(g_p - g_c) \quad (2.1)$$

where P is the number of neighborhood pixels and $\text{sgn}(x)$ represents the sign function. Let us denote as d a scale parameter corresponding to the radius of the neighborhood so that the coordinates of g_p are given by $(x_c + d \cos \theta_p, y_c + d \sin \theta_p)$, where $\theta_p = p \cdot \frac{2\pi}{P}$. Hence, now the displacement vector from the reference pixel to neighboring pixels is $\Delta \mathbf{r}_{\theta_p} = (d \cos \theta_p, d \sin \theta_p)$. In our implementation we set $P = 4$ to reduce the computational costs: as a result, each LBP has $N_P = 2^P = 16$ possible patterns and it returns an image with the same size $N \times M$ of I , where its pixel at coordinate $\mathbf{r} = (i, j)$ stores the value $LBP(\mathbf{r})$. The image texture can be now represented building a histogram:

$$H(n) = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} f(LBP(\mathbf{r}), k), \text{ with } k \in [0; N_P] \quad (2.2)$$

and

$$f(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

The basic LBP operator is neither robust to changes in spatial resolution nor to texture rotations. For this reason we adopt the rotation invariant co-occurrence among adjacent LBPs (CM-LBP) obtained as follows [40]. First, we compute the co-occurrence matrix of the output LBP image; using the notation introduced above, this returns the quantity $p_\theta(LBP(\mathbf{r}), LBP(\mathbf{r} + \Delta \mathbf{r}_\theta))$, where the subscript θ implies that the probability depends upon the angle θ . We consider four orientations, namely horizontal, vertical and two diagonals corresponding to $\theta = 0, \pi/4, \pi/2, 3\pi/4$. This approach produces four histograms, each with N_P^2 dimensions. Second, as the cells have different orientations in the image, we ensure rotation invariance by attaching a rotation invariance label to each LBP pair. In order to find the rotation equivalent LBP pairs we have to consider the case (a) where LBP pairs of $\theta = 0, \pi/4, \pi/2, 3\pi/4$ have rotation equivalence and the case (b) where the LBP pairs rotated by π have rotation equivalence. For the former case (a), we denote a LBP pair $P_\theta(\mathbf{r}, \Delta \mathbf{r}_\theta)$ as follows [41]:

$$P_\theta(\mathbf{r}, \Delta \mathbf{r}_\theta) = (LBP_\theta(\mathbf{r}), LBP_\theta(\mathbf{r} + \Delta \mathbf{r}_\theta)) \quad (2.3)$$

where

$$LBP_\theta(\mathbf{r}) = \sum_{p=0}^{P-1} 2^p \text{sgn}(I(\mathbf{r} + \Delta_{\theta,p}) - I(\mathbf{r}))$$

and

$$\Delta_{\theta,p} = (d \cos(\theta_p + \theta), d \sin(\theta_p + \theta))$$

Note that θ acts as the bias of the rotation angle in LBP. According to these definitions, the LBP pair of each configuration has the same value in terms of rotation. In order to consider the latter case (b), i.e. the case where the LBP pairs rotated by π have rotation equivalence, we set $P_\theta(\mathbf{r}, \mathbf{r} + \Delta\mathbf{r}_\theta) = (LBP_{\theta+\pi}(\Delta_\theta + \mathbf{r}), LBP_{\theta+\pi}(\mathbf{r}))$ using a mapping function M . Finally, a rotation invariant label for an LBP pair at \mathbf{r} , denoted as RIC-LBP, is given by:

$$P_\theta^{RI}(\mathbf{r}) = M(P_\theta(\mathbf{r}, \Delta\mathbf{r}_\theta)) \quad (2.4)$$

Similarly to what reported in equation 3.2, an RIC-LBP histogram is generated from $P_\theta^{RI}(\mathbf{r})$ for the image I . This histogram has dimension equal to $\frac{N_P(N_P+1)}{2} = 137$, and it is used as feature vector.

2.4.3 Principal Component Analysis (PCA)

The features presented above produces for each image a feature vector \mathbf{x} composed of hundreds of elements. However, it is well-known that a limited yet salient feature set simplifies both the pattern representation and the classifiers that are built on the selected representation [42].

Although we are aware that a reduction in the number of features may lead to a loss in the discrimination power and thereby lower the accuracy of the resulting recognition system, the limited number of samples available in our case suggests that a small number of features can alleviate the curse of dimensionality. It is worth noting that we also experimentally verified this hypothesis, but for the sake of presentation these results are omitted.

In order to determine an appropriate subspace of dimensionality m in the original feature space of dimensionality d , with $m \leq d$, we apply the linear transform given by the principal component analysis (PCA) that computes the m largest eigenvectors of the $d \times d$ covariance matrix of the n d -dimensional patterns.

Since PCA uses the most expressive features, i.e. the eigenvectors with the largest eigenvalues, it effectively approximates the data by a linear subspace using the mean squared error criterion. The interested reader may refer to [36] for further details on the PCA.

To decide how to set m , we look at the percentage of variance retained for different values of m and we choose m so as to retain the 95% of the variance. It is worth noting that what we have described in this subsection is applied in the training phase of the

Data	accuracy	precision	recall
PMC green channel	94.4	90.9	100.0
PMC red channel	94.4	90.9	100.0
RBC green channel	94.4	100.0	90.0
RBC red channel	94.4	90.9	100.0
Combiner	100.0	100.0	100.0

Table 2.1: Average performance rates of the classification system.

image classification block, i.e., in the uppermost part of Figure 2.4. In the test phase, i.e., the lowest part of the figure, once we have computed the d features from the images we simply project them into the PCA space with dimensions m .

2.4.4 Reliability computation of the base classifier

Measurement classifiers¹ besides labeling each sample, may supply other information typically related to the degree that the sample belongs (or does not belong) to the corresponding class, that can be used to compute the reliability score $\phi(\mathbf{x})$, where \mathbf{x} is the input pattern. Note that the assumption on measurement classifier is not a limitation as it is always possible to obtain a measurement for each classification act of any kind of classifiers. In our work the base classifier is a Support Vector Machines (SVM), whose kernel is a radial basis function; our choice is motivated by the fact that such a learning algorithm has demonstrated strong discriminative ability in binary classification problem [43]. We use the default values of classifier parameters available in the LIBSVM library [44]. We also apply the Platt's method [45] transforming the distance of the input pattern \mathbf{x} from the optimal separating hyperplane induced by the chosen kernel in the feature space into a posterior probability $\xi_i(\mathbf{x})$, for each class $i = 1, 2$. In practice, if the distance between the separating hyperplane and \mathbf{x} is very large, we can have a high degree of confidence that it was correctly classified, i.e. $\xi_i(\mathbf{x}) \gg \xi_j(\mathbf{x})$, with $i \neq j$. On the contrary, if a sample is very close to the hyperplane separating the two classes, both $\xi_1(\mathbf{x})$ and $\xi_2(\mathbf{x})$ will be close to 0.5. Then, these posterior probabilities are combined to compute the reliability score as $\phi(\mathbf{x}) = |\xi_1(\mathbf{x}) - \xi_2(\mathbf{x})|$: hence, the larger the difference between the class posterior probabilities, the larger the sample classification reliability.

¹The various classification algorithms can be divided into three categories: type I (abstract), that supplies only the label of the presumed class, type II (rank) that ranks all classes in a queue where the class at the top is the first choice, type III (measurement) that attributes each class a value that measures the degree that the input sample belongs to that class.

2.5 Patient classification

The patient classification block, either in the training or in the classification mode, takes as inputs both the classification labels and the reliability scores computed by the image classification stage. It aims at providing a unique label for each patient integrating all the labels assigned by each expert to patient images.

To tackle with this issue, the patient classifier adopts the weighted voting (WV) rule, among the many schemes proposed in the literature [28]. WV is a voting technique in which the opinion of each classifier about the class of the input pattern $O(I_{te}^i)$ is weighted by the reliability parameter $\phi(I_{te}^i)$.

Since each image classifier deals with a binary learning task, we now present a simplified notation of the WV rule where the weighted sum of the votes for each of the two classes is given by:

$$W_h(x) = \sum_{i:O(I_{te}^i)=h} \phi(I_{te}^i), \text{ with } h = \{0, 1\} \quad (2.5)$$

where x denotes the patient under test. Therefore the fusion output, $Y(I_{te})$, is 1 if $W_1(x) > W_0(x)$, 0 otherwise².

2.6 Results

The experiments were performed on the dataset presented hereinbefore (section 2.3) using a leave-one-person-out cross-validation approach. For each cross-validation run our approach is trained on all the data except for all the images belonging to one patient. In the test phase a classification is then made for that patient. This procedure guarantees that for each iteration the images of the same patient are not in the training and test set, that is, the separation of training and test data is done at patient level. The classifier is a SVM, as described in section 2.4.4. Its performances are measured in terms of accuracy, precision and recall and averaged out the cross-validation runs.

As reported in Figure 2.3 we use a multi-spectral approach leveraging on the information provided by four image types, namely RBC image fluidity, RBC image rigidity, PMC image fluidity, and PMC image rigidity: Table 2.1 shows the results achieved. The table is organized in two section: the first is composed of four rows and it reports the recognition performance achieved using only one of the four types of images. With

²In case of tie, i.e. if $W_1(x)$ is equal to $W_0(x)$, the output $Y(I_{te})$ is set arbitrarily to zero. Note that this case has never occurred in all tests we performed.

reference to Figure 2.3, these results are attained by each of the experts, i.e., each of the patient classification blocks. It is worth noting that each of the four experts achieves large accuracy, whose magnitude is always the same: further to the values of the figures of precision and recall, a careful investigation of the data reveals that each of them misclassifies samples that belong to two different patients. The second section of the table, composed of only one row named as “Combiner”, reports the final performances.

The results show that the use of a MES architecture leveraging on the multi spectral information provided by the four kinds of available images permit to successfully distinguish between controls and patients affected by T1DM. As a consequence, we deem that the physical state of RBC and PMC membranes contain information related to T1DM Progression. These findings reveal that the spatially resolved multispectral analysis of Laurdan fluorescence emission of blood cells has the potential to discriminate the presence of the disease with an higher degree of confidence with respect to traditional methods.

2.7 Conclusions

Here we have investigated if the physical state of PMC and RBC membranes can be used for positive/negative classifications in T1DM disease. Indeed, profiling of individuals' biomolecules of blood cells has provided new strategies for disease monitoring and treatment. The experiments carried out on a dataset of 800 blood cell images belonging to 18 subjects have shown that the proposed multi-experts system leveraging on multi-spectral information and using texture features correctly classifies all the available samples. Although the achieved results are quite promising, the path for the clinical practice is still long and complex. Indeed, there are many questions i.e. what could be the economical advantage of using this approach compared to the cost of current diagnostic procedures, or how much of the technology required by the proposed methodology is actually usable by non computer technician experts. Moreover, the effective improvement the patient would achieve by using this approach should be quantified in terms of prognostic results against the disease evolution. In our opinion, to allow the enter in the clinical practice, the entire pipeline, from the image acquisition to the patient classification, should be composed in an unique solution, possibly providing an optimized microscope with the artificial intelligence already embedded inside. Doing so, the interested physician wont be asked to know the internal functionalities of the diagnostic instrument to make it work, thus both the training and the procedural costs would be reduced and giving the patient only the benefit of a more accurate diabetes test.

3 T1DM disease second exploratory investigation

3.1 Chapter overview

Although the methods presented in previous chapter have already shown promising results, outperforming the diagnostic procedure currently employed in the clinical routine, they present some methodological lacks asking for a more exhaustive analysis. Therefore, in order to better understand the potential of membrane in detecting T1DM, we moved one step further and deepen more the effectiveness of our method, investigating if this classification approach permits not only to do positive/negative classifications, but also to distinguish between patients with and without any complications. Moreover, we collected the data from 9 new volunteers enlarging the dimension of the pool up to 27 patients. This translated into a more robust model and more reliable results. Finally, since the procedure that allows to separate blood's corpuscular components is usually more complicated and time expensive for PMCs than for RBCs, we investigated if we could use only the latter while obtaining again the satisfactory results.

On this basis, this chapter presents a decision support system that distinguishes between healthy subjects, type 1 diabetes mellitus patients, and type 1 diabetes mellitus patients with complications. It leverages on dual channel data computed from the physical state of human red blood cells membranes by means of features based on first- and second-order statistical measures as well as on rotation invariant co-occurrence local binary patterns. The experiments were carried out on a dataset of more than 1000 images belonging to 27 subjects.

Our method shows a global accuracy of 100%, outperforming also the state-of-the-art approach based on the glycosylated hemoglobin. The proposed recognition approach permits to achieve promising results.

3.2 Introduction

To improve the discriminative power between healthy subjects, T1DM patients, and T1DM patients with complications, in the last two decades decision support systems (DSSs) have been developed [26]. Indeed, the wide availability of analysis has offered plenty of data useful for diagnostics [46]. For instance, in [47] the authors designed a DSS that, using clinical features, detects if a patient shows signs of diabetes according to WHO criteria by means of the principal component analysis. Still using clinical attributes, other authors applied the AdaBoost algorithm to the same aim [48], whereas there are also attempts leveraging on a fuzzy approach [46]. Whilst most of the efforts have been directed towards DSSs employing clinical features and detecting signs of diabetes, the recent availability of fluorescence imaging-based diagnostic strategies has opened the chance to develop DSSs for T1DM diagnostics, trying to overcome the low sensitivity of traditional methods. Besides common immunohistochemical and morphological classifications, metabolic functional imaging method can detect changes in metabolism, regional chemical composition, and phase state of cells [49–51]. For T1DM diagnostics, the measurement of fluidity alterations can furnish a more sensitive index of disease progression than other detection methods [52]. Membrane fluidity is a quantitative measurement of the order of the lipid packing within the membranes. Uncontrolled glycemic fluctuations and oxidative stress commonly found in T1DM patients can impair membrane fluidity of blood cells, especially red blood cells (RBCs), and contribute to the development of T1DM complications [53]. RBC membrane fluidity can be investigated through Laurdan two-photon fluorescence microscopy of a liquid biopsy [54]: Laurdan is a lipophilic fluorescent probe whose emission spectrum undergoes to a shift towards longer wavelengths upon membrane fluidification. Through a dual channel acquisition of fluorescence a measure of fluidity can be obtained [18].

On these grounds, we present a DSS recognizing healthy subjects and T1DM patients with and without complications based on dual channel analysis of red blood cell membrane fluidity. Although preliminary results were reported in [55], this paper extends previous contribution in some respects. First, in [55] we used images of RBCs and peripheral mononuclear cells, whereas here we leverage only on RBC images to reduce the acquisition burden and to lower costs. Second, previous work studied how a classifier-based approach can discriminate healthy subjects and T1DM patients; conversely, here we consider also T1DM patients with complications. Third, the DSS presented here was tested on a dataset larger than the one used in [55]. Fourth, the performance of the DSS are compared here also against those attained by the HbA1c tests.

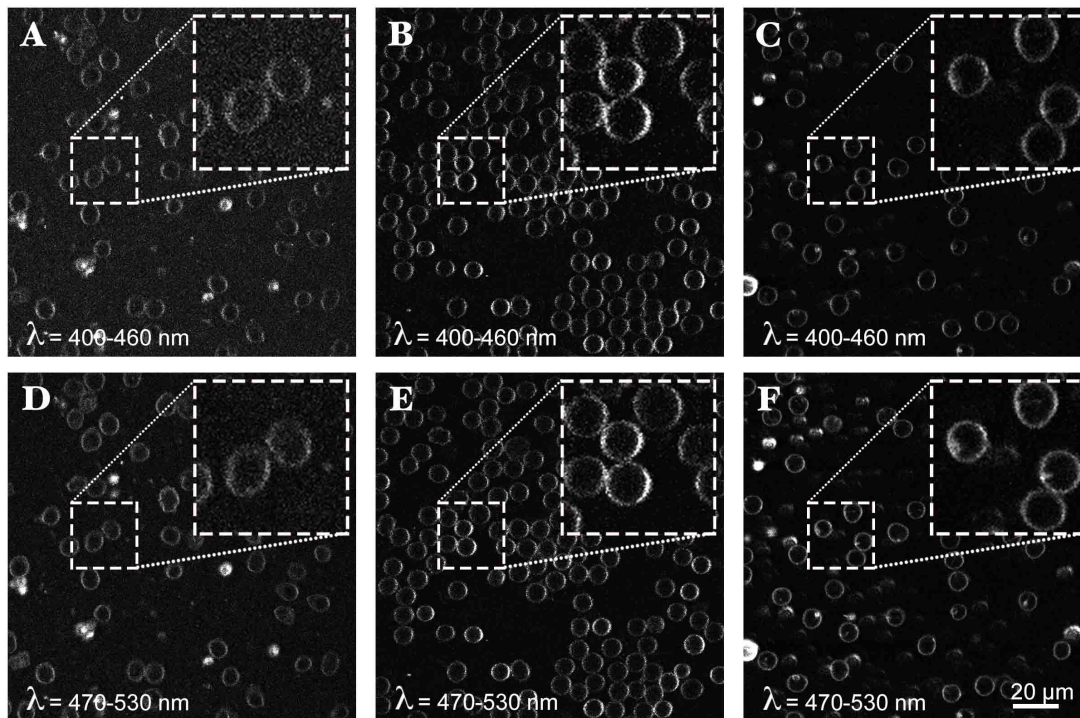


Figure 3.1: Portions of dual channel RBC images. Images in panels A and D belong to healthy subjects, images in panels B and E belong to T1DM patients, and panels C and F show images belonging to T1DM patients with complications. Furthermore, panels A, B and C show image portions from Ch1 ($\lambda = 400-460 \text{ nm}$), whereas images in panels C, D and E belong to Ch2 ($\lambda = 470-530 \text{ nm}$). The scale bar corresponds to $20\mu\text{m}$, while each particular shown in the dashed square is zoomed 2.5 times.

3.3 Dataset

Blood samples were obtained after consent from 8 healthy subjects, 10 T1DM patients and 9 T1DM patients with complications. In the following, these three classes are referred to as *Healthy*, *T1DM* and *T1DMwC*, respectively. Upon collection, the RBCs were isolated from blood by density gradient centrifugation, counted, seeded in an uncoated two well dish (RPMI-1640 5% FCS) and labeled with Laurdan, 2-dimethylamino (6-lauroyl) naphthalene (Laurdan, Molecular Probes, Inc., Eugene, OR, USA), a fluorescent dye highly sensitive to the presence and mobility of water molecules within the membrane bilayer, yielding information on membrane fluidity by a shift in its emission spectrum [24]. Laurdan exhibits an emission spectral shift depending on the lipid phase state, i.e. bluish in ordered, gel phases and greenish in disordered, liquid-crystalline phases. Laurdan distributes equally between lipid phases and does not associate pref-

erentially with specific fatty acids or phospholipid headgroups. Laurdan stock solution was prepared by dissolving the probe Laurdan (Molecular Probes, USA) in DMSO, at a final concentration of 1 mM. 1 μ l of Laurdan stock solution was added per milliliter of cell culture medium. Laurdan intensity images were obtained with an inverted confocal microscope (DMIRE2, Leica Microsystems, Germany) using a 63x oil immersion objective (NA 1.4) under excitation at 800 nm with a mode-locked Titanium-Sapphire laser (Chameleon, Coherent, Santa Clara, CA). The pixel size is $230 \times 230 \mu\text{m}^2$. An expert biologist selected 5 zones per well and, which are then focused by an internal photon multiplier collecting 4 images per zone at different z-axis positions. The images have of resolution equal to $512 \times 512 \text{ px}$, they are stored using 8 bit and saved as uncompressed tif. To detect the spectral shift, Laurdan intensity images were acquired simultaneously recording emission in the range of $400 \div 460 \text{ nm}$ and $470 \div 530 \text{ nm}$, while imaging was performed at 37°C . The two ranges are hereinafter referred to as *Ch1* and *Ch2*, respectively. At the end of the acquisition, we collected 1084 RBC images. Figure 3.1 shows some example of the images.

Furthermore, on the same blood samples we collected also the HbA1c data via high-performance liquid chromatography [56].

3.4 Methods

We are interested in discovering if the information contained in the two channels of RBC membrane images allows distinguishing between T1DM patients, T1DM patients with complications and healthy subjects.

Since free cholesterol in RBC plasma membranes can exchange bidirectionally with plasma lipoprotein cholesterol [26,27], we can assume that plasma cholesterol content reflects RBC cholesterol content. In the submicrometric fluidity RBC maps shown in Figure 3.2, we observe a two-colored scales spanning from red (liquid-like) to green (gel-like) representing the increase in the blood cholesterol content. The histograms shown in panel B confirm this visual inspection since they exhibit a shift of the mean value towards more fluid regions as the disease progresses, although there is also an increase in variance. These results suggest that the total cholesterol content is inversely correlated with RBC fluidity values and that, looking at the intra-group variation in G1 and G2, cells' fluidity increases together with the cholesterol and LDL content as well, as evident from the negative slopes of the linear regressions to the data in Figure 3.3 (G0, G1 and G2 slopes are 0.0002, -0.0001, -0.0004 dl/mg respectively). Moreover, the low R2 values (0.0829, 0.0587 and 0.2788 for G0, G1 and G2 respectively) indicate that, with

the exception of cholesterol, the linear model has to account for other factors. LDL content increases following the same trend.

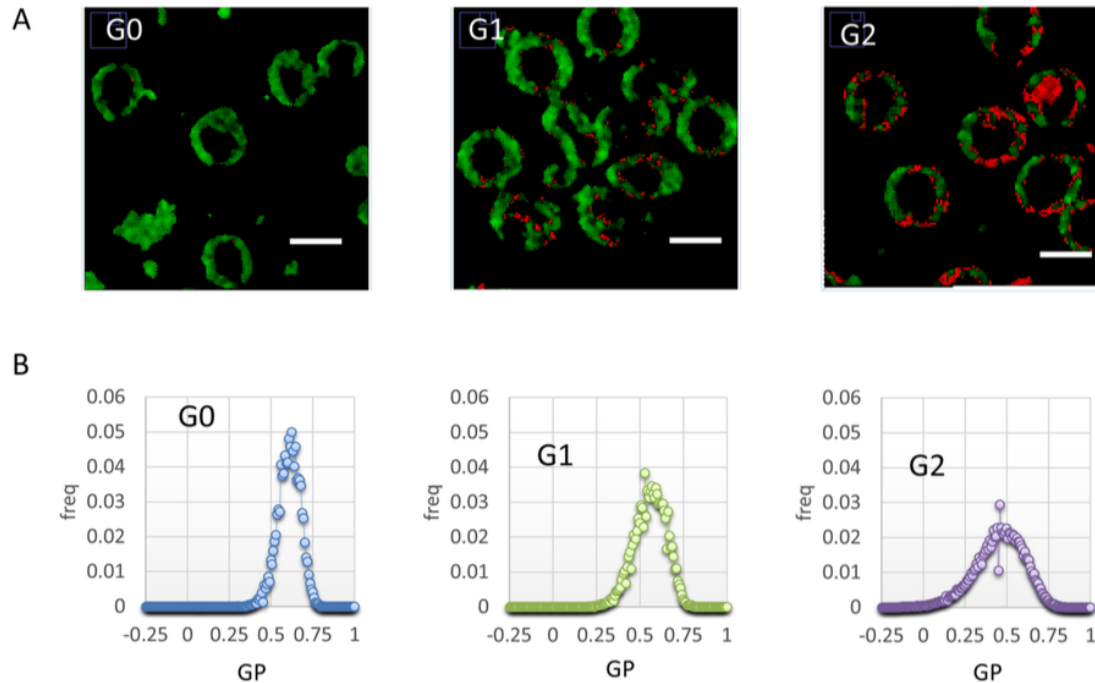


Figure 3.2: Sub-micrometric spatial distribution of RBC fluidity. (A) Submicrometric fluidity maps of RBC are reported, in a two-colored scale spanning from red (liquid-like) to green (gel-like). G0: control; G1: T1DM, no complications; G2: T1DM with complications. (B) GP Histograms of the corresponding images, showing a shift of the mean value towards more fluid regions going on with disease progression, but also an increase in variance.

On these grounds, we investigate how ML is useful to verify if RBCs can lead to the discrimination of T1DM patients, T1DM patients with complications and healthy subjects. In this respect, we adopted a dual channel approach, as depicted in Figure 3.4 A). Given a set of patient images, the *image classification* block classifies each image in the stack for each channel, assigning also a reliability measure to each decision. Next, for each channel, the labels and the reliabilities computed by this block are used by the *patient classification* block to set a label to the patient. Finally, the decisions provided by the two pipelines are aggregated in the *combiner* so that dual-channel information is exploited.

Next subsections now describe each of the block represented in Figure 3.4 A): section 3.4.1 describes the image classification block detailing the image features we considered as well as the classifier employed, whereas section 3.4.5 introduces how the patient

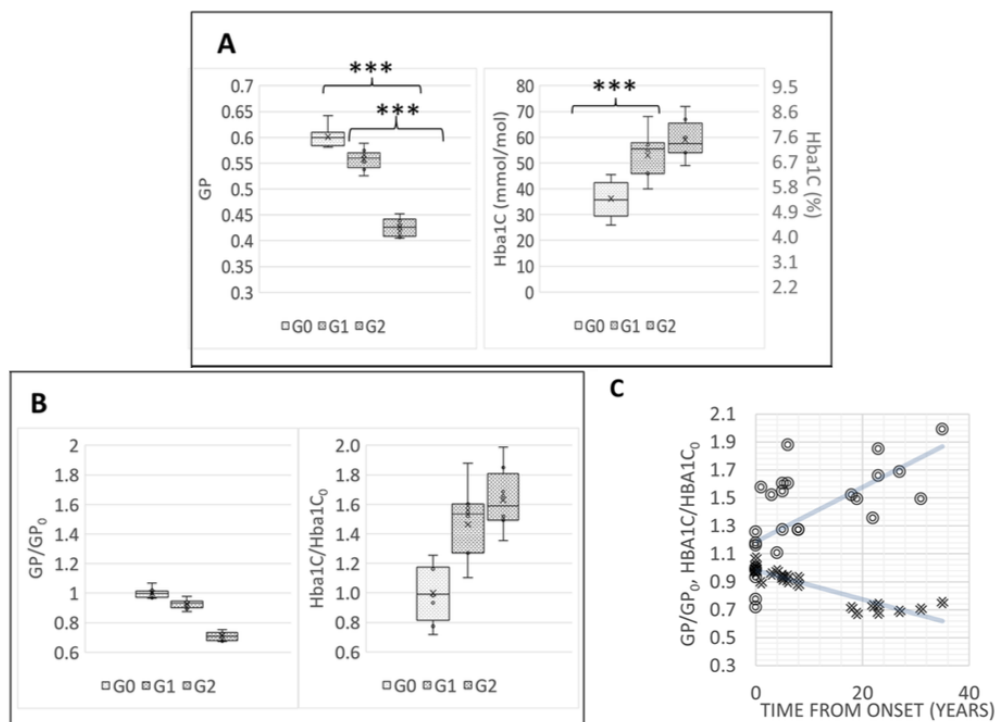


Figure 3.3: RBC fluidity allows to classify patients with and without complications, and increases with time after the first diagnosis. (A) Mean GP values and HbA1c values of different groups G0: control; G1: T1DM, no complications; G2: T1DM with complications. *** stays for $p < 0.001$. (B) Mean GP values and HbA1c values normalized to GP0 and HbA1c0 values (average over the G0) of different subgroups. (C) Mean GP values and HbA1c normalized values of RBC fluidity variations with time after the first diagnosis. Solid lines represent linear regression curves. The slopes of the HbA1C/HbA1C0 curve is 0.0195 y-1, while the slope of the GP/GP0 is -0.0104 y-1. R2 values are respectively 0.4213 and 0.8365 for the HbA1C/HbA1C0 and GP/GP0.

classification block works and finally section 3.4.6 shows the method used to combine the dual channel information to obtain the comprehensive categorization of the subject.

