Ricardo Rafael Baptista Gomes

Licenciado em Ciências de Engenharia Biomédica

LONG-TERM BIOSIGNALS VISUALIZATION AND PROCESSING

Dissertação para obtenção de grau de
Mestre em Engenharia Biomédica

Orientador: Prof. Dr. Hugo Gamboa, FCT – UNL

Júri:

Presidente: Dr. Mário Secca, Prof. Associado da FCT - UNL
Arguente: Mestre André Ribeiro Lourenço, Prof. Assistente do ISEL
Vogal: Dr. Hugo Gamboa, Prof. Auxiliar da FCT - UNL

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

October, 2011

Long-term biosignals visualization and processing

Advisor: Prof. Dr. Hugo Gamboa

Thesis submitted in the fulfillment of the requirements for the
Degree of Master in Biomedical Engineering.

Physics Department

Faculty of Sciences and Technology,
New University of Lisbon

2011

Copyright

Acknowledgments

At the end of this important step of my academic and personal life, I would like to thank many people that have accompanied my route through the last years of my life. The last five years have been amazing; learning with experienced professors and dedicated professionals was enriching at all levels.

First, I would like to thank professor Hugo Gamboa, for his support during this year, and for the opportunity he gave me: to have this great experience, with a team of motivated professionals. Thanks for the guidance throughout the time on which I did my master thesis and for always having good advices when they were necessary.

The opportunity to create new tools that can help to visualize, analyze and process information from our biosignals strongly encouraged me for this research work. In addition, it was very rewarding to observe the utility of the innovating developed tools being approved by healthcare specialists after making a demonstration at Hospital Santa Maria.

I also want to thank the staff working at PLUX - Wireless Biosignals, S.A. for sharing the last few months with me. The kindness and dedication that they demonstrated have really motivated me to work inside a team.

I want to thank Joana Sousa for her support during the last months, which was very important for me to accomplish the objectives that were defined in the beginning of this work. I also want to express my appreciation to Neuza Nunes, who has been more than a mentor in my academic career, with her wise advices and visible concern.

I want to show my gratitude to all my friends that were with me in these amazing years on which we shared knowledge and experience. My special thanks to Roque Soares, Gonçalo Lopes, Diogo Tendeiro, Mariana Almeida, Beatriz Rodolpho and Ana Nicolau.

In the end of this important chapter of my life, there are two people who deserve my thanks and appreciation; reaching this point was not easy, but it was possible, in part due to Dr. Marina Tavares and my friend Aníbal Mota. Your help in a critical phase of my life was priceless, and I will always be thankful for that.

Finally, I want to thank those who were always available to help me: my parents and Catarina Ferreira, for their unconditional support, and for always helping me.

I dedicate this work to my parents, and my brother Rodrigo.

Abstract

Long-term biosignals acquisitions are an important source of information about the patients' state and its evolution. However, long-term biosignals monitoring involves managing extremely large datasets, which makes signal visualization and processing a complex task.

To overcome these problems, a new data structure to manage long-term biosignals was developed. Based on this new data structure, dedicated tools for long-term biosignals visualization and processing were implemented.

A multilevel visualization tool for any type of biosignals, based on subsampling is presented, focused on four representative signal parameters (mean, maximum, minimum and standard deviation error).

The visualization tool enables an overview of the entire signal and a more detailed visualization in specific parts which we want to highlight, allowing an *user friendly* interaction that leads to an easier signal exploring.

The "map" and "reduce" concept is also exposed for long-term biosignal processing. A processing tool (ECG peak detection) was adapted for long-term biosignals. In order to test the developed algorithm, long-term biosignals acquisitions (approximately 8 hours each) were carried out.

The visualization tool has proven to be faster than the standard methods, allowing a fast navigation over the different visualization levels of biosignals. Regarding the developed processing algorithm, it detected the peaks of long-term ECG signals with fewer time consuming than the nonparalell processing algorithm.

The non-specific characteristics of the new data structure, visualization tool and the speed improvement in signal processing introduced by these algorithms makes them powerful tools for long-term biosignals visualization and processing.

Keywords: Biosignal, signal processing, long-term monitoring, data structure.

Resumo

Aquisições de biosinais de longa duração são uma importante fonte de informação acerca do estado e evolução dos pacientes. No entanto, a monitorização de longa duração de biosinais envolve a manipulação de bases de dados extremamente longas, o que torna a visualização e o processamento de sinais uma tarefa complexa.

De modo a superar estes problemas, uma nova estrutura de dados para manipulação de biosinais de longa duração foi desenvolvida. Baseado nesta nova estrutura de dados, ferramentas dedicadas à visualização e processamento de biosinais de longa duração foram implementadas.

Uma ferramenta de visualização multi-nível para qualquer tipo de biosinais, baseada em sub-amostragem é apresentada, focando-se em quatro parâmetros representativos do sinal (média, máximo, mínimo e desvio padrão).

A ferramenta de visualização permite uma visão geral da totalidade do sinal e uma visualização mais detalhada em trechos especificos em que se queira realçar, possibilitando uma interacção *user friendly* que leva a uma mais fácil inspecção do sinal.

O conceito de "map" e "reduce" é também apresentado para o processamento de biosinais de longa duração. Uma ferramenta de processamento (detecção de picos de ECG) foi adaptada para biosinais de longa duração. De modo a testar o algoritmo desenvolvido, aquisições de longa duração (aproximadamente 8 horas) de biosinais foram efectuadas.

A ferramenta de visualização provou ser mais rápida do que os métodos padrão, permitindo uma navegação rápida sobre os diferentes níveis de visualização dos biosinais. Em relação ao algoritmo de processamento desenvolvido, este detectou os picos de sinais de ECG de longa duração com menor consumo de tempo que os algoritmos de processamento não paralelo.

O carácter não-específico da nova estrutura de dados e da ferramenta de visualização e o

aumento de velocidade do processamento de sinais introduzido por estes algoritmos torna-os ferramentas potentes para a visualização e processamento de biosinais de longa duração.

Palavras-chave: Biosinais, processamento de sinal, monitorização de longa duração, estrutura de dados.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The growing demand for medical systems and applications for human welfare and quality of life is increasingly supported by the body signals monitoring of a subject.

There are several types of body signals, also called biosignals, including bioelectric (generated by nerve and muscle cells), bioimpedance (containing information about tissue composition, blood volume and distribution, endocrine activity, automatic nervous system and more), biomagnetic, bioacustic, biomechanical and biochemical signals [6]. These biosignals give the researcher or the clinician a perspective over the patient's state since they carry useful information for the comprehension of complex physiologic mechanisms underlying the behavior of living systems. The process of monitoring biosignals may be as simple as a physician estimating the patient's mean heart rate by feeling, with the fingertips, the blood pressure pulse. Biomedical signal analysis is nowadays a method of the greatest importance for data interpretation in medicine and research, since the manipulation and processing of data provide vital information about the condition of the subject or the status of the experiment.

Data visualization and inspection are an increasingly important part of understanding and explaining phenomena of everyday life. Besides acquiring biosignals it is desirable to visualize and extract information, either at real time, or by graphically displaying and analyzing them offline.

Signal visualization and processing techniques have been developed to help the examination of many different biosignals and to find important information embedded in them [22]. The advantage of visual data exploration consists in involving the researcher or clinician directly in the data mining process, since he can select visually the interesting parts of the

signal being analysed.

In clinical cases such as sleep disorders and neuromuscular diseases, a constant monitoring of the patient's condition is necessary. This requirement is due to the possible occurrence of sudden alterations in the patient's state. The demand for a correct and prompt diagnosis leads to a mandatory identification of insufficiency signs in the clinical context. With this intention, long-term biosignal acquisitions are one of the possible methods that allow a continuous monitoring of the patient [21]. However, long-term acquisitions generate large amounts of data. In order to analyze and follow up the patient's condition it is very important to acquire, visualize and extract relevant information from the signals. In patients with neuromuscular diseases, the heart rate variability, respiration, muscular and electrodermal activity signals are extremely important, since they indicate when a muscular crisis is occurring [46]. In a future perspective, the continuous monitoring of these signals will allow the health care providers to know beforehand when the patient needs assistance, assuring the patients' comfort and safety while they are continuously and remotely monitored in ambient assisted living conditions [41].

The long duration datasets obtained by these acquisitions exceed the capabilities for which standard analysis and processing software were designed. Besides processing problems related to the difficulty to manipulate large amounts of data, long-term biosignals are not easy to display using standard visualization software. The difficulties to visualize signals obtained in long acquisitions (e.g. recording for several hours) rise up from the lack of capability in the currently available tools to correctly visualize the entire signal [22].

Considering the described problems with the long-term biosignals visualization and processing, and the importance of this kind of signals in health and research areas, this work presents new solutions that aim at the development of tools that enable a simple visualization of very large biosignals and an effective processing of this kind of signals.

This dissertation was developed at PLUX - Wireless Biosignals, S.A. [37], which considers the main goals of its Research and Development (R&D) department to be the creation of new solutions for more comfortable and ergonomic biosignals monitoring.

1.2 State of the Art

The design of tools to visualize and analyze biosignals have been an active research area in the last years, due to the importance of monitoring a subjects' condition. Nowadays, it is possible to acquire a variety of these biosignals, either on a clinical context or during research studies, that can provide very useful information about the patients' or subjects' state. Biosignal

monitoring techniques aim at the perception and identification of important variables that can be extracted from biological data.

In order to save and exchange the acquired signals, it is primarily necessary to choose a correct data structure to store information. Thus, signals must be recorded in a format that allows *a posteriori* data accessing. There are several standard formats for biological data storage and exchange and one of the most common examples is the EDF (European Data Format) [12, 23]. This format allows to save multiple channels data, as well as information about the recording and the subject, being widely used for biomedical signal databases.

At the Harvard-MIT division of Health Science and Technology, an interesting web based research resource for complex physiologic signals was developed to help new investigations in studies of biological signals. This resource, exposed by Moody *et al.* [31], is a framework composed of three interdependent components: database of physiologic signals (PhysioBank), tools for biosignal analysis (PhysioToolkit) and a web based discussion forum (PhysioNet) [36]. The already mentioned biosignal database - PhysioBank - allows internet access to long-term biosignals that can be downloaded, but the necessary tools to provide a fast and easy visualization and processing of long-term biosignals are not user friendly.

Multilevel visualization of very large datasets has been developed over the last years by web mapping services that enable internet users to see images of the Earth in different levels of detail [47]. However, to our knowledge, this type of approach was not implemented in the biosignal analysis area.

Long-term biosignal visualization has been an area in constant development, as well as data mining algorithms [27]. Time series data mining approaches has led to the introduction of similarity measures, representations and algorithms [2, 24, 25]. Despite the advances in such techniques, those who deal with time series are confronted with difficulties in learning, implementating and manipulating the mentioned tools [27].

An open source tool to visualize and perform processing operations such as filtering, powerspectrum, or heart rate detection in Electrocardiography (ECG) data saved in the EDF format is the EDFbrowser. In spite of allowing to open very large data sets, the visualization tool is not fast, and does not enable the visualization of the entire long-term biosignals.

Processing very large datasets is another field where important results have been achieved. Parallel processing techniques (using several processors that process a part of a dataset each one) allow the division of one big, complex task, in several smaller and faster operations.

MapReduce is a programming model introduced by Google for processing large data sets with two simple concepts: the map (the ”split” step, on which the big input problem is parti-

tioned to be processed in smaller parts) and reduce (the ”merge” step, when the results of the processing partitions are combined to generate the output) [9]. This programming paradigm has been applied in some investigation areas, such as data intensive scientific analysis [13].

The visualization and processing solutions described in this thesis contribute for the development of new tools for biosignal analysis and processing specifically for long-term signals.

1.3 Objectives

The primordial objective of this thesis is to develop tools for the visualization and processing of long-term biosignals. To accomplish these objectives, biosignals obtained in long-term acquisitions are necessary, as well as tools capable to store and visualize the large amounts of acquired data and algorithms for large datasets processing.

Considering the main goals of this research, a new data structure which provides a novel way to store long-term biosignals and easily access them with dedicated visualization and processing tools needed to be designed and implemented. The visualization tool must offer the possibility to inspect biosignals with huge sizes in a fast and user friendly way. The proposed tools must have future perspectives to become powerful for biosignals inspection and analysis, accessible remotely in a web based environment. Regarding signal processing, algorithms for an efficient processing of very large datasets, that do not exceed start alone computer's capabilities are an objective.

The presented tools need to be tested on different time varying biosignals obtained from the human being, such as as electromyography (EMG) , electrocardiography, blood volume pressure (BVP), accelerometry (ACC) or respiration (Resp) signals. However, the goal of this thesis was not to develop tools to be applied in a specific type of signal but to be as general as possible.

1.4 Thesis overview

The main structure of this thesis is represented in Figure 1.1.

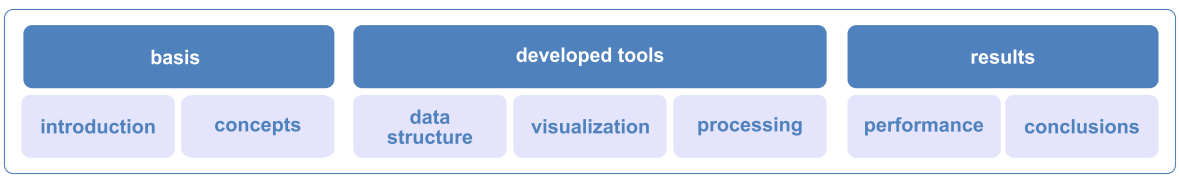

Figure 1.1: Scheme of the thesis structure.

The present chapter has an introductory role on the thesis context. The motivations and objectives that encouraged the development of this work are exposed, and an overview on the state of the art is provided. The "Basis" (Figure 1.1) of the thesis is completed with Chapter 2 (theoretical concepts), on which the fundamentals of our work and important concepts are explained.

Chapter 3 focuses on the new data structure implemented, while chapter 4 provides information on the developed tool to visualize long-term biosignals. In Chapter 5, the developed algorithms for long-term biosignals processing are exposed and explained. Chapters 3, 4 and 5 together form the second part of the thesis, the "Developed tools" (Figure 1.1). This is the part on which the developed tools are scrutinized and all the details about the implemented innovating algorithms are reported.

Finally, the "Results" (Figure 1.1) part of the thesis is composed by the performance evaluation (presented in Chapter 6) and the conclusions (exposed in Chapter 7).

The thesis has two additional appendixes. Appendix A presents the article that was submitted during this work and accepted for publication. Appendix B presents a work route, giving an insight on the different steps taken through this work until the final result.

Chapter 2

Concepts

In this chapter, contextual information about biosignals, data mining, data visualization and parallel processing mechanisms will be provided. The objective is to introduce relevant concepts that will help to understand the fundamental basis of the present work.

2.1 Biosignals

Biosignal is a term used for all kinds of signals which can be continuously measured from biological beings. Biosignals are space, time, or space-time records of biological events such as heart beats or the contraction of a muscle. The electrical, chemical or mechanical activity ocurring during these biological events often generates signals that are measurable and can be analyzed. For that reason, biosignals contain useful information that can be important for medical diagnosis, since they allow the comprehension of underlying physiological mechanisms of a specific biological event or system [14].

Biological signals can be acquired in a variety of ways and following signal acquisition, the recorded data is analyzed in order to extract signal's characteristics. The next section presents several biosignal types, focusing on a specific type: the ECG signals.

2.1.1 Biosignals Types

In the daily life, the acquisition of a specific biosignal may be of the utmost importance in order to understand the origin of a specific problem.

The physiological origins of biosignals can be various. Figure 2.1 presents possible physiological sources of these signals. Examples of this variety of biological signals are Blood Volume Pressure, which derives from the force exerted by blood on the walls of blood vessels, Electromyography, that is measured due to the electrical activity generated by nervous system

control of the muscle cells, Accelerometry, which tracks movement's acceleration, Electrodermal Activity (EDA) , that reflects changes in the electrical properties of the skin, Respiration signals (Resp), which arises from variations in the chest volume due to respiratory activity and Electroencephalography (EEG), which monitors the electrical activity of the scalp.

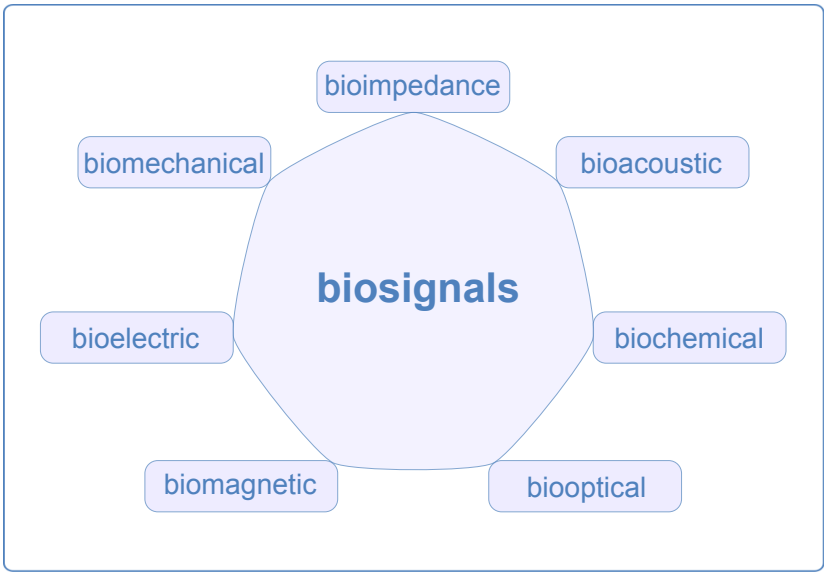

Figure 2.1: Different types of biological signals.