3.4.1 Image classification

This block operates in two modes, namely training and classification, which are depicted in Figure 3.4 B).

In the training mode, for each channel we have the training images $\{I_{tr}^i\}_{i=1}^m$ and the corresponding three classes labels $\{y_{tr}^i\}_{i=1}^m$, where y can be 0, 1 or 2. The *feature computation* block computes for each I_{tr}^i the set of features \mathbf{x}_{tr}^i that will be presented hereinafter (section 3.4.2); then, the *principal component analysis* (PCA) block finds the

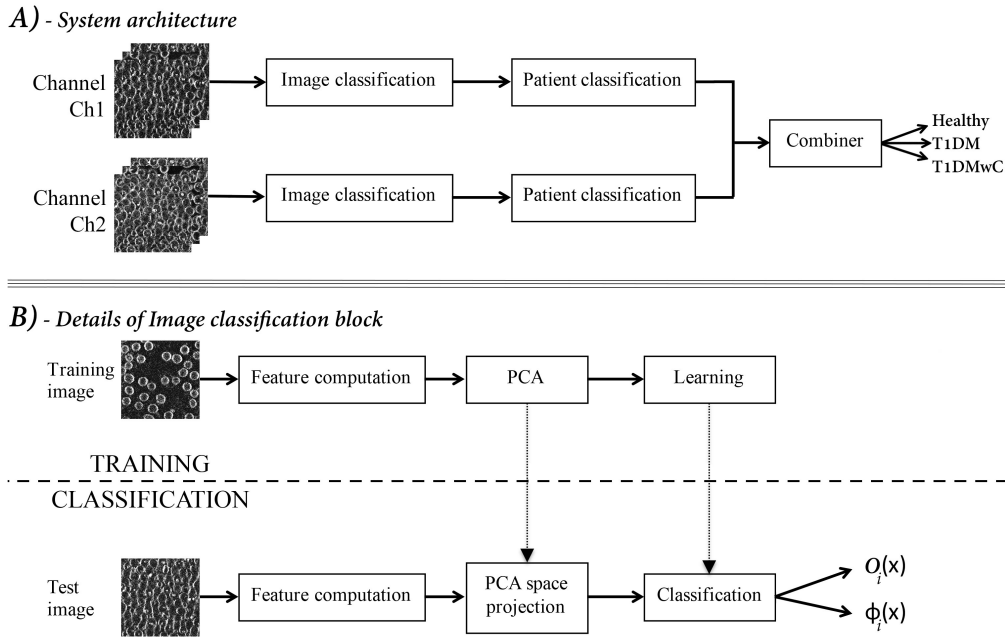


Figure 3.4: (A) Schematic representation of the dual-channel pipeline used to classify the RBC images, which labels each patient as an healthy subject (*Healthy*), or as a T1DM patient (*T1DM*), or a T1DM patient with complications (*T1DMwC*). (B) Details on image classification block: for each image it provides a label O_i and a reliability measure ϕ_i which are then used by the patient classification block.

principal components (section 3.4.3). Finally, a classifier learns how to partition the transformed feature space using training data.

In the classification mode, for each channel, given a patient and his/her collection of test images $\{I_{te}^i\}_{i=1}^n$, we first compute the features \mathbf{x}_{te}^i which are then projected into the feature space given by the PCA. The classifier then assigns to each I_{te}^i a label $O(\mathbf{x}_{te}^i)$, which can be 0, 1 or 2, and a reliability measure $\phi(\mathbf{x}_{te}^i)$. This latter quantity, that we will detail in section 3.4.4, ranges in $[0; 1]$ where 0 and 1 represent an unreliable and a very reliable classification, respectively. It is worth noting that the reliability has demonstrated its usefulness in different fields of pattern classification, such as the application of different fusion rules [28] and the assessment of the quality of the outputs provided by the classifier [29, 30, 57].

Next subsections will focus on the three main steps of the image classification block: first, in subsection 3.4.2 we will describe the features we considered; second, subsection 3.4.3 introduces the PCA, whereas subsection 3.4.4 describes how to compute the reliability of the classifier.

3.4.2 Feature computation

This section presents the set of descriptors computed in the image classification block (Figure 3.4 A) and B)). In the literature, there exist several image descriptors applied to medical image analysis derived from either morphological or textural analysis, usually integrated together to substantially improve the classification accuracy [31–34].

A priori knowledge of the domain suggests us to adopt a global approach that computes the features from the whole image. Indeed, differently from other cells, Laurdan labels only plasma membrane of RBCs since they lack intracellular organelles and, hence, a segmentation step of cell membranes is useless. Sometimes the peculiar biconcave shape of RBCs may give the impression that some organelles or other spots are in the inner part of the cells, but that contribution still comes from plasma membranes.

The features we consider can be divided into two main families. The former is composed of *statistical features*, which are based on first- and second-order measures [35].

We now qualitatively present such descriptors, whereas Tables 3.1 and 3.2 report the formal definitions according to the following notation:

The content of such tables adopt the following notation:

- G is the number of gray levels in the image I ;
- g_i and g_j denotes two gray-level values, with i and j lying in $[0; G - 1]$;
- \mathbf{d} is a displacement vector;
- $h(g_i)$ stands for the probability density function of the image I ;
- $g_{max\,ass} = \arg \max_{g_i} h(g_i)$ represents the absolute maximum of the probability density function;
- $\Gamma = \{g_i | h(g_i - 1) < h(g_i) \wedge h(g_i) > h(g_i + 1)\}$ is the set containing the relative maxima of the probability density function;
- $p(g_i, g_j, \mathbf{d})$ denotes the gray level co-occurrence matrix (GLCM) of I ;
- $\mu_{p(g_j, \mathbf{d})} = \sum_{i=0}^{G-1} p(g_i, g_j, \mathbf{d})$ and $\mu_{p(g_i, \mathbf{d})} = \sum_{j=0}^{G-1} p(g_i, g_j, \mathbf{d})$ denote the mean value of the one-dimensional marginal distributions of the GLCM;
- $\partial(g_i, g_j) = (g_i - g_j)^2$ is a measure of dissimilarity.

Feature	Definition
Mean ($\mu(I)$)	$\sum_{i=0}^{G-1} g_i \cdot h(g_i)$
Standard deviation	$\sum_{i=0}^{G-1} (g_i - \mu(I))^2 \cdot h(g_i)$
Skewness	$\sum_{i=0}^{G-1} (g_i - \mu(I))^3 \cdot h(g_i)$
Kurtosis	$\sum_{i=0}^{G-1} (g_i - \mu(I))^4 \cdot h(g_i)$
Width	$\max(g_i) - \min(g_i)$, for $g_i > 0$
Energy	$\sum_{i=0}^{G-1} h(g_i)^2$
Entropy	$-\sum_{i=0}^{G-1} h(g_i) \cdot \log[h(g_i)]$, for $h(g_i) > 0$
Absolute maximum	$\max(h(g_i))$
Gray-level absolute maximum	$\arg \max_{g_i} h(g_i)$
Energy around absolute maximum	$\sum_{i=0}^{G-1} h(g_i)^2$, for $g_i \in \{g_{max\,ass} \pm \delta\}$, with $\delta = 2$
Number relative maxima	$\dim(g_{m_rel})$
Energy around relative maxima	$\sum_{i=0}^{G-1} h(g_i)^2$, for $g_i \in \Gamma(j) \pm \Delta$, with $j = 1, \dots, \Gamma $ and $\delta = 2$

Table 3.1: Definition of the first-order statistical measures.

Feature	Definition
Autocorrelation	$\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} [p(g_i, g_j, \mathbf{d}) - \mu_{p(g_i, \mathbf{d})} \mu_{p(g_j, \mathbf{d})}]$
Homogeneity	$\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{p(g_i, g_j, \mathbf{d})}{1 + \vartheta(g_i, g_j)}$ for $g_i \neq g_j$
Entropy	$-\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} p(g_i, g_j, \mathbf{d}) \cdot \log[p(g_i, g_j, \mathbf{d})]$, for $p(g_i, g_j, \mathbf{d}) > 0$
Energy	$\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} p(g_i, g_j, \mathbf{d})^2$
Covariance	$\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (g_i - \mu_{p(g_i, \mathbf{d})})(g_j - \mu_{p(g_j, \mathbf{d})})p(g_i, g_j, \mathbf{d})$
Inertia	$\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \vartheta(g_i, g_j)p(g_i, g_j, \mathbf{d})$
Absolute contrast	$\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} g_i - g_j \cdot p(g_i, g_j, \mathbf{d})$

Table 3.2: Definition of the second-order statistical measures.

The first-order features are the statistical moments up to the fourth order of the first-order image histogram, i.e., the mean, the standard deviation, the skewness, and the kurtosis. Furthermore, this set contains also other measures extracted from the first-order image histogram, such as its width, energy, entropy, the value of its absolute maximum and the corresponding gray-level value, the energy around such maximum, the number of relative maxima in such histogram and their energy.

Second-order features are derived from the the gray level co-occurrence matrix (GLCM) of each image: this quantity represents the distribution of co-occurring values between neighboring pixels according to different displacements [37]. Its statistics correlate well with the image structure and texture [38]. The GLCM represents the distribution of probability of occurrence of a pair of gray-level values g_i and g_j separated by a given displacement vector \mathbf{d} , i.e., $p(g_i, g_j, \mathbf{d})$, with i and j lying in $[0; G - 1]$, where G is the number of gray levels in the image. From the GLCM we compute the autocorrelation, the homogeneity, also known as inverse difference moment, the entropy, the energy, the covariance, the inertia and the absolute contrast. The interested readers can find their formal definitions in Table 3.2. Shortly, the autocorrelation looks for repetition in the image: for instance, if the autocorrelation is a spike, it says that the image has no elements that are correlated, i.e. everything is unique. The homogeneity measures the smoothness of the gray level distribution of the image. The entropy measures the degree of disorder among pixels in the image and images with a larger number of gray levels have larger entropy. The energy measures the uniformity of the gray level distribution of the image, whereas the covariance measures the linear dependency of gray levels on those of neighboring pixels. Both the inertia and the absolute contrast accounts for the amount of local gray level variation in an image: a high value of these parameters may indicate the presence of noise or wrinkled textures in the image.

The latter family of feature is set of *rotation invariant co-occurrence local binary patterns*, where Local Binary Pattern (LBP) is a grayscale texture descriptor that assigns to each pixel of the image a label obtained comparing it with its neighborhood matrix. Different types of LBP operators can be computed varying both the pixel arrangement and the number of neighbors, respectively. The standard LBP operator works in a 3×3 neighborhood square matrix [39] as follows. Defining as g_c the value of the pixel with coordinates $\mathbf{r} = (x, y)$, and with g_p the value of its p -th neighbor, the LBP operator is defined as:

$$LBP(\mathbf{r}) = \sum_{p=0}^{P-1} 2^p \text{sgn}(g_p - g_c) \quad (3.1)$$

where P is the number of neighborhood pixels, $\text{sgn}(x)$ represents the sign function and

coordinates of g_p are given by $(x + \delta \cos \theta_p, y + \delta \sin \theta_p)$, where $\theta_p = p \cdot \frac{2\pi}{P}$ and δ denotes the radius of the neighborhood. Hence, the displacement vector from the reference pixel to neighboring pixels is $\Delta \mathbf{r}_{\theta_p} = (\delta \cos \theta_p, \delta \sin \theta_p)$. In our implementation we set $P = 4$ to reduce the computational costs: as a result, each LBP has $N_P = 2^P = 16$ possible patterns and it returns an image with the same size $N \times M$ of I , where its pixel \mathbf{r} stores the value $LBP(\mathbf{r})$. The image texture can be now represented building a histogram:

$$H = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} f(LBP(\mathbf{r}), k), \text{ with } k \in [0; N_P] \quad (3.2)$$

and

$$f(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

The basic LBP operator is neither robust to changes in spatial resolution nor to texture rotations. For this reason we compute the rotation invariant co-occurrence among adjacent LBPs (CM-LBP) obtained as follows [40]. First, we computed the co-occurrence matrix of the output LBP image; using the notation introduced above, this returns the quantity $p_\theta(LBP(\mathbf{r}), LBP(\mathbf{r} + \Delta \mathbf{r}_\theta))$, where the subscript θ implies that the probability depends upon the angle θ . We consider four orientations, namely horizontal, vertical and two diagonals corresponding to $\theta = 0, \pi/4, \pi/2, 3\pi/4$. This approach produces four histograms, each with N_P^2 dimensions. Second, as the cells have different orientations in the image, we ensure rotation invariance by attaching a rotation invariance label to each LBP pair. In order to find the rotation equivalent LBP pairs we have to consider the case (a) where LBP pairs of $\theta = 0, \pi/4, \pi/2, 3\pi/4$ have rotation equivalence and the case (b) where the LBP pairs rotated by π have rotation equivalence. For the former case (a), we denote a LBP pair $P_\theta(\mathbf{r}, \Delta \mathbf{r}_\theta)$ as follows [41]:

$$P_\theta(\mathbf{r}, \Delta \mathbf{r}_\theta) = (LBP_\theta(\mathbf{r}), LBP_\theta(\mathbf{r} + \Delta \mathbf{r}_\theta)) \quad (3.3)$$

where

$$LBP_\theta(\mathbf{r}) = \sum_{p=0}^{P-1} 2^p \text{sgn}(I(\mathbf{r} + \Delta_{\theta,p}) - I(\mathbf{r}))$$

and

$$\Delta_{\theta,p} = (\delta \cos(\theta_p + \theta), \delta \sin(\theta_p + \theta))$$

Note that θ acts as the bias of the rotation angle in LBP. According to these definitions, the LBP pair of each configuration has the same value in terms of rotation. In order to consider the latter case (b), i.e. the case where the LBP pairs rotated by π have

rotation equivalence, we set $P_\theta(\mathbf{r}, \mathbf{r} + \Delta\mathbf{r}_\theta) = (LBP_{\theta+\pi}(\Delta\theta + \mathbf{r}), LBP_{\theta+\pi}(\mathbf{r}))$ using a mapping function M . Finally, a rotation invariant label for an LBP pair at \mathbf{r} , denoted as RIC-LBP, is given by:

$$P_\theta^{RI}(\mathbf{r}) = M(P_\theta(\mathbf{r}, \Delta\mathbf{r}_\theta)) \quad (3.4)$$

Similarly to what reported in equation 3.2, an RIC-LBP histogram is generated from $P_\theta^{RI}(\mathbf{r})$ for the image I , whose dimension is equal to $\frac{N_P(N_P+1)}{2} = 136$. We computed three variations of the rotation invariant co-occurrence LBPs by setting the pair $(\delta, \Delta\mathbf{r}_\theta)$ equal to (1, 2), (2, 4) and (4, 8): hence, the feature vector is composed of 408 elements.

3.4.3 Principal Component Analysis (PCA)

The features presented above produces for each image a feature vector \mathbf{x} composed of 427 elements¹. However, it is well-known that a limited yet salient feature set simplifies both the pattern representation and the classifiers that are built on the selected representation as it alleviates the curse of dimensionality [42].

In this respect, PCA is a simple, non-parametric method for extracting relevant information from data sets [36]. Indeed, with small efforts, it helps reducing a complex data set to a lower dimension to reveal the sometimes hidden, simplified structures that often underlie it. We therefore applied the PCA to the training set, retaining a number of principal components mapping the 99% of the variance, which correspond to 31 new orthogonal features.

It is worth noting that this procedure is applied in the training phase of the image classification block, depicted in the upper part of Figure 3.4 B). In the test phase (the lower part of the same figure) our proposal first computes the whole feature set from the test images, then it projects such features into the PCA space and, finally, it retains the selected components that will feed the classifier.

3.4.4 Classification reliability

As reported at the beginning of section 3.4.1, in the classification mode the classifier assigns to each input image a reliability score denoted as $\phi(\mathbf{x}_{te}^i)$, which is computed exploiting information provided by any measurement classifier² that the sample \mathbf{x}_{te}^i belongs (or not) to the corresponding class.

¹As described in section 3.4.2, the feature set is composed of 12 first-order measures, 7 second-order measures and 408 LBP measures.

²The various classification algorithms can be divided into three categories: type I (abstract), which supplies only the label of the presumed class, type II (rank) that ranks all classes in a queue where the class at the top is the first choice, type III (measurement) that attributes each class a value that

In this work we adopted a Support Vector Machine (SVM) with a radial basis kernel. To deal with the multi-class learning problem at hand we used a pairwise coupling approach [58], which decomposes the polychotomies into $\binom{3}{2}$ dichotomies, where each binary learner is specialized in discriminating between pair of classes. Binary predictions are aggregated to a final decision using a voting criterion [44]. To compute $\phi(\mathbf{x}_{te}^i)$ we applied the algorithm presented in [58] that optimizes an objective function whose argument is the sum of the pairwise differences between the reliabilities of each binary learner computed using the Platt's method [45].

3.4.5 Patient classification

For each channel, the patient classification block assigns to each patient under test a label Y and a reliability ψ , computed as follows on the basis of the outputs provided by the image classification block. The availability of several images collected in different xyz positions (as presented in section 3.3) permits us to exploit a degree of redundancy that lowers the effect of image misclassifications. To this aim we adopted the weighted voting (WV) rule where the opinion $O(\mathbf{x}_{te}^i)$ on each image I_{te}^i is weighted by the reliability parameter $\phi(\mathbf{x}_{te}^i)$. The weighted sum of the votes for each of the three output classes of our problem is given by:

$$W_h = \sum_{i:O(\mathbf{x}_{te}^i)=h} \phi(\mathbf{x}_{te}^i), \text{ with } h = \{0, 1, 2\} \quad (3.5)$$

and the output of the patient classification block³ is $Y = \arg \max_h W_h$. The reliability ψ of this decision is computed as

$$\psi = 1 - \frac{\max(W_h \setminus (\max W_h))}{\max W_h} \quad (3.6)$$

where the symbol \setminus denotes the relative complement of two sets. It lies in $[0; 1]$ so that the larger the difference between the sum of weighted votes for the winning class and those assigned to the second one, the larger its value.

3.4.6 Combiner

The last block of the pipeline aggregates the spectral information as it takes the final decision L on a patient on the basis of the outputs provided by the two patient classification blocks, which are $Y_{Ch1}, Y_{Ch2}, \psi_{Ch1}$ and ψ_{Ch2} . Indeed, subscripts $Ch1$ and $Ch2$

measures the degree that the input sample belongs to that class.

³Note that ties never occurred in all tests we performed.

stand for the two channels acquired and processed in the pipeline. The final label is computed as:

$$L = \begin{cases} Y_{Ch1}, & \text{if } \psi_{Ch1} \geq \psi_{Ch2} \\ Y_{Ch2}, & \text{otherwise} \end{cases} \quad (3.7)$$

3.5 Experimental setup

The experiments were performed on the dataset presented hereinbefore using a leave-one-person-out cross-validation procedure, as described in section 2.6.

As anticipated in section 3.4.4, we adopted an SVM with a radial basis kernel, using default values of its parameters. Nevertheless, we ran experiments varying such parameters, and we always got the performance shown below. Although the No Free Lunch Theorems for Optimization tells us that it is impossible to tune an algorithm such that it will have optimal settings for all possible problems [59], our finding agrees with [60], where the authors empirically observed that in many cases the use of tuned parameters can not significantly outperform the default values of a classifier suggested in the literature.

The proposed approach was compared against three other methods. The first threshold the value of HbA1c that, as reported in section 3.2, is the method currently endorsed in many countries as a diagnostic test for diabetes as well as for disease monitoring. Indeed, current approach measures the concentration of HbA1c in the plasma to identify the three-month average glucose concentration in the blood. If HbA1c concentration exceeds a threshold, the patient undergoes further analysis to define the medical picture [61]. Specialized societies for diabetes provide guidelines with the value of such a threshold [62]. In the following, this method is referred to as *thresholding HbA1c*.

As a second method to be compared, we investigated the use of HbA1c concentration when the value of the threshold is learnt from the data applying a Bayesian approach. This case is hereinafter referred to as *Bayesian HbA1c*. Let us introduce the Bayesian conditional risk $R(\omega_i|v)$ of predicting one the class $\omega_i \in \Omega$, with $\Omega = \{Healthy, T1DM, T1DMwC\}$, given the HbA1c value v :

$$R(\omega_i|v) = \sum_j \lambda(\omega_i|\omega_j)P(\omega_j|v) \quad (3.8)$$

where $\lambda(\omega_i|\omega_j) : \Omega \times \Omega \rightarrow \mathfrak{R}$ denotes the loss function measuring the cost for predicting ω_i when the true class is ω_j . Assigning a unit cost to a misclassified sample, and a zero cost to a correctly classified one, i.e. $\lambda(\omega_i|\omega_j) = 0$ if $\omega_i = \omega_j$ and 1 otherwise, we get

$$R(\omega_i|v) = \sum_{j \neq i} P(\omega_j|v) = 1 - P(\omega_i|v) \quad (3.9)$$

The predicted class is the one that minimizes the overall risk $R = \int R(\omega|v)p(v)dv$, which is equal to select the class ω_i for which $R(\omega_i|v)$ is minimum. The third method, hereinafter denoted to as *GPmap DSS*, consists of a DSS employing a classification pipeline similar to the one presented in Figure 3.4 A) where, however, the input image is the GPmap, i.e. a greyscale synthetic image used by biologists and physicians for the visual inspection of membrane as it enhances the fluidity and rigidity distribution [52]. Once the artifacts, such as sediments or other extracellular aggregates, were removed from the two images I_{Ch1} and I_{Ch2} simultaneously recorded from the two emission channels (section 3.3), the GPmap pixel intensity at location (x, y) is computed as:

$$\text{GPmap}(x, y) = \frac{I_{Ch1}(x, y) - \alpha \cdot I_{Ch2}(x, y)}{I_{Ch1}(x, y) + \alpha \cdot I_{Ch2}(x, y)} \quad (3.10)$$

where the parameter α is a calibration factor obtained by measuring both the emission channels in DMSO [52, 63]. In summary, from the GPmap, this DSS first computes the same feature set introduced in section 3.4.2 and, second, it applies the PCA to reduce the feature set dimensionality. Finally, an SVM with a radial basis kernel classifies each GPmap image of a patient, and the patient classification block combines these labels. It is straightforward to note that this DSS does not employ the combiner block as the dual-channel information is not available in this case.

3.6 Results

The results of the experiments are reported in Table 3.3, where the performance are measured in terms of accuracy, precision, recall and F1 score per each of the three classes *Healthy*, *T1DM* and *T1DMwC* (section 3.3). This table shows the results attained by our proposal and by the three other approaches described in previous section.

To analyze how much the PCA and the multispectral approach affect the results, Table 3.4 reports the performance with and without the PCA step achieved by: the proposed MS-DSS, each channel (Ch2 and Ch1) at the end of the patient classification block (Figure 3.4), an alternative way to combine the information provided by the two channels referred to as *concatenated DSS* (C-DSS). C-DSS concatenates the feature vectors computed by each channel in a single array, which is then passed through the PCA. Next, the selected descriptors feed a single classifier. Notice also that the columns of Table 3.4 are organized in three sections, each reporting the results attained using three different classification paradigms, i.e. an SVM as a kernel machine, a Multi-Layer Perceptron (MLP) as a neural network, and kernel density-based k-Nearest Neighbor (kernel kNN) as a statistical classifier. While the SVM setting is reported in section 3.5,

Performance metrics		METHODS			
		Our proposal (<i>MS-DSS</i>)	<i>Thresholding HbA1c</i>	<i>Bayesian HbA1c</i>	<i>GPmap DSS</i>
Accuracy		1.00	0.65	0.65	0.93
Healthy	Precision	1.00	0.67	0.88	0.82
	Recall	1.00	1.00	0.88	1.00
	F1 score	1.00	0.80	0.88	0.90
T1DM	Precision	1.00	0.56	0.56	1.00
	Recall	1.00	0.50	0.60	1.00
	F1 score	1.00	0.53	0.57	1.00
T1DMwC	Precision	1.00	0.80	0.57	1.00
	Recall	1.00	0.50	0.50	0.80
	F1 score	1.00	0.62	0.53	0.89

Table 3.3: Recognition performance of our proposal and of three compared methods, which are described in section 3.5. The figures of precision, recall and F1 are reported for each of the three classes, namely healthy subjects (*Healthy*), T1DM patients (*T1DM*) and T1DM patients with complications (*T1DMwC*).

	SVM				MLP				kernel kNN			
	<i>MS-DSS</i>	<i>Ch1</i>	<i>Ch2</i>	<i>C-DSS</i>	<i>MS-DSS</i>	<i>Ch1</i>	<i>Ch2</i>	<i>C-DSS</i>	<i>MS-DSS</i>	<i>Ch1</i>	<i>Ch2</i>	<i>C-DSS</i>
Accuracy	1.00 (1.00)	1.00 (1.00)	1.00 (0.96)	1.00 (1.00)	0.93 (0.78)	0.89 (0.70)	0.85 (0.70)	0.87 (0.81)	0.93 (0.85)	0.89 (0.48)	0.89 (0.82)	0.87 (0.78)
F1 Healthy	1.00 (1.00)	1.00 (1.00)	1.00 (0.93)	1.00 (1.00)	0.88 (0.57)	0.88 (0.57)	0.80 (0.50)	0.93 (0.81)	0.89 (0.76)	0.84 (0.53)	0.84 (0.76)	0.82 (0.73)
F1 T1DM	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	1.00 (0.84)	0.86 (0.70)	0.90 (0.73)	0.84 (0.80)	0.89 (0.76)	0.95 (0.46)	0.82 (0.82)	0.82 (0.82)
F1 T1DMwC	1.00 (1.00)	1.00 (0.95)	1.00 (0.95)	1.00 (1.00)	0.89 (0.86)	0.89 (0.80)	0.84 (0.78)	0.85 (0.83)	1.00 (0.86)	0.88 (0.36)	1.00 (0.88)	0.97 (0.80)

Table 3.4: Recognition performance with and without the PCA step achieved by: the proposed MS-DSS, each channel (*Ch1* and *Ch2*) at the end of the patient classification block, an alternative way to combine the information provided by the two channels (*C-DSS*). Three different classifiers are considered (SVM, MLP, and kernel kNN). The performance are reported in terms of accuracy and F1 for each of the three classes, namely healthy subjects (*Healthy*), T1DM patients (*T1DM*) and T1DM patients with complications (*T1DMwC*). Results attained without the PCA are in round parentheses.

after preliminary tests looking for the best configuration, the MLP was designed using an input layer with a number of neurons given by the number of the retained features, and two hidden layers with five and two neurons. In the case of kernel kNN, we used a Gaussian kernel. The MLP and kernel kNN classification reliabilities are estimated as reported in [64].