In spite of the existence of several types of biosignals and the general purpose of the developed work (to developed tools which are signal-independent) a study was made involving a specific type of biosignals, the bioelectric signals (application of an ECG peak detection algorithm to long-term biosignals).

Bioelectric signals are generated by nerve and muscle cells, due to electrochemical changes that occur between cells. Stimulating a cell with a strong enough stimulus causes an action potential (ions flowing through the cell membrane) that can be measured. The excited cell can transmit the action potential to the neighboring cells, inducing the propagation of this potential. The activation of a large number of cells generates an electrical field, measurable on the surface of the tissue. Examples of this type of signals are ECG, EEG and EMG.

In the next section a detailed view of Electrocardiography and ECG signals is provided. The focus on these specific biosignals is justified by the applications concentrated specially in problems involving them (see Chapter 5.2).

Electrocardiography

One of the most familiar biosignals to be measured is the electrocardiogram, which is the recording of the electrical activity of the heart, associated with its mechanical activity (re-

polarization and depolarization of the atrial and ventricular chambers of the heart).

In that way, diagnostic analysis of the mechanical function of the heart is achieved through the assessment of the ECG. Electrocardiography takes care of the detection and amplification of electrical changes on the skin, with surface electrodes, that are caused by heart muscle depolarization which occurs on each heart beat. In general, ECG's important parts consist of P, QRS and T waves, shown in Figure 2.2.

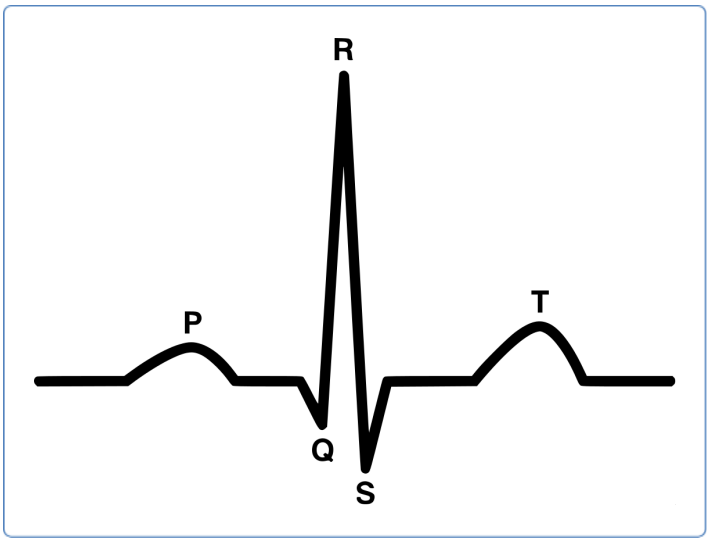

Figure 2.2: Normal shape of an ECG signal. The P, QRS and T waves are highlighted.

The P-R interval is a measure of the time from the beginning of atrial activation to the beginning of ventricular activation, ranging from 0.12 to 0.20 seconds [20].

The QRS complex' duration lies between 0.06 and 0.10 seconds [20] and its abnormal prolongation maybe a signal of blocking in the normal conduction pathways through the ventricles. The S-T interval reflects the depolarization of the ventricular myocardium, while the T wave indicates its repolarization.

ECG processing techiques and parameters extraction play an important role in the monitoring of patients. Several measures and analysis proceedings can be done with ECG signals. One of the most important examples is the QRS peaks detection. This waveform is the most easily identifiable within the electrocardiogram. Since it reflects the electrical activity within the heart during the ventricular contraction, the time of its occurrence and its shape provide much information about the current state of the heart [26]. Other important parameter is the heart rate variability (HRV) [29]. This physiological phenomenon is characterized by the variation in the time interval between heart beats.

In order to study and analyze biological signals, it is necessary to have access to these data. The biological information can be used for instantaneous analysis with real-time processing

(heart rate calculation) or for a posterior inspection in search for useful information.

In the present work our focus resides on the analysis, visualization, and processing of biosignals *a posteriori*, i.e. not simultaneously with signal acquisition. Therefore, in the next section, an introduction to biosignals acquisition is presented.

2.1.2 Biosignals Acquisition

Biosignals are often analog and with small amplitudes when compared to the surronding noise. To enable the extraction of meaningful information from biosignals (which are crucial to understand biological systems), powerful data acquisition techniques and equipment are commonly used.

Normally, these signals contain unwanted interference or noise that mask relevant information [6]. Thus, high-precision low-noise equipment is necessary to minimize the effects of noise. The basic components in a bioinstrumentation system are shown in Figure 2.3.

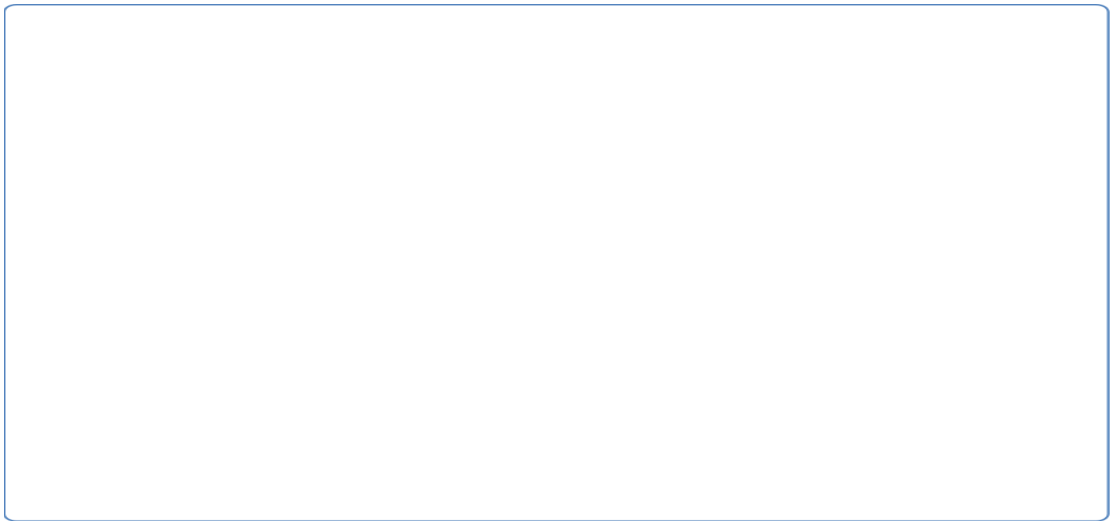

Figure 2.3: Scheme of the acquisition of biosignals. Biosignals are ”read” by a specific sensor; the sensor converts physical information into an electric output, allowing the conversion from biological data to electrical records. From [14].

A sensor converts physical *phenomena* into an electric output. There are several types of sensors; its objective is normally to transduce the observed biosignal into an electrical analog signal that is measurable using a data acquisition system.

The data acquisition system converts the analog signal into a digital signal that can be stored. Digital signal processing techniques are applied to the saved signals in order to reduce noise and extract additional information which can be useful for the comprehension of the physiological meaning the acquired signals.

Any stage of the biosignal acquistion chain (amplification, analog filtering, and Analog-to-digital conversion (ADC)) shall generate misleading or untraceable distortions. Distortions in a signal measurement may lead to an improper diagnosis, which represents an high risk, since these signals carry biological information which might be used for medical interpretion.

Since data is stored in computers in the form of discrete values, the analog signals need to be converted into discrete units in an analog-to-digital conversion. This conversion is composed by two steps: sampling and quantization. The continuous values are observed (sampled) at fixed intervals and rounded (quantized) to the nearest discrete values, as it is shown in Figure 2.4. The generated time values of this proccess are called ”samples”.

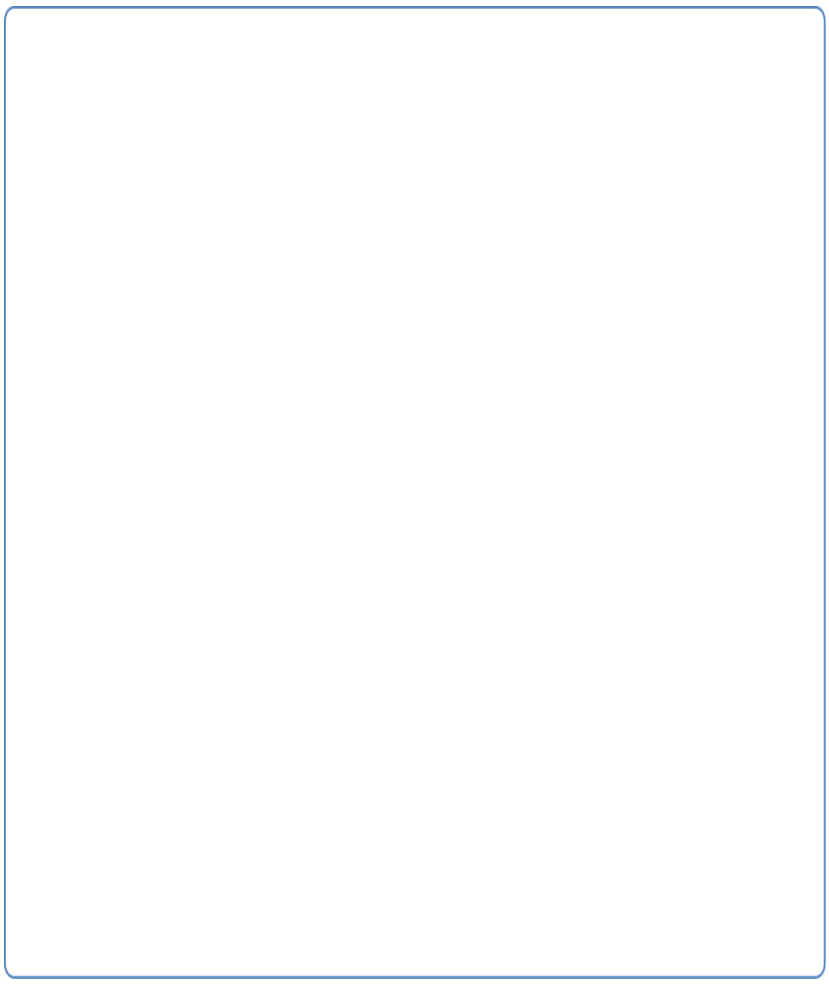

Figure 2.4: Sampling and quantization of an analog signal. From [14].

ADC has two important parameters that influence how the digital data represents the original signal: the precision (accuracy level of a sample observation) and the frequency (defines the observation rate) with which the signal is recorded and sampled [30].

The present work has as main objective the creation of tools for the storage, display and processing (the last two steps in Figure 2.3) of long-term biosignals. Biosignal processing is

focused in the extraction of important information from the signals, by manipulating them so that relevant data can be can be extracted. The next section presents the biosignals processing concept.

2.1.3 Biosignals Processing

The representation, transformation and manipulation of biosignals and extraction of significant information are subjects of biomedical signal processing. Biosignal processing tools have supported the development of medical monitoring systems that provide an overview over the human body's functioning.

The processing of biomedical signals is usually composed of three stages [40]:

- Signal transformation;
- Signal parameters extraction;
- Signal interpretation and/or classification.

After the acquisition of biosignals, described in section 2.1.2, one of the main goals is to obtain abstract information from the acquired signals.

The parameters extraction step might provide medical care professionals with important information, which was not visible by looking to the recorded biosignals. This process allows the interpretation and classification of biosignals, providing a perception of the patients' state. A variety of parameters extraction techniques can be used and several parameters can be extracted from biosignals, such as the heart rate and HRV analysis from ECG signals.

Since there are several health conditions which require a long-term monitoring of patients' biosignals, large data sets are obtained. Issues related with the signal sizes problem are presented in the next section.

2.2 Long-term datasets

In the last years, our capability to collect and store data has overtaken our ability to process, analyze and explore it. Scientists and engineers from a broad range of working areas have been capturing increasingly complex experimental data sets, such as high spatial, temporal and spectral-resolution remote sensing systems, and other environmental monitoring devices [8]. The medicine and biomedical engineering areas are not an exception, since nowadays an increasing amount of biological data is saved. One of the examples is the acquisition of biosignals in a long-term perpective, for a close monitoring of the subjects' biological signals.

This rise in the quantity of saved data has led to the necessity of finding data mining techniques which allow the analysis, processing and visualization of the recorded data. The next chapter covers the data mining concept and gives an insight on data mining techniques that have been developed.

2.2.1 Data mining

During the last years databases have been growing in size, as well as in the varieties of data and its applications. This upgrowth causes the development of tools to extract useful information (knowledge) from the huge volumes of data to be of the highest importance. The most relevant process in these required tools is the application of specific data-mining methods for pattern discovery and extraction. [15]

Several data mining techniques have been developed in order to allow data analysis and display, since they are fundamental for decision support in many application contexts.

One of the possible fields where data mining is necessary is the exploring of spatio-temporal data sets, which are often very large and difficult to analyze. An example is given in [8] with algorithms for spatio-temporal data sets data mining and visualization.

In the health care area, a data mining approach to policy analysis in a health insurance domain [7] was addressed in 2001 in order to demonstrate how these algorithms can be used to predict health outcomes and provide policy information for hypertension management.

Data mining techniques have also been developed in the biosignals area. An example of application of this technique is the *signal clustering*, which consists of assigning a group of objects into groups (called clusters) so that the objects in the same cluster are more similar to each other than to those in other clusters [32].

Specifically in health care data field (particularly for biosignals), data mining algorithms are very useful tools to analyze current trends and changes in the data.

2.2.2 Multilevel visualization techniques

Web mapping services and spatial visualization features are a growing area nowadays. Services like Bing Maps [1] and Google Earth [11], that provide multi-resolution visualization tools to explore our planet are widely used by the internet general public. The concept associated to this services is to allow the user to explore the Earth by ”overflying” and being able to have a closer or farther view of the planet’s surface.

These tools provide the possibility to visualize images of the Earth with multiple resolution levels, according to the area that is being selected. The bigger that area is, the less detail is

contained in the images being shown.

Besides web mapping and geo-spatial exploring tools, other multilevel visualization methods have been created. One example is the application of this idea for accessing Electronic Health Records (EHR) in phases [3, 42], according to the level of detail that is necessary.

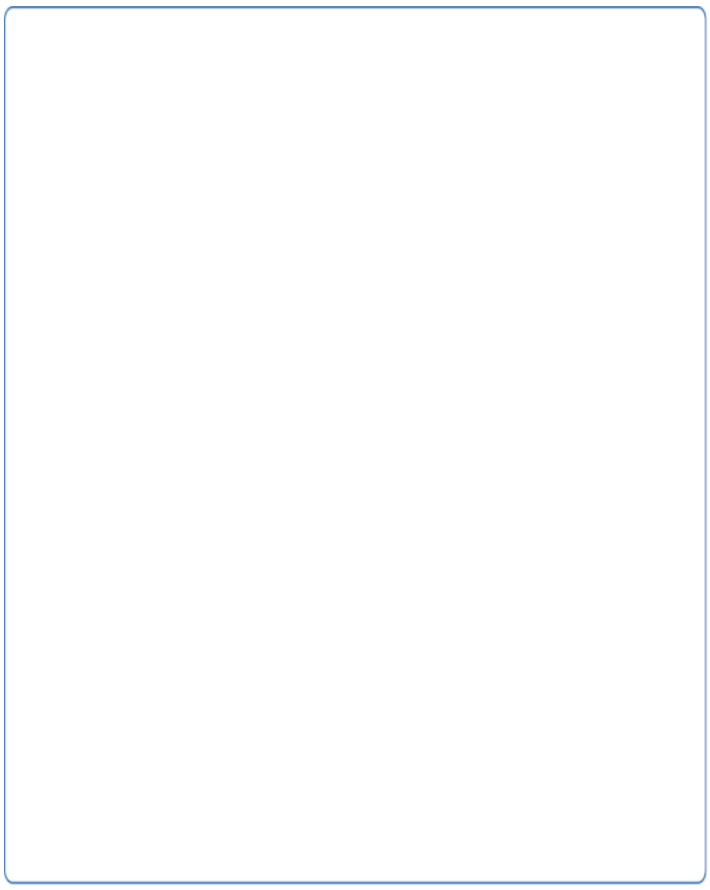

Figure 2.5: Schematic representation of an Electronic Health Record exploring tool. From [3].

This particular application was explored and a representative scheme is shown in Figure 2.5 which introduces an exploring tool that enables organizing an EHR according to Levels Of Detail (LOD). The different levels of detail are organized as it is described below:

- **LOD1:** the top (less detailed) level is the problem set. All the data in EHR is categorized into several problem sets (each set has all the problems related to a specific organ sytem);
- **LOD2:** this is the problem level. Specific health problems of each problem set are gathered in this level;
- **LOD3:** in this level, all the visits that the patient did in order to solve the health problem of the subsequent level are organized chronologically;

- **LOD4:** the fourth level is related with summary data (exam results, patient condition and treatment procedures) of each visit.

- **LOD5:** is the most detailed level and allows assessing all the specific data of each visit.

As an application example for the LOD EHR exploring tool described above, the first level of detail may contain a circulatory system set. Consequently, the heart is one of the possible organs to be affected, and Coronary artery disease is one of the possible problems to affect this organ; this is the second level of detail. In the third detail level, the description of the patient's visits related to the problem mentioned in the second level is given. The fourth level of detail provides information about a specific visit that the patient did in the Coronary artery disease context; Access to specific exams (e.g. ECG), medical reports on the patient condition or clinical procedures is possible in this level of detail. The fifth detail level is where all the data regarding the specific visit related to the Coronary artery disease.

As it was presented, some applications of the multilevel visualization concept have been developed. However, this concept could also be applied to biosignals, since the acquisitions of this type of signals are increasingly growing, making signal analysis and visualization difficult tasks.

2.3 Parallel computing

Parallel computing relies on the division of a complex problem in several smaller problems which can be solved in parallel; these smaller problems can be solved using multiple Central Processing Units (CPU's). The parallel processing concept has started to become widely known since the beginning era of the multiprocessor computers [4].

The traditional developed software was written for serial computations. By other words, the standard software architectures are developed to run on a single computer, using one central processing unit. The problems were splitted in series of instructions executed in sequence so that the instructions run in separate moments in time. However, with the parallel computing concepts, the problems are splitted in different parts that can be solved separately [5]. This problem simplification allows to divide a complex task in multiple subtasks which can be undertaken separately, using multiple processors. All the processors run a sequence of instructions at the same time, producing a result faster than serial computations.

The computer resources that enable parallel processing may arise from the use of a computer with more than one processor, a network of computers, or even both sources together.

An example of a parallel processing architecture is given in Figure 2.6.

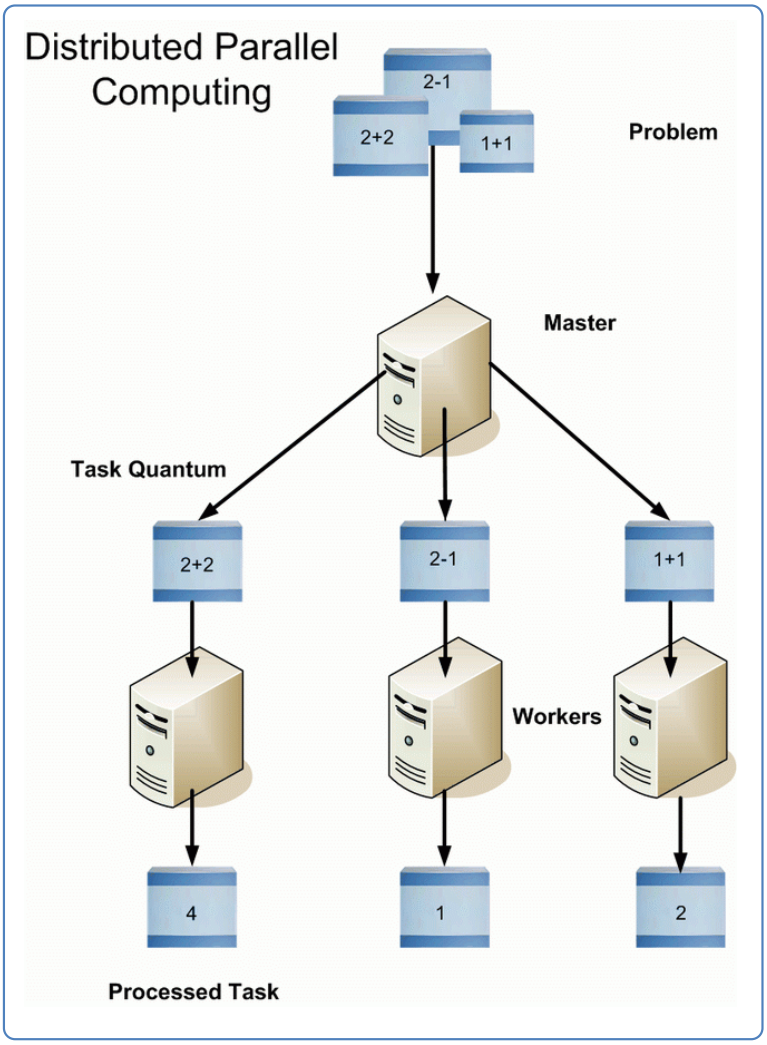

Figure 2.6: Schematic representation of a parallel computing architecture. From [10].

Figure 2.6 represents the parallel computing concept; a master computer may have three independent tasks to be performed. These tasks are distributed by a network of computers, so that each one only has to process one simple task.

One well known parallel computing approach (the MapReduce) is described below.

2.3.1 MapReduce algorithms

MapReduce is a programming model and its associated implementation for processing and generating large data sets [9] developed by the Google team.

The programs developed using this model are parallelizable and thus can make use of a cluster of computers.

The programming model presented by Dean and Ghemawat is based on the following, simple concepts: a ”map” function is specified by the user; this function processes a key/value pair to generate a set of intermediate key/value pairs. All the intermediate key/value pairs with the same key are merged by a ”reduce” function.

As an example, consider the problem of counting the number of occurrences of each word in a set of files (for example, in a directory with several different files). A MapReduce approach to solve this problem could be described by the following:

1. The map function iterates over the files in the specified directory;
2. For each file, the key/value pairs are computed. Each time a word occurs a new pair <word, 1 >is computed;
3. All the intermediate values are grouped by key;
4. Iteration over the resulting groups;
5. Each group is reduced by adding the number of occurrences of each specific word.

Besides the simplicity, this programming model enables a serial algorithm to become parallel operations (”map” and ”reduce”).

Chapter 3

Data structure

The visualization, analysis and processing of long-term biosignals are mandatory tasks to continuously monitor and understand electrophysiological data from patients.

As the present work deals with long-term biosignals, a tool to store and display large amounts of data is of great importance in order to enable a prompt, easy and correct signal analysis.

This chapter exposes the proposed data architecture, aiming at a dedicated tool for long-term biosignals visualization, analysis and processing. This chapter also gives an insight on the work already done in this area, in order to combine the already designed concepts and achieve the established objectives.

3.1 Overview

One of the main goals of this work was to create tools that could answer the problem of long-term biosignals analysis. To do so, this work presents a new architecture for biosignal data and, supported by this new data structure, a new software to access, visualize and process long-term biosignals. For that, some requirements were defined such as:

- rapid access to the files on which data is recorded;
- a fast and user-friendly multi level visualization tool to view entire biosignals with large sizes, featuring advanced navigation options to access specific parts of the biosignals;
- an intuitive and fast biosignals processing tool, applying the concept of parallel processing to biosignals.

However, one of the main problems in long-term biosignals resides on the file format on which the acquired biosignals are stored to be accessed, visualized and processed posteriorly.

The chosen file format must not compromise the access to the acquired data. Thus, it is very important to study and understand the advantages and limits of each file format to save the recorded signals so that an easy and fast data access is guaranteed.

Since several types of standard formats for biomedical data were already created with this objective, a survey of those file formats was carried out in order to understand which one represents the best solution to meet the needs of the proposed work.

3.2 Standard file formats

Biomedical signal databases are used in several areas, such as engineering, scientific research and healthcare. Database standardization facilitates multicenter collaboration and data sharing. As a consequence of the benefits brought by data format standardization, a large number of standards for biological data recording have been created [44].

Following, a list of standard file formats for biosignal databases (as well as other file formats that are not specific for biosignals) that can represent an alternative solution for the mentioned problem is presented.

- ***.mat** - MATLAB

  MATLAB file format (.*mat) is one of the most known file formats in the biomedical engineering area. This file format saves data in binary (not human-readable) form. *.mat files have a 128 byte header (with information about the file) followed by one or more data elements. Each data element is composed of an 8 byte tag followed by the data in the element. The tag function is to specify the number of bytes in the data element and how these bytes should be interpreted (as 16 bit values, 32 bit values, floating point values or other). The tags provide fast access to individual data elements within a *.mat file, since they map the data. Once found a tag when exploring a file, it is possible to skip ahead a chosen number of bytes until the next tag [28].

  Despite being a powerful tool, MATLAB is not Open Source and thus cannot be used freely, making it necessary to look for other solutions regarding biosignal databases management.

- ***.txt**

  Text files are globally known by users with a broad range of usage applications since they are computer files, existing inside of the file system. This type of file format is not specific to save biological data or any other type of data; hence it can be used in many areas. *.txt files are structured as a sequence of lines, with special characters to indicate when a new line starts, and when the end of the file was reached.

  In addition to the advantage of being more portable than binary files, both across systems and programs, text files can be more easily accessed and modifed, for example, using one of the many available text editors.

  There are some acquiring system devices which use this file format to store data. In this research work an equipment to acquire biosignals that records data in a text file was used (bioPLUX wireless acquisition unit [37]). This is a portable, small sized and light-weighted system with a 12 bit ADC and a sampling frequency of 1000 Hz. Data is saved in text files (*.txt extension) composed by a header with 8 lines of information about the record (date, time, sampling frequency, sampled channels, sampling resolution and acquiring device mac address) and a section with the sampled channels (digitized values), organized in columns.

  The main disadvantage of this file format is the difficulty to have a fast access to the data. In order to access data from a specific line of the file, all the previous lines need to be read, making data accessing a slow process.

- ***.edf** - EDF

  EDF is a simple format for archiving and exchanging of biological and physical signals. The signals can have any (and different) physical dimensions and sampling frequencies.

  An EDF file has an ASCII header containing mainly patient and recording time identification, the number of signals, the duration of the data records and the characteristics (mainly dimension, calibration values, sampling frequency) of each signal. Following the header are subsequent data records, each of the same duration, that contain the recorded signals in 2 byte integer values.

  The EDF file format, that can also accommodate annotations, markers, and events has become standard for EEG and PSG (Polysomnography) acquisitions [12]. However, the study of its specifications and the implementation of an EDF import/export unit is a time consuming task (which may take a few days of work) [44].

- ***.dat** - PhysioBank

  PhysioBank is an archive with characterized digital recordings of physiologic signals and related data for use by the biomedical research community. PhysioBank contains biomedical signals from healthy subjects and patients with a variety of conditions.

  Each PhysioBank database can contain more than one record, and each recording might have three files: the header information (*.hea file), a short text file that describes the signals (with the name or URL of the signal file, storage format, number and type of signals, sampling frequency, calibration data, digitizer characteristics, record duration and starting time), the annotation file, with the description of features of one or more signals in the record and a binary (*.dat) signal file, containing digitized samples of one or more signals.

- ***.h5** - HDF5 (Hierarchical Data Format 5)

  HDF5 is a file format for storing and managing data (not specific for biosignals). A variety of datatypes is supported by HDF5, which is portable and extensible, allowing applications to evolve in their use of this tool [18].

  *.hdf5 (or in a simpler way, *.h5) files have a simple structure. The architecture includes only two major types of objects:

  - **Datasets**, which are multidimensional arrays of a homogenous type;
  - **Groups**, which are container structures that can hold datasets and other groups.

  An intuitive Python interface for this file format is available through the *h5py* package [17], allowing fast and robust storage of enormous amounts of data, organized by name. The HDF5 file format allows fast and random access to any point of the data.

Taking into account the features, advantages and disadvantages of each file format mentioned before, HDF5 is the format that better fulfills the requirements defined for the purposed work, i.e., to develop a tool that allows a fast access, visualization and processing of long-term biosignals. Due to its features, HDF5 is a powerful tool for storing and managing large amounts of data since this file format allows the necessary random access to any point in the signal in a fast way.

In the next section, the developed data structure based on the chosen file format is presented.

3.3 Proposed data structure

In this section, the fundamentals of the proposed data structure are presented and the designed algorithm for its creation is also explained.

3.3.1 Basis idea

Since an acquisition equipment that stores data in text files (*.txt) was used (the bioPLUX unit), the major obstacle that appeared as a consequence of the file format that stored the biosignals was the impossibility to have random access to a specific time window of the recording, chosen by the user to visualize.

In order to overcome this difficulty, a new data structure that enables accessing the data in a fast way was developed. As mentioned in the previous section, the data structure developed in the present work was based on the HDF5 file format.

The data structure architecture is represented in Figure 3.1.

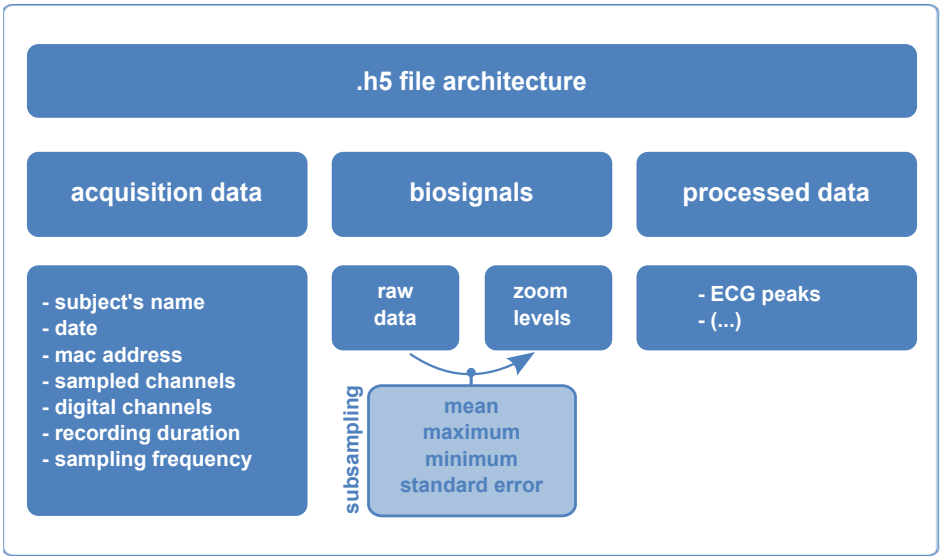

Figure 3.1: Proposed data structure for biosignals.

The data architecture (Figure 3.1) is based on three main blocks: acquisition data, biosignals and processed data.

The first block (acquisition data) provides the user with the general information about the acquisition: signal acquisition parameters (sampling frequency, sampled channels, digital channels, sampled channels, acquiring device mac address), recording information (date, duration) or the subject characteristics (name, age).

The biosignals block, represented in Figure 3.1, contains the acquired biosignals (raw data) and the different zoom levels. Thus, from this second block, the user can visualize the

raw signals or specific parts of them, using the detail levels of visualization.

The different levels of visualization are stored in the new data structure so they only need to be calculated once, allowing the user to start visualizing any specific part of the signals instantaneously and at any time as it was required.

To obtain the different zoom levels, four subsampling parameters (mean, maximum, minimum, standard deviation), shown in Figure 3.1, are extracted from the signal.

The choice of these four specific parameters to represent several zoom levels of the signals was based on:

- **mean:** identifies the biosignals' central location and provides a representative measure of the signals' shape. The mean of a discrete signal X with n samples is represented by $E[X]$ and is calculated as it is given in equation 3.1.

$$E[X] = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{3.1}$$

- **maximum and minimum:** are the parameters that define the envelope on which the sampled signal is restrained;

- **standard deviation error:** gives information about the signal's spreading, indicating signal variation zones.

The zoom levels concept is the key for the visualization of long-term biosignals, since with this kind of approach the tool provides rapidly a general overview of the entire signals showing the mean, maximum, minimum and standard deviation. Visualizing the signal's morphology with more detail is also possible. For this, the user can access the higher zoom levels, which contain a larger amount of data, allowing a thorough analysis of the signal in specific time windows.

The third block in Figure 3.1, represents the processed data. In this part of the data structure, processed data can be stored for further analysis. As an example, the "processed data" block allows saving the QRS peaks times extracted from ECG signals.

The process that enables the creation of the different zoom levels is described in the next section.

3.3.2 Visualization levels creation algorithm

The visualization of biological signals is normally done by displaying an entire signal (operation that might take a long time to be carried out) or by displaying only a portion of the entire

signal. However, displaying an entire signal (specifically a long-term biosignal) is a time consuming operation. In the other hand, visualizing only a portion of the entire signal can lead to mistakes in signal analysis, increasing the risk of a bad diagnosis. Thus, the multi-level visualization architecture aims at overcoming this problem, enabling the navigation over the different zoom levels that represent the entire signal.

In signal processing, subsampling is a technique to reduce the amount of data of a signal. A subsampling-based algorithm was developed in this work in order to create the biosignals' zoom levels. Each zoom level provides a different resolution of the signal. The first (and more detailed) level of visualization is the raw data. This level has the entire biosignal recording and gathers the largest amount of information available about the biosignal being visualized. The subsequent zoom levels provide less detailed information than the preceding one since they have a smaller number of samples (because of the subsampling operations). In spite of having a smaller amount of samples, thus a fewer quantity of data, which leads to less detail, the different levels of zoom represent the same time interval.

Each subsampling operation is carried out by splitting the input signal in groups with a selected number of samples - the resampling factor, that is going to be hereafter denoted by r, and for each group the representative signals' measures are calculated. The resampling factor can be, for example 10, which means that the maximum, minimum, mean and standard deviation will be computed from 10 to 10 samples. This way, the data length will be divided by r, since each group with r samples will be represented by one new sample.

To allow an overview of the developed algorithm, a fluxogram representing the creation of the biosignals data structure is shown in Figure 3.2.

There are three main parts in this procedure. The first task of this sequence is the reception of the different inputs - the raw data to be converted to the new data structure (with N samples), the different recording attributes (such as the sampling frequency, the precision number of bits, the sampled channels and the recording date) that are read from the .txt file on which the biosignals were recorded and finally, two optional parameters: the resampling factor (r) and the maximum number of points that can be drawn on each zoom level ($npviz$). This optional parameters are predifined to be 10 and 1000 respectively, if no value is given by the user.

The second step of the present algorithm consists of writing the attributes of the acquisition (Figure 3.1 in the "acquisition data" block). Besides the recording attributes read from the acquisition file, other attributes given by the user (name of the patient, recording place, and any other information that might be considered necessary) are written.