Table 3.5 reports the results of further experiments showing that the redundancy introduced by the proposed approach is beneficial to get a system more robust than using only a channel or than concatenating the features, as discussed in the next section.

With reference to the computation time, all the experiments described so far were performed on a 13-inch MacBook Pro with Retina display, 3.1 GHz Intel Core i7, 16 GB 1867 MHz DDR3, 512GB Flash Storage, Intel Iris Graphics 6100 1536 MB. All the

Poisson noise	METHODS			
	<i>MS-DSS</i>	<i>Ch1</i>	<i>Ch2</i>	<i>C-DSS</i>
Accuracy	1.00 (0.96)	0.96 (0.89)	0.96 (0.93)	0.98 (0.94)
F1 Healthy	1.00 (0.93)	1.00 (0.88)	0.93 (0.86)	1.00 (0.93)
F1 T1DM	1.00 (1.00)	0.95 (0.89)	1.00 (0.95)	0.97 (0.93)
F1 T1DMwC	1.00 (0.95)	0.95 (0.90)	0.95 (0.95)	0.97 (0.97)

Table 3.5: Recognition performance on images corrupted by Poisson noise with and without the PCA step achieved by: the proposed *MS-DSS*, each channel (*Ch1* and *Ch2*) at the end of the patient classification block, an alternative way to combine the information provided by the two channels (*C-DSS*). The performance attained by an SVM are reported in terms of system accuracy and F1 for each of the three classes, namely healthy subjects (*Healthy*), T1DM patients (*T1DM*) and T1DM patients with complications (*T1DMwC*). Results attained without the PCA are in round parentheses.

data were processed and analyzed using our in-house software tools coded in MATLAB 2015a (Mathworks Inc, MA, U.S.A.), whereas for the SVM classifier we used the LIBSVM library [44]. Averagely, the computation time to extract the features and to classify each patient is $\sim 12s$.

3.7 Discussion

Based on the results, in Table 3.3 we first observed that the proposed approach, referred to as *MS-DSS*, provides a perfect classification of the samples: nevertheless, the same performance are not attained by the state-of-the art technique for early detection of complications in T1DM. Indeed, thresholding the values of HbA1c on the cohort of patients contained in our database provides the performance reported in the second column of Table 3.3. Turning our attention to the results attained by the Bayesian HbA1c method, shown in the third column of the same table, we notice that they are still lower than those attained by the proposed method. Furthermore, although it performs better than thresholding HbA1c on the healthy subjects, it poorly discriminates T1DM patients with complications. This suggests that values of HbA1c do not carry enough information to distinguish between T1DM and T1DMwC. It is worth noting also that such results attained by thresholding HbA1c and Bayesian HbA1c agree with the limitations of using the glycosylated hemoglobin reported in the literature [48].

To better understand the aforementioned comparison with HbA1c deserves a more in-depth discussion from a biological point of view. In fact, insulin therapy maintains protein glycation as close as possible to normal to very low performance when HbA1c is used as discrimination threshold. Conversely, the treatment is not able not target

directly the dysregulation of the lipid metabolism, that instead is one of the main player in the disease progression. This latter aspect of the diabetes could indeed determine the effectiveness of the presented technique, in fact, it relies on the detection of alterations in the lipid membrane composition, polarity and phase whose pathological trends are not inhibited by the insulin therapy, thus provide a less biased signature for detecting subtle changes associated to diabetes presence and complications.

To deepen the results of our proposal, we also tested the GPmap DSS described in section 3.5, thus investigating what happens if we use the synthetic image usually inspected by biologists and physicians [52] rather than leveraging on the multispectral approach proposed here. The performance achieved are reported in the last column of Table 3.3: on the one side, they are larger than those attained using the HbA1c concentration, confirming that RBC membrane carry useful data for detecting complications in diabetes patients. On the other side, our proposal has better performance suggesting that the pipeline benefits from the use of a dual-channel approach.

The observations reported so far have shown the promising results achieved by our proposal in comparison with two methods leveraging on the glycosylated hemoglobin, i.e. the biomarker currently used in the diagnostic practice, and with another approach based on images usually employed for the visual inspection of the RBC images. For the sake of completeness, although the most crucial part of the classification is the ability of the DSS to discriminate against diabetic patients who have complications or not, it worth noting that the additional recognition of the *Healthy* class guarantees the robustness of the approach. Actually, whereas a pipeline leveraging on a cascade method, thus performing the dichotomous discrimination of *T1DM* and *T1DMwC* only after a previously *T1DM* patients classification would have probably worked well, a more challenging task and therefore, the achieved results support the validity of the extracted features.

We are now interested in discussing how much the PCA and the multispectral approach affect the results. Let us focus on the results of additional tests shown in Table 3.4, and in particular on those attained by the SVM. When comparing the values shown in each tabular outside and inside the parentheses, which correspond to the performance attained with and without the PCA, we observe that the PCA does not improve the performance (except for Ch1). We deem that this can happen as the SVM mitigates the curse of dimensionality [65]. This observation is supported also by the results attained by the MLP and the kernel kNN, which are two classifiers suffering more from this effect. As expected, PCA is beneficial as the performance improve. Second, let us focus again on SVM results, and compare now the values in the columns shown outside the round

parentheses, which were achieved with the PCA. We notice that the two channels and the C-DSS perform as the MS-DSS, being an alternative way to combine the multispectral information. Furthermore, this observation does not hold for the MLP and the kernel kNN: indeed, in a large part of the cases, MS-DSS performance are better than those returned by Ch1, Ch2 and C-DSS.

Let us now discuss the results investigating the benefit of the redundancy given by the proposed multispectral approach. We first observe that the internal photon multiplier used to acquire the images suffers from Poisson noise [66], as also the analysis of our images confirmed. In our experiments, whose results are reported in Table 3.5, we therefore added Poisson noise to the images, where the noise was generated from the data themselves, i.e. the output pixel is generated from a Poisson distribution with mean equal to the value of the input pixel. In these tests, we first computed the features from the noisy images, and then we trained the classifiers mixing in equal proportion original and noisy images. Also, test images were similarly mixed. We deem that such a choice represents a reasonable and challenging scenario to test the robustness of the proposed approach, and it also permitted us to deeper investigate how the PCA affects the results. Interestingly, the performance in Table 3.5 show that the PCA can help better recognizing the samples, providing a set of more discriminative orthogonal features. Furthermore, these results show that the MS-DSS attains better performance than each channel and than C-DSS, being therefore a more robust approach for the problem at hand.

As a final issue of the discussion, we turn our attention to the computation time reported at the end of section 3.6. As the time needed to extract the features and to classify each patient is quite small, let us now discuss the time required to prepare and to image the samples as described in section 3.3. The time required for sample preparation is a little bit higher for the proposed technique with respect to HbA1C, which relies on High Performance Liquid Chromatography and it does not require RBC extraction and labeling (overall time needed $\sim 45min$). However, the parallel preparation of many samples with automated machines and high throughput imaging systems ($\sim 5s$ per sample) can significantly speed up of the overall process, making it reliable for translation in clinical settings. The slightly higher time cost is repaid in the long term, since it guarantees an higher sensitivity and specificity.

3.8 Conclusion

We have presented a DSS leveraging on dual channel metabolic functional imaging data computed from the physical state of RBC membranes, a novel biomarker that allows to monitor T1DM progression. It uses features based on first- and second-order statistical measures as well as on rotation invariant co-occurrence local binary patterns. The dual channel approach also introduces a certain degree of redundancy that lowers the effect of image misclassifications. The recognition approach permits to achieve promising results in a field of medicine characterized by large impact on the population.

Future efforts are directed towards enlarging the patient dataset, assessing the possibility of the method to describe long-term diabetes evolution, and lowering the acquisition cost of the proposed approach becoming independent from two-photon microscopy. To this last aim, future research will be directed towards the use of a cheaper microscope with UV source. Indeed, the raising quality of this light source has the potential to be useful in this application in a near future. Furthermore, if we develop a fully automated system able to carry out the whole procedure, that is imaging and processing, we will have to face the issue of automatic image acquisition. In this context, we should consider the chance to add a pre-processing stage determining if a zone of the well contains enough information to be classified by the proposed DSS.

From a biological point of view, it would be interesting to understand if there is the possibility of further improve the number of classes to maybe detect new variants of the disease or its complications and therefore correlate the images with the response to therapies of to understand if they could be used as predictors of the different disease progressions. Actually, it would be interesting to test the effectiveness of our pipeline even in the presence of gestational diabetes, thus try to distinguish a stable state of the pathology, as for the T1DM and the T2DM, from a diabetic condition that should completely disappear after giving birth. This would allow also a more specific and personalized therapy tailored on each patient and also would prove the ability to give a predictive outcome over the time. Finally, in order to achieve more robust results, it will be necessary to increase the number of acquired samples, since the analysis would then reach a higher level of generalization, not only focusing on a specific group of patients to detect but performing its predictions based on a larger and more heterogeneous pool of subjects.

4 Insulin granules tracking

4.1 Chapter overview

In chapter 2 and 3 we described an AI-based method that can be used as a diagnostic tool to monitor T1DM and to forecast possible complications. Our investigation was covered out at the mesoscale since, from a biological point of view, we tackled the phenomena of cells adaptation in response of a change on the physiological state, considering the environmental variation as well.

In this chapter we approach the analysis at the micro-scale, studying dynamics of the insulin granules inside the β -cells. In fact, in the literature, most of the efforts have focused on the study of phenomena in the proximity of cells membrane and, to the best of our knowledge, no further explorations were conducted towards the internal part of the β -cells. This quantitative intracellular investigation opens up the possibility to study the physiological cellular dynamics and the related insulin carrier system, in order to better assess the basic phenomena involved in both T1DM and T2DM.

Pancreatic beta cells biosynthesize and package insulin in insulin granules, whose secretion is regulated to maintain blood glucose homeostasis. The detailed knowledge of the dynamics of insulin granules could reveal defects in the intracellular handling and secretion of these granules, leading to impaired insulin secretion and consequently to the development of several metabolic diseases, including type-2 diabetes and the metabolic syndrome. The use of spinning disk confocal and light sheet microscopy with fast sequential scanning that enable rapid volumetric imaging, allows to monitor at high spatial and temporal resolution the dynamics of insulin granules. However, obtaining all the information for accurate 3D imaging and particle tracking within a single cell is complex and challenging, and extracting information from the particle tracking data requires to analyse the segments of motion trajectories. To this aim, we present a quantitative analysis of the 4D motion of insulin granules in glucose-stimulated INS-1E beta cells. First, we tracked each granule inside the cells via a computer-based automatic approach relying on a two-step iterative process. Next, we removed the artifacts and introduced a set of quantitative cinematic features describing granule dynamics. Finally, we implemented

an unsupervised machine learning based exploratory data analysis, which allows to distinguish two sets of granules marked by distinct dynamics: a first pool is characterized by a diffusive dynamic behavior, and a second pool that is characterized by a more directed and targeted movement. These pools may have distinct functional roles and/or interactions with other structures and organelles in beta cells that could be selectively impaired in pathological settings.

4.2 Introduction

The pancreas is an elongated organ nestled next to the first part of the small intestine. Its gross anatomy and the structure of pancreatic exocrine tissue and ducts are discussed in the context of the digestive system, that is beyond the purpose of this thesis. Whilst, the endocrine pancreas refers to those cells within the pancreas that synthesize and secrete hormones, including the insulin. This endocrine portion takes the form of many small clusters of cells called *islets of Langerhans*, that are roughly one million islets in human pancreas.

Pancreatic islets (one of which is depicted in Fig. 4.1) house three major cell types, each of which produces a different endocrine product:

- *Alpha cells* secrete the hormone glucagon.
- *Beta cells* produce insulin and are the most abundant of the islet cells.
- *Delta cells* secrete the hormone somatostatin, which is also produced by a number of other endocrine cells in the body.

Islets are richly vascularized, allowing their secreted hormones ready access to the circulation. Although islets comprise only 1 – 2% of the mass of the pancreas, they receive about 10to15% of the pancreatic blood flow. Additionally, they are innervated by parasympathetic and sympathetic neurons, and nervous signals clearly modulate secretion of insulin and glucagon.

Focusing on the beta cells, insulin is synthesized in respective nuclei. Here the insulin mRNA is translated as a single chain precursor called preproinsulin, and removal of its signal peptide during insertion into the endoplasmic reticulum generates proinsulin. Within the endoplasmic reticulum, proinsulin is exposed to several specific endopeptidases, thereby generating the mature form of insulin. Insulin is then packaged in the Golgi into special vesicles, referred to as insulin granules (IG) which accumulate in the cytoplasm. When the beta cell is appropriately stimulated, insulin is secreted from the

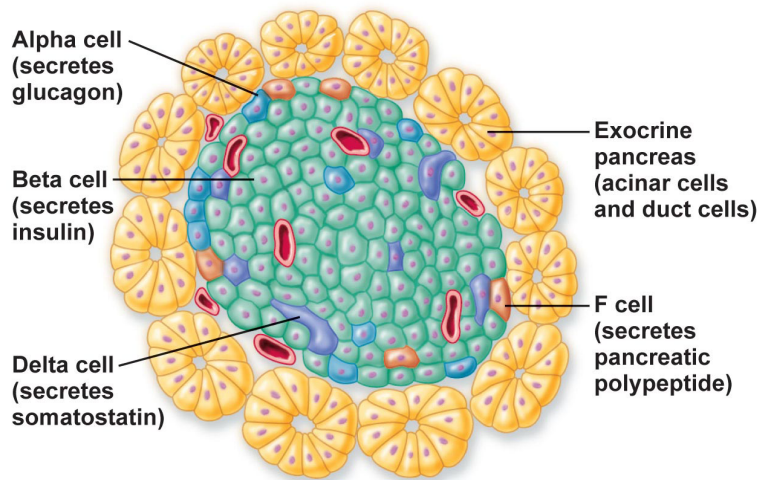


Figure 4.1: Representation of an Islet of Langerhans.

cell by exocytosis and diffuses into islet capillary blood, in order to the control of intermediary metabolism having effects on both carbohydrate and lipid metabolism, and significant influences on protein and mineral metabolism as well.

These essential functions are highly regulated, while defects in them can lead to impaired insulin secretion and be involved in the etiology of Type 2 Diabetes Mellitus and other features of the metabolic syndrome [67], [68]. Peripheral insulin resistance due to obesity or other metabolic disturbances results in chronic compensatory stimulation of pancreatic beta cells to secrete insulin. This condition may further degrade to reduced insulin biosynthetic and secretory capacities in the over-stimulated cells. In severe cases this condition culminates in hypoinsulinemia that requires parenteral insulin therapy to curb the uncontrolled hyperglycemia that develops in such diabetic patients [69], [70].

Traditionally, insulin secretion has been measured biochemically in whole animals and individuals, in isolated islets of Langerhans and beta cell lines following glucose stimulation or under the impact of different secretagogues. Such experiments report the rates and amounts of insulin being secreted, but do not furnish insight into the insulin granule subcellular trafficking following stimulation. To elucidate these mechanisms, several imaging techniques were developed and employed as powerful tools for detecting granule diffusion and trafficking, and to relate them to functional events in cells, such as membrane fluidity modifications, calcium release and uptake and metabolic pathway activations [52], [49], [71], [72], [73], [74]. During the last decade, these techniques have been applied to study insulin secretion and enabled monitoring at high temporal resolution, down to the single-granule, the release and the pre-exocytotic as well as

post-exocytotic events [75], [76], [77], [78].

In particular, the retrieved time trajectories revealed complex behaviors: combinations of random, directed, transiently stalled and constrained motions [77]. Different pools of granules' paths were separated in two classes according to their dynamic behavior (random walk motion and microtubule-driven motion) [79]. These different types of motion reflect the interplay of various molecular components in crowded environments. However, these imaging methods that monitor trafficking pathways or fusion event, of IG are constrained to a single bi-dimensional focal plane and lack crucial information for revealing the relation between IG intracellular dynamics and fusion in stimulated live beta cells [80], [81] [78].

Recent developments in imaging technologies, such as spinning disk confocal and light sheet microscopy with fast sequential scanning [82], [83], enable rapid volumetric imaging. These technologies also allow high resolution IG tracking and a detailed reconstruction of the intracellular trajectories. The resulting individual time trajectories that contain more information than the mean values extracted from the outlined traditional approaches, can potentially be decipher yet unknown mechanisms involved in beta cell secretory function. On the other hand, the challenge to produce an accurate 3D imaging and to track all particles within a single cell simultaneously is very complex. Moreover, the extraction of quantitative information from the obtained tracks needs the analysis of all particles' motion trajectories.

Although the works mentioned so far have achieved promising results, an approach for 4D IG tracking and analysis is still at infancy, being often limited either to local processing or to 2D image analysis.

To this aim, we present in this work our early experiences in quantitatively analysis of 4D motion and the dynamics of insulin granules in glucose-stimulated INS-1E beta cells detected by spinning disk imaging, where insulin granules were specifically labeled with an IAPP-mCherry protein [84]. To this end, we had to address several challenges, which are presented in this manuscript. First, we tracked each granule inside the cells via a computer-based automatic approach relying on a two-step iterative process, which tracks and cross-frame links the detected points. However, due to the large amount of noise affecting the images there was a need to filter out artifacts, a task performed analyzing the performances of the tracking software. Next, we introduced a set of quantitative cinematic features describing the routes and the granule dynamics. Further to the previous artifacts removal, there was a need to detect and remove outliers, a task we performed by developing a k-NN based algorithm. Finally, we present an unsupervised machine learning based exploratory data analysis, which, at the end, reveals two clusters

of granules with different motion characteristics.

On these grounds, we present a software procedure able to analyze induced alteration of IG three-dimensional dynamics induced in beta cells as detected by spinning disk imaging of glucose-stimulated INS-1E beta cells, in which insulin granules were labeled with IAPP-mCherry protein. This software allows for the 1) reconstruction of IG intracellular trajectories, 2) extraction of dynamical features from the trajectories and, ultimately, 3) classification of two sub-types of IG granules according to their dynamical behavior.

4.3 Materials

In the following subsections we report the steps we performed to collect the experiment's data starting from the β -cells' preparation.

4.3.1 Cell cultures and glucose stimulation

INS-1E cells were transfected with IAPP-mCherry protein targeted to insulin granules (IG) (courtesy of Dr. Patrick E. MacDonald, University of Alberta, Edmonton, Canada), as described [84]. Cells were plated at a density of 10⁵ cells/cm² in small IBIDI plates, in RPMI 1640 medium containing 11 mM glucose. After one day, the medium was changed to media supplemented with 5, 11, or 25mM glucose. Cells were then loaded with Laurdan and emission images were acquired at zero time (when the cells were fed with the different glucose concentrations), 24- and 48h. Other plates that were incubated for 32h at the same glucose concentrations (5, 11, and 25mM glucose) received 0, 200, 300, and 500 lM, The incubation medium with palmitic acid (PA) was serum free supplemented with 0.5% (w/v) fatty acid-free BSA PA during the last 16h of incubation. In addition, other cells incubated for 48h at 5, 11, or 25mM glucose for 48h received the different PA concentrations for 3h only.

4.3.2 Images Acquisition

Images were obtained by using a X-light spinning disk confocal scan head (Crysel Instruments, Italy) mounted on an inverted Nikon Ti-E microscope (Nikon Japan) fitted with a 60X immersion objective (NA 1.4) and Metamorph acquisition Software (Molecular Devices, USA). A EMCCD camera Evolve 512 (Photometrics, USA) collected 16 bit, unsigned images at a 0.11fps. mCherry was excited by a Lumencor LED high intensity fluorescence illuminator (Lumencor, USA; excitation peak 560nm). Volumetric imaging

was performed by acquiring an average volume 20 by 15 by $0.5 \mu m$ constructed collecting a sequence of 9 parallel planes ($Dz = 0.11 \mu m$), to ensure a count of the puncta over the whole cell volume. An entire volume is collected in $t = 3s$. A 4D time series (100 volumes, total acquisition time 45 min) was performed to allow 3D particle tracking. These settings were determined empirically to match a reasonable trade-off among maximization of both the signal to noise ratio and the temporal resolution of the time-lapse experiment.

4.3.3 Dataset

Six samples were incubated for 48 h in standard conditions (see paragraph 2.1). Then 6 different GSIS glucose-stimulated insulin secretion (GSIS) assay were conducted following the protocol outlined above. After adjusting the glucose concentration the acquisition was performed for 15 min when the cells were maintained at 5.5mM glucose, and for 30 min after the glucose concentration was increased up to 16.7 mM; the total acquisition time was 45 min. To perform a global dynamic analysis (sections 6.2 and 6.3), all the traces collected in all the six different experiments were merged and analyzed as a single dataset. In section 6.3 the traces collected in all six experiments were merged into a single dataset since we aimed at studying the global IGs behavior; consequently, we did not analyze the effects of the glucose addition (*5.5 mM glucose* and *16.7 mM glucose* stacks) which will be investigated in a future study.

4.4 Methods

This section illustrates the methods employed in assessing the dynamic of insulin granules inside beta cells in response to glucose stimulation. In order to extract quantitative information from the collected 4D stacks of images, we first tracked each granule inside the cells (as described in subsections 4.5 and 5), next we calculated and analyzed several cinematic features from their routes (as detailed in subsection 6.1). Finally, using a machine learning approach, we distinguished between two families of insulin granules with distinct dynamical behaviors (section 6.2).

4.5 Automatic tracking

The track of an object is completely defined by the coordinates of its position in the space for each frame in which it is located. Manual tracking of the insulin granules

in a stack is an unfeasible task owing to the mean granules' density (in the order of 10000 *granules/cell*) and the size of cells' clusters in the given preparations.

Some commercial software tools (*e.g.* *Velocity*¹, *ImarisTrack*² and *Metamorph*³) and open source software tools (the most popular are *DiaTrack*⁴, *Ilastik*⁵ and *TrackMate*⁶) have been developed to cope with the challenge to handle bi-dimensional and three-dimensional datasets acquired over the time, and to extract particles positional information.

Therefore, to overcome the limitation of working using only planar visualizers to deal with such a complex scenario, we developed a computer-based automatic approach, which is based on a two-step iterative process and uses *TrackMate*, a freeware package available within the *Fiji* software platform [85]. This software is able to track the huge number of IGs in the collected volumes.

The first step of this approach, named as “particle-segmentation”, detect all the granules and organized them into a sequence, *i.e.*, in this step the system identifies the positions of granules in the stack by providing the point coordinates (x, y, z, t) . This first step leverages on the *Difference of Gaussians* algorithm [86], a feature enhancement algorithm. It involves the subtraction of one blurred version of an original image from the same image less blurred. The original grayscale images are convoluted with Gaussian kernels having differing standard deviations. This operation suppresses only high-frequency spatial information, and subtracting one image from the other and preserves spatial information that lies between the range of frequencies. The result is a denoise effect and particle highlight at the same time [87].

In the second step, named as “particle-linking”, the system computes a cross-frame connection, where the point P_f identifying the granule g at frame f is linked with the point P_{f+1} at frame $f + 1$, where P_{f+1} is considered the most likely identification of g on the frame $f + 1$. Formally, this identification is computed on the basis of the particle-to-particle square distance approach, where P_{f+1} is linked to P_f if their linking cost is the minimum respect to the costs calculated for all other granules' segmentations in f and if their cross-frame distance is not more than a specific threshold [88]. The analyzed traces guarantee the absence of split or merge events during their motion: therefore, to alleviate the computational burden, for the tracking phase we utilized a simplified version

¹<http://cellularimaging.perkinelmer.com>

²<http://www.bitplane.com/Imaris/ImarisTrack>

³<https://www.moleculardevices.com>

⁴<http://www.diatrack.org/index.html>

⁵<http://ilastik.org/index.html>

⁶<http://imagej.net/Getting-started-with-TrackMate>

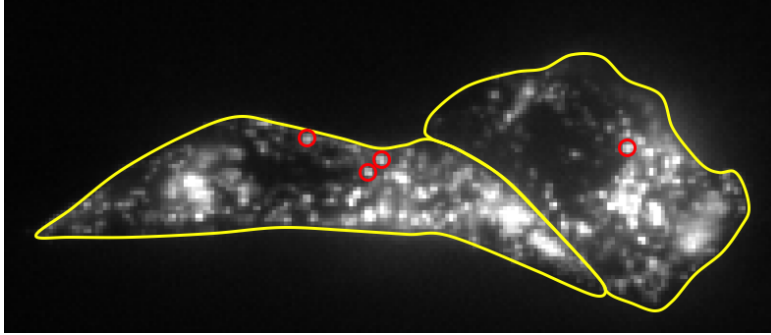


Figure 4.2: IAPP-mcherry-labeled IG tracking in INS-1E beta cells ($14.3\mu\text{l}$ glucose).

of the described method implemented in the *Simple LAP tracker* [89]. We set $5px$ as the maximum linking distance on the basis of the microscope's parameters. Moreover, the maximum number of allowed frames with no g detection is fixed to the value of 2 for possible temporal overlap with other granules. All the links of g over the different frames represent the three-dimensional path the granule.

Although we set the best microscope's parameter combination to optimize the acquisitions, the resulting 4D images were affected by noise and exhibited low contrast, as shown in Figure 4.2. This straightforwardly impacts granules' detection, which in some cases could result in the extraction of a cloud of points, which hinder the segmentation and increase the number of short and disconnected traces that were definitively regarded as artifacts. Such artifacts decrease the reliability of the tracking process, having the potential to degrade the following cinematic analysis. To mitigate this problem, we set the duration of the trace as a trace reliability criterion. On this basis, we grouped the set of all of the detected traces in four homogeneous subsets that were defined as follows: (i) a very low quality subset containing all the tracks where $L \geq 0.25F$, where L is the length in frames of the trace and F is the total number of frames in the dataset, (ii) a low quality subset composed of all the traces for which $0.25F < L \leq 0.50F$, (iii) an average quality subset containing all the traces for which $0.50F < L \leq 0.75F$, (iv) a good quality subset composed of all the traces satisfying $L \geq 0.75F$. The results are presented in section 6.

5 Visual4DTracker: a software for manual tracking

Although in previous section we report that manual tracking of the whole 4D stack is unfeasible, we needed to validate the results of the automatic tracking approach. This is required since we used the tracked traces to compute quantitative features to describe the dynamic of the granules and to reveal other properties that cannot be revealed by a visual inspection of the specimen. Since the performances of automatic tracking approach is inevitably affected by the noise in the 4D images, we considered only the granules whose track is reliable enough rather than using all tracks. Thus, given a specific quality subset, we compared their traces set of randomly selected granules that were computed by the automatic approach described in previous section, using a gold standard, which contains a set of traces annotated by a human expert. To collect the gold standard we manually labeled the x , y , z and t position of a set of granules. To the best of our knowledge, all the available open-source software for manual tracking perform a frame by frame segmentation, within a single layer of the stack. In our case, this forced the operator to perform a bi-dimensional tracking that is affected by severe limitations due to the nature of the task, the amount of noise affecting the acquisition, and the frequency of granule overlap that hinder the optical flow of each particle. We therefore developed a Matlab-based software that permits the user to set or to proofread the position of a single granule frame by frame directly in a 3D environment, i.e. the user visualizes the whole volumetric stack and can surf the data over the time. Moreover, once the gold standard of a track is available, our software compares such a trace with the corresponding one provided by the automatic approach to get a measure of performance.

The following sections present the developed software.

Visual4DTracker is a software package to visualize, handle and manually track $3D + t$ stacks of microscopy images containing objects such cells, granules etc..

The software allows users to track the movement of each blob in the $3D + t$ stack, marking it frame by frame and providing the corresponding trace over the entire microscopy acquisition period. Moreover, it allows users to proof-read and to evaluate the traces

with respect to a given gold standard.

The software is implemented in MatLab 2015b, and it runs on Windows, Unix and Linux platforms. It is freely available upon request at *e.cordelli@unicampus.it*.

5.1 Background

Microscopy images are widely used to study biological phenomena since they can provide a variety of useful information about morphology, structure, functional issues, and dynamics. In particular, dynamic investigations make use of time-lapse microscopy, where microscope image sequences are recorded and then observed at a greater speed to give an accelerated view of the process. This technique permits the study of space and time evolution of sets of particles, such as cells, granules or vesicles (hereinafter also referred to as blobs). Particle tracking, which consists in blobs' localization and trajectories' reconstruction, is a basic step useful also to extract quantitative features from image series [90], [91], [92], [93], [94], [95] and [96].

Although biological phenomena have in general a $3D$ nature, time-lapse experiments traditionally collect a series of $2D$ images over time, referred to as $2D+t$ datasets. However this approach may lose relevant $3D$ information, and, to overcome this limitation recent advances in high-throughput microscopy have made possible to image multiple z -positions of the sample in a time that is at least an order of magnitude smaller than the time scale of the biological process. This makes possible to collect $3D$ image stacks of the sample at different times, generating $3D+t$ datasets (also referred to as $4D$ in the following). These datasets are typically fairly large and their size tends to further grow as long as acquisition technologies improve. Moreover, in many cases in the field of view of the acquisition system there is a large number of particles, sometimes of more types, whose spatial and temporal behavior should be extracted and properly characterized to derive the information of interest. Particle tracking and feature extraction in these cases necessarily require some kind of assistance. For these reasons several commercial and open source software tools have been developed to handle $2D+t$ and $3D+t$ datasets, and to assist biologists in the extraction of quantitative information. In this respect, this manuscript presents *Visual4DTracker*, a software handling $3D+t$ stacks and offering the possibility to proofread the data in $3D$. It overcomes the main limitations of open source tools currently available, as described in the following sections.

5.2 Computational methods

We will focus here only on $3D + t$ particle tracking, as $3D + t$ datasets may carry more information on the biological processes under investigation than $2D + t$ data volumes. To the best of our knowledge, DiaTrack [97], ilastik [98] and TrackMate [85] are the most popular open source tools performing automatic tracking of multiple particles, whose main properties are summarized in Table 5.1. Furthermore, DiaTrack and TrackMate permit the computation of a predefined set of quantitative features from blob traces (e.g. velocity, acceleration, etc.).

On the one side, these tools provide the possibility to rotate, zoom and surf the $3D$ volume at a given t ; on the other side, fixed the view of the volume in terms of zoom, angle of view, rotation and pan, the users can surf its time evolution. However, they lose the ability to show the $3D$ view when the data has to be annotated: in this case, the user is limited to work with the orthographic projections on the frontal, lateral and horizontal vistas. The use of an own data structure for storing the tracking results as well as any other computed quantitative data is a common limitation of such tools. Straightforwardly, this prevents data sharing between different softwares, platforms and, eventually, researchers.

In many practical applications, once a tool has automatically provided the particle traces, researchers and practitioners may need to proofread these data, as the results can be far to be fully satisfactory. In this respect, only TrackMate offers this functionality, but the user has to interact only with $2D$ views of the $3D + t$ data. Moreover, there is no possibility to proofread traces computed by other tools. In other cases the users may need to manually track a particle or a limited set of particles: to this aim, a researcher can also try to use other open source tools, such as Fiji or Vaa3D [99], but still there are some limitations. For instance, in Fiji the interaction is again only with $2D$ views of the $3D + t$ data, while in Vaa3D there is no chance to visualize time evolution.

To overcome these limitations we present Visual4DTracker, which can handle $3D + t$ stacks further to be used as a tool to proofread $3D$ data. The software shows the stack sequences into a $3D$ space, and a graphical user interface (GUI) allows users to navigate and to interact with the data in space and time. The users can therefore manually trace blobs, proofread an existing trace, view all the traces directly in $3D$, and store the results in an xml file formatted to facilitate data sharing.

Software	<i>ilastik</i> [98]	<i>DiaTrack</i> [97]	<i>TrackMate</i> [85]	<i>Visual4DTracker</i> (proposed)
Manual tracking	YES: after segmentation, clicking frame by frame the moving object	NO	NO	YES
Automatic tracking	YES: it uses the Least Square, Distance association method	YES: after segmentation, it uses the Least Square Distance association method	YES: after segmentation, it uses the Least Square Distance association method	NO
Single particle tracking	YES	NO	NO	YES
Multiple particle tracking	YES	YES	YES	NO
Particle occlusion handling	YES: it is possible to manually define split events of single objects	NO	YES: it automatically searches for multiple objects split events	NO
Pre-processing	YES (several filters)	YES (only Gaussian Filter, luminosity threshold)	NO	YES (several filters, user defined functions)
Proof-reading	NO	NO	YES: it is only possible to load the traces and to modify the tracking parameters	YES
Tracking processing	NO	YES: it is possible to compute kinematic parameters from the traces	NO	NO
3D tracking visualization	NO: only 2D multiple orthogonal layers representation	NO: only 2D multiple orthogonal layers representation	NO: 2D multiple layers representation surfable in quote and frames	YES
3D data reconstruction	YES	YES	YES	YES
Manual tracking in 3D visualization	NO	NO	NO	YES
User developed pre/post processing functions	NO	NO	NO: not directly, however the software is a Fiji plug-in	YES: it is possible to upload custom functions in Matlab
Output	File in .hdf5 format (or others)	File in .txt format trajectories and data and .eps format for the images	File in .xml format	File in .xml format

Table 5.1: Summary of the main properties of the open source softwares for 4D tracking.

5.3 Software description

For an easier comprehension Figure 5.1 shows the main GUI's appearance where the functional panels are labeled with numbers and referred to in the following text.

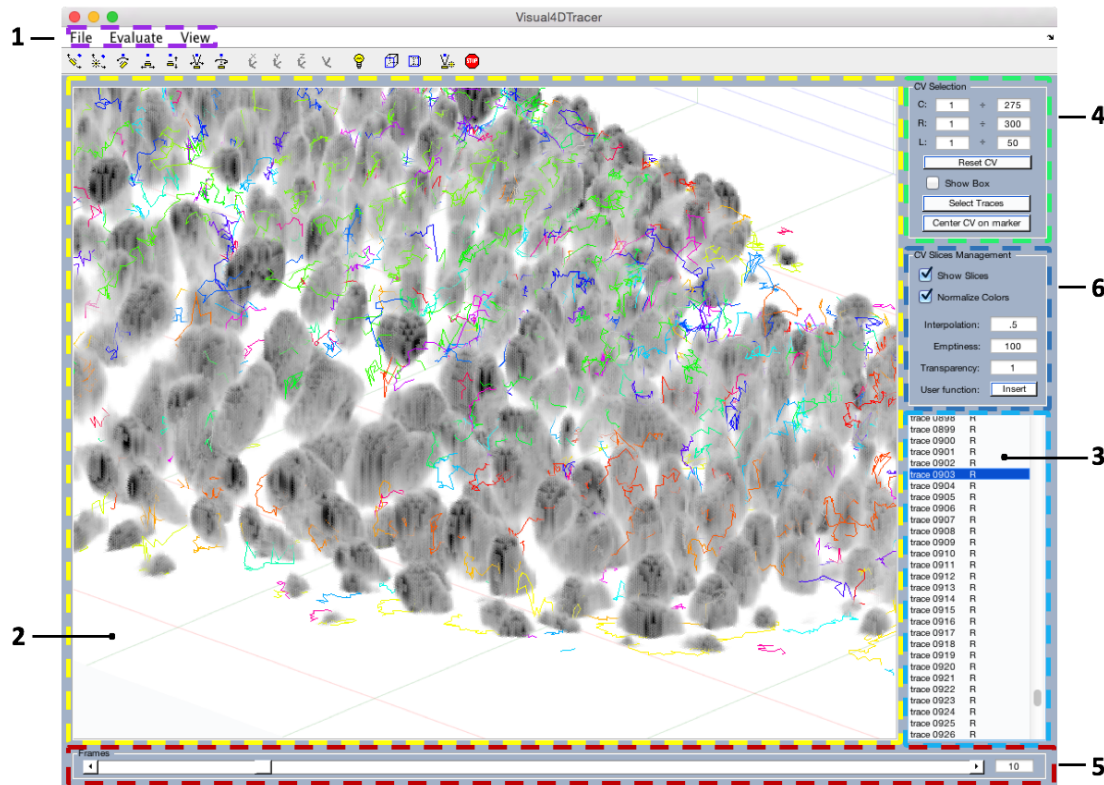


Figure 5.1: Main GUI's appearance: the numbers labels the functional panels. This example shows the 3D representation of *Drosophila* cells growing inside a larva, where each trace over the time is represented with a different color.

Visual4DTracer is a software that allows to handle, to visualize and to surf 4D stacks of images, containing objects moving in a 3D space and acquired over time at a microscope. Moreover, the Graphical User Interface (GUI) of the tool allows the user to mark the objects' positions frame by frame and to save their traces in a file that can be exported; there is also the possibility to load ((1) in Figure 5.1) a file containing a set of traces and to proceed further with data annotation. This functionality yields the possibility to proofread traces already marked by other users or provided by a software for automatic tracing, thus being able to compare two different traces belonging to the same experiment.

Visual4DTracker consists in a main window (MW, (2) in Figure 5.1) where the 3D

volume at a given frame ((5) in Figure 5.1) f is shown. The volume can be rotated, zoomed and panned using the mouse, the keyboard or a set of specific buttons in the toolbar. Other graphical objects at the bottom and at the left of the MW allow the user to interact with the 4D data. On the right side there are three panels. The first ((4) in Figure 5.1) permits users to visualize a parallelepiped shaped ROI in the MW, helping user focus the attention to a certain volume, and freeing up the memory to improve the user experience. The second panel ((6) in Figure 5.1) offers some basic image processing functionalities to reduce noise, to improve the contrast in the volume, and to resample the stack. The third panel ((3) in Figure 5.1) lists either the traced blob traces or those

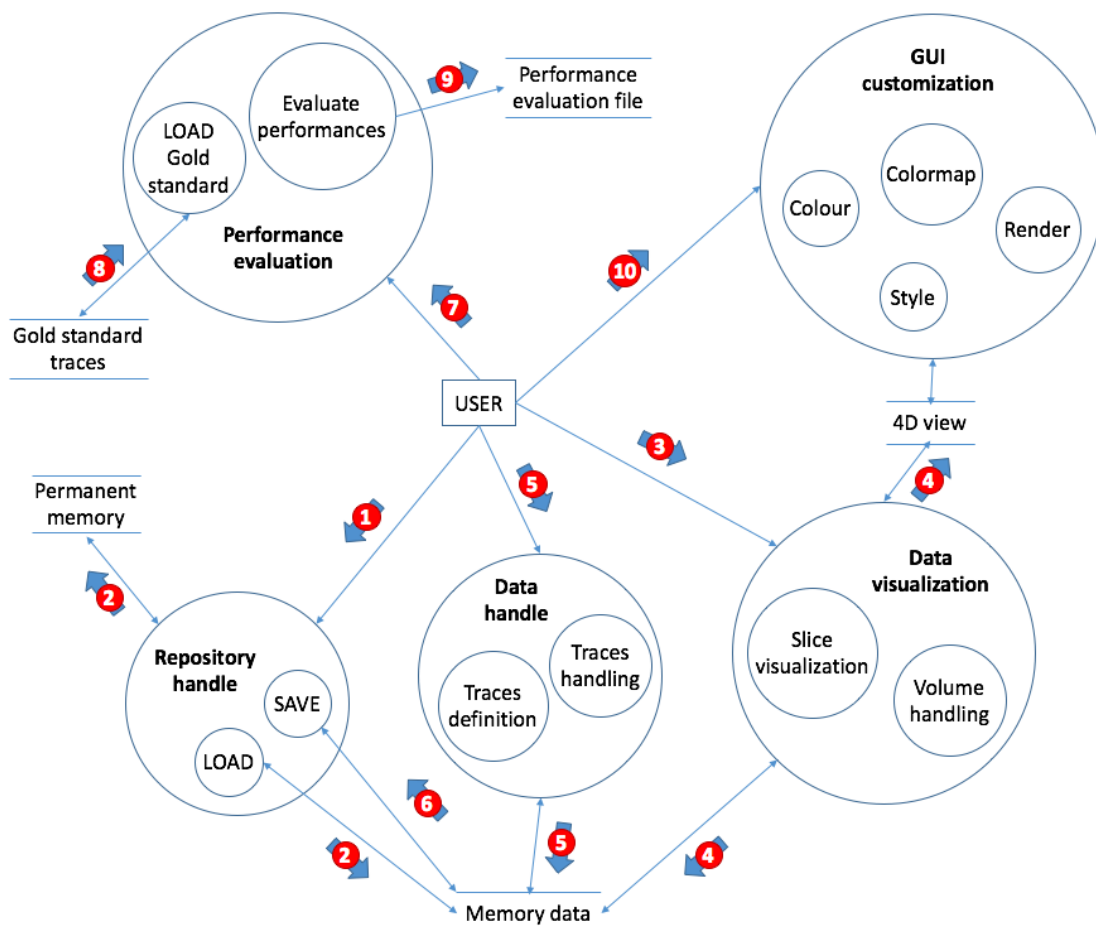


Figure 5.2: Tool's data flow diagram: the circles represent each process of the GUI linked each other with slim arrows, the coupled horizontal lines symbolize data repositories and the wide arrows with number show the typical actions and data flow.

loaded from an external file. The user can select which trace has to be shown, and the visualization engine shows the trace as a whole together with a subset of its markers

(eventually all). A top menu allows the user to load or to save a set of traces, to set several markers and traces graphical properties, to load the ground truth of the traces in the $4D$ stack (a feature useful for performance evaluation).

Next subsections describe the main characteristics of the software, while the guide provides a detailed description. Additionally, in order to describe the tool's features Figure 5.2 depicts the Visual4dTracker's data flow diagram, where the data transformation circles (i.e. the circles in light blue) represent each process of the GUI. These elements are connected each other by slim arrows. Horizontal parallel lines represent data stores, whereas the thick numbered arrows illustrate the typical flow of actions and data due to the interaction of the user with the application. Note also that the sequence of transactions depicted in the figure matches the toy example's steps described in the guide.

5.4 Volume visualization

Tools for volume visualization permit to set the ROI and offer some basic image processing functionalities. By default, the negative of the original grey level image is shown to enhance white or grey details embedded in dark regions, as in microscopic images they usually dominate in size.

When the $4D$ data is pretty huge, visualization and interaction with can decrease the user experience. Hence, there is the chance to select a $3D$ ROI of the volume by setting in the upper right panel the voxel intervals for the x - y - z axes or by using the mouse directly in the GUI. Note also that such a ROI also helps user focus the attention to a certain volume.

On the visualized volume, Visual4DTracker permits to apply three image processing operations to improve the perceived rendering quality. The first allows to set to zero the opacity of all the voxels whose visualized grey level intensity exceeds a threshold, set by the user. This functionality removes some artifacts in the image and filters out possible salt and pepper noise. The second operation applies a transparency to all the voxels in the visualized volume regardless of their grey level intensity, setting τ in $[-1; +\infty)$. If τ lies in $[-1; 0]$, the software sets the opacity of all the voxels to $1 + \tau$. If $\tau > 0$, the opacity of each voxel at coordinate (x, y, z) is set to $i^\tau(x, y, z)$, where $i(x, y, z)$ is the original grey level voxel intensity. Hence, the brighter the voxel, the larger the transparency. This helps the user visualize darker objects in the image, which usually correspond to particles. This operation also removes the noise usually present in the smooth borders of the blobs, facilitating the detection of a single blob in a cluster. The third operation linearly interpolates the intensity of the voxels, improving the spatial resolution and, in

some cases, the perceived smoothness of the image. Note that previous 3D ROI selection reduces the computational burden of the interpolation.

5.5 Handling, proofreading and evaluating the traces

Visual4DTracker inherently works with $3D + t$ images. For a given t , the user can mark each blob position directly in $3D$ via two left mouse clicks as the intersection in \mathbb{R}^3 between two incident segments. To this aim, the first user's click on the blob will produce a first piercing segment whose direction lies on the line orthogonal to the visualization plane. Once the volume is rotated, the user performs a second click on the same blob which yields a second piercing segment. However, since this procedure usually provides two skew segments, the blob marker is set to the middle point of the minimum distance segment linking the two piercing segments. A right click on a detected particle deletes its marker.

To mark the positions of a blob over time the user can repeat the aforementioned two-left-clicks procedure for all the different t in the $4D$ stacks. It is also possible to show more than one trace in the same window.

As a key feature the software helps user proofread traces loaded from an external file, whose content is formatted as reported in the guide. This can be very helpful as in many applications some existing software (e.g. TrackMate in Fiji) can be employed to trace the blobs: however, although in some cases they offer the possibility to tune the algorithmic parameters, the results can be affected by errors. In these cases, proofreading is a fast approach to correct the errors and move forwards with data analysis. To this aim, the aforementioned two-clicks procedure permits to modify blob markers positions, as described in the guide. Visual4DTracker would help user do this job via the offered volume visualization functionalities in conjunction with the chance to select the traces and the corresponding blobs under investigation.

The possibility to evaluate the tracing performance is another useful functionality. Each trace available in the workspace can be compared against a gold standard, loaded from an external file. For a proper comparison, we need to ensure that each predicted blob is uniquely associated with one gold standard blob, respecting also the temporal evolution of the movements. For this purpose, let us denote with C and G the set of marker positions for a trace in the test set and for the corresponding trace in the gold standard. The software first constructs an undirected bipartite graph with vertex set $C \cup G$. For each pair $\vec{c} \in C$ and $\vec{g} \in G$ it adds an edge with weight $w_{cg} = \frac{1}{\epsilon + \|\vec{c} - \vec{g}\|}$ if $\|\vec{c} - \vec{g}\| < d$, being d a distance parameter that can be roughly estimated as the

expected diameter of an object, and ϵ a small constant preventing numerical overflows. It then computes the maximum weight bipartite matching. A test set marker position \vec{c} is considered to be a *true positive* (TP) if it is matched to a gold standard position \vec{g} such that $\|\vec{c} - \vec{g}\| < d/2$. Unmatched predictions are counted as *false positives* (FP) and unmatched gold standard markers are counted as *false negatives* (FN). From these metrics, we derive P, R, and F_1 score as $P = TP/(TP + FP)$, $R = TP/(TP + FN)$, and $F_1 = \frac{2P \cdot R}{P + R}$.

Next sections now present the software with further details, guiding the reader through a detailed practical toy example and describing the steps to follow from loading to proofreading the data in a tracking task.

The interested reader can also watch the clip movies showing software functionalities using a real example.

5.6 Step by step software guide

This *step-by-step* guide shows how to use Visual4DTracer, a software that allows to handle, to visualize and to surf 4D stacks of images, containing objects¹ moving in a 3D space and acquired over time at a microscope. Moreover, the Graphical User Interface (GUI) of the tool allows the user to mark the objects' positions frame by frame and to save their traces in a file that can be exported; there is also the possibility to load a file containing a set of traces and to proceed further with data annotation. This functionality yields the possibility to proofread traces already marked by other users or provided by a software for automatic tracing, thus being able to compare two different traces belonging to the same experiment.

This quick guide intends to support the reader through the use of the software, and it therefore makes use of a toy (synthetic) example to present how the user can interact with the GUI, load a stack, handle the 4D volume, visualize the data, handle and define traces, and assess the tracing performances. The presentation of the toy example is offered in the light blue boxes in the following. Furthermore with the intent of describing the processes given by the interaction of the user with the GUI, this guide in the light red boxes refers to figure 2 of the main manuscript, which depicts the data flow diagram. The software as well as the data of the toy example can be downloaded. Furthermore, some videos presenting this tool are available at the same link.

¹We use the general term objects, although in the literature they are often referred to as corpuscles or blobs.

5.7 Installation

The software tool is a Matlab package written using the R2015b version of the software. It is therefore a cross-platform application working on Windows, Unix and Linux operating systems and, aside from Matlab, it does not require any further installation.

Visual4DTracer accepts a series of stacks of 2D grayscale images² of any Matlab supported format, i.e. *BMP*, *GIF*, *HDF*, *JPEG*, *JPEG2000*, *PBM*, *PCX*, *PGM*, *PNG*, *PNM*, *PPM*, *RAS*, *TIFF* and *XWD*.

5.8 Loading of 4D stack and traces

To run the application navigate to the “Visual4DTracer” folder and type “Visual4DTracer” in the Matlab Command Window. Once the software has been launched, the main GUI with all the panels disabled will appear, as shown in Figure 5.3.

The user can interact with the global features of the application using the items located in the menu list on the top of the GUI, marked by (1) in Figure 5.3.

First, to load a 4D stack click on the “File” menu and then on the “Load” > “Stack..” item. A dialog box pointing to the current Matlab’s folder will appear, but the user has to select the main folder containing the data of the experiment. Note that the application will read and write data (being either the 4D volume, the traces, the report on the assessed performances or the log files) located only in that main folder. Such a folder is conceptually a frames’ container, that are actually stacks of 2D images stored in subfolders and where each image is a slice of the same stack. Each subfolder corresponds to a given time step, and it has to be named as *name you’d like-time*. Furthermore, all the images must have the same sizes, and the number of images per subfolder cannot vary. Once the user has selected the main data folder containing all the stacks, click on the “Open” button in the dialog box. The application will load in memory the data contained in all the subfolders. Next, there is the possibility to specify the voxel spacing of the images and, by default, all the values are set to 1. There is also a chance to load an xlm file containing a set of traces already available for this stack at hand.

The described steps correspond to **arrow number 1** in the data flow diagram.

This file could have been saved as presented in section 5.13, and its format is described in the Appendix A.

²The maximum stacks’ dimensions admitted are determined from the user’s Matlab memory usage setting and depends on the color depth of images loaded.

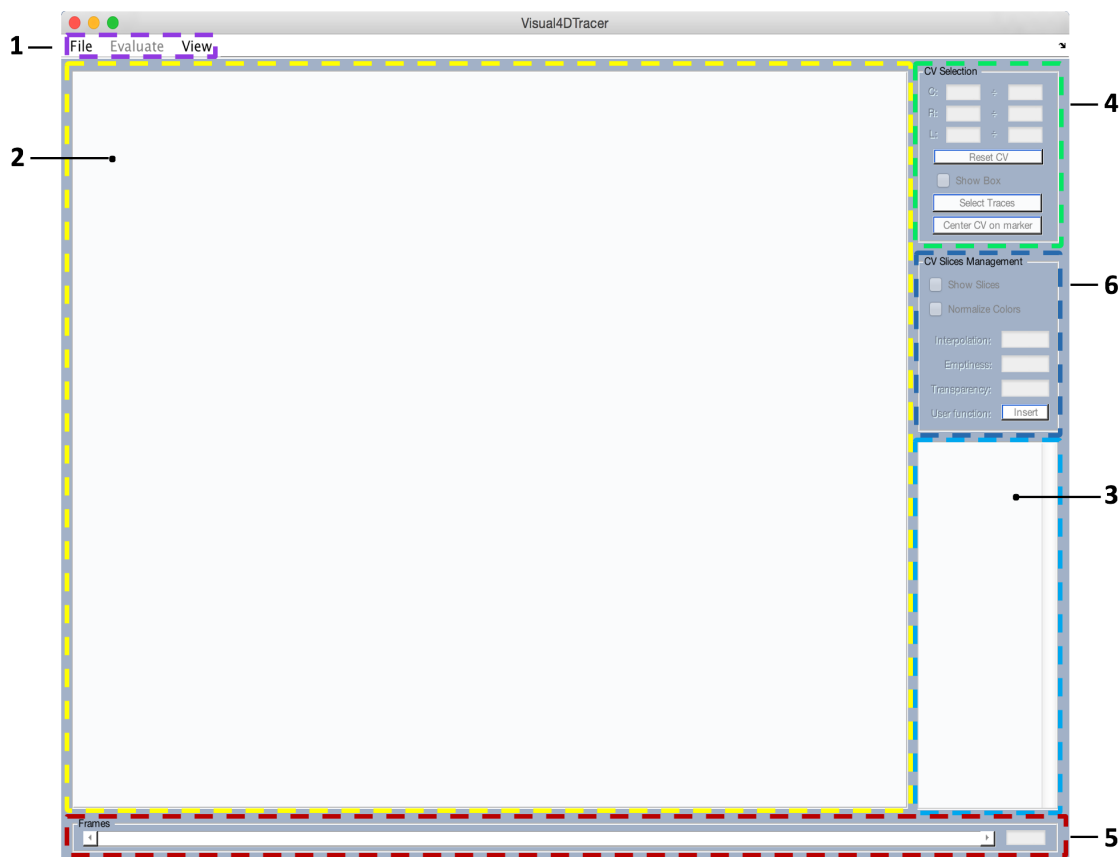


Figure 5.3: Main GUI once the application has been launched. Different areas of the GUI are highlighted by different colors to facilitate its description through the text.

To load the stacks of the toy example, click on “File” > “Load” > “Stack..”, navigate to the “toy example” main folder, select the “4D images” subfolder and then click on the “Open” button.

To load the stacks of the toy example, click on “File” > “Load” > “Stack..”, navigate to the “toy example” main folder, select the “4D images” subfolder and then click on the “Open” button. Next, keep the default values (1) for the voxel spacing and click “Ok”. Now, to load the related traces, select the “traces.xml” file located in the same folder.

Once the dialog box has closed, the data are loaded in the GUI’s variables workspace, and within the GUI main panel (2) by default the application shows only the empty euclidean space and the axes with scales, whose ranges fit with the stack dimensions.

The described steps correspond to **arrow number 2** in the data flow diagram.

This allows to unburden the graphics memory, while waiting for the user who can set up the desired point of view and the portion of the volume to be shown by selecting a control volume (CV) within the whole stack. This procedure will be described in section 5.9. Furthermore, the slide bar at the bottom of the GUI is set to the first frame.

The user should now interact with the trace panel, numbered as (3) in the figure. This panel lists all the traces reported in the xml file; note that the last element of the list, named as “new trace” corresponds to an empty trace and it can be used to add a new trace, as described in section 5.12.

The toy example contains 4 traces, and therefore the fifth item in the list allows to add a new trace.