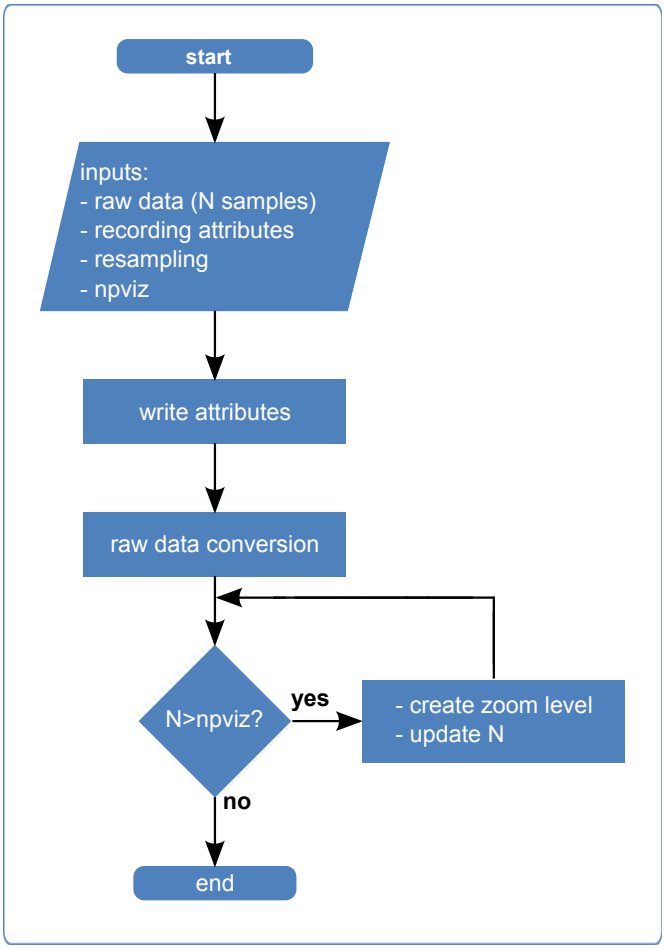

Figure 3.2: Fluxogram of the developed algorithm for the creation of the proposed data structure.

After writing the recording attributes, the algorithm proceeds with raw data conversion. Considering the potential of the HDF5 file format, described in section 3.2, the algorithm creates a new group called ”raw”, on which the raw data is saved. The ”raw” group contains different datasets. Each dataset saves one of the sampled channels. The raw data conversion algorithm consists of reading a *.txt file line by line, saving the data into *.h5 file format, i.e. the conversion step converts the *.txt file into the new data structure, which, as it was mentioned, allows an easier and faster access to the acquired data.

The third step of the data structure algorithm is the creation of the zoom levels. The first zoom level is obtained taking the raw data as the input signal and extracting the mean, maximum, minumum and standard deviation error parameters (see Figure 3.1). For higher zoom levels, the same parameters are extracted, but instead of using raw data, the algorithm uses as input the data from the last zoom level to be created. In this case the algorithm calculates the mentioned parameters taking advantage of the data reduction that is done on each level computation. Thus, the algorithm is simplified, since it calculates the mean of

means, the maximum of maxima, minimum of minima and the standard deviation error. It should be noted that the standard deviation error, (*std*), is obtained taking into account the expression given in equation 3.2, where $E[X]$ represents the expected value for the random variable X.

$$std(X) = \sqrt{E[X - E[X]]^2} = \sqrt{E[X] - E[X]^2} \tag{3.2}$$

As it is represented in the algorithm's fluxogram (Figure 3.2), the subsampling operations are carried out while the new subsampled signal has a bigger number of samples than the *npviz* parameter, that represents the limit (maximum) number of samples that the outermost zoom level can have. If we take a raw signal with, for example, 1×10^6 samples and we choose $r = 10$ and $npviz = 500$, taking into account that each zoom level iteration will divide the number of samples of the signal by the given subsampling factor, the algorithm will iterate 4 times, until the subsampled signal has less than 500 samples. This example is represented in Table 3.1.

Table 3.1: Example of different zoom levels and the respective number of samples

Subsampling iterations	Zoom level	Number of samples
0	raw data	1×10^6
1	1	1×10^5
2	2	1×10^4
3	3	1×10^3
4	4	1×10^2

After creating the different zoom levels according to the signal length (N), the resampling factor (r) and the maximum number of points that can be used to represent the signal on each visualization level (*npviz*), the data structure creation routine stops.

The visual effect and data reducion that the described subsampling technique provides are shown on Figure 3.3. As is is visible, the same signal can be represented by a smaller amount of samples, with high visual resemblance and allowing a perfect perception of the signal's shape.

Thanks to the parameters that were already mentioned, it is possible to see the main morphology of the signal (given by the mean), where the signal has passed through stronger variations (indicated by the standard deviation shaded area), and the extreme values of the signal (represented by the maximum and minimum lines).

Besides the creation of the "acquisition data" and "biosignals" sections of the new data

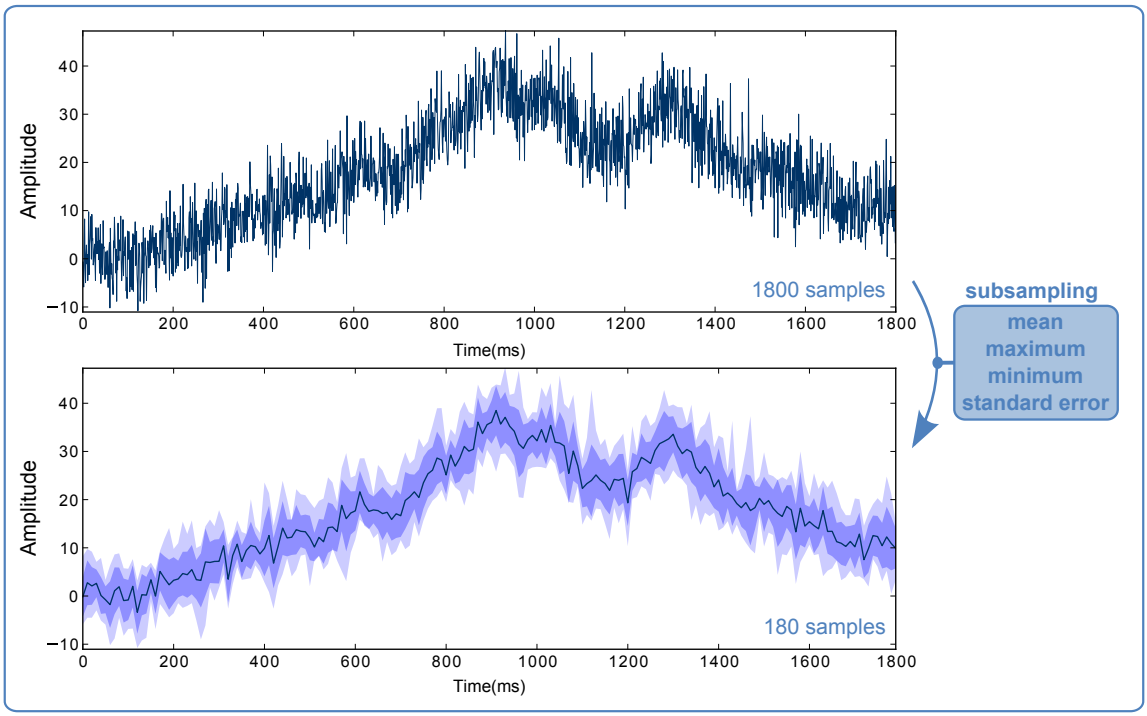

Figure 3.3: Illustration of the effect produced by a subsampling operation over a random signal (adimensional amplitude).

structure, the possibility of saving important processed data is provided. In the next section this issue is explored.

3.3.3 Processed data saving

Besides data visualization and analysis, biosignals data processing plays a very important role in the understanding of patients state and its evolution. After processing biosignals, an important task is to save the data obtained with the processing operations, so it can be accessed *a posteriori*, without running the processing algorithms again.

The developed data structure has a section to save processed biosignals data, such as the ECG peaks detected by a QRS peak detection algorithm. A group called ”processed data” is available in the data structure, and inside, several groups can be saved (following the example, the processed data group could be called ”ECG peaks” and in this group a dataset with the detected peaks could be saved for posterior access).

Such as long-term biosignals visualization, processing this specific type of biosignals is challenging. If standard (non-parallel) processing algorithms are used, processing biosignals with very large sizes becomes unfeasible due to memory errors. In this context, and to bridge this problem, this work presents a new approach regarding the processing of long-term biosig-

nals: parallel computing algorithms. In Chapter 5 this concept will be described with more detail.

The new data format provides a broader approach to the visualization and processing of biosignals, allowing the user to save the results of the biosignals processing tasks besides the raw data from the acquisition and other information about the subject or the recorded signals.

Chapter 4

Long-term biosignals visualization

In this chapter, the developed signal visualization methods created for this study are described. A tool to visualize long-term biosignals was implemented, based on the new data structure introduced in Chapter 3.

4.1 Visualization tool purpose

The main idea of the visualization tool for long-term biosignals is to allow a general overview of the entire signal in the first instance, giving the user the possibility to zoom in and out to a specific time window, showing more or less detail.

This follows the concept of multiple levels of visualization, depicted in the previous chapter. Displaying an entire long-term biosignal on the computer monitor would not give much information to the user and would exceed the capabilities of the visualization device, since this means trying to draw millions of points in a screen with only some thousands of available pixels. Due to this difficulties, a tool to enable the fast assessment of biosignals morphology, allowing the inspection of interesting portions of the signal with a greater level of detail was designed and developed in the present work.

This approach is comparable to a web mapping service; however, instead of viewing images of the Earth's surface it enables the visualization of large electrophysiological signals.

A client-server model was selected for the implementation of the visualization tool, given that data transmission via Internet is getting more common every day. Thus, a web environment application was developed, giving the tool higher portability. A client-server model, using Python as a way to manage data from the long biosignals and Javascript and HTML (HyperText Markup Language) to create the visualization platform was implemented.

4.2 Visualization tool properties

The developed visualization tool enables the visualization of long-term biosignals which were converted to the biosignals data structure that was already explained in Chapter 3. This biosignal visualization tool gives the possibility to open multiple channels (corresponding to different biosignals) from a recording.

The initial display shows the entire signals that are being visualized. This is done by drawing the outermost, or by other means, the lowest zoom level, thus the one with less detailed information about the signals.

The developed visualization tool gives the user the capability to analyze biosignals in an easy and prompt way. The most important operations are the zooming and panning actions; these actions allow an efficient navigation through the signals being visualized, either by selecting smaller signal portions to be viewed or by moving through the temporal axis (accessing different values of the signal in different time intervals). Zooming and panning operations promote the assessment of the evolution in signal's shape through time.

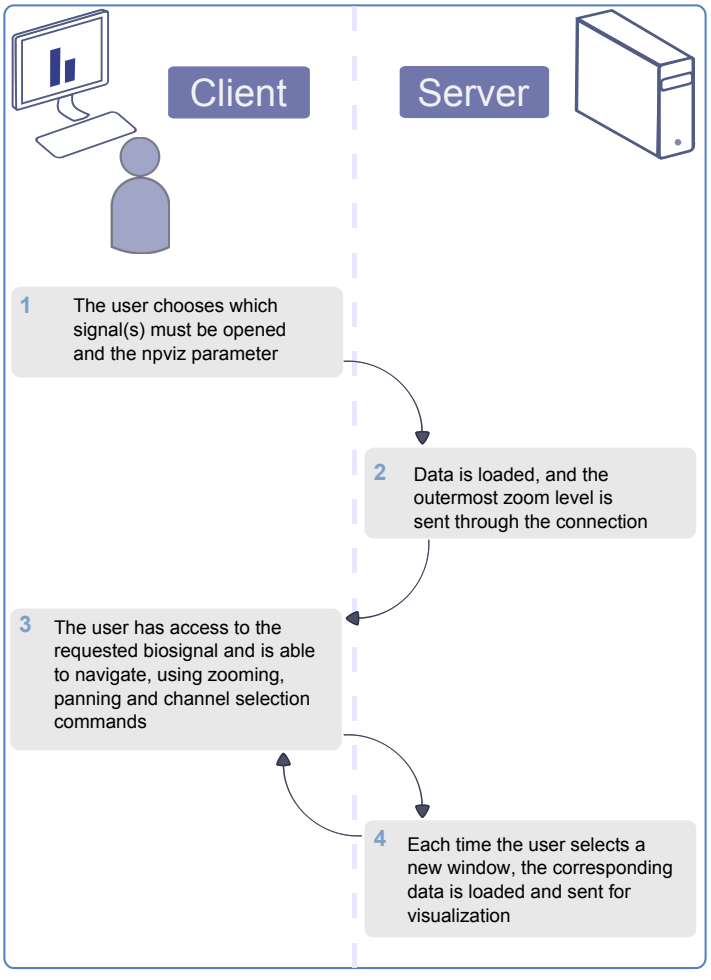

Figure 4.1: Developed client-server model for the biosignal visualization interface.

The developed client-server model, which allows an *user friendly* interface for biosignals visualization is shown in Figure 4.1, presenting the different steps of the developed client-server model for biosignals visualization. The first step represented in this figure involve the inputs that the user must give in order to view the desired biosignals. In the first step, besides choosing a recording to be explored, the user chooses the *npviz* parameter. This parameter defines the maximum number of points that the visualization tool can use to represent the signals on each zoom level. The second step one comprises the data loading process that precedes data displaying. Steps 3 and 4 represent the signal navigation operations.

The developed visualization tool offers a set of possible operations that enhance the signal navigation. The options provided as well as other interactive features are described below:

- **Zoom:** Performing zoom operations is possible using ±keys (+ key produces a 2× zoom in and − key allows a 2× zoom out);
- **Pan:** Panning through the signal is possible by pressing the arrow keys of the keyboard. This operation allows to go forward and backward in time, exploring the temporal evolution of the signal;
- **Expand channel:** By pressing each channel being visualized with a double click, an expanded view of the selected channel will appear. The ”normal” view can be reset by pressing the same key.
- **Select displayed channels:** it is possible to select the channels to be visualized *on-the-fly*, by pressing the key that corresponds to the desired channel. If channel 1 is being shown, pressing ”1” will make it fade out; otherwise, if channel 1 is not already being shown, pressing this key will make it show up.
- **Select time window:** this option is available by dragging the time window borders in a overview window presented in the visualization tool (dragging these borders allows a precise selection of the selected time window). Moving this time window forward or backwards allows the user to explore the signal in different intervals of time with the same length.

When the user presses the navigation keys, the signal being shown is updated to the new position selected. Using one of these signal exploring operations, a new specific time-window of the signal to be displayed is asked in each iteration. With that request, a command is sent through the client-server connection, which returns the new data correspondent to the

selected interval. In the case of the zooming operations, the selected time window to zoom could belong to the same zoom level or to another.

As the user explores the signal, navigating through the different visualization levels, the tool calculates the correct zoom level according to the time window that is being selected, gets data from the data structure, and displays it.

The correspondence between the selected time window and the zoom level that should be accessed, depending on the *npviz* parameter and the resampling factor with which the zoom levels were calculated, was defined as presented in in table 4.1.

Table 4.1: Zoom level selection according to the size of the selected window to be viewed

npviz (samples)	Resampling factor	Selected window (samples)	Zoom level
1000	10	[0,1000]]1000, 10000]]10000,100000]	0 (Raw) 1 2
2000	20	[0, 2000]]2000, 40000]]40000,800000]	0 (Raw) 1 2

As it is shown in the first example given in table 4.1, if the *npviz* parameter is set up to be 1000, the raw data (zoom level 0) will only be accessed for time windows with less than 1000 samples (if the sampling frequency is 1000Hz, this is equivalent to the visualization of less than 1 second of the signal). If, in the other hand, the user is trying to visualize more than 1000 samples and less than 10000, the second zoom level should be accessed. This is because in the second zoom level, 10000 samples of the original signal are represented by 1000 points due to the resampling effect, explained in Chapter 3.3.2.

An expression for the calculation of the correct zoom level, z, corresponding to each selected zoom window, was defined taking the examples of data window selections and the correspondent zoom level, shown in table 4.1. The obtained expression is shown in equation 4.1, where N is the selected number of points, r is the resampling, *npviz* is the maximum number of points to be displayed and $\lfloor x \rfloor$ represents the largest integer lower than x.

$$
\begin{aligned}
z &= \left\lfloor \left(\frac{log(N)}{log(r)} - \frac{log(npviz)}{log(r)} + 1 \right) \right\rfloor \\
&= \left\lfloor \left(\frac{log(N) - log(npviz)}{log(r)} + 1 \right) \right\rfloor \\
&= \left\lfloor \left(\frac{log(\frac{N}{npviz})}{log(r)} + 1 \right) \right\rfloor \\
&= \left\lfloor \left(log_r \left(\frac{N}{npviz} \right) + 1 \right) \right\rfloor
\end{aligned} \tag{4.1}
$$

This zoom level calculation formula, given in equation 4.1 works correctly for values of N that comply with the expression:

$$
N > \frac{npviz}{resampling} \tag{4.2}
$$

For the values of N that do not fulfill the condition given above, the visualization algorithm assumes that the user is trying to access the raw data, since this situation occurs when small time windows are selected.

This web environment tool lets the user explore signals using its different zoom levels and there are two drawing stages in the process:

- **Preview:** This is the first drawing stage of the visualization tool, on which the signals' informations to be drawn are only maximum and minimum (aiming for a fast and representative overview). The information shown in the preview allows the perception of the "envelope" on which the signal values are constrained;

- **Detailed view:**

  The detailed view must appear after the preview with an adequate delay. This delay was defined to be 300 miliseconds. Thus, 0.3 seconds after the first drawing stage, the second step draws the signal's mean, as well as the maximum, minimum, and the error shade (defined by mean±standard deviation error) with the intention of showing all the signals' characteristics. The delay between the preview and the detailed view was chosen taking two important requirements into account: it should allow the user to correctly distinguish the two drawing stages and it should be enough so that the the visualization tool could access data in this time gap.

The existence of two drawing steps allows the user to have a fast view of the signal's shape (represented by the maximum and minimum lines) on each interaction. This phased

drawing technique enables a faster navigation through the signal, since the user can ask for new time windows to be displayed almost instantly. The detailed data is shown only when the viewer stops in a specific time window, providing the user with the complete information about the signal being observed.