5.9 Volume handling

Once the data are loaded, GUI panels will be enabled and they allow to interact with the 4D stacks. To this aim, the user works with the *CV Selection* (panel (4) in Figure 5.3), where he/she can set the boundaries of a cropped sub-volume of the data. By default, the ranges for each of the three C, R and L axes³ reported in the text-boxes are equal to the ranges shown in the main window. Furthermore, to visualize within the main window the selected sub-volume the user has to select the *show box* toggle button: the boundaries of the parallelepiped region are reported in red, green and blue, which correspond to R, C and L, respectively. To set the boundaries of the sub-volume to be shown, the user can set the CV principal diagonal coordinates by typing the values in six text boxes. For each of the three axes (denoted as C, R and L in the figure) the first text box corresponds to the sub-volume corner closest to the origin, whilst the second box corresponds to the farthest corner, as depicted in the example shown in Figure 5.4. Alternatively, the user can set the boundaries using the main window: to this aim, right click on the *main panel* at a desired location, than a contextual panel will appear allowing the user to select which of CV's borders to shift at clicked coordinates.

If you have already selected a blob within a trace (as described in sections 5.11 and 5.12) there is also the possibility to automatically center the parallelepiped region at the selected blob in the selected frame; furthermore the size of the parallelepiped is

³C, R and L stand for columns, rows and layers, respectively, and the same labels are shown in the main window close to each axis.

equal to the values shown in the C, R, L text boxes. To use this functionality press the “Center CV on marker” button or use the key *F*.

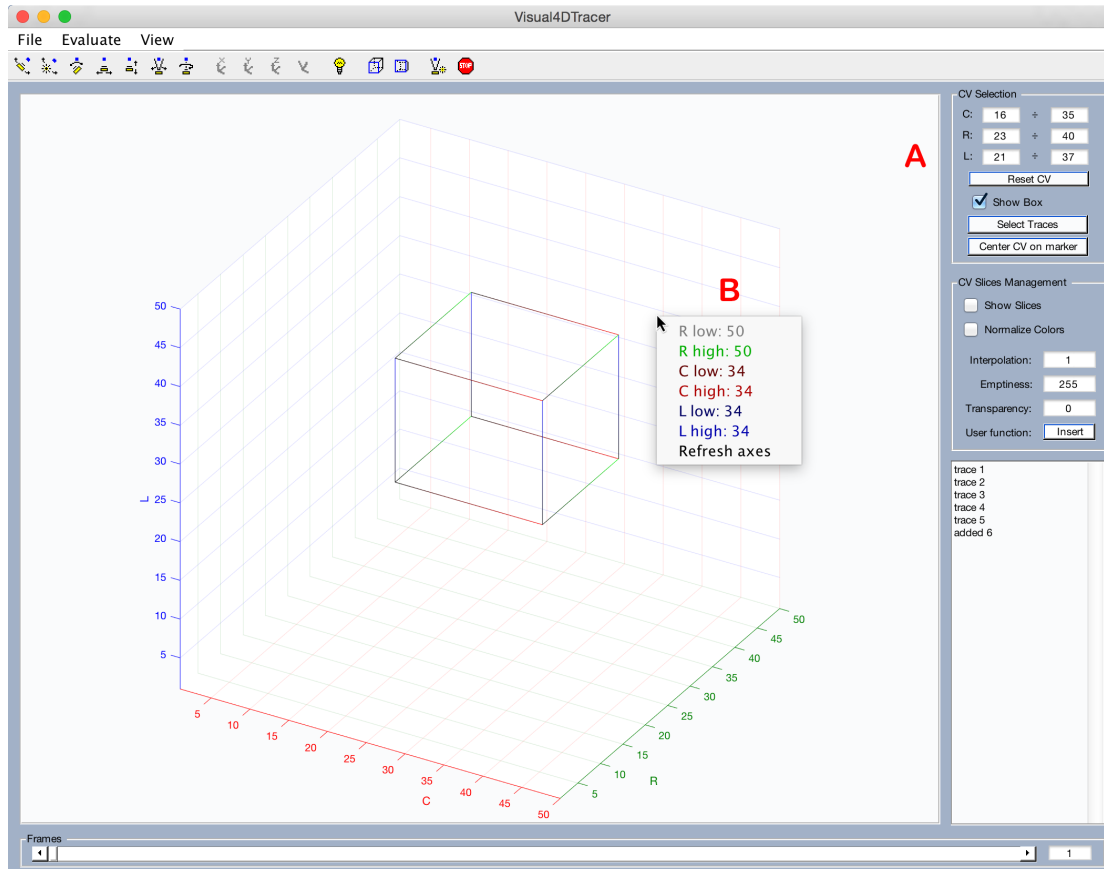


Figure 5.4: CV boundaries definition in the “toy example”: A) using *CV Slices Management* panel or B) using *main panel’s* contextual menu.

The tool also allows users to explore the 4D stack in space and in time. Indeed, the user can *rotate* the volume around the point at the center of the main panel, *translate* the stack and *zoom* in or out. These operations can be performed using the navigation toolbar shown on the top of the main panel. To improve the visual experience and to speed up the process, rather than using the toolbar the user can rotate, translate or zoom the volume via three main shortcuts based on the *T* or *Y* or *U* keys, respectively, that must be held down while clicking and dragging the mouse around. As reported above, the temporal domain can be controlled using the *Frames* panel (5) at the bottom of the main window: it is possible to move the slider to scan the experiment frames or by typing the specific number in the bottom-right editable space. Moreover the user can use the *spacebar* to rapidly move the frame by one step forward and *shift + spacebar* to

move one step backward.

The described steps correspond to **arrow number 3** in the data flow diagram.

5.10 Slice visualization

Once the sub-volume of interest has been set, the user can visualize the portion of its slices within the main window by selecting the *Show Slices* toggle button in the *CV Slices Management* panel (6). The result is a box completely filled with all voxels belonging to the selected part of the volume, including the objects of interest of the experiment together with the background and, eventually, the acquisition noise. This panel is specifically designed to process the images and to remove from the stack all non informative elements as much as possible.

To enhance the quality of the image shown, the user can perform one (or at least all) of the three following procedures⁴.

The first parameter is named as *Interpolation*, and it provides a sharper visualization of the whole sub-volume by performing a trilinear interpolation over all the voxels to simulate the increase or the reduction of the global level of detail of acquisition. The value in the edit box allows to set the distance of two consecutive planes normalized to the original distance, admitting values lying in $[0.1; 10]$. For instance, a value of 0.5 will insert an interpolated plane between each couple of original plane for all the dimensions of the stack, while a value equal to 3 will show one plane every two. Use this parameter wisely: a too thin interpolation will cause an high graphical computational load.

Try to set the interpolation value equal to 0.5 in the toy example.

Second, user can apply a threshold to voxels' intensity, removing all the voxels brighter than the threshold because part of the background. To this aim, use the *Emptiness* parameter, where the user can set a value within 0 and $2^b - 1$, where b denotes the number of bits used to represent the image intensity. Pixels whose intensity is larger than the value of the emptiness are set as transparent, and they are not visualized in the volume.

Try to set the emptiness value equal to 150 in the toy example.

Third, the user can set the *Transparency*, a less “aggressive” parameter that tunes the transparency of voxels whose intensity is below the emptiness. In the textbox, values

⁴In the following we assume to be working with images containing bright objects on a dark background.

must lie in $[-1; +\infty]$: when using a value in $[-1; 0)$ the software reduces the global opacity of the visualized region. For instance, if you set -0.3 , the opacity is reduced by 30%, i.e. the transparency is set equal to 70%.

When typing in the text box a value larger than 0, the voxels opacity is defined with a power law function of their normalized grey level intensity⁵, where the user sets the power exponent. Its behavior is shown in Figure 5.5.

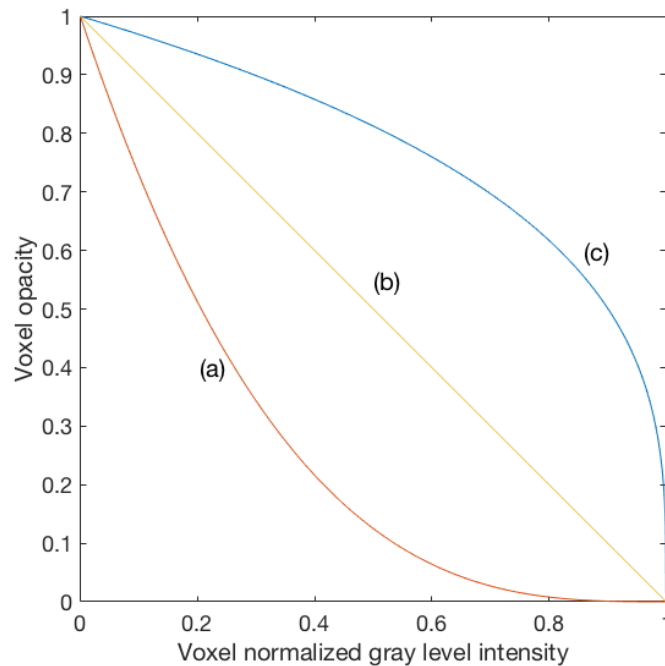


Figure 5.5: *Transparency* non-negative behavior show. Voxel opacity vs. Voxel normalized grey level intensity using three different parameter's value: 3 (a), 1 (b) and 0.3 (c).

Try to set the transparency value equal to 2 in the toy example.

The described steps correspond to **arrow number 4** in the data flow diagram.

5.11 Traces handling

The panel in the lower-right corner permits to handle objects' traces (panel (3) in Figure 5.3). It shows the list of all traces contained in the loaded file, that is four traces in the example. Let us recall that a trace records the center of an object moving in the

⁵The grey level value of each voxel is normalized to the color depth of the stacks.

space over time: hence, given a trace, it would be interesting to contemporary visualize the position of the object at a certain time and its trace from the beginning to the end of the observation.

As for the volume slices, once the data are loaded, all the traces are not shown in the axes: the user can plot in the main panel either the object's position marker or the line showing the whole object's positions marked inside all the frames, which in the following we refer to as route. The user can click the *G* key to show the object's position marker, after selecting on one or more traces from the list. To hide the marker, the user can simply use the combination *shift + G*.

Select the first trace from list and press *G*: a green circle located at the 3D coordinates of the trace will be shown in the corresponding frame, and it will be updated when frames changes.

To plot the entire route of the trace, the user first selects the trace in the list and then it presses the *R* key, whereas the combination *shift + R* should be used for hiding.

Press the *R* key on a trace: the software will show a black broken line, and the marker position will be updated when moving the frame slider.

If the user has selected a specific CV, traces outside the sub-volume are not of interest. It is possible to automatically select all the traced objects inside the box by clicking the *Select Traces* button at the bottom of the *CV Slices Management* panel. This operation highlights in the trace list the traces contained in the CV, hence, the user can press *G* and *R* keys to visualize them in the main panel.

The last item in the list, "new trace", is reserved to the definition of a new trace, which is considered as a trace with no object spotted in any frame.

Finally, the user can add to "Visual4DTracer" any image processing filter: to this aim, press the *Insert* button and select the Matlab's .m file with the filter code. The algorithm of the added filter must process the entire stack frame by frame; furthermore, the input parameter of the function must be a 3D matrix represented in double format, and it must return the processed 3D matrix of doubles. To remove the filter press the same button, which is now named as *Remove*; furthermore, if the user puts the mouse over this button, the full path of the current filter will be shown in a while.

5.12 Traces definition

This package offers a set of functionalities to trace objects 4D movements in a 3D environment. The basic tool allows to generate points in the 3D environment, which can be used to add or to modify the object position at a given time stamp. To this aim, the user clicks twice the same point in the 3D space by two different points of view. The rationale of this technique lies in the fact that a point in the 3D space can be identified as the intersection of two coplanar non parallel lines. The first line is defined by the user mouse click on the target object: the software sets the line on the direction perpendicular to the screen plane at mouse coordinates. The user then changes the point of view by rotating the volume and defines the second line by clicking again on the target object (Figure 5.6).

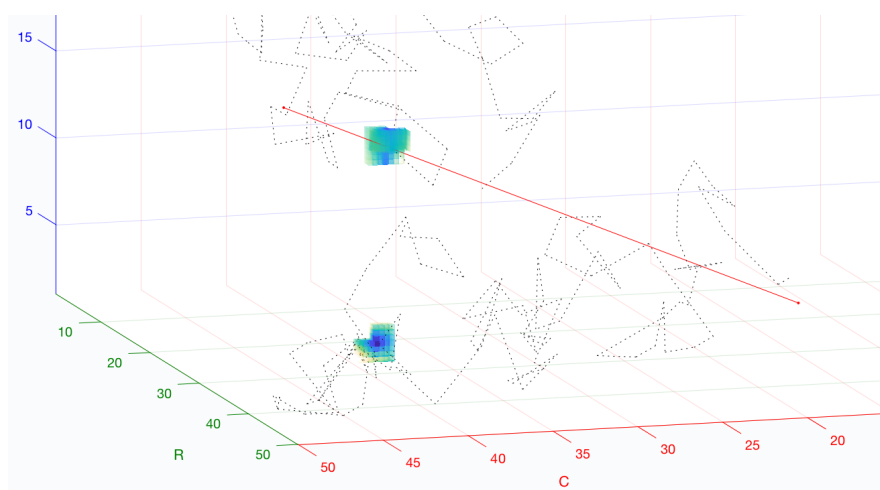


Figure 5.6: Zoom over the object's marking procedure: the user has defined the first line (red) and rotated the camera to define the second line and mark the object in the stack.

In practice, it seems unlikely the user sets two coplanar lines using the mouse, while it seems reasonable he/she sets two skew lines. On this ground, the target's coordinates are calculated as the middle point of the segment representing the minimum distance between the lines.

In the toy example set the frame number to 1 and focus your attention on the object closest to the ground, panning the object at the center of the screen to easily rotate it.

In the *traces list* now select the “new trace” item and press *G* to enable the visualization of markers you are defining. Hold down *T* and then hold down the mouse button to rotate the volume and to choose a desired point of view of the object, and then release

T. Next use the combination *shift* (a cross will substitute the arrow pointer) + *mouse button* to set the first line: a red line perpendicular to the screen will appear. Now change the direction of the camera rotating the volume and define another line using again *shift* + *mouse button*: the old red line will disappear and a marker will be set at the line intersection point.

The package offers a fast way to do the same task. Once you have selected the “new trace” item and pressed *G*. You are in a first default view automatically set, and then use *shift* + *mouse button* to set the first line. Press *C* to move to a second default view and then define another line. Note that you can update the values of the angles' view by rotating the camera using *T* and the mouse button, as described before.

In the toy example set the frame number to 1 and focus your attention on the object closest to the ground, panning the object at the center of the screen to easily rotate it. In the *traces list* now select the “new trace” item and press *G* to enable the visualization of markers you are defining. Hold down *T* and then hold down mouse button to rotate the volume and choose a desired point of view of the object, and then release *T*. Next use the combination *shift* (a cross will substitute the arrow pointer) + *mouse button* to set the first line: a red line perpendicular to the screen will appear. Now change the direction of the camera rotating the volume and define another line: the old red line will disappear and a marker will be set at the line intersection point.

Now that the first marker is set, use the *Frames* panel to move to next frame (or press *space* once) where no markers exist and repeat the aforementioned procedure. Once you have marked the object's coordinates for a certain number of frames trying to follow the object's movement inside the stack, press the *R* key on the trace list after selecting the “new trace” item. The result will be a broken line connecting detected object positions, which represents object path in the space and defines the new trace.

The user can repeat all steps described above to define other new traces: to add an entirely new trace click the *add new empty trace* button at the bottom of the trace list.

Given a trace, no more than a marker per frame can exist: therefore, if you set a new marker in a frame where the marker already exists, the software overwrite the marker coordinates. This permits to proofread an existing trace, as well as to handle any possible annotation errors.

Finally, to delete a marker right click on any position in the the main volume area and select the item *delete marker* from the contextual menu. If you are interested in deleting the entire trace, first select a trace in the trace list, second, right click on it and,

third, select the item *delete all markers* from the contextual menu.

At last, to distinguish an already analyzed trace from the others, in the trace list the user can right click on it and select the item “Mark as checked” from the contextual menu. This checking operation is mandatory to evaluate the performances, as described in 5.14.

The described steps correspond to **arrow number 5** in the data flow diagram.

5.13 Saving 4D traces

Once the user has traced and/or proofread the object positions, the data can be saved in an xml file, allowing to write and to exchange text files containing all information on traces and markers 4D coordinates⁶.

To this goal, click on the *File* menu and then on *Save* item: a dialog box pointing to the file folder where original data are stored will be shown. The user can choose a name for its file and save the traces.

When the number of objects to trace is quite large, or if the user needs to suspend the work, there is the possibility to save the traces, and then load it back later to continue. To agevolate this procedure, the user can export (and than import) the local variables of the GUI's setting.

To do it in our toy example click on “File” > “Export GUI's configuration file..” item, than choose a name and location for the file.

The file will contain all GUI's parameters set by the user before saving the local variables related to the *CV Slices Managment panel*, the *CV Slices Management panel*, the *Frames panel*, the traces list and the *View* menu.

To load back the local variables from a file, select “File” > “Import GUI's configuration file..” item and choose the previously exported file.

The described steps correspond to **arrow number 6** in the data flow diagram.

5.14 Performance evaluation

Given a stack, the software allows to estimate the accuracy of 4D object detection by comparing for all the traces the positions of their markers with the reference positions

⁶This format is adopted by many popular softwares, as *ImageJ*, *OsiriX*, etc.

of the markers of the same stack. For simplicity, the reference set of traces is referred to as *gold standard*, whereas the set of traces under investigation is named as *test set*. In practice, the test set can be the output of any 4D tracing software, and the gold standard can be the proofread stack.

Select from the trace list the trace number 3, then press *R* and *G* to plot the route of the trace and the object's marker. Now navigate the frames 11 - 15 and notice that the markers do not correctly match the object's position in the stack. Please proofread the volume correcting the wrong marker positions in these five frames, and then save again the file (creating a new xml file or overwriting the existing one). Experiment also the possibility to check this trace as marked and to save the local variables in another file.

The described steps correspond to **arrow number 7** in the data flow diagram.

Now it is possible to evaluate the performance of the original trace number 3 with respect to the proofread one.

In the example, we can consider the original toy data as the test set and the new set of traces as the gold standard. First load the original toy data in the software, second click on the *Evaluate* menu item, third select the item *Load xml gold standard traces...*, fourth select the recently proofread and saved xml file, finally import the local variables' file associated to the traces: the software will compare only the checked traces in the list. Finally, select the *Evaluate performances..* menu item in the same *Evaluate* menu.

To properly compare test set positions against the gold standard we need to ensure that each test set marker is uniquely associated with at most one gold standard marker.

The described steps correspond to **arrow number 8** in the data flow diagram.

Given a trace and time t^* , let us denote with (\mathbf{x}_m, t^*) and (\mathbf{x}_g, t^*) the 4D coordinates for a predicted blob and a gold standard blob, respectively, where \mathbf{x} stands for the 3D coordinates. A predicted blob is considered a *true positive* if $\|\mathbf{x}_m - \mathbf{x}_g\| \leq d$, where d is a distance in pixels that the user can set exploiting a-priori knowledge on the biological domain under investigation. In the other case, the predicted blob is considered a *false positive*, whereas we have a *false negative* when the predicted blob is missing in comparison with the gold standard. As occlusions during blob movements can occur, the evaluation tool also counts how many times a blob disappears both in the predicted and in the gold standard traces. The results are saved in a csv file; at the beginning it

reports a brief parameters description, followed by the the list of performance computed for each trace. Furthermore, it also reports the average value of all the performances scores.

In the previous example, once the gold standard has been selected, a window will pop up so that the user can set the value of D : for our toy data, set the value to 3 and press *Ok*. In the next dialog box, insert a name for the output file.

The described steps correspond to **arrow number 9** in the data flow diagram.

5.15 Other utilities

The GUI allows to set other minor parameters related to how images and traces are rendered. With reference to Fig. 5.4, these options are located in the *View* menu and consist in the following five main items:

Axes: it allows to set the axes ratio. The default value is *Equal*.

Linetype: this changes the line-type of both markers and traces. The former can be represented as dot and circles, whereas the latter are represented using solid or dashed lines. The default values are *circle* and *solid*, respectively.

Slices: it is composed by two items. The first, named as *Colormap*, changes the colormap and representation of slices. The implemented maps are *gray*, *gray inverted* (which inverts the gray color map), *highlighted* and *highlighted inverted* (which again inverts the colormap). The second item is named as *Direction* and it shows the slices along all the dimensions or only along the Z axis to alleviate the memory usage for big images. The default values are *gray* and *all*, respectively.

Render: it permits to select between two rendering engines, namely *OpenGL* and *Painter*. The former uses the hardware acceleration to perform faster image rendering; the latter produces a more detailed, but even slower, result. The default value is *OpenGL*.

GUI color: it permits to set basic theme colors for the GUI. The default value is *blue*.

The described steps correspond to **arrow number 10** in the data flow diagram.

6 T1DM - Tracks features extraction and dynamics clustering analysis

In this section, we resume the description of tracks' exploratory analysis, which we interrupted after section 4.5 to present the tracking tool. The following sections focus on the ML analysis we computed on the traces extracted from the insulin granules and filtered using the criteria described in section 6.2.

In this work in fact the gold standard annotation was carried out manually by two human experts, using the described software: we randomly selected 103 traces from the four subsets mentioned at the end of section 4.5. The subsets in fact contain traces with different quality, *i.e.* 6403 *very low quality* traces, 930 in the *low quality* pool, 476 in *average quality* pool and 938 *good quality* traces. For each of these traces, F_1 scores were computed varying the value of the threshold R . The results are reported in Figure 6.2, showing that the subsets containing *very low quality* and *low quality* traces should be rejected. Indeed, the low values of F_1 scores suggest that they are not reliable enough to be representative of the genuine granules' dynamics and, hence, they could lead to ambiguous considerations if employed in the following data analysis.

6.1 Features extraction

This subsection describes the features extracted from traces belonging to the *average quality* and *good quality* subsets, as motivated in previous section 5. Such features aim at quantifying the dynamic behavior of the granules inside their beta-cells. We defined 14 descriptors, which, for the sake of presentation, were divided in two groups, named as *Route* and *Time function features*. The former describes the route of the granule, whereas the latter computes quantities related to the hourly law of the motion. We now qualitatively presents such measures, whilst Table 6.1 introduces their formal definition.

Route features:

- *Total displacement*: it is the sum of length of segments composing the broken line

of granule path and it represents the distance the granule travels within the cell;

- *Angles average*: this is the average of the variation, frame by frame, of the angle between consecutive segments of granule path; it represents the extent of track wiggling and it measures the tendency to follow a straight line;
- *Angles standard deviation*: this is the standard deviation of granule's frame by frame angular variation of direction, and it measures the stability of swirling of the track;
- *Tortuosity*: it is the ratio between the *total displacement* and the Euclidean distance between the first and the last track point; it quantifies the complexity of the route with respect to its linear displacement vector;
- *Energy*: it is the square power of the density distribution of granule frame by frame displacements;
- *Entropy*: it quantifies the state of disorder the track introduces.

Time function features:

- *Average velocity*: it is the discrete derivative of the *total displacement* and it describes the average velocity of the granule;
- *Velocity standard deviation*: it measures the difference between the punctual velocity and the *average velocity*;
- *Velocity skewness*: it measures the asymmetry in punctual velocities' distribution;
- *Velocity kurtosis*: it gives an index of preference of a specific punctual velocity in the track;
- *Average acceleration*: it is the discrete derivative of the punctual velocity;
- *Acceleration standard deviation*; it measures the difference between the punctual acceleration and the *average acceleration*;
- *Acceleration skewness*: it measures the asymmetry in the distribution of punctual accelerations;
- *Acceleration kurtosis*: it gives an index of preference of a specific punctual acceleration in the track;

Preliminary symbols definitions

\mathbf{F} is the set including the index of frames which contain one marker of the granule under study
\mathbf{x} , \mathbf{y} and \mathbf{z} are vectors containing the coordinates of the marker of the granule under study at each frame $f \in \mathbf{F}$
$\mathbf{p}_f = (\mathbf{x}_f, \mathbf{y}_f, \mathbf{z}_f)$, where $f \in [0; \mathbf{F} - 1]$
Set of displacements: $\mathbf{D} = \{\ \mathbf{p}_{f+1} - \mathbf{p}_f\ , f \in [0; \mathbf{F} - 1]\}$
$pdf(\cdot)$: probability density function of \cdot
Set of angles: $\mathbf{R} = \left\{ \arccos \left(\frac{(\mathbf{p}_{f+1} - \mathbf{p}_f) \cdot (\mathbf{p}_{f+2} - \mathbf{p}_{f+1})}{\ \mathbf{p}_{f+1} - \mathbf{p}_f\ \cdot \ \mathbf{p}_{f+2} - \mathbf{p}_{f+1}\ } \right), f \in [0; \mathbf{F} - 2] \right\}$
Set of velocity: $\mathbf{V} = \left\{ \frac{\mathbf{p}_f}{\mathbf{F}(f+1) - \mathbf{F}(f)}, f \in [0; \mathbf{F} - 1] \right\}$
Set of acceleration: $\mathbf{A} = \left\{ \frac{\mathbf{V}_f}{\mathbf{F}(f+1) - \mathbf{F}(f)}, f \in [0; \mathbf{F} - 2] \right\}$

Features

<i>Route-based</i>	<i>Hourly law-based</i>
Total displacement (Δ) = $\sum_{i=1}^{ \mathbf{D} } D_i$	Average velocity ($\bar{\mathbf{V}}$) = $\frac{1}{ \mathbf{V} } \sum_{i=1}^{ \mathbf{V} } \mathbf{V}_i$
Average angles ($\bar{\mathbf{R}}$) = $\frac{1}{ \mathbf{R} } \sum_{i=1}^{ \mathbf{R} } \mathbf{R}_i$	Velocity standard deviation = $\sqrt{\frac{\sum_{i=1}^{ \mathbf{V} } (\mathbf{V}_i - \bar{\mathbf{V}})^2}{ \mathbf{V} - 1}}$
Angles standard deviation = $\sqrt{\frac{\sum_{i=1}^{ \mathbf{R} } (\mathbf{R}_i - \bar{\mathbf{R}})^2}{ \mathbf{R} - 1}}$	Velocity skewness = $\frac{\frac{1}{ \mathbf{V} } \sum_{i=1}^{ \mathbf{V} } (\mathbf{V}_i - \bar{\mathbf{V}})^3}{\left(\sqrt{\frac{1}{ \mathbf{V} } \sum_{i=1}^{ \mathbf{V} } (\mathbf{V}_i - \bar{\mathbf{V}})^2} \right)^3}$
Tortuosity = $\frac{\Delta}{ \mathbf{p}_{ \mathbf{F} -1} - \mathbf{p}_1 }$	Velocity kurtosis = $\frac{\frac{1}{ \mathbf{V} } \sum_{i=1}^{ \mathbf{V} } (\mathbf{V}_i - \bar{\mathbf{V}})^4}{\left(\sqrt{\frac{1}{ \mathbf{V} } \sum_{i=1}^{ \mathbf{V} } (\mathbf{V}_i - \bar{\mathbf{V}})^2} \right)^2}$
Energy = $pdf(\mathbf{x})^2 + pdf(\mathbf{y})^2 + pdf(\mathbf{z})^2$	Average acceleration ($\bar{\mathbf{A}}$) = $\frac{1}{ \mathbf{A} } \sum_{i=1}^{ \mathbf{A} } \mathbf{A}_i$
Entropy = $-(pdf(\mathbf{x}) \log_2(pdf(\mathbf{x})) + pdf(\mathbf{y}) \log_2(pdf(\mathbf{y})) + pdf(\mathbf{z}) \log_2(pdf(\mathbf{z})))$	Acceleration standard deviation = $\sqrt{\frac{\sum_{i=1}^{ \mathbf{A} } (\mathbf{A}_i - \bar{\mathbf{A}})^2}{ \mathbf{A} - 1}}$
	Acceleration skewness = $\frac{\frac{1}{ \mathbf{A} } \sum_{i=1}^{ \mathbf{A} } (\mathbf{A}_i - \bar{\mathbf{A}})^3}{\left(\sqrt{\frac{1}{ \mathbf{A} } \sum_{i=1}^{ \mathbf{A} } (\mathbf{A}_i - \bar{\mathbf{A}})^2} \right)^3}$
	Acceleration kurtosis = $\frac{\frac{1}{ \mathbf{A} } \sum_{i=1}^{ \mathbf{A} } (\mathbf{A}_i - \bar{\mathbf{A}})^4}{\left(\sqrt{\frac{1}{ \mathbf{A} } \sum_{i=1}^{ \mathbf{A} } (\mathbf{A}_i - \bar{\mathbf{A}})^2} \right)^2}$

Table 6.1: Formal definition of extracted features.