When the user reaches the raw data level there are no statistical parameters of the biosignal and the visuzlization presentes the raw original signal.

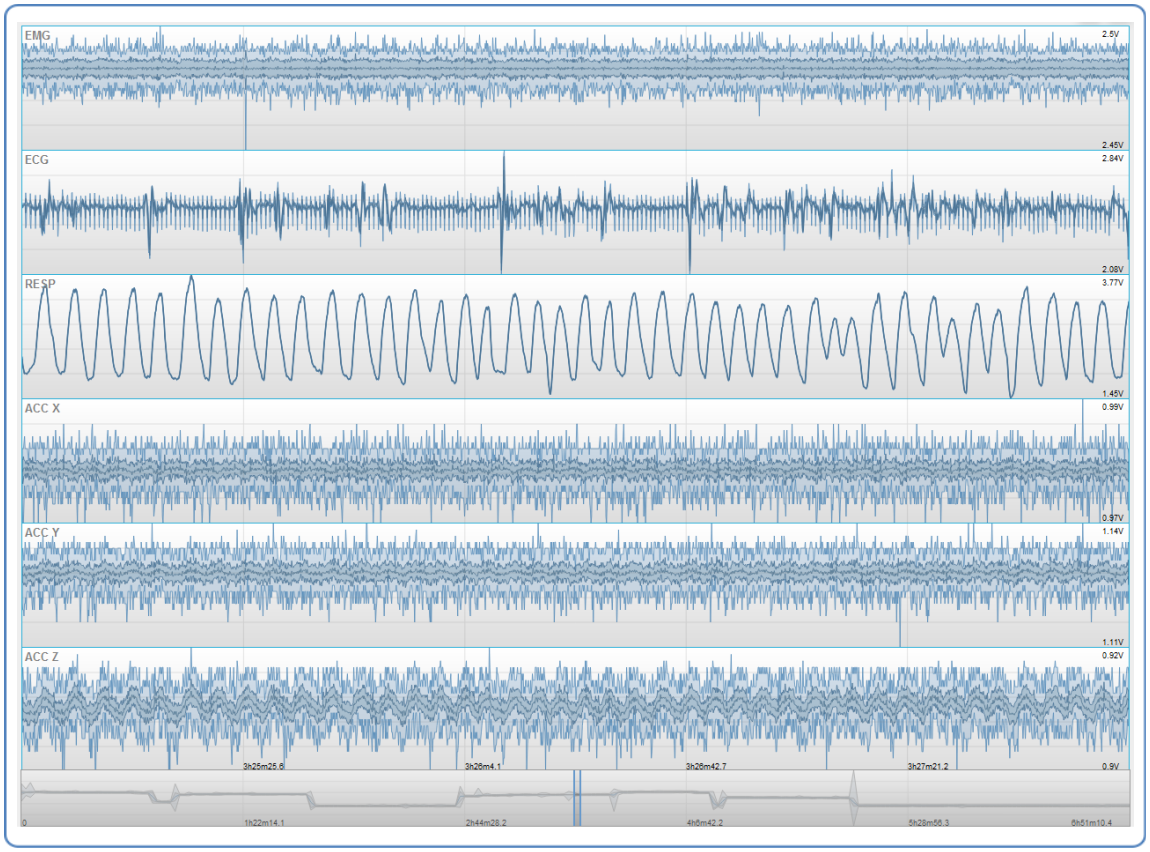

Figure 4.2: Developed biosignal visualization tool.

Figure 4.2 shows the aspect of the designed visualization tool. Six channels (EMG, ECG, Respiration and the three accelerometer components - x, y and z) are visible. At the bottom of the image, there is an overview window with the entire signal drawn, and a rectangle indicating the current time window selected.

Besides identifying the current portion of the signal that is being visualized, the overview window enables the user to select precise time windows in the signal to be displayed in detail. This overview window enables the user to understand "where" is the signal being explored, i.e., identifies the time interval of the signal that is currently selected, and also enables this time interval to be moved. Moving the time window allows the selection of a fixed lenght window to visualize different sections of a signal (for example, the user is able to select a ten

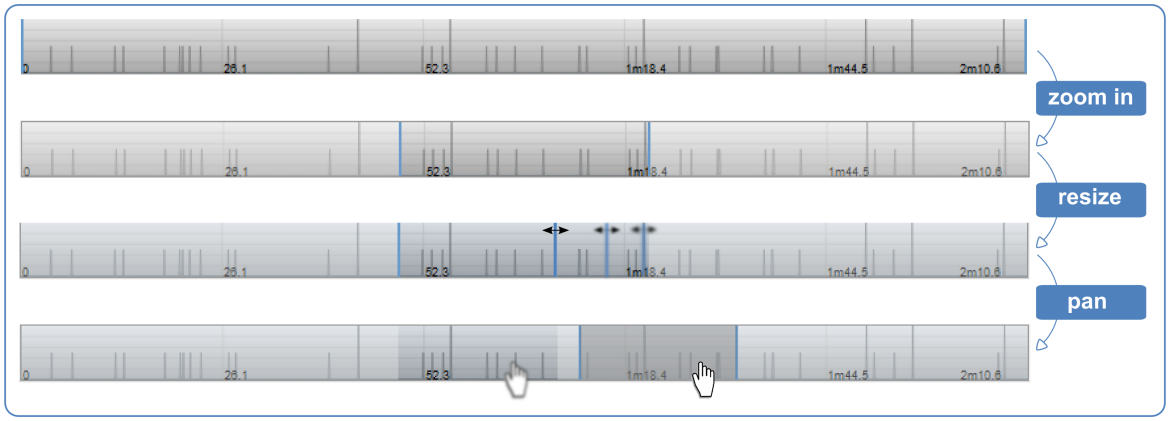

Figure 4.3: Representation of the functionalities featured by the overview window of the visualization tool.

seconds visualization window, that shows the signal being explored from the instant 1'20" until the instant 1'30" and then move this window forward, to visualize the same signal from the instant 2'20" to 2'30"). The overview window of the developed visualization tool is shown with more detail in Figure 4.3.

Using the overview window or the zooming and panning options enables a simple and fast signal navigation. An outlook of the possibility to visualize signals through different detail levels is shown in Figure 4.4. The evolution in the visual perception of a signal using different detail levels of visualization is demonstrated. As the selected part of the signal shrinks, the amount of data gets shorter and the detail level increases. In the top (a) the visualization of an entire signal with more than 6 hours is provided. The below images (b,c,d) show the effects produced by successive zooming in operations, with an increase in the level of detail visualized. The bottom image (d) allows the visualization of a short window of the biosignal, in which only four ECG QRS peaks are visible.

Besides the effect of the existence of different zoom levels on the visualization experience, the *npviz* is also an important parameter that changes the way how the visualization is undertaken. When this parameter is set to a small value, the visualization tool displays detailed information later than when it set to larger values. As an example, if the user is trying to visualize a segment of a biosignal and the *npviz* parameter is defined to be 1000, the detail in the displayed information will be less than it would be with $npviz = 3000$. Figure 4.5 exhibits the differences in the way how the same biosignals are displayed (for the same time window) using two distinct *npviz* values (1000 points for the upper image and 2000 for the lower).

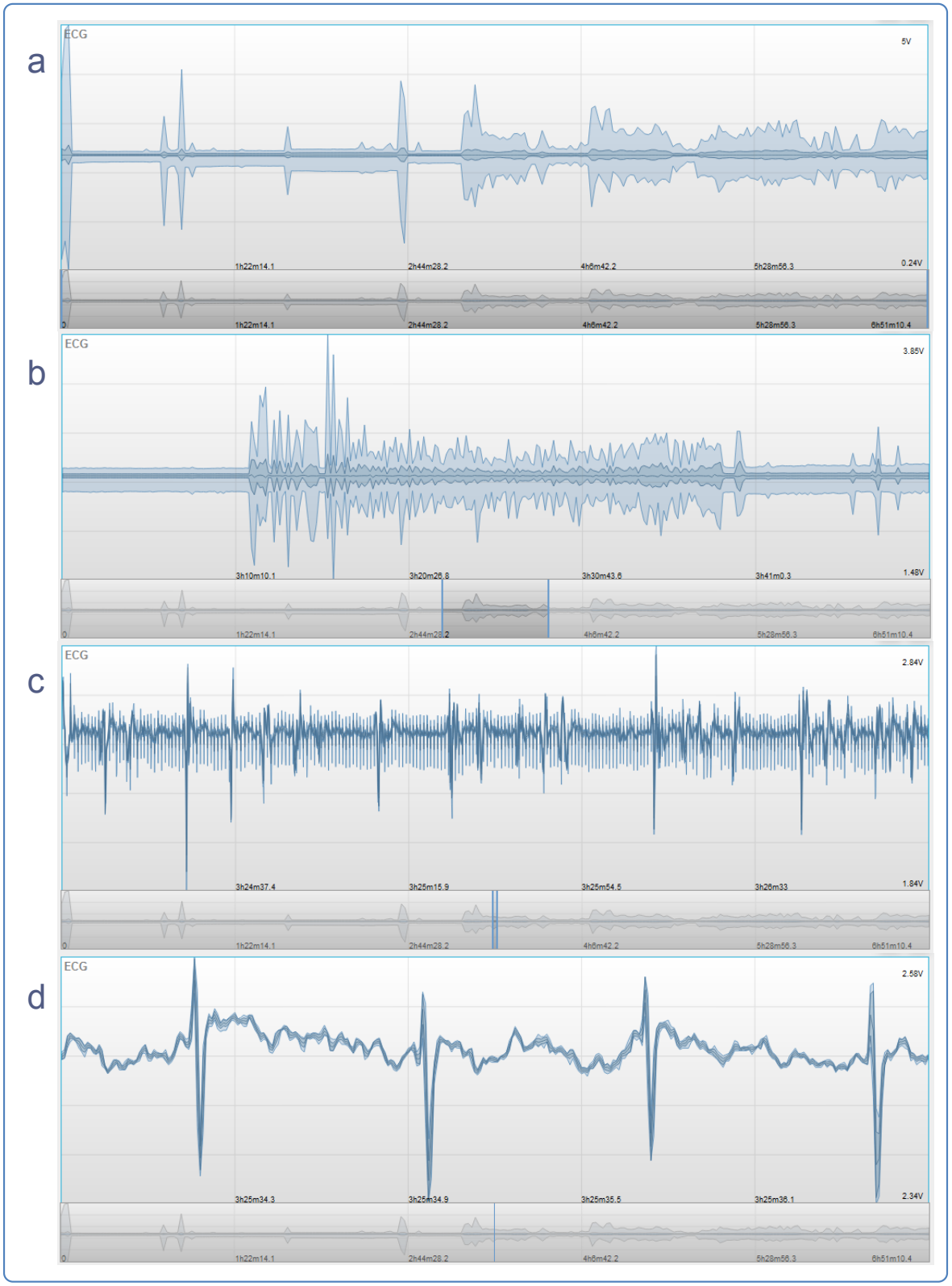

Figure 4.4: Perspective of the evolution of signal visualization according to the selected zoom window.

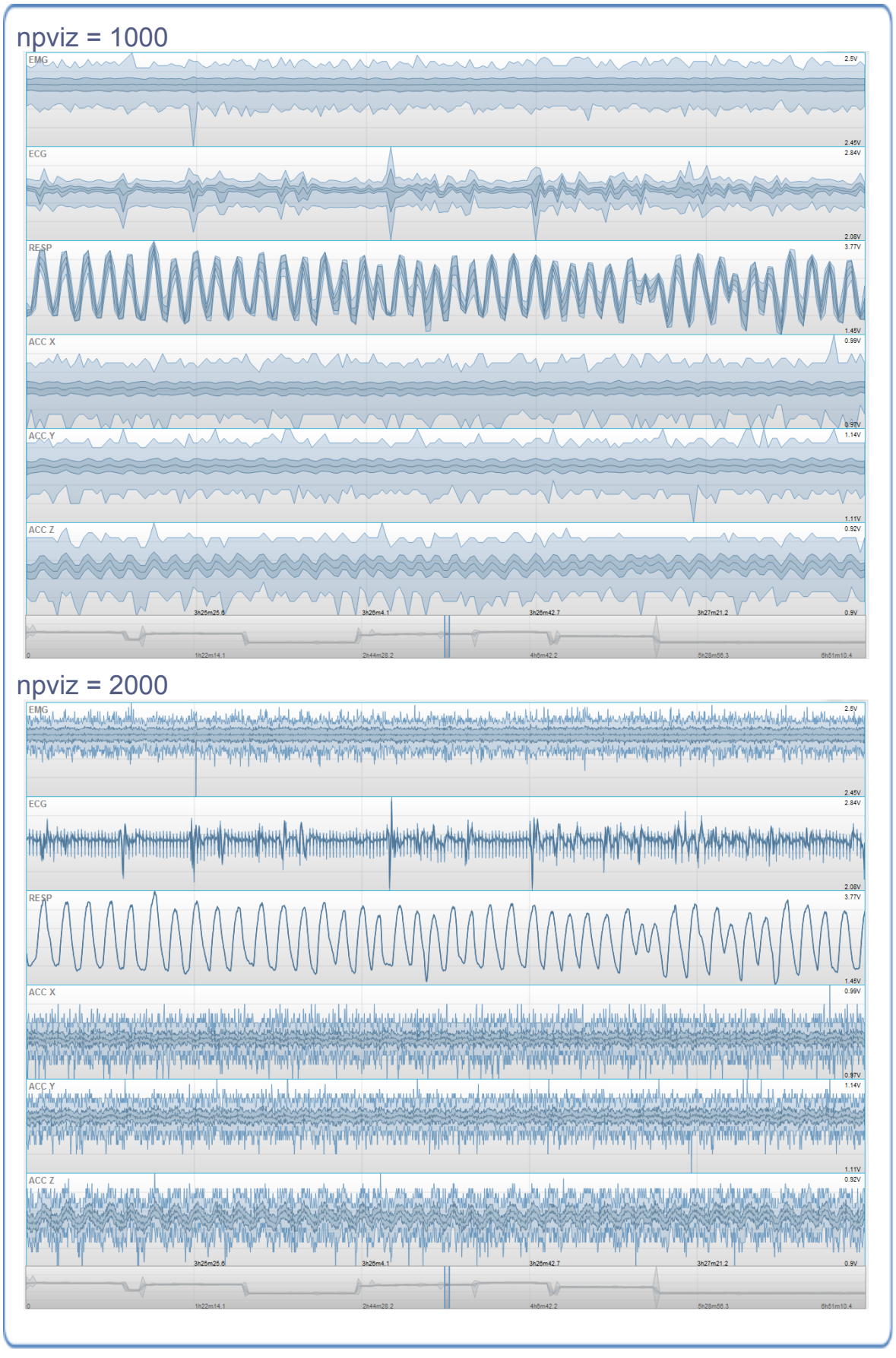

Figure 4.5: Effect of the *npviz* parameter in the visualization.

A new biosignal visualization concept was presented in this chapter. This concept enables the user to explore and analyze long-term biosignals, but its advantages are not limited to large sized signals. Besides providing the possibility to open large datasets, the developed tool allows several operations, such as zooming to multiple visualization levels, panning, and other interactive features.

Beyond signal visualization and analysis, signal processing is also very important in order to assure that relevant information is extracted from biosignals. This information is the basis for the comprehension of the patients's state, thus a correct identification of the ocurring physiological events is a demand.

Chapter 5 presents an approach to parallel processing apllied to long-term biosignals, in an attempt to overcome processing difficulties related to the signal sizes.

Chapter 5

Long-term biosignals processing

Besides the problems related to visualization, long-term biosignals also need different approaches regarding signal processing.

Taking into account the sizes of long-term biosignals (which can reach several hours or even days), a parallel computing solution was implemented. This chapter exposes the designed and implemented method for long-term biosignals processing. The concepts of "map" and "reduce" are presented and its general application in algorithms for biosignals processing is explored. Concluding this chapter, we present a specific application of this parallel processing concept to ECG peaks detection.

5.1 The `MapReduce` algorithm

Since the goal is to process very large datasets, standard processing algorithms cannot support the sizes of signals with huge sizes. This problems arise from the impossibility of loading huge size biosignals to the computer's memory for processing. In this case, the input for the processing algorithms must not be the entire signal. In order to overcome the signal size problems, a parallel processing solution was designed and is presented in the next section.

5.1.1 Overview

Parallel computing allows the division of a long task in several simpler subtasks, as it is described in chapter 2.3.

Since the sizes of these signals are extremely large, running a processing algorithm using a single CPU (Central Processing Unit) is not a plausible solution. Therefore, an algorithm that can run using multiple CPU's, was designed. This algorithm breaks a problem into

discrete parts that can be solved concurrently (instructions from each part are executed simultaneously on different processing units).

The implemented parallel processing algorithm is based on a ”map” and ”reduce” process. On this process, a long signal is divided in parts (the input signal is mapped in several portions with fixed length), that are processed independently - the processing function is applied to each input. After processing all the separated parts, the results are merged (”reduce” step).

With the designed approach, an answer to the long-term biosignal processing problem with a parallel processing method is provided. This implementation enables a large scale task distribution architecture, with a group of computers processing in parallel, each one of them with a small part of the signal (see Figure 5.1).

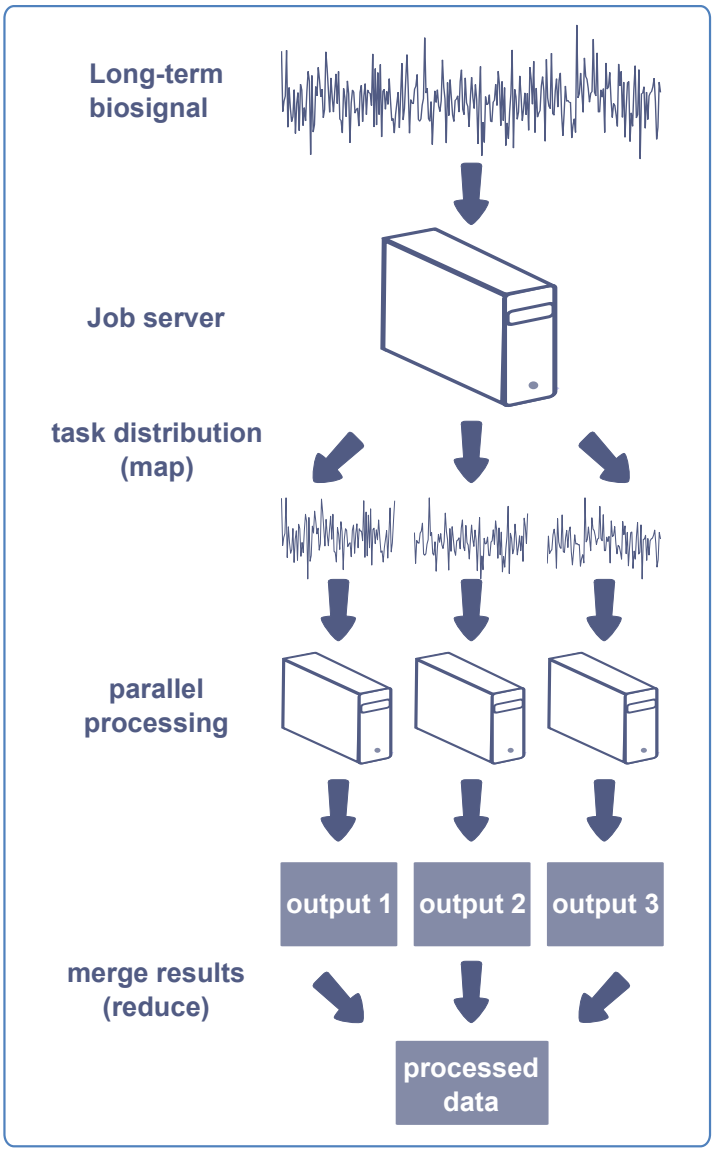

Figure 5.1: Representation of the parallel processing concept applied to long-term biosignals.

The objective is to map the signal in intervals with fixed length and process each mapped interval, using algorithms that work efficiently with shorter signals. After processing each interval, the results are merged together. If the operation is working correctly, merged data should be the same as the data that the detection algorithm would retrieve in case it could receive the entire signal as input.

5.1.2 Algorithm design

The developed processing algorithm is presented below.

Hereafter, the discrete biosignal to be processed, X, described in equation 5.1 is considered, where k is an integer value that represents the signal's number of samples.

$$X = \{x_1, x_2, \ldots, x_k\} \tag{5.1}$$

The processing operation can be represented by equation 5.2.

$$Y = F(X) \tag{5.2}$$

The operator F receives an entire biosignal (X) as input and returns Y. Since the input signal might be very long, the need to map it in several smaller regions to be processed separately becomes imperative.

However, X can be splitted in subgroups with a fixed number of samples - L. The signal mapper is then a list of pairs that define the several subgroups (time intervals) to be processed separately. Let us call this list of pairs J. J is described on equation 5.3.

$$\begin{aligned} J = \{&(0, L), (L - v, L - v + L), \\ &(2L - 2v, 2L - 2v + L), \\ &\ldots, \\ &(mL - mv, mL - mv + L)\} \end{aligned} \tag{5.3}$$

with v being the number of samples to be overlapped, and m an integer.

Selecting the signal (X) in the time intervals defined by J, the signal will be mapped. Each subsignal can be defined by equation 5.4.

$$\begin{aligned} x^0 &= \{x_0, \ldots, x_L\} \\ x^1 &= \{x_{L-v}, \ldots, x_{L-v+L}\} \\ \cdots x^m &= \{x_{mL-mv}, \ldots, x_{mL-mv+L}\} \end{aligned} \tag{5.4}$$

In the borders where the signal is splitted to be processed, there might occur some problems. Consider a function that needs to analyse a fixed length "region" in order to localize a specific event. If the signal is splitted in this "region", the event may not be detected, and relevant information might be lost due to this splitting operation. In order to overcome this kind of questions, the implemented algorithm has an overlapping number of samples, v (everytime the algorithm runs for a selected time window, there is a number of samples from the end of the last time window that is considered in the beggining of the actual one).

After mapping the signal, the processing algorithm is applied to the various intervals mapped from the signal. Giving each interval to the input of the processing routine, we will obtain a group of outputs, that can be defined by equation 5.5.

$$y^j = f(x^j) \tag{5.5}$$

On this last step, in which j represents a subprocessing group, the results from the independent separate processing tasks are merged together, in the "reduce" operation. The function that correctly joins together the outputs from the subprocessing tasks is denoted by G, and the final result is given by equation 5.6.

$$Y = G(y^0, y^1, \ldots) \tag{5.6}$$

Parallel processing in computers enables to divide a long operation in several smaller tasks that can be carried out by different computer core processors. Considering a processing operation with a fixed start time (T_s) , that takes a time T to be carried out by one processor and that the processing is going to be divided by N_s processors, the total parallel processing time (T_p) will be given by equation 5.7.

$$T_p = T_s + \frac{T}{N_s} \times (1 + Ov) \times 2 \tag{5.7}$$

On equation 5.7, the overlap (Ov) is defined by the expression given in 5.8, where v is the overlapping number of samples and N_{slice} is the number of samples of each processing slice.

$$O_v = \frac{v}{N_{slice}} \tag{5.8}$$

Since the existence of the overlap means that there are samples being processed in two different subtasks, larger overlaps cause the processing to last longer, while smaller overlaps lead to shorter processing duration. However, the overlap not only influences the processing time but also the processing efficiency. When the overlap is not big enough, for some processing functions there is the danger of occurring processing errors. An example is the application of a causal filter to long-term biosignals. These type of filters make use of past samples, thus an overlap that enables passing all the necessary information for the filter to be applied is required.

5.2 Application: ECG processing algorithm

An example of a mature processing algorithm [35], which does not work properly taking long-term biosignals as input, was adapted to this type of signals: the peak detector to be applied on ECG signals.

A representation of the ECG peak detection algorithm is made in Figure 5.2.

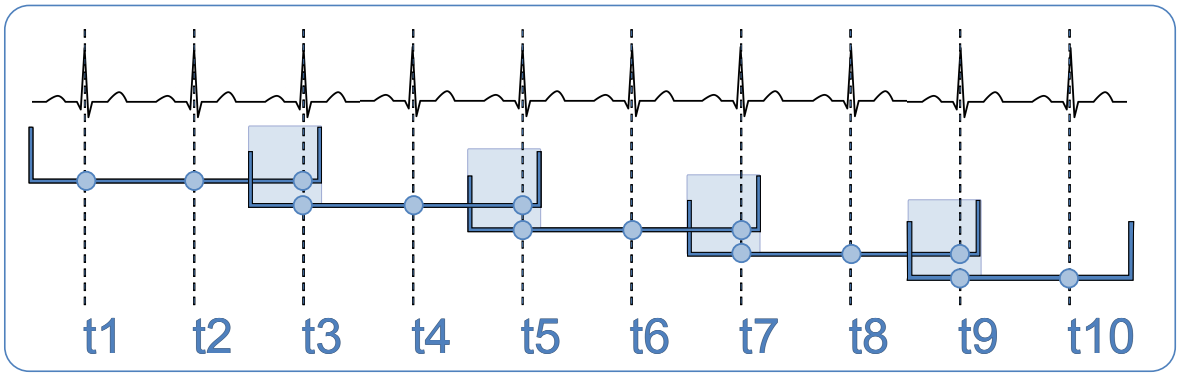

Figure 5.2: Representation of the processing algorithm for the ECG peaks detection.

The times $(t_1, t_2, ..., t_{10})$ in Figure 5.2 indicate the peaks detected by the algorithm, while the shaded areas represent the overlapping of the algorithm (where two processing windows intersect).

With the mapping and reducing technique implementation, the large ECG signals are mapped in a series of fixed time intervals and the processing function takes this intervals as input (see Figure 5.1). For each interval, the processing results (detected ECG QRS

peaks) are calculated and the outputs of the group of "subprocessing" steps are merged, thus resulting in the final result, which is an array with the ECG peaks detected from the input ECG signal.

For the peak detection function in ECG signals, the considered overlap was defined by making use of physiological information. Since the normal duration of a QRS complex varies in the interval $0.06 - 0.10s$ [20], the overlap was set to be 200 milliseconds, twice as large as the upper limit of the QRS duration variation interval.

Chapter 6

Performance Evaluation

In this chapter, a performance evaluation of the developed visualization and processing tools for long-term biosignals was undertaken. In order to test the visualizing and processing algorithms several types of biosignals have been acquired in long-term recordings. A case study of the application of the developed tools to real long-term biosignals is presented for a better understanding of its potential.

The developed tools were tested in order to evaluate their performance, using the acquired biosignals. The creation of the multiple visualization levels was monitored; the performance evaluation of the conversion algorithm is presented and discussed below.

6.1 Data structure creation evaluation

All the performance tests were made with the same computer - a Intel Core i7 720QM with a 1.60GHz processor.

In all the presented results, it should be noted that a file with a size of, for example, 346,8 MB is equivalent to have a biosignal recording obtained in a 21 hours long acquisition with a sampling frequency of $1000Hz$.

Table 6.1: Data structure creation times.

text file size (MB)	Conversion times (s)	
	raw data	zoom levels
346,8	41	85
435,1	50	104
954,6	91	217
1.021,2	109	234
1.297,3	157	357

Table 6.1 presents the performance of the developed conversion tools that transform the data from the acquiring data format to the new data format dedicated to long-term biosignals visualization and processing. As one can see, the conversion process is not instantaneous, being a time consuming task in signal analysis.

Table 6.2: Load times for .txt and .h5 files

file size (MB)	Load times (s)	
	.h5 file	.txt file
14	0.01	6.35
144	0.04	64.33
347	0.57	349.33
424	0.79	(Memory Error)

However, the benefits of this conversion step are evidenced by the results presented in Table 6.2; here, the focus goes to the differences in time consuming when loading data from a file in the *.txt format or in the *.h5 file format, which was the chosen file format to be used in the developed biosignals database architecture.

The presented results demonstrate that opening text files with biosignal acquisitions of several hours by loading them on python would take a very long time or even cause a memory error, the presented results (see table 6.2) are an evidence of the benefits of the developed data structure on data accessing/visualization. Since the data conversion only has to be carried out once, this benefits are even more easily observable.

Besides the data files creation performance, the visualization tool capabilities were also tested. The results of such tests are available in the next section.

6.2 Visualization tool evaluation

The performance of the visualization tool is independent of the type and size of the signal being visualized as well as of the zoom level on which the user is ”navigating” with the developed tool.

Operations like zooming and panning over long-term biosignals, that take several seconds using python visualization methods, are practically instantaneous using the developed tools.

Since the conversion only has to be carried out once, and acessing data from the new structure takes only miliseconds, it is possible to understand the advantages brought by the presented tools.

After assessing the visualization tool performance, the developed processing algorithm performance is going to be scrutinized in the next section.

6.3 Processing tool evaluation

Performance tests were carried out with the processing tools, so it could be confirmed if the results of the new processing algorithms were obtained faster than using the standard algorithm.

The processing results of the application of the developed MapReduce algorithm for the detection of ECG peaks were compared with the output of the standard algorithms (which do not use parallel processing), in order to analyze the efficiency of the new tools. The developed processing algorithm works correctly. This fact was expected, since the two processing algorithms use the same root processing algorithm. The main difference between them is that the parallel processing algorithm processes smaller inputs than the literature method.

The performance tests consisted of processing an entire signal by the non-parallel processing algorithm and by the ”map” and ”reduce” algorithm using an ECG signal with 602613 samples (approximately 10 minutes because the sampling frequency of the acquiring device is 1000 Hz). The results are shown in Table 6.3. In this specific case it was expected that the parallel processing algorithm would spend less time processing the signal than the non-parallel one.

Table 6.3: Comparison of the ECG processing times for the parallel and standard algorithms (applied to a 10 minutes signal)

Processing algorithm	Time consumed (s)
non-parallel	10.7
parallel (2 processors)	6.46
parallel (3 processors)	5.61
parallel (4 processors)	4.99

The performance results shown in Table 6.3 allow the perception of speed-up introduced by the use of parallel processing. However, the processing times were obtained for a small testing signal in order to allow using the non-parallel algorithm.

A different test was performed in order to better understand the influence of the utilization of multiple processors for parallel processing. A long-term biosignal with approximately 10 hours of duration. The results of this test are shown in Table 6.4.

The results presented in Table 6.4 evidence the speed improvement introduced by the developed parallel processing. This results are in compliance with the expected, since using a larger amount of computing resources should accelerate the processing process. However, these results do not show a linear increase in the speed of the parallel processing algorithm.

Table 6.4: Comparison of the ECG processing times for the parallel and standard algorithms (applied to a 10 hours signal)

Processing algorithm	Time consumed (s)
parallel (2 processors)	309
parallel (3 processors)	243
parallel (4 processors)	214

This fact might be due to the factors already referred, such as the algorithms starting time.

The presented and discussed results allow the perception of the developed tools' potential. In order to better understand performance of these tools when applied in real-life situations, the next section presents a case study with the implementation of the tools developed in this work.

6.4 Case Study

In this section a case study involving the application of the developed data structure, visualization and processing tools to real long-term biosignals is presented. [1]

6.4.1 Protocol

Figure 6.1 presents the three steps that composed this study. The first step of this study was the acquisition phase. A set of biomedical sensors were worn by the patients in specific anatomic regions. These sensors were connected to the bioPLUX wireless acquisition unit, which sent data via Bluetooth to a mobile phone, in which data was saved in a .txt file to be processed after acquisition.

The second step was the conversion of the .txt file to the developed data structure and the calculation of the zoom levels, which enabled the visualization of the acquired signals.

The third step was the signal processing. For that, the algorithm implemented in this work was used to extract important physiological parameters from an ECG signal.

Following, the different stages of this study are minutely described. This study represents an example of how to use the tools designed in this work.

Biosignal acquisition layout

Before acquiring biosignals, a set of electrophysiological signals to be acquired and sensor placements were defined in order to create an acquisition protocol that does not negatively

[1] This study was done in collaboration with wiCardioResp project [46], which is supported by the National Strategic Reference Framework program (NSRF-QREN). The signal acquisitions and the tests to the presented data structure were performed in collaboration with Hospital Santa Maria.

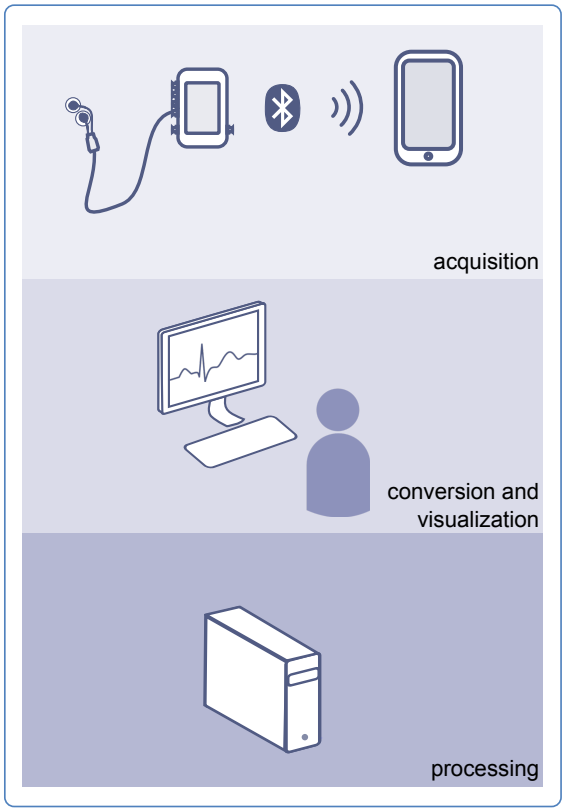

Figure 6.1: Representative scheme of the sequence of tasks done during this case study.

affect the quality of life of the population that was monitored. The different characteristics of the acquisition process are exposed below.

The population of this study was composed of three Amyotrophic Lateral Sclerosis (ALS) patients and four healthy people (the control group) that volunteered for the mentioned research project.

The acquisitions were carried out in the subjects' homes, and during one night sleep (each recording had the approximate duration of 8 hours).

These acquisitions aimed at the study of electrophysiological parameters that could indicate potential ALS crisis throughout the night . All the subjects, patients and the control group, were informed about the objectives of the research on which they were invited to participate, and gave their approval to proceed with the biosignals acquisitions.

The acquisition equipment that was used to acquire the biosignals was the bioPLUX research system, a wireless signal acquisition unit shown in Figure 6.2.

Figure 6.2: Acquisition device used during the biosignal recordings of this study.

Acquired signals

Several types of biosignals such as as Electromyography, Electrocardiography, Electrodermal activity, Acceleration and respiration were acquired. The way how the different sensors were displayed in the body of the patients to be monitored is represented in Figure 6.3.