Noteworthy, although exist other cinematic features from literature we could calculate from the analyzed tracks [100], we chose all fundamental quantities to characterize our study without implementing derivative more complex formulas. This allowed us to keep the approach basic enough to easily explain all possible results by the cinematic point of view and allowing for deeper considerations for further related works.

6.2 Exploratory clustering analysis

As mentioned in section 2.4, we aimed at quantitatively investigate granules' dynamics by looking for possible distinct subgroups of granules with different motion behaviors. We merged the images collected under *5.5 mM glucose* and *16.7 mM glucose*. This permitted us to analyze the data without any particular a priori assumption on any model underlying them: indeed, the use of an unsupervised learning approach would permit us to extract previously unknown relationships. We first visually inspected the data looking for clouds of points. We observed 143 scatter plots given by the all the possible triplets of the extracted features. In these charts, each point represents the trace of a granule, and the centers of any detected cloud indicates the points average values, i.e. the common cinematic behavior of the corresponding grouped traces. This inspection revealed the presence of some outliers: this could be expected due to the intrinsic noise affecting the traces (four scatter plots are depicted in Figure 6.1, showing the presence of possible outliers). Moreover, some feature combination suggests the existence of two distinct clusters as it can be observed in the two charts at the bottom of Figure 6.1, where most of the dots are concentrated around two distinct centroids.

Algorithm 1 K-NN based outlier search algorithm

```
1: for  $p$  in allPoints do
2:    $kNn =$  firsts  $k$  neighbour of  $p$ 
3:
4:   for  $n$  in  $kNn$  do
5:      $pComp = pComp + 1/\mathbf{dist}(n, p)$ 
6:   end for
7:
8:   if  $pComp/k \leq thComp$  then
9:      $pComp$  is an outlier
10:  end if
11: end for
```

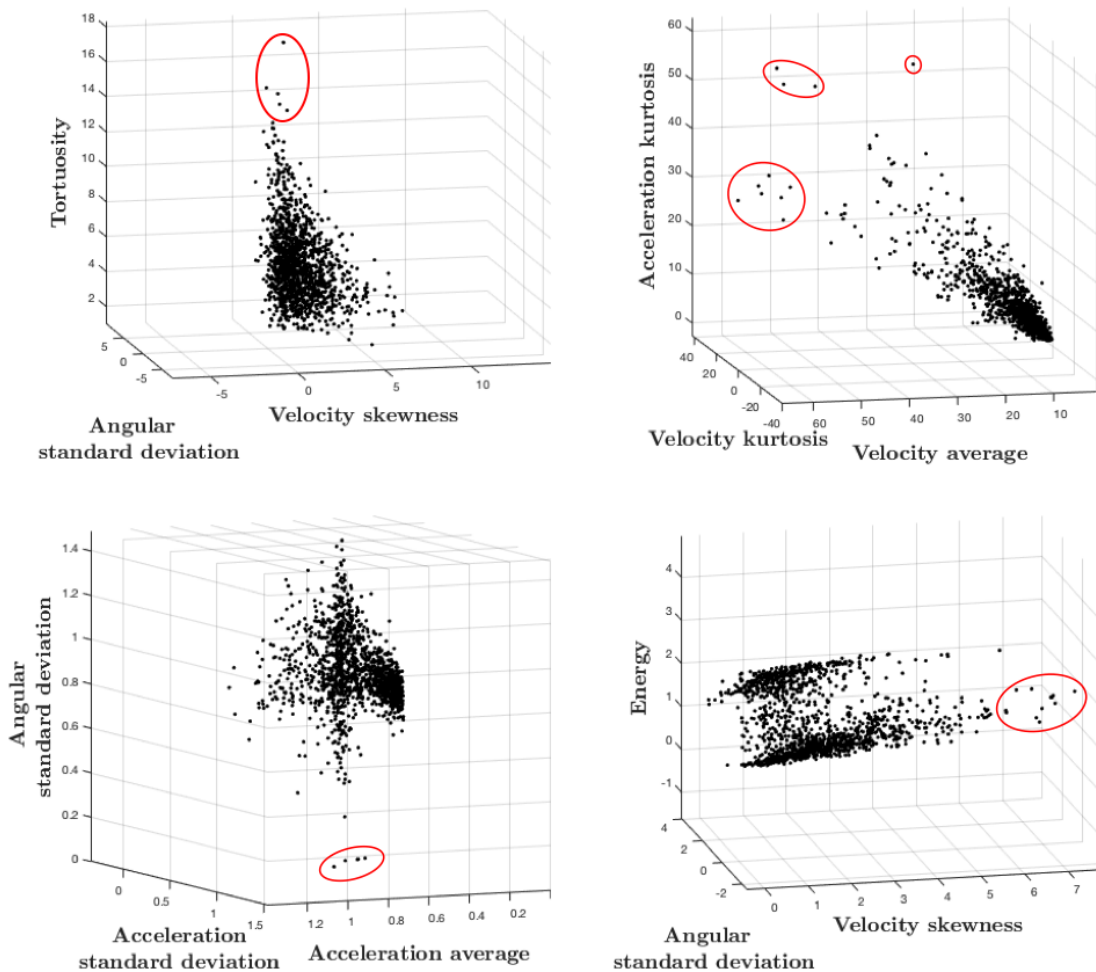


Figure 6.1: Four examples of triplets of the features randomly extracted from the 143 different combinations. The red circles show the outliers.

Noteworthy, having already rejected the two unreliable subsets of traces resulted in plots with fewer outliers. We therefore removed all the points with a low probability to belong to the cloud applying a machine learning approach which implements for compactness measurement a variation of the k-NN algorithm (algorithm 1). In the code, *allPoints* is the set containing all the points represented in the selected feature space, p is a counter, k is the number of p -th neighbors to be stored in kNn , which are also the points used to estimate the compactness $pComp$ of p . The compactness is computed on the basis of the distance of p with respect to its neighborhood's points. If the compactness of p is not greater than a superimposed threshold $thComp$, p is labeled as an outlier. The parameters k and $thComp$ were empirically set to 18 and 171, respectively, after a preliminary investigation.

After the outliers removal we quantitatively confirm the existence of two clouds of points already identified by the visual inspection. In this respect, by applying the *Expectation Maximization* clustering algorithm to the remaining data. This method was selected because it is able to identify Bayesian-like and density-based distributions [101], as suggested observing of the scatter plots. Finally, we statistically validated the resulting clusters using a *Mann Whitney U-test*.

6.3 Results and Discussion

The dataset presented in section 4.3.3 is composed of six days scheduled acquisition sessions, consisting of 45 minutes experiments per session. The optical system set introduced a temporal delay, which led to time-lapsed series of around 220 three-dimensional stacks per experiment. Splitting each time series at the moment of the addition of glucose, we average, by have 80 initial stacks for *5.5 mM glucose*, 120 ending stacks for *16.7 mM glucose*¹ and 20 rejected stacks where the osmotic volumetric variation during nutrient addition was captured.

Following the procedure described in previous section, we analyzed the tracks' quality; a broad section of thousands of granules found by the software revealed to be artifact introduced by background noise and cytoplasmic dust. In order to filter the less reliable tracks we first divided the entire pool into quartiles using tracks' duration to estimate their quality² (section 4.5), then we proceeded with the automatic tracking validation step, which allowed us to keep only the most reliable quartiles. We evaluated the average tracking performances of the automatic approach adopted with respect to a gold standard manually annotated. In this respect, the evaluation procedure described in subsection 5 dichotomizes each track's point into "valid" or "invalid" on the basis of a threshold measuring of the distance from the gold standard equivalent point. Since the "true value of the threshold is unknown, we measured the performances varying such parameters between 1 and 20 voxels. The corresponding performances achieved are shown in Fig. 6.2. It shows that when we consider a distance threshold equal or larger than 5 voxels we obtained a F1-score higher than 80% and 90% for average and good quality traces, respectively. If we consider only these sets of traces, since the average of granules'

¹Due to the fact that not all the *16.7 mM glucose* group acquisitions have the same number of frames, when we specifically calculated the *Displacement* feature (subsection 6.1) we ignored the last 40 stacks in order to not bias the result.

²In fact, besides the number of stacks (*i.e.* the number of frames) into *5.5 mM glucose* and *16.7 mM glucose* stacks was often unbalanced, the duration of a track could still be considered a good meter of tracking reliability.

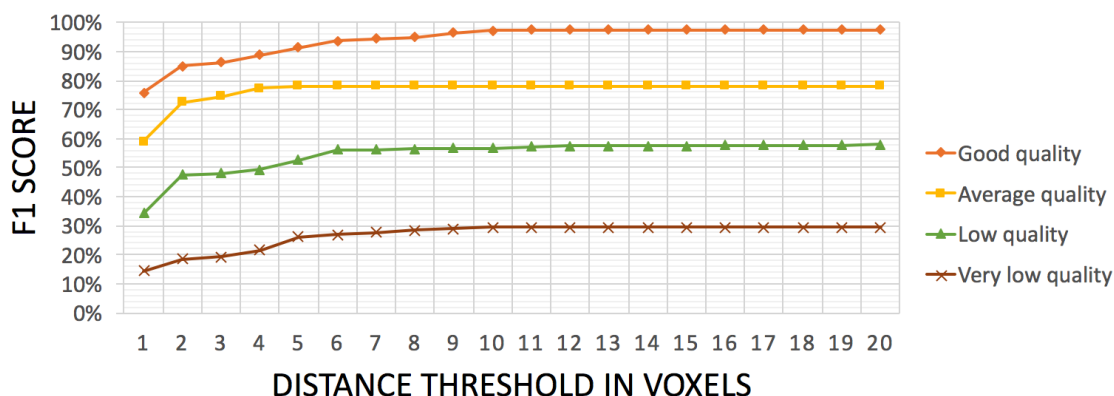


Figure 6.2: Automatic tracking evaluation: all the four tracks' pools are evaluated sweeping the threshold distance parameters from 1 to 20 voxels and the plot reports their corresponding average *F1-score*.

diameters is 5 voxels, the automatic tracker is uncertain about the segmented positions no more than 20% of times and it can make wrong classification between two types of granules only if they collide. Furthermore, it is worth noting that when the threshold distance parameters exceed a certain value of voxels, the performances get an upper bound. This reflects the inability of the tracker to detect the granule in some of the collected frames *i.e.* it counts the number of absent detections of the automatic tracker, which are part of the evaluation criteria, and represents a segmentation error not involved with software's tracking performances rather than the overall images acquisition noise obtained at the beginning of the pipeline.

After the validation step we had 995 tracks and we computed the 14 described features. Let us shortly recall that we aim was analyze granules' dynamics inside the cells and reveal different types of granule motions through the cinematic quantities extracted and visualized in clustered clouds of points. Moreover, we were interested in assessment of different particles' behavior independently from their cells' incubation period and nutrients' stimulation: on this ground we combined all collected tracks in a single pool regardless of *5.5 mM glucose* and *16.7 mM glucose* conditions.

As mentioned in section 4.4 above, we first visually inspected the data observing the tendency to group together. This allowed us to identify three quantities, namely *Angles average*, *Acceleration standard deviation* and *Entropy*, which show two clouds of traces, as depicted in Fig. 6.3-A. Next, we identified the outliers into the data using the approach described in subsection 6.2. Corresponding results are reported in Fig. 6.3-B.

Having removed the outliers, we applied the Expectation Maximization algorithm to quantify the observed subdivision of data in two classes with different dynamic. This

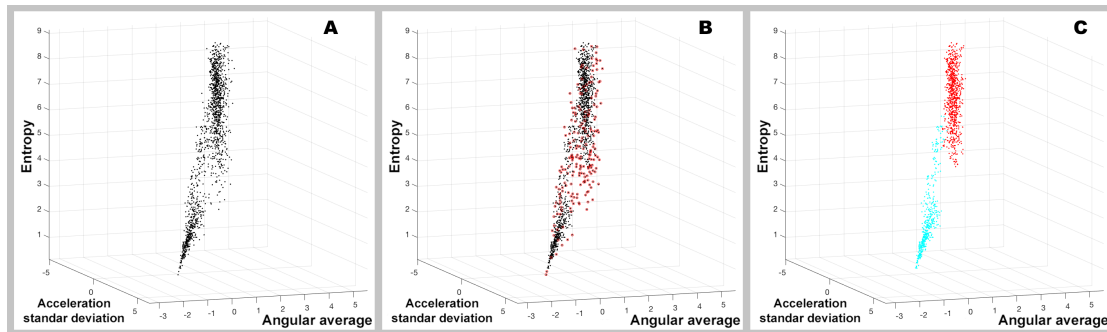


Figure 6.3: Three-dimensional combination of features showing the presence of two distinct family of granules. **A** panel - raw data, **B** panel - tracks with outliers identified, **C** panel - data clustered in two distinct classes (red and cyan dots) after the outlier removal.

gaussian-oriented method distinguished the two distributions on the basis of their different densities. Indeed, each cloud of points seemed to represent a group of granules moving with a common driving criterion, visually identified by the centroids of the data, i.e., the means of their distributions, and actually hindered by the presence of mutual collisions and intracellular gradients, shown by the spreads around the clouds' centers, i.e., the distributions' standard deviations. The results are shown in Fig. 6.3-C. Finally, we verified the quality of our results and we quantitatively proved, from a statistical point of view, that the two clouds of point were distinct. To this end, we performed a Mann Whitney U-test and considered the granules' classes as samples belonging to the same population for the null hypothesis. We get $p < 0.001$, thus confirming the separability of the two groups.

From a biological perspective, these results show the presence of two distinct families of granules coexisting inside the beta-cells that are identifiable by their different dynamics. The first is a family characterized by a diffusive motion, showing a higher angular average and a lower standard deviation of their acceleration at the same time, but resulting to have a greater movement entropy (depicted as red in the plot shown in Fig. 6.3-C). The second group includes granules with a more linear dynamics, which exhibits a lower angular average, as well as lower entropy and a more spread acceleration distribution.

6.4 Fully quantitative clustering analysis

Although the aforementioned clustering method showed a promising discriminative approach, able to distinguish two different pools of dynamic behaviors of the granules, in order to increase the reliability and the robustness of these results we performed an extra

analysis, leaving the visual inspection approach after the dynamical feature computation step and introducing a more quantitative data assessment.

In fact, the previous exploratory approach was limited since we visually inspected the data, being therefore limited to a 3D analysis. This forced us to neglect any possible feature combination composed of more than three descriptors. Oppositely, now we perform an exhaustive search to find out the feature set that best clusterize the data accordingly to a given performance score. Again, the clustering algorithm is the *Expectation Maximization*.

The number of different features' combinations assessed in the exhaustive analysis is:

$$\sum_{s=1}^N \binom{s}{N} = 2^N - 1 \quad (6.1)$$

with N the number of extracted features.

For each set we fix the number of clusters k to 2, 3 and 4; any cluster could group together all granules with a common dynamic behavior and, to the best of our knowledge, in a physiological state the granules should roughly divide into two different families [102]. In fact, in order to evaluate the number of clusters to be investigated, we performed some preliminary experiments and we ran the *elbow* test [103] on the data using the *K-means* algorithm³. We randomly extracted 10 subgroups among all the possible sets of features' combinations given by Equation 6.1. On this set we applied the *K-means* varying k parameter in [1; 10]. Then, for each obtained data partition we computed the global variance as the sum of its clusters' variance and stored it into a global variance vector in function of k . Finally, we computed the average elbow vector as the average of all subsets' global variances vectors⁴. The plot of the average elbow vector is shown in Figure 6.4. As depicted on the plot, the k values in the proximity of the elbow are equal to 2, 3 and 4. On this ground, in our exhaustive analysis we limited the number of clusters to such values, since a large number of partitions in the dataset would lead to a degradation of the clustering compactness producing not enough robust results.

After the exhaustive feature search step, we need to compare which features combination permits to better separate the groups. To this aim we use a metric to evaluate the clustering performance. Since there was no external reference to assess the goodness of the results, we used an internal validation measure to establish the clustering reliability. First we performed a track-oriented evaluation, computing the *Silhouette index* to know the specific distribution of the level of membership of each track to the own cluster.

³The choice of this clustering algorithm is motivated by its low computational burden.

⁴During this latest calculation we rejected all not-elbow-like vectors because considered poorly discriminative.

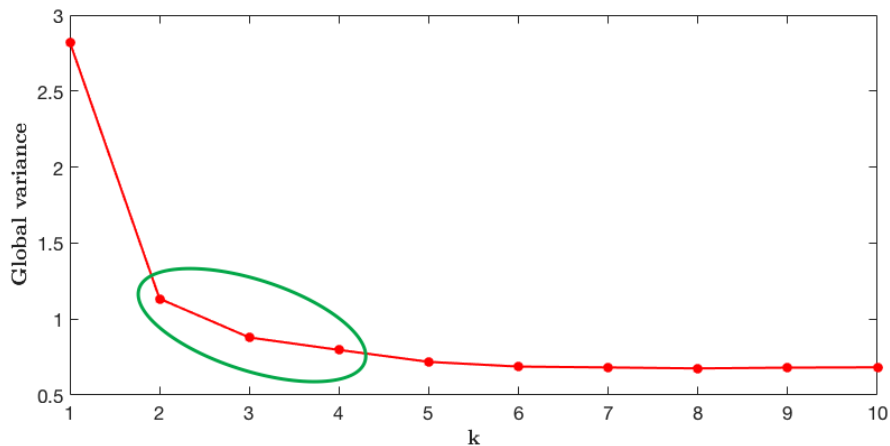


Figure 6.4: Average elbow vector on randomly selected feature combination subsets. The clustering algorithm used is the K -means and the k parameter lies in \mathbb{N} , ranging in $[1; 10]$, the green ellipse indicates the k values in the proximity of plot's elbow.

Second, to have a more heterogeneous evaluation of the clusters against the features' set variation we computed three global-oriented metrics, namely the *Davies-Bouldin index*, the *Calinski-Harabasz index* and the *Simmety_Distance between and within index*. A brief description of each index is reported in the next section.

6.5 Internal clustering evaluation indexes

When assessing the goodness of a clustering approach, a quantitative index is required in order to evaluate all analysis performed with a high number of features, where a direct visual inspection is not allowed. When no external information of the data structure is available, the internal measures evaluate the validity of a clustering. In order to evaluate the ability of an index to evaluate the goodness of a group of clusters, it is possible to introduce two basic concepts of internal validation measures: (i) *compactness*, it measures how closely related the objects in a cluster are, (ii) *separation*, that measures how distinct or well-separated a cluster is from other clusters. Moreover, a good clustering should be fulfill several main property, as monotonicity, noise, density, subclusters and skewed distributions [104]. According to such issues, we identified 4 metrics suited to measure clustering results. They are defined below according to following notations:

- n : number of objects;
- NC : number of clusters;

- C_i : the i -th cluster;
- n_i : number of objects in C_i ;
- $d(x, y)$: Euclidean distance between x and y ;
- w : the middle point of the line segment defined by the cluster's center;
- $stdev$: average standard deviation of the clusters

Silhouette index [105]: each cluster is represented by a so-called silhouette, which is based on the comparison of its tightness and separation. This silhouette shows which objects lie well within their cluster, and which ones are merely somewhere in between clusters. The Silhouette index is defined for each sample and is composed of two scores: the mean distance between a sample and all other points in the same class and the mean distance between a sample and all other points in the next nearest cluster. Comparing the two scores we calculate the final score in the range $[-1; 1]$, for each point into the dataset, where a positive value remarks a stronger affinity with its belonging cluster rather than with the others.

Its formal definition is:

$$\frac{1}{NC} \sum_i \left\{ \frac{1}{n_i} \sum_{x \in C_i} \frac{b(x) - a(x)}{\max[b(x), a(x)]} \right\} \quad (6.2)$$

with:

$$a(x) = \frac{1}{n_i - 1} \sum_{y \in C_i, y \neq x} d(x, y),$$

$$b(x) = \min_{j, j \neq i} \left[\frac{1}{n_j} \sum_{y \in C_j} d(x, y) \right]$$

The graphical interpretation of these quantities is shown in Figure 6.5:

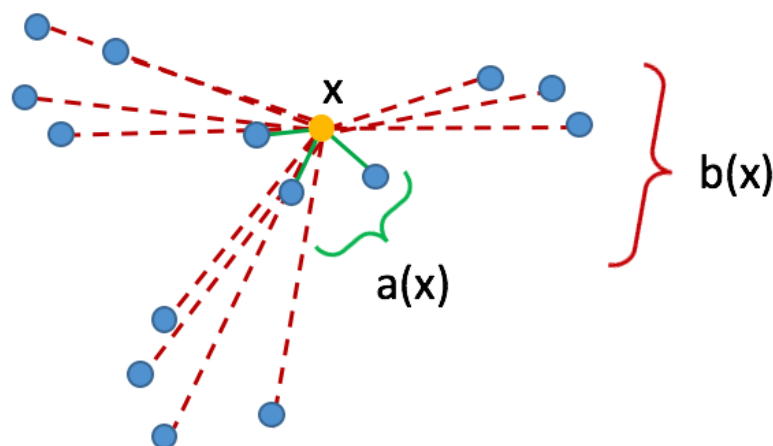


Figure 6.5: Representation of the Silhouette index's calculation.

Davies-Bouldin index [106]: it is an internal evaluation scheme, where the validation of how well the clustering has been done is made using quantities and features inherent to the dataset. The measure can be used to infer the appropriateness of data partitions and can therefore be used to compare relative appropriateness of various divisions of the data. The measure does not depend on either the number of clusters analyzed nor the method of partitioning of the data and can be used to guide a cluster seeking algorithm. The score is in range $[0; +\infty]$ and values closer to zero indicate a better partition.

Its formal definition is:

$$\frac{1}{NC} \sum_i \min_{j, j \neq i} \left\{ \left[\frac{1}{n_i} \sum_{x \in C_i} d(x, c_i) + \frac{1}{n_j} \sum_{x \in C_j} d(x, c_j) \right] / d(c_i, c_j) \right\} \quad (6.3)$$

Calinski-Harabasz index [107]: it can be used to evaluate the model and a higher score relates to a model with better defined clusters. The range of possible values is $[0; +\infty]$ and the score is higher when clusters are dense and well separated, which relates to a standard concept of a cluster and another advantage is that the score is fast to compute if compared with the latter indexes and it is often referred to as one of the most efficient techniques for determining the number of clusters.

Its formal definition is:

$$\frac{\sum_i n_i d^2(c_i, c) / (NC - 1)}{\sum_i \sum_{x \in C_i} d^2(x, C_i) / (n - NC)} \quad (6.4)$$

Simmetry Distance between and within index [108]: it relies on the clusters' compactness (in terms of intra-cluster variance), and on the density between clusters (in terms of inter-cluster density). It evaluates the average density in the region among clusters in relation with the density of the clusters and then, considers also an inter-cluster density measure leveraging on the number of points in the neighborhood of cluster's centroids. It has been shown that this index is the only internal validation measure which performs well in all five mentioned aspects. The score ranges in $[0; +\infty]$ where a lower value indicates a better clustering.

Its formal definition is:

$$Scat(NC) + Dens_bw(NC) \quad (6.5)$$

with:

$$Scat(NC) = \frac{1}{NC} \sum_i \|\sigma(C_i)\| / \|\sigma(D)\|,$$

$$Dens_bw(NC) = \frac{1}{NC(NC - 1)} \sum_i \left[\sum_{j,j \neq i} \frac{\sum_{x \in C_i \cup C_j} f(x, u_{ij})}{\max\{\sum_{x \in C_i} f(x, C_i), \sum_{x \in C_j} f(x, C_j)\}} \right],$$

$$f(x, u) = \begin{cases} 0, & \text{if } d(x, u) > stdev \\ 1, & \text{otherwise} \end{cases}$$

6.6 Results and Discussion

After the quartile selection step applied to all the tracks automatically extracted, we had 1414 tracks, *i.e.* all tracks belonging to the *average quality* and *good quality* groups and we computed the 14 described features. In our first experience after rejecting the tracks without enough quality we ran an additional filtering step, that identifies and removes the outliers via a manually tuned algorithm (section 6.2). Oppositely, the quantitative approach how introduced allowed us to find features discriminant enough to exclude the latter algorithm from the pipeline, since all possible outliers are handled by the clustering algorithm and don't affect the scatters' assessment as instead could happened via visual inspection. As described in our first approach, firstly we divided the elaboration in two phases.

First, we combined all collected tracks in a single pool regardless of *5.5 mM glucose* and *16.7 mM glucose* conditions. This way all granules where analyzed during their entire acquisition period, allowing the clustering algorithms to identify the clouds of

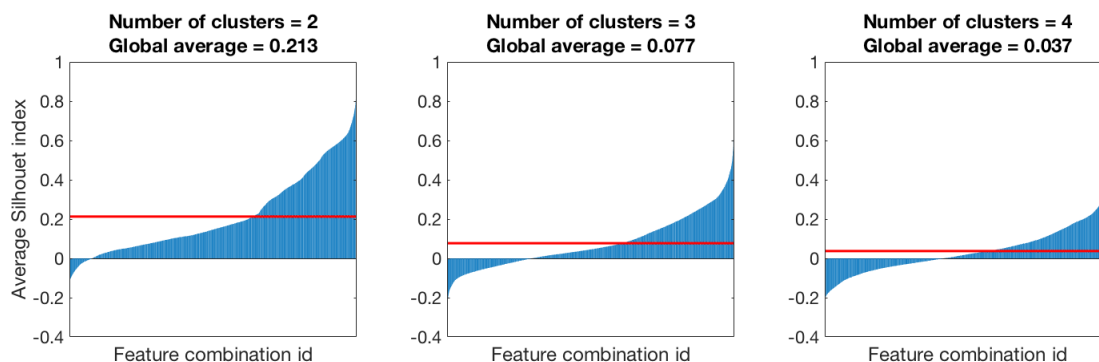


Figure 6.6: Sorted *Silhouette indexes* means vs. the number of clusters identified from the clustering algorithm. The red lines represent the global average of the plots.

points leveraging on all their dynamics after have considered both the physiological ambient and the ambient with an higher concentration of glucose.