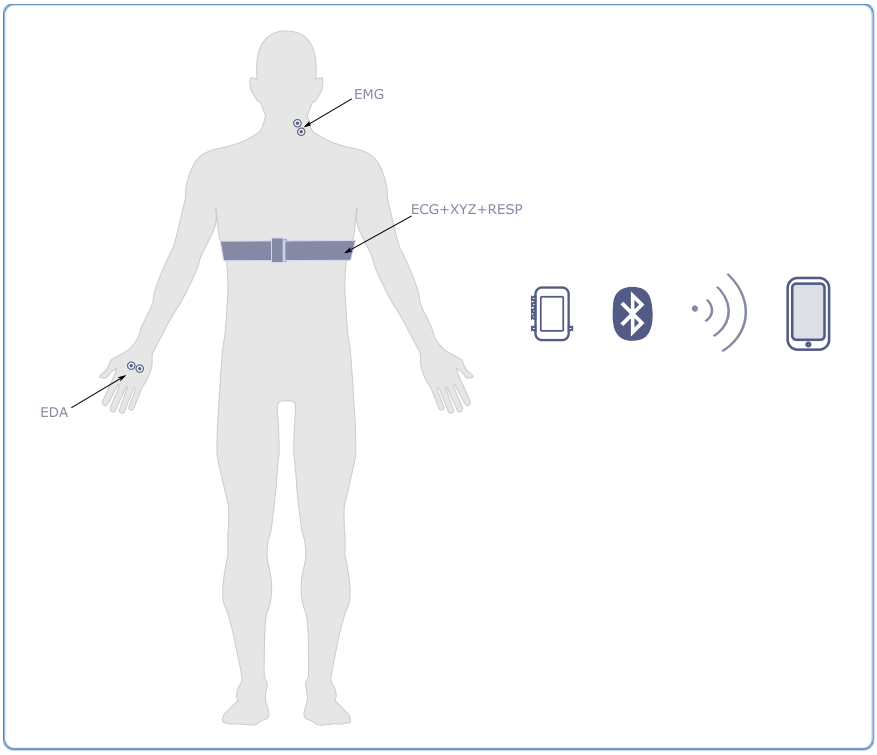

Figure 6.3: Layout of the acquisition setup.

The biosignal acquisitions were undertaken using an wireless acquisition equipment, that sent the recorded data to a smartphone by a bluetooth connection.

Given that three different biosignals to be acquired implied the placement of sensors in the chest area (ECG, Respiration and ACC), these sensors were integrated in a chest strap. Such configuration allowed not only the reduction of the interference in the signals to be acquired, but also increased the usability of the system and the patients' comfort. Thus, the display of the various sensors was made as it is shown in Figure 6.3. The EDA sensor was placed in the palm of the hand of the patient, the EMG sensor in the sternocleidomastoid muscle and the ECG, ACC and respiration sensors were placed in a chest band, as it was already said. Through this setup, the sweat signal, which is related to the sympathetic nervous system (SNS), was monitored, allowing the extraction of relevant events associated with the activity of this system; the EMG signal permitted the assessment of the muscular activations in the neck (potential crisis indicators), the ECG was used for heart rate extraction and the accelerometers monitored the patients' movements during their sleep (when abrupt, these may also indicate the ocurrence of a muscular crisis); finally, the respiration sensor's purpose was to extract the respiratory rate.

6.4.2 Biosignal Analysis

In spite of having biosignal recordings from 7 volunteers in this study, only one of them is going to be addressed in this section. Since this work deals with long-term biosignals, this section will only cover the analysis of the longest acquisition carried out. This recording has approximately 10 hours of duration and the size of the acquisition file (*.txt) is $1,3Gb$.

Before visualizing and processing the acquired biosignals, the data conversion and creation of the visualization levels were carried out.

The *.txt file with the acquired data was converted into the adopted (*.h5) format. For that, the conversion tool developed in this work and described in Chapter 3 was used. This conversion process took 557 seconds (approximately 10 minutes) to be concluded. The 10 minutes spent in the conversion included:

- 188 seconds for the raw data conversion;
- 369 seconds for the creation of the zoom levels.

After converting the signals to the developed data structure, they were analyzed with the designed visualization tool. The visualization *npviz* parameter was set to 2000 points; a

detailed assessment of the signals morpholophy was possible when a time window with 140 seconds of the signalwas selected. Since the sampling frequency of the acquired signal was $1000Hz$, this time interval is equivalent to 140000 samples. Using the zoom level calculation formula mentioned in Chapter 4, the zoom level corresponding to this time window is the second.

Since one of the acquired signals was the ECG, the parallel processing algorithm was applied to this signal. The algorithm took 213.8 seconds to run, using one computer with 4 active processing units and detected 45061 ECG R-peaks. This number of peaks (equal to the number of heartbeats in the same period) allows to calculate the mean heart rate, that in this case was 75 bpm (beats per minute).

Taking into account the time spent on each step of the implemented architecture and considering a continuous analysis of the acquired signals, from the conversion up to the signal processing an analysis of a long-term biosignal spent approximately 15 minutes, using the tools developed and implemented in this work.

Due to the acquired biosignals characteristics (signals that imply an huge amount of data to be saved, converted, visualized and processed) and considering that a physician needs to have more than one tool to carry out all these processes, 15 minutes is considered an good results result, given that some of these computing could be done even during the signal acquisiton phase that took 10 hours. The developed tools integrate all the necessary processes for the physician to make a fast and accurate analysis of the patients' signals. Furthermore, with the developed visualization tool, the physician can visualize an entire electrophysiological signal, being allowed to increase the zoom level to obtain more detail about a specific part of the biosignal. All these features are available in a user-friendly software.

Chapter 7

Conclusions

In this concluding chapter, a summary of the developed work, its general results and accomplishments will be presented. The future objectives for the continuation of the already achieved results are also highlighted in this chapter.

7.1 General achievements

The main goal of this thesis was to develop dedicated tools that enable long-term biosignals visualization and processing.

In order to accomplish the purposed objectives, the first step was to create a new data structure for biosignals that could provide the possibility to have a fast access to data. Besides this requirement, the designed data structure was projected to provide a multilevel visualization of data.

An algorithm to calculate multiple zoom levels of long-term biosignals was implemented. This algorithm produces subsampled levels, with the support of parameters enable to represent data using a smaller amount of samples: the mean, maximum, minimum and standard deviation. The mentioned algorithm creates the different detail levels according to two parameters: maximum number of samples that represent the signal in the outermost zoom level and the resampling factor.

The developed visualization tool is general, which means that any type of signal represented by a time series can be explored with it. Besides being general, the visualization tool allows the visualization of long-term biosignals acquired during several hours.

In addition to the developed work on the visualization levels and the biosignals visualization tool, a processing approach for long-term biosignals was also developed and implemented.

The problem of long-term biosignals processing resides in the impossibility of carrying out

the processing of an entire signal with this dimension. In order to bridge this problem, an implemented processing tool based on the concept of parallel processing was presented. With this concept, a `MapReduce` algorithm for independent multi task processing was created. This algorithm allows long-term biosignals to be efficiently processed by dividing the processing task in multiple feasible processing subtasks. The results of each subtask are then gathered, generating the processed data.

The performance of the developed tools was evaluated. Several long-term biosigals acquisitions were carried out in order to perform these performance tests.

The time consuming of data conversion and visualization levels creation was monitored. Regarding signal processing, the presented algorithm was applied to a specific case of ECG QRS peaks detection.

The conversion tool results indicated that it is faster (up to double speed) than the existing standard file converter regarding raw data conversion. The visualization tool also proven to be more efficient than standard methods, thanks to the implemented multi-level architecture and to fast data accessing. With respect to biosignal processing, the implemented parallel algorithm exhibited great results, with an increase in the processing speed when compared to the non-parallel algorithm.

Besides allowing the user to save the results of the parallel biosignal processing algorithms, saving the raw data from the acquisition and possible important information about the subject or the recorded signals, this format allows a new way of exploring biological data, in a fast and intuitive multi-level visualization of the biosignals, compatible with the web environment.

Considering standard formats for storage and exchange of biological and physical signals, it is possible to conclude that the new and innovative developed data structure allows a broader approach to the visualization and processing of biosignals (particularly for long-term biosignals).

7.2 Future work

The present thesis does not answer all the problems related to long-term biosignals visualization and processing. Some aspects can be improved, therefore a list of future work suggestions is presented below.

- **Processed data visualization:** The visualization tool is prepared to display biosignals data. However, graphical display of processed data as a way of link the extracted

properties of signals and signals. We intend to make it possible to visualize at the same time the signal and important processed data, such as the ECG peaks detected.

- **New processing algorithms adapted to long-term biosignals:** Other future goal is to develop new processing algorithms adapted to long-term biosignals, such as the heart rate variability (HRV), since it's parameters are of great importance in clinical cases that need long-term monitoring, such as neuromuscular diseases.

- **Automatic calculation of the overlap:** a necessary improvement to the processing algorithm is an automatic calculation of the indicated number of overlapping samples to be considered for each processing operation. This will remove one input parameter from the processing algorithm.

The growing need to monitor patients, particularly in a long-term perspective leads to the obligation of visualizing and processing very large biosignals. Patients in ambient assisted living (AAL) are an example of the growing urgency of developing tools that allow a correct and prompt tracking of the health state and its evolution.

Due to this demands, dedicated tools for long-term biosignal analysis were developed. Since biosignal analysis and processing is a promising area in medicine, sports and research, the opportunity to give a contribute with innovating techniques for the evolution of this field was a very gratifying and enriching experience.

Bibliography

[1] Bing Maps [online] available at: http://www.bing.com/maps/ [accessed 17 october 2011].

[2] J. Aach and G. M. Church. Aligning gene expression time series with time warping algorithms. *Bioinformatics*, 17(6), 2001.

[3] J. An, Z. Wu, H. Chen, X. Lu, and H. Duan. Level of Detail Navigation and Visualization of Electronic Health Records. *Healthcare Informatics*, (Bmei):2516–2519, 2010.

[4] K. Asanovic, R. Bodik, J. Demmel, T. Keaveny, K. Keutzer, J. Kubiatowicz, N. Morgan, D. Patterson, K. Sen, J. Wawrzynek, D. Wessel, and K. Yelick. A view of the parallel computing landscape. *Commun. ACM*, 52:56–67, October 2009.

[5] B. Barney. Introduction to parallel processing [online] available at: https://computing.llnl.gov/tutorials/parallel_comp/ [accessed 17 october 2011], 2011.

[6] J. Bronzino. *The biomedical engineering handbook*. Number vol. 1 in The Biomedical Engineering Handbook. CRC Press, 2000.

[7] Y. Chae. Data mining approach to policy analysis in a health insurance domain. *International Journal of Medical Informatics*, 62(2-3):103–111, July 2001.

[8] P. Compieta, S. D. Martino, M. Bertolotto, F. Ferrucci, and T. Kechadi. Exploratory spatio-temporal data mining and visualization. *Journal of Visual Languages Computing*, 18(3):255–279, 2007.

[9] J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. *Commun. ACM*, 51:107–113, January 2008.

[10] Distributed and Parallel Processing using WCF. Distributed and parallel processing using wcf [online] available at http://www.codeproject.com/kb/wcf/wcfparallelprocessing.aspx [accessed 17 october 2011], 2011.

[11] G. Earth. GoogleEarth [online] available at: http://www.google.com/intl/en/earth/index.html [accessed 17 october 2011]. http://www.google.com/intl/en/earth/index.html, 2011.

[12] EDF. EDF - European Data Format [online] Available at: http://www.edfplus.info/ [Accessed 5 September 2011], 2007.

[13] J. Ekanayake, S. Pallickara, and G. Fox. MapReduce for Data Intensive Scientific Analyses. In *Proceedings of the 2008 Fourth IEEE International Conference on eScience*, pages 277–284, Washington, DC, USA, 2008. IEEE Computer Society.

[14] J. Enderle, J. Bronzino, and S. Blanchard. *Introduction to biomedical engineering*. Academic Press series in biomedical engineering. Elsevier Academic Press, 2005.

[15] U. Fayyad, G. Piatetsky-shapiro, and P. Smyth. Knowledge Discovery and Data Mining: Towards a Unifying Framework. pages 82–88. AAAI Press, 1996.

[16] D. Flanagan. *JavaScript: the definitive guide*. Definitive Guide Series. O'Reilly, 2006.

[17] H5PY. h5py - a python interface to the hdf5 library [online] available at: http://code.google.com/p/h5py/ [accessed 17 september 2011], 2011.

[18] HDF group. HDF5 - HDF group [online] Available at:http://www.hdfgroup.org/HDF5/ [Accessed 5 September 2011], 2007.

[19] J. Hunter and D. Dale. The Matplotlib User′s Guide.

[20] R. Istepanian, S. Laxminarayan, and C. Pattichis. *M-health: emerging mobile health systems*. Topics in Biomedical Engineering. Springer, 2006.

[21] H. Kayyali, S. Weimer, C. Frederick, C. Martin, D. Basa, J. Juguilon, and F. Jugilioni. Remotely attended home monitoring of sleep disorders. *Telemed J E Health*, 14(4):371–4, 2008.

[22] D. A. Keim. Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics*, 8, January 2002.

[23] B. Kemp and J. Olivan. European data format "plus" (EDF+), an EDF alike standard format for the exchange of physiological data. *Clinical Neurophysiology*, 2003.

[24] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Locally adaptive dimensionality reduction for indexing large time series databases. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, SIGMOD '01, New York, NY, USA, 2001. ACM.

[25] E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: a survey and empirical demonstration. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, New York, NY, USA, 2002. ACM.

[26] B.-U. Köhler, C. Hennig, and R. Orglmeister. The principles of software QRS detection. *IEEE Engineering in Medicine and Biology Magazine*, 21(1):42–57, 2002.

[27] N. Kumar, N. Lolla, E. Keogh, S. Lonardi, and C. A. Ratanamahatana. Time-series bitmaps: a practical visualization tool for working with large time series databases. In *SIAM 2005 Data Mining Conference*, 2005.

[28] MATLAB. MATLAB: MATFile Format [online] available at: http://www.mathworks.com/help/pdf_doc/matlab/matfile_format.pdf [accessed 17 october 2011], 2011.

[29] J. M. Medeiros. Development of a Heart Rate Variability analysis tool (Master Thesis). https://estudogeral.sib.uc.pt/bitstream/10316/14091/1/Development

[30] J. Millman. *Microelectronics Digital and Analog Circuits and Systems*. McGraw-Hill Book Company, 1979.

[31] G. B. Moody, R. G. Mark, and A. L. Goldberger. PhysioNet: a Web-based resource for the study of physiologic signals. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):70–75, 2001.

[32] N. Nunes, T. Araújo, and H. Gamboa. Two-Modes Cyclic Biosignal Clustering based on Time Series Analysis. In *Procedings of the 4th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2011)*, Rome, Jan. 2011.

[33] T. Oliphant. *SciPy Tutorial*. http://www.tau.ac.il/∼kineret/amit/scipy_tutorial, 2004.

[34] T. Oliphant. *Guide to Numpy*. Tregol Publishing, 2006.

[35] J. Pan and W. Tompkins. A real-time QRS detection algorithm. *Biomedical Engineering, IEEE Transactions on*, (3):230–236, 1985.

[36] Physionet. PhysioNet - the research resource for complex physiologic signals [online] Available at: http://www.physionet.org/ [Accessed 17 September 2011], 199.

[37] PLUX. PLUX - Wireless Biosignals [online] Available at: http://plux.info/ [Accessed 5 September 2011], 2007.

[38] D. M. Ritchie, S. C. Johnson, M. E. Lesk, and B. W. Kernighan. *The C programming language*, pages 85–113. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1986.

[39] Scipy. Weave [online] available at: http://www.scipy.org/weave [accessed 17 october 2011].

[40] J. Semmlow. *Biosignal and biomedical image processing: MATLAB-based applications*. Number vol. 1 in Signal processing series. Marcel Dekker, 2004.

[41] J. Sousa, S. Palma, H. Silva, and H. Gamboa. aal@ home: a New Home Care Wireless Biosignal Monitoring Tool for Ambient Assisted Living. published in: http://www.plux.info/files/ftp/docs/aal_home_vf.pdf, 2010.

[42] E. Sundvall, M. Nyström, M. Forss, R. Chen, H. Petersson, and H. Ahlfeldt. Graphical overview and navigation of electronic health records in a prototyping environment using Google Earth and openEHR archetypes. *Studies In Health Technology And Informatics*, 129(Pt 2):1043–1047, 2007.

[43] G. van Rossum. *The Python Language Reference Manual*. Network Theory Ltd., 2003.

[44] A. Varri, B. Kemp, T. Penzel, and A. Schlogl. Standards for biomedical signal databases. *Engineering in Medicine and Biology Magazine, IEEE*, 20(3):33 –37, may/jun 2001.

[45] Vitalli Vanovschi. Parallel Python [online] Available at: http://www.parallelpython.com/ [Accessed 17 September 2011], 2005.

[46] wiCardioResp. wiCardioResp [online] Available at: http://wicardioresp.plux.info/ [Accessed 5 September 2011], 2011.

[47] L. Zhang, C. Yang, D. Liu, Y. Ren, and X. Rui. A web-mapping system for real-time visualization of the global terrain. *Computers & amp; Geosciences*, 31(3):343 – 352, 2005.

Appendix A

Publications

During the development of this project, one article was submitted and accepted to an international conference. This publication is entitled "Long-term biosignals visualization and processing" and will be presented in *BIOSIGNALS 2012*, of the "5th International Joint Conference on Biomedical Engineering Systems and Technologies" (BIOSTEC 2012).

LONG TERM BIOSIGNALS VISUALIZATION AND PROCESSING

Ricardo Gomes[1], Neuza Nunes[2], Joana Sousa[2] and Hugo Gamboa[1,2]
[1]*Physics Department, FCT-UNL, Lisbon, Portugal*
[2]*PLUX Wireless Biosignals, Lisbon, Portugal*
ricardo27gomes@gmail.com, nnunes@plux.info, jsousa@plux.info, hgamboa@fct.unl.pt

Keywords: Biosignal, Subsampling, Signal-Processing algorithms, long term monitoring, data structure.