To select the number of clusters that maximizes the clustering performance we use the *Silhouette index*: Figure. 6.6 plots the sorted distribution of silhouette indices for each feature calculated for the number of clusters equal to 2, 3 and 4, respectively. It also reports the global averages of the silhouettes (red line in the plots). The greater predominance of positive averages belonging to the result attended with $k = 2$, suggests to search for only two different clouds of points.

After have determined the number of clusters, we proceeded with the identification of the most discriminative features, among the initial set of 14. In order to make more robust the approach, the *Davies-Bouldin index* and the *Calinski-Harabasz index* was calculated over all the features combinations: the first common features matching the highest scores of all indexes were the *Average angles*, *Angles standard deviation* and *Energy*, as reported in Table 6.2.

For the sake of completeness, when ranking the intersection of features' combinations produced by the indexes' computation, we did not considered the possible combination of one single feature, in fact the discriminative contribution offered by only one feature could be easily affected by noise, therefore misleading the analysis. The scatter plot of the aforementioned features is shown in Fig.6.7

In order to quantitatively determine how the data splits into two clusters with different motions, we chose the Expectation Maximization algorithm because of its ability to identify Bayesian-like and density-based distributions: observing the centroids of the clustered clouds of points we can thus identify two distinct dynamical behaviors of the tracks. From a cinematic point of view, both groups of granules present a specific driving criterion that is visually identifiable with their own zones of high density, *i.e.*

Feature name vs. Index	SH	DB	CH	S_Dbw
Average velocity	–	–	–	2
Average acceleration	–	–	–	1
Acceleration standard deviation	–	–	–	3
Average angles	1	1	1	1
Angles standard deviation	2	3	–	1
Energy	3	–	2	–
Entropy	–	2	–	1

Table 6.2: Rank of features chosen by the *Silhouette* index (SH), *Davies-Bouldin* index (DB), *Calinski-Harabasz* index (CH) and *Simmerty-Distance between and within* index (S_Dbw) index when evaluating all the feature sets' clusters; given the number of clusters equal to 2.

the centroids of each cloud where the most of granules accumulate. Then, the scatter shows a minority of points surrounding each clusters' kernel: the equivalent granules present slightly divergent dynamics probably caused by their mutual collisions inside the cells that hinder their spontaneous motion and by several different intracellular gradients. These samples spread around the clouds' centers and they determine the standard deviations of the two distributions.

From a biological perspective, these results show the presence of two distinct families of granules coexisting inside the beta-cells that are identifiable by their different dynamics. The first is a family characterized by a diffusive motion and referred to as *Diffusive granules*, showing a higher angular average and a lower standard deviation of their acceleration at the same time, but resulting to have a greater movement energy (depicted as cyne in the plot shown in Fig. 6.7). The second group instead, referred to as *Tubular granules*, includes granules with a more linear dynamics, which exhibits a lower angular average, as well as lower entropy and a more spread acceleration distribution.

After the clustering subdivision we proceeded with the second phase of the analysis and removed the limitation of considering all the acquired tracks belonging to the same pool. Now we investigated the influence of an higher glucose concentration in the physiology of the cells. In both the clouds of points we split the *5.5 mM glucose* tracks from the *16.7 mM glucose* ones and we perform a *Mann-Whitney U-test* to verify if the dynamic observed in normal conditions (*5.5 mM glucose* concentration) is statistically different from dynamic obtained with a higher glucose concentration. The obtained *p-values* are shown in Tab. 6.3: they reject the null hypothesis for all the selected features except when considering the *Angles standard deviation* of the *Diffusive granules* cluster and therefore suggest the existence of a statistically significative variation in dynamics produced by

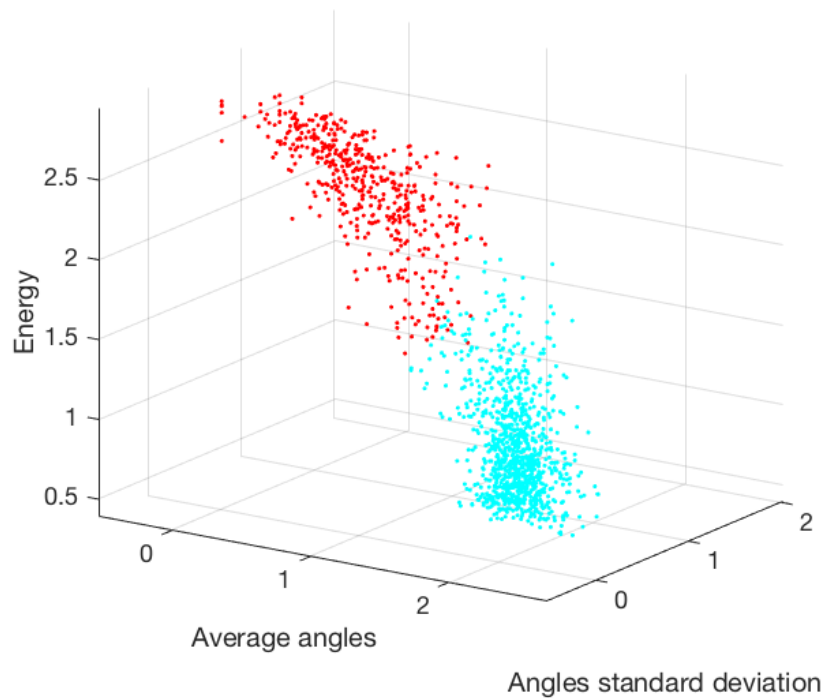


Figure 6.7: Scatter plot of the tracks after the clustering analysis and the feature selection. The two clouds of points are depicted in different colors.

an increment of glucose concentration.

A possible explanation of the similarity between this dynamical behaviors, could be that when considering a possible diffusive transportation the granules assume mainly chaotic motions with different directions thus result less affected by a nutrient variation in the surrounding ambient, thus producing a more disordered dynamical reaction.

P-value	Average angles	Angles standard deviation	Energy
<i>Diffusive granules</i>	0.018	0.394	< 0.001
<i>Tubular granules</i>	< 0.001	0.003	< 0.001

Table 6.3: *P-values* resulting from the *Mann-Whitney U-test* performed on the *5.5 mM glucose* and the *16.7 mM glucose* tracks for the two clustered clouds of points.

6.7 Conclusion

In this study we described an approach to detect insulin granules' dynamic inside the beta cells by employing confocal spinning disk microscopy. It works with 4D images and it mines the data via a machine learning approach.

To tackle the huge number of collected granules we tested several free softwares for automatic tracking, and finding out the *TrackMate* software best fits our goal. The statistical validation was performed with a homemade manual tracking application and the work of two operators. Next we extracted several cinematic features to quantify the dynamic of each track and we clustered their behavior, thus identifying two classes of tracks with distinct motion typologies.

We are aware that the main limitation of this method resides in the number of analyzed tracks that leads complex statistical analysis. Moreover, further development is planned to improve the spatial resolution of the images and producing a finer granules' tracking as well.

The resulting findings of this preliminary study show the presence of two sets of granules, marked by distinct dynamics: the first pool is characterized by a diffusive dynamic behavior, and the second pool is characterized by a more directed and targeted movement. The developed software will allow monitoring selectively these two classes unraveling if they have distinct functional roles and/or interactions with other structures and organelles in beta cells upon glucose stimulation and whether, they could selectively be impaired in pathological conditions. In this, the insulin resistance typical of T2DM could also have an impact on the beta cells and the behavior of the granules within them, thus altering their physiological dynamics and changing the balance between the two identified families. More specifically, the accumulation of glucose in the blood, could be anticipated by the presence of a constant greater number of granules in the pancreatic cells, ready to exocytosis to induce the body to metabolize more sugar. This could help the therapist to focus on more specific diets to give to the patient in order to prevent the hyperglycemia state.

7 Conclusions

With respect to the first issue, we have investigated if the physical state of PMC and RBC membranes can be used for positive/negative classifications in T1DM disease. Indeed, profiling of individuals' biomolecules of blood cells has provided new strategies for disease monitoring and treatment. The experiments carried out on a dataset of 800 blood cell images belonging to 18 subjects have shown that the proposed multi-experts system leveraging on multi-spectral information and using texture features correctly classifies all the available samples.

Next, we have presented a DSS leveraging on dual channel metabolic functional imaging data computed from the physical state of RBC membranes, a novel biomarker that allows to monitor T1DM progression. It uses features based on first- and second-order statistical measures as well as on rotation invariant co-occurrence local binary patterns. The dual channel approach also introduces a certain degree of redundancy that lowers the effect of image misclassifications. The recognition approach permits to achieve promising results in a field of medicine characterized by large impact on the population. Future efforts are directed towards enlarging the patient dataset, assessing the possibility of the method to describe long-term diabetes evolution, and lowering the acquisition cost of the proposed approach becoming independent from two-photon microscopy. To this last aim, future research will be directed towards the use of a cheaper microscope with UV source. Indeed, the raising quality of this light source has the potential to be useful in this application in a near future.

We described an approach to detect insulin granules' dynamic inside the beta cells by employing confocal spinning disk microscopy. It works with 4D images and it mines the data via a machine learning approach. To tackle the huge number of collected granules we tested several free softwares for automatic tracking, and find out the *TrackMate* software best fits our goal. The statistical validation was performed with a homemade manual tracking application and the work of two operators. Next we extracted several cinematic features to quantify the dynamic of each track and we clustered their behavior, thus identifying two classes of tracks with distinct motion typologies. We are aware that the main limitation of this method resides in the number of analyzed tracks that leads

complex statistical analysis. Moreover, further development is planned to improve the spatial resolution of the images and producing a finer granules' tracking as well. The resulting findings of this preliminary study show the presence of two sets of granules, marked by distinct dynamics: the first pool is characterized by a diffusive dynamic behavior, and the second pool is characterized by a more directed and targeted movement. The developed software will allow monitoring selectively these two classes unraveling if they have distinct functional roles and/or interactions with other structures and organelles in beta cells upon glucose stimulation and whether, they could selectively be impaired in pathological conditions.

Bibliography

- [1] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016.
- [2] D. Faggella. (2018) Machine learning healthcare applications - 2018 and beyond. [Online]. Available: <https://emerj.com/ai-sector-overviews/machine-learning-healthcare-applications/>
- [3] S. Rizzo, F. Botta, S. Raimondi, D. Origgi, C. Fanciullo, A. G. Morganti, and M. Bellomi, "Radiomics: the facts and the challenges of image analysis," *European radiology experimental*, vol. 2, no. 1, p. 36, 2018.
- [4] World Health Organization. (2016) Diabetes. [Online]. Available: <http://www.who.int/mediacentre/factsheets/fs312/en/>
- [5] A. Amutha, R. M. Anjana, U. Venkatesan, H. Ranjani, R. Unnikrishnan, K. V. Narayan, V. Mohan, and M. K. Ali, "Incidence of complications in young-onset diabetes: comparing type 2 with type 1 (the young diab study)," *Diabetes research and clinical practice*, vol. 123, pp. 1–8, 2017.
- [6] D. C. Klonoff, J. L. Parkes, B. P. Kovatchev, D. Kerr, W. C. Bevier, R. L. Brazg, M. Christiansen, T. S. Bailey, J. H. Nichols, and M. A. Kohn, "Investigation of the accuracy of 18 marketed blood glucose monitors," *Diabetes Care*, vol. 41, no. 8, pp. 1681–1688, 2018.
- [7] S. E. Jackson and J. D. Chester, "Personalised cancer medicine," *International journal of cancer*, vol. 137, no. 2, pp. 262–266, 2015.
- [8] W. G. Jerome and R. L. Price, *Basic confocal microscopy*. Springer, 2018.
- [9] S. Erich, "Biological membranes architecture and function," *Handbook of Biological Physics*, (ed. R. Lipowsky and E. Sackmann, vol. 1, Elsevier, 1995.

- [10] K. Yazdanbakhsh, C. Lomas-Francis, and M. E. Reid, "Blood groups and diseases associated with inherited abnormalities of the red blood cell membrane," *Transfusion medicine reviews*, vol. 14, no. 4, pp. 364–374, 2000.
- [11] J. El Hokayem, H. N. Cukier, and D. M. Dykxhoorn, "Blood derived induced pluripotent stem cells (ipscs): benefits, challenges and the road ahead," *Journal of Alzheimer's disease & Parkinsonism*, vol. 6, no. 5, 2016.
- [12] G. M. Hon, M. S. Hassan, S. J. van Rensburg, S. Abel, R. T. Erasmus, and T. Matsha, "Peripheral blood mononuclear cell membrane fluidity and disease outcome in patients with multiple sclerosis," *Indian Journal of Hematology and Blood Transfusion*, vol. 28, no. 1, pp. 1–6, 2012.
- [13] G. Lenaz, "Lipid fluidity and membrane protein dynamics," *Biosci. Rep.*, vol. 7, no. 11, pp. 823–837, Nov 1987.
- [14] R. Scherz-Shouval and Z. Elazar, "ROS, mitochondria and the regulation of autophagy," *Trends Cell Biol.*, vol. 17, no. 9, pp. 422–427, Sep 2007.
- [15] G. Maulucci, G. Pani *et al.*, "Investigation of the spatial distribution of glutathione redox-balance in live cells by using Fluorescence Ratio Imaging Microscopy," *Biosens Bioelectron*, vol. 25, no. 4, pp. 682–687, Dec 2009.
- [16] —, "Compartmentalization of the redox environment in PC-12 neuronal cells," *Eur. Biophys. J.*, vol. 39, no. 6, pp. 993–999, May 2010.
- [17] F. R. Maxfield and I. Tabas, "Role of cholesterol and lipid organization in disease," *Nature*, vol. 438, no. 7068, pp. 612–621, Dec 2005.
- [18] G. Maulucci, D. Troiani, S. L. M. Eramo, F. Paciello, M. V. Podda, G. Paludetti, M. Papi, A. Maiorana, V. Palmieri, M. De Spirito *et al.*, "Time evolution of noise induced oxidation in outer hair cells: role of nad (p) h and plasma membrane fluidity," *Biochimica et Biophysica Acta (BBA)-General Subjects*, vol. 1840, no. 7, pp. 2192–2202, 2014.
- [19] C. Angelucci, G. Maulucci *et al.*, "Stearoyl-CoA desaturase 1 and paracrine diffusible signals have a major role in the promotion of breast cancer cell migration induced by cancer-associated fibroblasts," *Br. J. Cancer*, vol. 112, no. 10, pp. 1675–1686, May 2015.

- [20] G. M. Hon, M. S. Hassan *et al.*, “Red blood cell membrane fluidity in the etiology of multiple sclerosis,” *J. Membr. Biol.*, vol. 232, no. 1-3, pp. 25–34, Dec 2009.
- [21] —, “Peripheral blood mononuclear cell membrane fluidity and disease outcome in patients with multiple sclerosis,” *Indian J Hematol Blood Transfus*, vol. 28, no. 1, pp. 1–6, Mar 2012.
- [22] E. Zanni, G. Maulucci *et al.*, “ER stress induced by the OCH1 mutation triggers changes in lipid homeostasis in *Kluyveromyces lactis*,” *Res. Microbiol.*, vol. 166, no. 2, pp. 84–92, 2015.
- [23] G. Maulucci, M. Chiarpotto *et al.*, “Quantitative analysis of autophagic flux by confocal pH-imaging of autophagic intermediates,” *Autophagy*, vol. 11, no. 10, pp. 1905–1916, Oct 2015.
- [24] L. A. Bagatolli and E. Gratton, “Two photon fluorescence microscopy of coexisting lipid domains in giant unilamellar vesicles of binary phospholipid mixtures,” *Biophys. J.*, vol. 78, pp. 290–305, Jan 2000.
- [25] G. Balogh, G. Maulucci *et al.*, “Heat stress causes spatially-distinct membrane re-modelling in K562 leukemia cells,” *PLoS ONE*, vol. 6, no. 6, p. e21182, 2011.
- [26] K. Donsa, S. Spat *et al.*, “Towards personalization of diabetes therapy using computerized decision support and machine learning: some open problems and challenges,” in *Smart Health*. Springer, 2015, pp. 237–260.
- [27] E. Boland, T. Monsod *et al.*, “Limitations of conventional methods of self-monitoring of blood glucose,” *DIABETES CARE*, vol. 24, no. 11, pp. 1858–1862, November 2001.
- [28] L. I. Kuncheva, “Switching between selection and fusion in combining classifiers: an experiment,” *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 32, no. 2, pp. 146–156, April 2002.
- [29] T. G. Dietterich and G. Bakiri, “Solving multiclass learning problems via error-correcting output codes,” *Journal of Artificial Intelligence Research*, vol. 2, pp. 263–286, 1995.
- [30] G. Iannello, G. Percannella *et al.*, “On the use of classification reliability for improving performance of the one-per-class decomposition method,” *Data & Knowledge Engineering*, vol. 68, no. 12, pp. 1398–1410, 2009.

- [31] Y. Huang, Z. Wu *et al.*, “Feature coding in image classification: A comprehensive study,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 3, pp. 493–506, 2014.
- [32] P. Soda, “Early experiences in the staining pattern classification of HEp-2 slides,” in *Computer-Based Medical Systems, 2007. Twentieth IEEE International Symposium on*. IEEE, 2007, pp. 219–224.
- [33] P. Soda, L. Onofri, and G. Iannello, “A decision support system for crithidia luciliae image classification,” *Artificial intelligence in medicine*, vol. 51, no. 1, pp. 67–74, 2011.
- [34] S. Di Cataldo, A. Bottino *et al.*, “Subclass discriminant analysis of morphological and textural features for HEp-2 staining pattern classification,” *Pattern Recognition*, vol. 47, no. 7, pp. 2389–2399, 2014.
- [35] B. Julesz, “Experiments in a visual perception of texture,” *Scientific American*, vol. 232, pp. 34–43, April 1975.
- [36] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley-Interscience, 2000.
- [37] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural features for image classification,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.
- [38] A. P. Dhawan, Y. Chitre *et al.*, “Analysis of mammographic microcalcifications using gray-level image structure features,” *IEEE Transactions on Medical Imaging*, vol. 15, no. 3, pp. 246–259, June 1996.
- [39] Ojala, T. and Pietikäinen, M. and Harwood, D., “A comparative study of texture measures with classification based on feature distribution,” *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.
- [40] R. Nosaka and K. Fukui, “HEp-2 cell classification using rotation invariant co-occurrence among local binary patterns,” *Pattern Recognition*, vol. 47, no. 7, pp. 2428–2436, 2013.
- [41] R. Nosaka, C. H. Suryanto, and K. Fukui, “Rotation invariant co-occurrence among adjacent LBPs,” in *Computer Vision-ACCV 2012 Workshops*. Springer, 2013, pp. 15–25.

- [42] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, January 2000.
- [43] T. Van Gestel, J. A. Suykens *et al.*, "Benchmarking least squares support vector machine classifiers," *Machine Learning*, vol. 54, no. 1, pp. 5–32, 2004.
- [44] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.
- [45] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.
- [46] L. Chang-Shing and W. Mei-Hui, "A fuzzy expert system for diabetes decision support application," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 41, no. 1, pp. 139–153, 2011.
- [47] K. Polat and S. Gunes, "An expert system approach based on principal component analysis and adaptive neuro-fuzzy inference system to diagnosis of diabetes disease," *Digital Signal Processing*, vol. 17, no. 4, pp. 702–710, 2007.
- [48] V. Veena Vijayan and C. Anjali, "Prediction and Diagnosis of Diabetes Mellitus - A Machine Learning Approach," in *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*. IEEE, 2015, pp. 197–202.
- [49] G. Maulucci, G. Bačić, L. Bridal, H. H. Schmidt, B. Tavitian, T. Viel, H. Utsumi, A. S. Yalçın, and M. De Spirito, "Imaging reactive oxygen species-induced modifications in living systems," *Antioxidants & redox signaling*, vol. 24, no. 16, pp. 939–958, 2016.
- [50] G. Maulucci, G. Pani, S. Fusco, M. Papi, G. Arcovito, T. Galeotti, M. Fraziano, and M. De Spirito, "Compartmentalization of the redox environment in pc-12 neuronal cells," *European Biophysics Journal*, vol. 39, no. 6, pp. 993–999, 2010.
- [51] G. Maulucci, G. Pani, V. Labate, M. Mele, E. Panieri, M. Papi, G. Arcovito, T. Galeotti, and M. De Spirito, "Investigation of the spatial distribution of glutathione redox-balance in live cells by using fluorescence ratio imaging microscopy," *Biosensors and Bioelectronics*, vol. 25, no. 4, pp. 682–687, 2009.

- [52] G. Maulucci, O. Cohen, B. Daniel, A. Sansone, P. Petropoulou, S. Filou, A. Spyridonidis, G. Pani, M. De Spirito, C. Chatgialoglu *et al.*, “Fatty acid-related modulations of membrane fluidity in cells: detection and implications,” *Free radical research*, vol. 50, no. sup1, pp. S40–S50, 2016.
- [53] P. F. Pilch, P. A. Thompson, and M. P. Czech, “Coordinate modulation of D-glucose transport activity and bilayer fluidity in plasma membranes derived from control and insulin-treated adipocytes,” *Proceedings of the National Academy of Sciences*, vol. 77, no. 2, pp. 915–918, 1980.
- [54] T. Parasassi, G. De Stasio, A. d’Ubaldo, and E. Gratton, “Phase fluctuation in phospholipid membranes revealed by Laurdan fluorescence.” *Biophysical journal*, vol. 57, no. 6, p. 1179, 1990.
- [55] E. Cordelli, G. Pani, D. Pitocco, G. Maulucci, and P. Soda, “Early experiences in using blood cells biomembranes as markers for diabetes diagnosis,” in *2016 IEEE 29th International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE, 2016, pp. 197–202.
- [56] B. L. Karger, “Hplc: Early and recent perspectives,” *J. Chem. Educ*, vol. 74, no. 1, p. 45, 1997.
- [57] G. Iannello, G. Percannella, P. Soda, and M. Vento, “Mitotic cells recognition in hep-2 images,” *Pattern Recognition Letters*, vol. 45, pp. 136–144, 2014.
- [58] T.-F. Wu, C.-J. Lin, and R. C. Weng, “Probability estimates for multi-class classification by pairwise coupling,” *Journal of Machine Learning Research*, vol. 5, no. Aug, pp. 975–1005, 2004.
- [59] D. H. Wolpert and W. G. Macready, “No Free Lunch Theorems for Optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
- [60] A. Arcuri and G. Fraser, “Parameter tuning or default values? an empirical investigation in search-based software engineering,” *Empirical Software Engineering*, vol. 18, no. 3, pp. 594–623, 2013.
- [61] K. Miedema, “Standardization of hba1c and optimal range of monitoring,” *Scandinavian Journal of Clinical and Laboratory Investigation*, vol. 65, no. sup240, pp. 61–72, 2005.

- [62] A. D. Association *et al.*, “6. glycemic targets,” *Diabetes care*, vol. 40, no. Supplement 1, pp. S48–S56, 2017.
- [63] E. Zanni, G. Maulucci, D. Pomata, F. Buiarelli, E. K. Krasnowska, T. Parasassi, M. De Spirito, H. J. Heipieper, and D. Uccelletti, “Er stress induced by the och1 mutation triggers changes in lipid homeostasis in *kluveromyces lactis*,” *Research in microbiology*, vol. 166, no. 2, pp. 84–92, 2015.
- [64] L. P. Cordella, P. Foggia, C. Sansone, F. Tortorella, and M. Vento, “Reliability parameters to improve combination strategies in multi-expert systems,” *Pattern Analysis & Applications*, vol. 2, no. 3, pp. 205–214, August 1999.
- [65] O. L. Mangasarian *et al.*, “Generalized support vector machines,” *Advances in Neural Information Processing Systems*, pp. 135–146, 1999.
- [66] J. Pawley, *Handbook of Biological Confocal Microscopy*. Springer Science & Business Media, 2010.
- [67] G. Maulucci, B. Daniel, O. Cohen, Y. Avrahami, and S. Sasson, “Hormetic and regulatory effects of lipid peroxidation mediators in pancreatic beta cells,” *Molecular aspects of medicine*, vol. 49, pp. 49–77, 2016.
- [68] G. Cohen, Y. Riahi, O. Shamni, M. Guichardant, C. Chatgililoglu, C. Ferreri, N. Kaiser, and S. Sasson, “Role of lipid peroxidation and ppar- δ in amplifying glucose-stimulated insulin secretion,” *Diabetes*, p. DB_110347, 2011.
- [69] K. Tanabe, K. Amo-Shiinoki, M. Hatanaka, and Y. Tanizawa, “Interorgan crosstalk contributing to β -cell dysfunction,” *Journal of diabetes research*, vol. 2017, 2017.
- [70] R. A. DeFronzo, E. Ferrannini, L. Groop, R. R. Henry, W. H. Herman, J. J. Holst, F. B. Hu, C. R. Kahn, I. Raz, G. I. Shulman *et al.*, “Type 2 diabetes mellitus,” *Nature reviews Disease primers*, vol. 1, p. 15019, 2015.
- [71] G. Maulucci, M. Chiarpotto, M. Papi, D. Samengo, G. Pani, and M. De Spirito, “Quantitative analysis of autophagic flux by confocal ph-imaging of autophagic intermediates,” *Autophagy*, vol. 11, no. 10, pp. 1905–1916, 2015.
- [72] G. Maulucci, F. Di Giacinto, C. De Angelis, O. Cohen, B. Daniel, C. Ferreri, M. De Spirito, and S. Sasson, “Real time quantitative analysis of lipid storage and lipolysis pathways by confocal spectral imaging of intracellular micropolarity,”

- Biochimica et Biophysica Acta (BBA)-Molecular and Cell Biology of Lipids*, vol. 1863, no. 7, pp. 783–793, 2018.
- [73] A. S. Rambold, S. Cohen, and J. Lippincott-Schwartz, “Fatty acid trafficking in starved cells: regulation by lipid droplet lipolysis, autophagy, and mitochondrial fusion dynamics,” *Developmental cell*, vol. 32, no. 6, pp. 678–692, 2015.
- [74] M. Hölttä-Vuori, R.-L. Uronen, J. Repakova, E. Salonen, I. Vattulainen, P. Panula, Z. Li, R. Bittman, and E. Ikonen, “Bodipy-cholesterol: a new tool to visualize sterol trafficking in living cells and organisms,” *Traffic*, vol. 9, no. 11, pp. 1839–1849, 2008.
- [75] A. Ivanova, Y. Kalaidzidis, R. Dirkx, M. Sarov, M. Gerlach, B. Schroth-Diez, A. Müller, Y. Liu, C. Andree, B. Mulligan *et al.*, “Age-dependent labeling and imaging of insulin secretory granules,” *Diabetes*, p. DB.121819, 2013.
- [76] M. Ohara-Imaizumi, Y. Nakamichi, T. Tanaka, H. Ishida, and S. Nagamatsu, “Imaging exocytosis of single insulin secretory granules with evanescent wave microscopy distinct behavior of granule motion in biphasic insulin release,” *Journal of Biological Chemistry*, vol. 277, no. 6, pp. 3805–3808, 2002.
- [77] S. A. Tabei, S. Burov, H. Y. Kim, A. Kuznetsov, T. Huynh, J. Jureller, L. H. Philipson, A. R. Dinner, and N. F. Scherer, “Intracellular transport of insulin granules is a subordinated random walk,” *Proceedings of the National Academy of Sciences*, p. 201221962, 2013.
- [78] A. Müller, H. Mziaut, M. Neukam, K.-P. Knoch, and M. Solimena, “A 4d view on insulin secretory granule turnover in the β -cell,” *Diabetes, Obesity and Metabolism*, vol. 19, pp. 107–114, 2017.
- [79] X. Zhu, R. Hu, M. Brissova, R. W. Stein, A. C. Powers, G. Gu, and I. Kaverina, “Microtubules negatively regulate insulin secretion in pancreatic β cells,” *Developmental cell*, vol. 34, no. 6, pp. 656–668, 2015.
- [80] A. T. Heaslip, S. R. Nelson, A. T. Lombardo, S. B. Previs, J. Armstrong, and D. M. Warshaw, “Cytoskeletal dependence of insulin granule movement dynamics in ins-1 beta-cells in response to glucose,” *PloS one*, vol. 9, no. 10, p. e109082, 2014.

- [81] J. A. Steyer and W. Almers, "Tracking single secretory granules in live chromaffin cells by evanescent-field fluorescence microscopy," *Biophysical journal*, vol. 76, no. 4, pp. 2262–2271, 1999.
- [82] S. Stehbens, H. Pemble, L. Murrow, and T. Wittmann, "Imaging intracellular protein dynamics by spinning disk confocal microscopy," in *Methods in enzymology*. Elsevier, 2012, vol. 504, pp. 293–313.
- [83] M. Weber, M. Mickoleit, and J. Huisken, "Light sheet microscopy," in *Methods in cell biology*. Elsevier, 2014, vol. 123, pp. 193–215.
- [84] G. M. Pigeau, J. Kolic, B. J. Ball, M. B. Hoppa, Y. W. Wang, T. Rückle, M. Woo, J. E. M. Fox, and P. E. MacDonald, "Insulin granule recruitment and exocytosis is dependent on p110 γ in insulinoma and human β -cells," *Diabetes*, 2009.
- [85] J.-Y. Tinevez, N. Perry, J. Schindelin, G. M. Hoopes, G. D. Reynolds, E. Laplantine, S. Y. Bednarek, S. L. Shorte, and K. W. Eliceiri, "TrackMate: An open and extensible platform for single-particle tracking," *Methods*, vol. 115, pp. 80–90, 2017.
- [86] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.
- [87] W. Davidson and M. Abramowitz, "Molecular expressions microscopy primer: Digital image processing-difference of gaussians edge enhancement algorithm," *Olympus America Inc., and Florida State University*, 2006.
- [88] K. Jaqaman, D. Loerke, M. Mettlen, H. Kuwata, S. Grinstein, S. L. Schmid, and G. Danuser, "Robust single-particle tracking in live-cell time-lapse sequences," *Nature methods*, vol. 5, no. 8, p. 695, 2008.
- [89] ImageJ. (2016) Trackmate algorithms. [Online]. Available: https://imagej.net/TrackMate_Algorithms
- [90] M. M. Frigault, J. Lacoste, J. L. Swift, and C. M. Brown, "Live-cell microscopy—tips and tools," *J Cell Sci*, vol. 122, no. 6, pp. 753–767, 2009.
- [91] N. Streibl, "Three-dimensional imaging by a microscope," *JOSA A*, vol. 2, no. 2, pp. 121–127, 1985.

- [92] Z. Aspridou, A. D. Balomenos, P. Tsakanikas, E. S. Manolakos, and K. Koutsoumanis, "Monitoring individual cell death using time-lapse microscopy: Application to stochastic modeling of microbial inactivation," 2016.
- [93] B. Neumann, T. Walter, J.-K. Hériché, J. Bulkescher, H. Erfle, C. Conrad, P. Rogers, I. Poser, M. Held, U. Liebel *et al.*, "Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes," *Nature*, vol. 464, no. 7289, p. 721, 2010.
- [94] T. Moschakis, B. S. Murray, and E. Dickinson, "Particle tracking using confocal microscopy to probe the microrheology in a phase-separating emulsion containing nonadsorbing polysaccharide," *Langmuir*, vol. 22, no. 10, pp. 4710–4719, 2006.
- [95] R. Bise, K. Li, S. Eom, and T. Kanade, "Reliably tracking partially overlapping neural stem cells in dic microscopy image sequences," in *MICCAI Workshop on OPTIMHisE*, vol. 5, 2009, pp. 67–77.
- [96] N. Kramer, A. Walzl, C. Unger, M. Rosner, G. Krupitza, M. Hengstschläger, and H. Dolznig, "In vitro cell migration and invasion assays," *Mutation Research/Reviews in Mutation Research*, vol. 752, no. 1, pp. 10–24, 2013.
- [97] P. Vallotton and S. Olivier. (2002) Diatrack. [Online]. Available: <http://www.diatrack.org/index.html>
- [98] C. Sommer, C. Strähle, U. Köthe, and F. A. Hamprecht. (2010) ilastik. [Online]. Available: <http://ilastik.org/index.html>
- [99] P. Hanchuan and H. Howard. (2006) Vaa3d. [Online]. Available: <http://home.penglab.com/proj/vaa3d/home/index.html>
- [100] P. Masuzzo, M. Van Troys, C. Ampe, and L. Martens, "Taking aim at moving targets in computational cell migration," *Trends in cell biology*, vol. 26, no. 2, pp. 88–110, 2016.
- [101] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal processing magazine*, vol. 13, no. 6, pp. 47–60, 1996.
- [102] G. Devis, E. Van Obberghen, G. Somers, F. Malaisse-Lagae, L. Orci, and W. Malaisse, "Dynamics of insulin release and microtubular-microfilamentous system," *Diabetologia*, vol. 10, no. 1, pp. 53–59, 1974.

- [103] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, no. 2, pp. 159–179, 1985.
- [104] Y. Liu, Z. Li, H. Xiong, X. Gao, J. Wu, and S. Wu, "Understanding and enhancement of internal clustering validation measures," *IEEE transactions on cybernetics*, vol. 43, no. 3, pp. 982–994, 2013.
- [105] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.
- [106] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.
- [107] M. Kozak, "A dendrite method for cluster analysis by caliński and harabasz: A classical work that is far too often incorrectly cited," *Communications in Statistics-Theory and Methods*, vol. 41, no. 12, pp. 2279–2280, 2012.
- [108] M. Halkidi and M. Vazirgiannis, "Clustering validity assessment: Finding the optimal partitioning of a data set," in *Proceedings 2001 IEEE International Conference on Data Mining*. IEEE, 2001, pp. 187–194.

Appendix A

This appendix shows below a schematic example of the xml format model used in Visual4DTracer to read and store the traces coordinates in 4D.

Note that depending on objects' dynamic the occurrences of spots may not match with the duration of experiment, therefore FRAME_ID could not be stored as progressive consecutive indices. This can happen, for instance, when the delay between two consecutive frames is larger than one second.

Notations:

- NUM_TRACES: Number of traces contained in the file.
- DATE: File's creation or last modification date coded in "day, day_number month year hh:mm:ss".
- FRAME_ID, X_COORD, Y_COORD, Z_COORD frame id, X, Y and Z coordinates, respectively, written using a floating point type.
- NUM_SPOTS: Number of times each blob was detected.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Tracks nTracks="NUMTRACES" spaceUnits="pixel" frameInterval="1.0"
timeUnits="frame" generationDate="DATE" from="Visual4DTracer">
  <particle nSpots="NUMSPOTS">
    <detection t="FRAME_ID" x="X_COORD" y="Y_COORD" z="Z_COORD" />
    <detection t="FRAME_ID" x="X_COORD" y="Y_COORD" z="Z_COORD" />
    <detection t="FRAME_ID" x="X_COORD" y="Y_COORD" z="Z_COORD" />
    <detection t="FRAME_ID" x="X_COORD" y="Y_COORD" z="Z_COORD" />
    ...
    <detection t="FRAME_ID" x="X_COORD" y="Y_COORD" z="Z_COORD" />
  </particle>
```

Tesi di dottorato in Bioingegneria e bioscienze, di Ermanno Cordelli,
discussa presso l'Università Campus Bio-Medico di Roma in data 13/12/2019.
La disseminazione e la riproduzione di questo documento sono consentite per scopi di didattica e ricerca,
a condizione che ne venga citata la fonte.

```
<particle nSpots="NUM.SPOTS">  
  ...  
</particle>  
  ...  
</Tracks>
```

Appendix B

Radiomics

(1) **Reference** Ramella, S., Fiore, M., Greco, C., Cordelli, E., Sicilia, R., Merone, M., *et al.* (2018). A radiomic approach for adaptive radiotherapy in non-small cell lung cancer patients. PloS one, 13(11), e0207455.

Abstract The primary goal of precision medicine is to minimize side effects and optimize efficacy of treatments. Recent advances in medical imaging technology allow the use of more advanced image analysis methods beyond simple measurements of tumor size or radio- tracer uptake metrics. The extraction of quantitative features from medical images to characterize tumor pathology or heterogeneity is an interesting process to investigate, in order to provide information that may be useful to guide the therapies and predict survival. This paper discusses the rationale supporting the concept of radiomics and the feasibility of its application to Non-Small Cell Lung Cancer in the field of radiation oncology research (Figure 7.1 shows a region of interest example in a 3D image). We studied 91 stage III patients treated with concurrent chemoradiation and adaptive approach in case of tumor reduction during treatment. We considered 12 statistics features and 230 textural features extracted from the CT images. In our study, we used an ensemble learning method to classify patients' data into either the adaptive or non-adaptive group during chemoradiation on the basis of the starting CT simulation. Our data supports the hypothesis that a specific signature can be identified (AUC 0.82, as reported in Figure 7.2). In our experience, a radiomic signature mixing semantic and image-based features has shown promising results for personalized adaptive radiotherapy in non-small cell lung cancer.

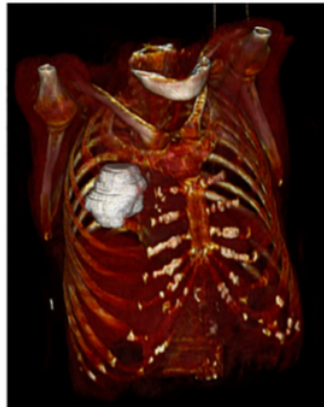


Figure 7.1: Example of region of interest in a 3D image.

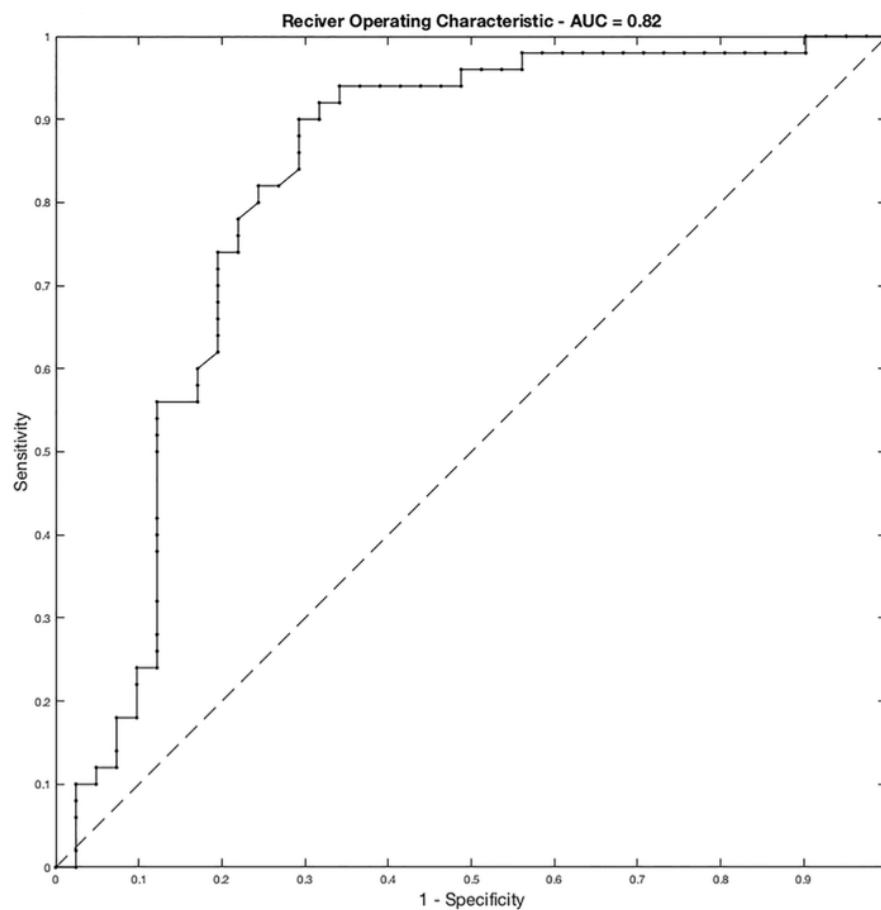


Figure 7.2: ROC curve of the proposed system.

(2) **Reference** D'Amico, N. C., Merone, M., Sicilia, R., Cordelli, E., D'Antoni, F., Zanetti, I. B., *et al.* (2019). Tackling imbalance radiomics in acoustic neuroma. *International Journal of Data Mining and Bioinformatics*, 22(4), 365-388.

Abstract Acoustic neuroma is a primary intracranial tumour of the myelinating cells of the 8th cranial nerve. Although it is a slow growing benign tumour, symptoms in the advanced phase can be serious. Hence, controlling tumour growth is essential and stereotactic radiosurgery, which can be performed with the CyberKnife robotic device, has proven effective for managing this disease. However, this approach may have side effects and a follow-up is necessary to assess its efficacy. To optimise the administration of this treatment, in this work we present a machine learning-based radiomics approach that first computes quantitative biomarkers from MR images routinely collected before the CyberKnife treatment and then predicts the treatment response. To tackle the challenge of class imbalance observed in the available dataset we present a cascade of cost-sensitive decision trees. The presented pipeline is depicted in Figure 7.3. We also experimentally compare the proposed approach with several approaches suited for learning under class skew. The results achieved, with a global accuracy of 0.92, demonstrate that radiomics has a great potential in predicting patients response to radiosurgery prior to the treatment that, in turns, can reflect into great advantages in therapy planning, sparing radiation toxicity and surgery when unnecessary.

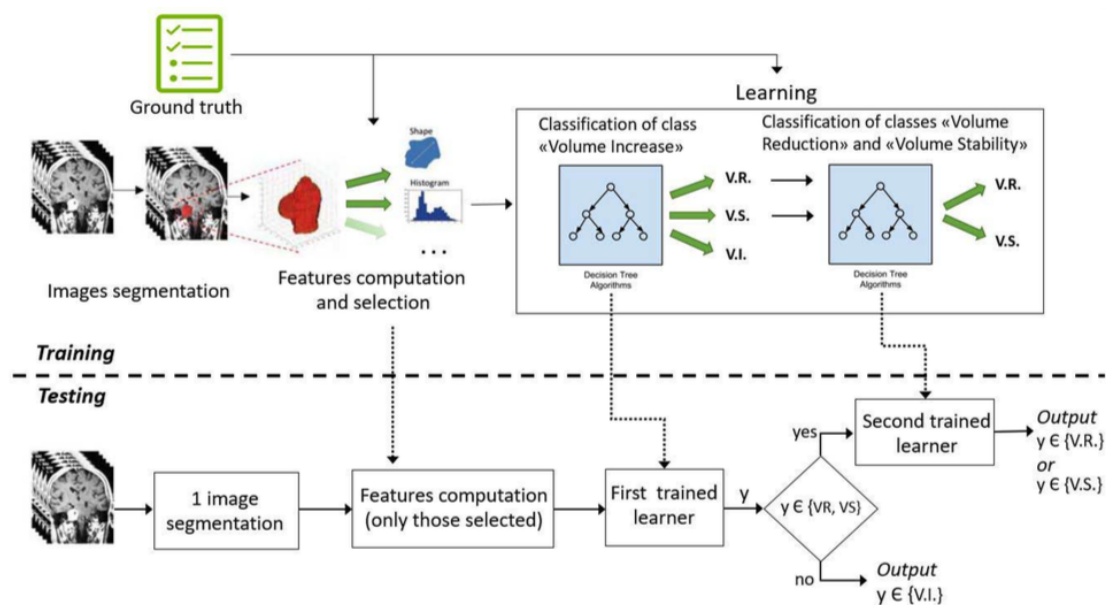


Figure 7.3: Schematic representation of the proposed machine learning approach.

(3) **Reference** Sicilia, R., Cordelli, E., Merone, M., Luperto, E., Papalia, R., Iannello, G., & Soda, P. (2019, June). Early radiomic experiences in classifying prostate cancer aggressiveness using 3D local binary patterns. In 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS) (pp. 355-360). IEEE.

Abstract Prostate cancer is the most common form of cancer in Western countries and there is the need to develop clinical decision support systems able to support physicians in the diagnosis of clinical relevant prostate cancer and avoid useless invasive prostate biopsies. In this respect, this paper introduces a radiomic approach that classifies the prostate cancer aggressiveness by combining Three Orthogonal Planes-Local Binary Pattern (TOP-LBP) with other texture measures. Furthermore, to combat the skewed nature of class priors, our proposal employs a data augmentation technique (Figure 7.4). The results achieved on 99 samples are up-and-coming, they favorably compare against conventional PI- RADS-based approach, and they show also the benefit given by the introduction of TOP-LBP in the radiomic signature, reaching an accuracy of 0.82.

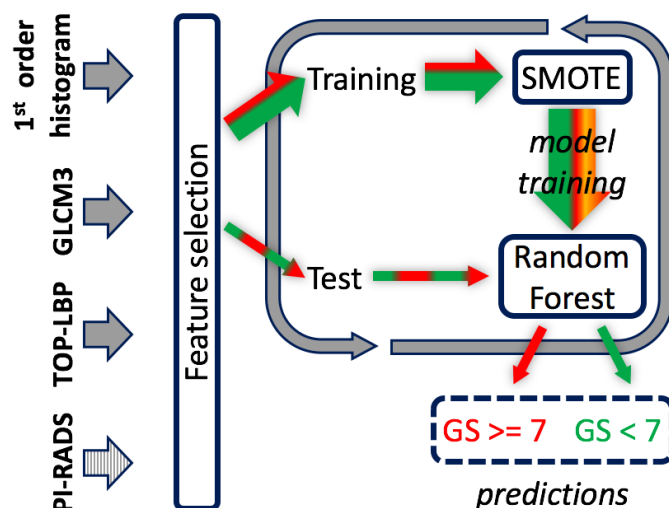


Figure 7.4: Schematic representation of the proposed approach.

Image processing

Reference Soda, P., Acciai, L., Cordelli, E., Costantini, I., Sacconi, L., Pavone, F. S., *et al.* (2015, August). Computer-based automatic identification of neurons in gigavoxel-sized 3d human brain images. In 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 7724-7727). IEEE.

Abstract Achieving a comprehensive knowledge of the human brain cytoarchitecture is a fundamental step to understand how the nervous system works, i.e., one of the greatest challenge of 21st century science. The recent development of biological tissue labeling and automated microscopic imaging systems has permitted to acquire images at the micro-resolution, which produce a huge quantity of data that cannot be manually analyzed. In case of mammals brain, automatic methods to extract objective information at the microscale have been applied until now to mice, macaque and cat 3D volume images. Here, referring to Figure 7.5, we report a method to automatically localize neurons in a sample of human brain removed during a surgical procedure for the treatments of drug resistant epilepsy in a child with hemimegalencephaly, whose neurons and neurites were fluorescence labelled and finally imaged using the two-photon fluorescence microscope. We exploit spectral information of the image volume improving the image quality through ad hoc pre- and postprocessing operations. The method provides the map of both parvalbuminergic neurons and all other cells nuclei and we obtained a value of f-score equal to 0.83 and 0.90, respectively.

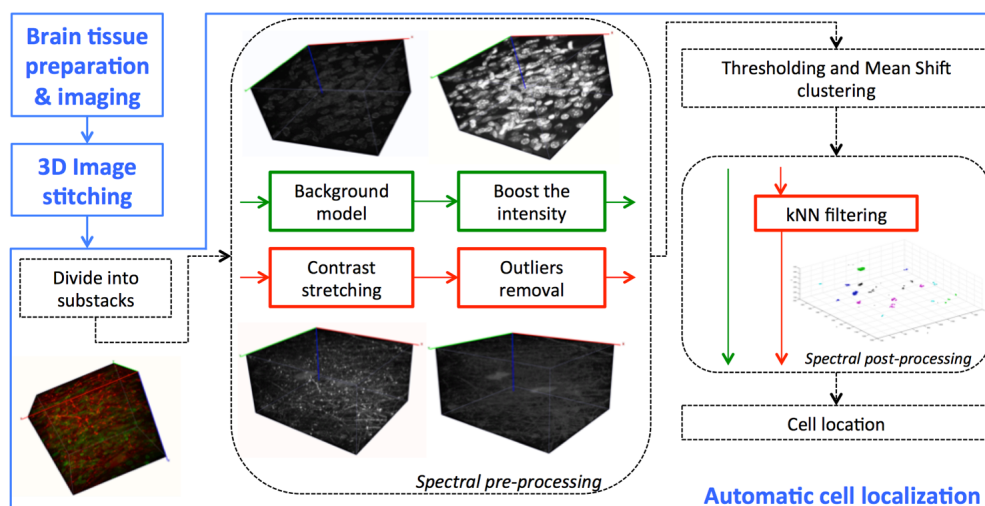


Figure 7.5: Schematic representation of the pipeline employed to localize the neurons.

Text analysis

Reference Sicilia, R., Merone, M., Valenti, R., Cordelli, E., D'Antoni, F., De Ruvo, V., *et al.* (2018, December). Cross-topic Rumour Detection in the Health Domain. In 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 2056-2063). IEEE.

Abstract Nowadays information diffusion has become more and more immediate and fast thanks to social media and its services. However, lack of controls and moderation in resources as social microblogs often leads to spread unverified information, such as rumours, which can become a threat to the society. To improve life quality and good information diffusion, various automatic systems have been studied for rumour detection in microblogs at level of aggregation of posts, whereas a few effort has been tried to the most challenging scenario where the rumour has to be recognized at level of each single post. In this work, we direct our efforts towards individual post rumour detection: we investigate how features describing influence potential, personal interest and network characteristic perform on two different datasets of posts collected from Twitter using two different health-related keywords (the pipeline of the process is shown in Figure 7.6). As a further contribution, we study what happens in cross-topic tests, i.e. when the rumour detection system is trained with posts with an hashtag and tested on samples with a different one.

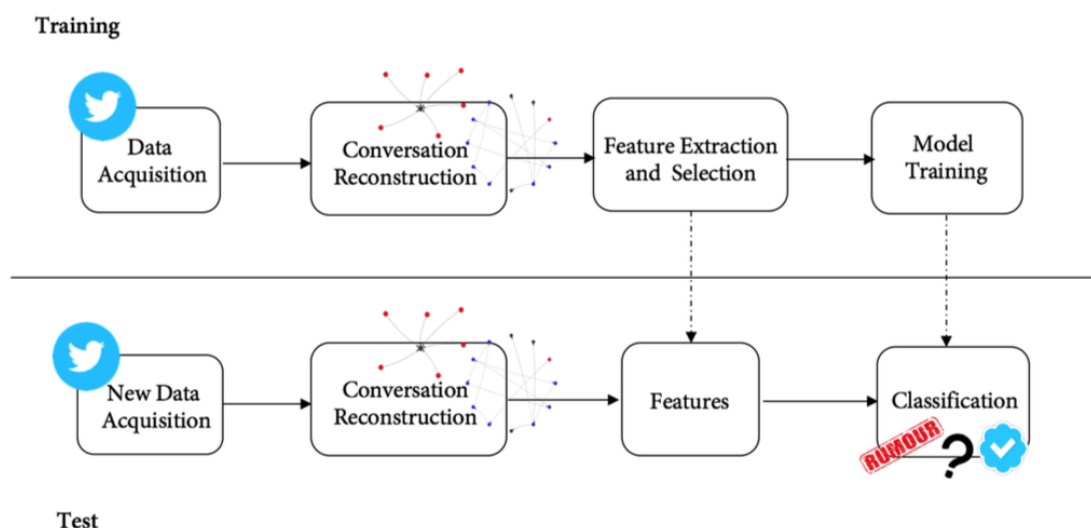


Figure 7.6: Pipeline of the rumour detection process.

Internet of Things

Reference Cordelli, E., Pennazza, G., Sabatini, M., Santonico, M., & Vollerò, L. (2018). An Open-Source Smart Sensor Architecture for Edge Computing in IoT Applications. In Multidisciplinary Digital Publishing Institute Proceedings (Vol. 2, No. 13, p. 955).

Abstract Smart sensors are sensing devices that include computational and communication functionalities. In this work we present a reference model aimed at simplifying the implementation of smart sensors and their integration in IoT applications. The proposed model is micro-controller agnostic and it is viable in different scenarios ranging from the management of a single analog sensor to the orchestration of a heterogeneous array of sensors. The model is open-source and the implementation is available online as reference for the development of custom smart sensors. The evaluation of our framework shows that it can be implemented with limited overhead. In Figure 7.7 is presented the smart sensor's architecture.

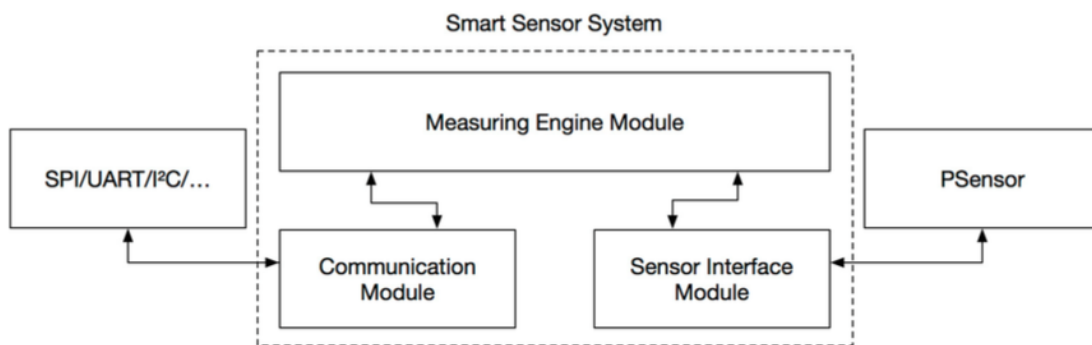


Figure 7.7: Simple Smart Sensor Architecture.