Abstract: Long term acquisitions of biosignals are an important source of information about the patients' state and evolution, but in some situations involves managing very large datasets, which makes signal visualization and processing an hard task. To overcome these problems, we introduce a new data structure to manage long term biosignals. A fast multilevel visualization tool for any biosignal, based on the concept of subsampling is presented, with focus on the representative signal parameters (mean, maximum, minimum and standard deviation error). The visualization tool enables an overview of the entire signaland a more detailed visualization in specific parts which we want to highlight. The "Split and Merge" concept is also exposed for long term biosignal processing. A processing tool (ECG peak detection) was adapted for long term biosignals and several types of biosignals were used to test the developed algorithm. The visualization tool has proven to be faster than the standard methods and the developed processing algorithm detected the peaks of long term ECG signals fast and efficiently. The non-specific character of the new data structure and visualization tool, and the speed improvement in signal processing techniques introduced by these algorithms makes them useful tools for long term biosignals visualization and processing.

1 INTRODUCTION

The increasing development of medical systems and applications for human welfare and quality of life has been supported by patients' body signals monitoring. There are several types of body signals, also called biosignals, including bioelectric, bioimpedance, biomagnetic, bioacustic, biomechanical and biochemical signals (Bronzino, 2000). These biosignals give the researcher/clinician a perspective over the patient's state since they carry useful information for the comprehension of complex physiologic mechanisms underlying the behavior of living systems. The process of monitoring biosignals may be as simple as a physician estimating the patient's mean heart rate by feeling, with the fingertips, the blood pressure pulse. Biomedical signal analysis is nowadays a method of the greatest importance for data interpretation in medicine and biology, since the manipulation and processing of data provide vital information about the condition of the subject or the status of the experiment.

Signal visualization and processing techniques have been developed to help the examination of many different biosignals and to find important information embedded in them. In clinical cases, such as sleep disorders and neuromuscular diseases, a constant monitoring of the patient's condition is necessary (Pinto et al., 2010). This requirement is due to the possible occurrence of sudden alterations in the patient's state. The demand for a correct and prompt diagnosis leads to a mandatory identification of insufficiency signs in the clinical context; With this intention, long term biosignal acquisitions are one of the possible methods that allow a continuous monitoring of the patient (Kayyali et al., 2008). However, long term acquisitions generate large amounts of data. In order to analyze and follow up the patient's condition it is very important to visualize the acquired signals and extract relevant information from them. In patients with neuromuscular diseases, the heart rate variability, respiration, muscular and electrodermal activity signals are extremely important, since they indicate when a muscular crisis is occurring. The electrodermal activity signals are also very important when monitoring epilepsy patients or for the diagnosis of bipolar disorders, since the nervous system is a major intervenient in this cases (Poh et al., 2010), (Kappeler-Setz et al., 2010). In a future perspective, the continuous monitoring of these signals would allow the health care

providers to know beforehand when the patient needs assistance, assuring the patients' comfort and safety while they are continuously and remotely monitored in an ambient assisted living conditions (Sousa et al., 2010).

The long duration datasets obtained with these acquisitions exceed the capabilities for which standard analysis and processing software were designed. In addition to processing problems related to the difficulty of handling large amounts of data, displaying long term biosignals using standard visualization software is not feasible. Difficulties to visualize signals obtained in long acquisitions (e.g. recording for several hours) rise up from our inability to correctly visualize the entire signal displayed.

Considering the described problems with the long term biologic signals visualization and processing and the importance of this type of signals in health and research areas, we propose a new solution, by developing tools that enable a simple visualization of very large biosignals and an effective processing of these signals.

In this paper we present a new data structure designed for long term biosignals and we describe the tools developed to provide the possibility of having dedicated software for the visualization of biosignals in a fast and user friendly way (not only for very long biosignals, but also beneficial for smaller signals).

These tools have future perspectives to become powerful for biosignals inspection and analysis, accessible remotely with a web based tool. Regarding signal processing, we have implemented algorithms for an efficient processing of very large datasets based on parallel processing approaches.

In order to test the developed tools, several biosignals were acquired. Different time varying biosignals obtained from human volunteers, such as as electromyography (EMG), electrocardiography (ECG), electrodermal activiy (EDA), accelerometry or respiration signals. However, our goal was not to develop tools to be applied to a specific type of signal but to be as general as possible.

The following section presents the developed tools and the new data structure, designed for long term biosignals. There are three distintc parts: the first details the designed data structure, the second provides information on the implemented tools to visualize long term biosignals, and in the third one, the developed algorithms for long term biosignals processing are exposed and explained. In section 3 we present the methods of the developed work and discuss the results and algorithm's performance. Finally, we conclude the work in section 5.

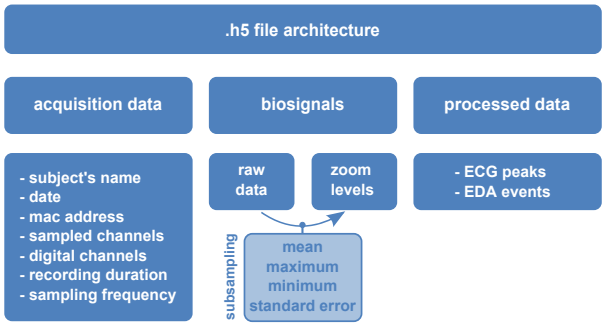

Figure 1: Proposed data structure for biosignals

2 PROPOSED DATA STRUCTURE AND DEVELOPED TOOLS

2.1 Long term biosignals data structure

The visualization of long term biosignals is very important in order to monitor electrophysiological data from patients. As we are dealing with very long signals, a tool to display large amounts of data is necessary.

Since we used acquisition equipment that saves (raw) data in text files, the major obstacle that appeared as a consequence of the file format that stored the signals was the impossibility to have random access to a specific time window of the recording, chosen by the user to visualize. In order to overcome this difficulty, we decided to create a new data structure that enables accessing the data fast. This structure was based on the HDF5 file format, which is a powerful tool for storing and managing different types of data allowing the necessary random access to any point in the signal being visualized (HDF group, 2007).

The data structure architecture (represented in Figure 1) is based on a section containing the biosignals, and a section for the processed data. Besides the two mentioned sections, a third one exists, containing information about the data, such as the acquisition date, the sampled channels or the sampling frequency.

The biosignals section is composed by the raw data and the different "zoom levels". These levels of zoom are the key for the phased visualization of signals that is developed. To obtain the different zoom levels, the four subsampling parameters (mean, maximum, minimum, standard deviation) shown in figure 1 are extracted from the signal.

There are fundaments for the choice of this four specific parameters to represent several zoom levels of the signals. Data mean identifies its central location. It is a representative measure of the signals shape. Maximum and minimum parameters define the

envelope on which the sampled signal is restrained, while the standard deviation error provides information about the signal's spreading.

The different zoom levels are created by a subsampling proccess. Each zoom level provides a different resolution of the signal. The first (and more detailed) level is the raw data, and the subsequent zoom levels are less detailed than the preceding one, having a smaller number of samples (because of the subsampling operations) but representing the same time interval. The subsampling operation is carried out by splitting the input signal in groups with a selected number of samples - the resampling factor (this factor can be for example 10, which means that the maximum, minimum, mean and standard deviation will be computed from 10 to 10 samples) and for each group calculating the representative signals' measures.

The first zoom level is obtained taking the raw data as input signal, while for higher zoom levels, the same four parameters are extracted, but instead of using raw data, the algorithm receives as input the data from the last zoom level to be created. In this case the algorithm calculates the mentioned parameters taking advantage of the data mining that is done on each level computation. Thus, the algorithm is simplified, since it calculates the mean of means, the maximum of maxima, minimum of minima and the standard deviation error. It should be noted that the standard deviation error, (sd), is obtained taking into account the expression given in equation 1, where $E[X]$ represents the expected value for the random variable X.

$$sd(X) = \sqrt{E[X - E[X]^2]} = \sqrt{E[X] - E[X]^2} \quad (1)$$

The visual effect and data mining of the described subsampling technique are shown on Figure 2.

The new data format provides a broader approach to the visualization and processing of biosignals, allowing the user to save the results of the biosignals processing tasks besides the raw data from the acquisition and other information about the subject or the recorded signals.

2.2 Long term biosignals visualization

A tool to visualize view long term signals was implemented, based on the new data structure.

The main idea of the visualization tool for long term biosignals is to allow a general overview of the entire signal in the first instance, giving the user the possibility to zoom in and out to a specific time window. This approach is comparable to a web mapping service, however, instead of viewing images of

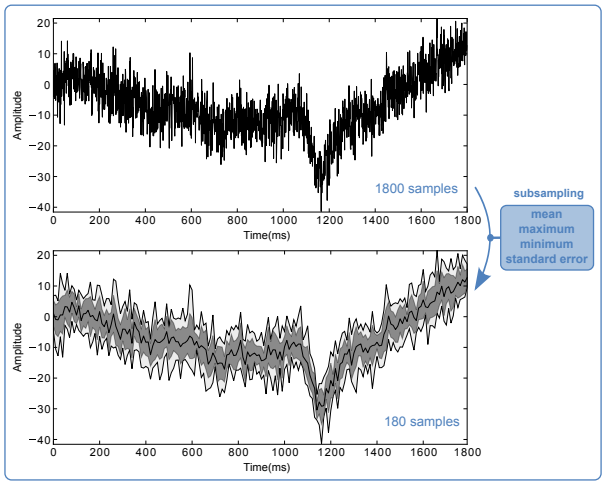

Figure 2: Illustration of the effect produced by a subsampling operation over a random signal (adimensional amplitude).

the Earth's surface it enables the visualization of large electrophysiological signals.

Data transmission via Internet is getting more common every day, and so a web environment application able to work in the web was developed, in order to provide a tool to visualize signals on the internet, by uploading or downloading them. A client-server model, using python as a way to manage data from the long biosignals and javascript to create the visualization platform has been implemented.

The tool enables the visualization of long term biosignals which have been converted to the already mentioned data structure. When the user runs the tool, the initial display is done by drawing the entire signal that is being visualized. This is the outermost, or by other means, the biggest zoom level, thus the one with less detailed information about the signal. When the user presses the navigation keys the signal being shown is updated to the new position selected. Signal navigation is facilitated by an overview window, that indicates the selected region of the signal and enables the user to select precise time windows in the signal to be visualized in detail.

This web environment directed tool lets the user explore signals using its different zoom levels and there are two drawing stages:

- **Preview:** on which the signals' informations to be drawn are only the maximum and minimum (aiming for a fast and representative overview);
- **Detailed view:** that draws the signal's mean, as well as the maximum, minimum, and the error shade (defined by mean±standard deviation error) with the intention of showing all the signals' characteristics.

The existence of two drawing steps allows the user to have a fast view of the signal's shape (represented by the maximum and minimum lines) on each interaction. This phased drawing technique enables a faster navigation through the signal, since the user can ask for new time windows to be displayed almost instantly. The detailed data is shown only when the viewer stops in a specific time window, providing the user with the complete information about the signal being observed.

When the user reaches the raw data level, no detailed information is shown, since there are no statistical parameters of the biosignal - the detail is the signal itself.

As the user "navigates" through the signal, the tool calculates the correct zoom level according to the time window that is being selected, gets data from the data structure, and displays it. The correct zoom level z corresponding to each selected zoom window is obtained with the equation 2.

$$z = \left\lceil \left(\frac{log(N)}{log(R)} - \frac{log(V)}{log(R)} + 1 \right) \right\rceil \quad (2)$$

Where N is the number of points that we are trying to see, R is the resampling, V is the maximum number of points to be displayed and $\lceil x \rceil$ represents the ceiling operation (rounding for the next integer).

2.3 Long term biosignals processing

Besides the problems inherent to visualization, long term biosignals also need different approaches regarding signal processing. In this work we introduce a new method to process long term biosignals. Since we are working with very large datasets, the input for these algorithms cant be the entire signal. In order to overcome the signal size problems, we suggest a block processing solution.

The implemented processing algorithms are based on a "split and merge"' process, in which a long signal is divided in parts (split inputs), that are processed independently - the processing function is applied to each input - and connect the various results (merge outputs).

Taking into account that the signals sizes can reach many hours, a parallel processing solution has been implemented. With our approach we answer to the long term biosignal processing problem with a parallel processing method (dividing a large problem into smaller ones that can be solved independently).

This implementation enables a large scale task distribution architecture, with a group of computers processing in parallel, each one of them with a small part of the signal.

The objective is to map the signal in intervals with fixed length and process each mapped interval, using an algorithm that works efficiently with shorter signals. After processing each interval, the results are merged together. If the operation is working correctly, merged data should be the same as the data that the detection algorithm would retrieve in case it could receive the entire signal as input.

Hereafter, we consider the discrete biosignal to be processed, X, described in equation 3, where k is an integer value that represents the signal's number of samples.

$$x(n) = \{x_1, x_2, \ldots, x_k\} \quad (3)$$

The processing operation can be represented by equation 4.

$$Y = F(X) \quad (4)$$

The operator F receives an entire biosignal (X) as input and returns Y. Since the input signal might be very long, the need to map it in several smaller regions to be processed separately becomes imperative.

However, X can be splitted in subgroups with a fixed number of samples - L. The signal mapper is then a list of pairs that define the several subgroups to be processed separately. Let us call this list of pairs J. J is described on equation 5.

$$\begin{aligned} J = \{&(0, L), (L - v, L - v + L), \\ &(2L - 2v, 2L - 2v + L), \\ &\ldots, \\ &(mL - mv, mL - mv + L)\} \end{aligned} \quad (5)$$

with v being the number of samples to be overlapped, and m an integer.

Selecting the signal (X) in the time intervals defined by J, the signal will be mapped. Each subsignal can be defined by equation 6.

$$\begin{aligned} x^0 &= \{x_0, \ldots, x_L\} \\ x^1 &= \{x_{L-v}, \ldots, x_{L-v+L}\} \\ &\cdots \end{aligned} \quad (6)$$

In the borders where the signal is splitted to be processed, there might occur some problems. In order to overcome this kind of questions, the implemented algorithm has an overlapping number of samples, v (everytime the algorithm runs for a selected time window, there is a number of samples from the end of the last time window that is considered in the beggining of the actual one).

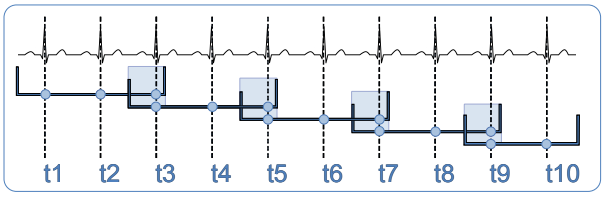

Figure 3: Representation of the processing algorithm for the ECG peaks detection.

After mapping the signal, the processing algorithm is applied to the various intervals mapped from the signal. Giving each interval to the input of the processing routine, we will obtain a group of outputs, that can be defined by equation 7.

$$y^j = f(x^j) \tag{7}$$

On this last step, in which i represents a subprocessing group, the results from the independent separate processing tasks are merged together, in the "reduce" operation. The function that correctly joins together the outputs from the subprocessing tasks is denoted by G, and the final result is given by equation 8.

$$Y = G(y^0, y^1, \dots) \tag{8}$$

An example of a mature processing algorithm (Pan and Tompkins, 1985), which do not work properly on long term biosignals was adapted to this type of signals: the peak detector to be applied on ECG signals. A representation of the ECG peak detection algorithm is made in figure 3. The times $(t_1, t_2, ..., t_{10})$ indicate the peaks detected by the algorithm, while the shaded areas represent the overlapping of the algorithm (where two processing windows intersect). With the mapping and reducing techinque implementation, the large signals are mapped in a series of time intervals and the processing functions take this intervals as input. For each interval, the processing results are calculated and the outputs of the group of "subprocessing" steps are merged.

Parallel processing in computers enables to divide a long operation in several smaller tasks that can be carried out by different computer core processors. Considering a processing operation with a fixed start time (T_s), that takes a time T to be carried out by one processor and that the processing is going to be divided by N_s processors, the total parallel processing time (T_p) will be given by equation 9.

$$T_p = T_s + \frac{T}{N_s} \times (1 + Ov) \times 2 \tag{9}$$

On equation 9, the overlap (Ov) is defined by the expression given in 10, where v is the overlapping number of samples and N_{slice} is the number of samples of each processing slice.

$$O_v = \frac{v}{N_{slice}} \tag{10}$$

Since the existance of the overlap means that there are samples being processed in two different subtasks, a bigger overlap causes the processing to last longer. However, if the overlap is too small, there is the danger of ocurring processing errors. In order to prevent these errors, our ECG peak detection algorithm only considers the data to be efficiently processed when there are coincident peaks in the output (adjacent subtasks detect at least one common peak).

3 PERFORMANCE EVALUATION

3.1 BIOSIGNAL ACQUISITION METHODS

Several types of biosignals such as as electromyography (EMG), electrocardiography (ECG), electrodermal activiy (EDA), accelerometer and respiration have been acquired in cooperation with the WiCardioResp project (wiCardioResp, 2011) in order to test the visualizing and processing algorithms. The acquisitions were carried out at the patients' homes, with their approval, during the night (each recording had the approximate duration of 8 hours).

The equipment used to acquire the biosignals necessary for this work was the bioPLUX research system, a wireless signal acquisition unit (PLUX - Wireless Biosignals, 2011). This system is portable, small in size and light-weighted, has a 12 bit ADC and a sampling frequency of 1000 Hz, and creates text files with the acquired (raw) data.

With the acquired biosignals, the developed tools have been tested in order to evaluate their performance. Times of the creation of visualization levels, the visualization tool performance, and time consuming in processing algorithms were monitored. Besides the speed tests, efficiency tests were carried out with the processing tools, so it could be confirmed if the results of the new processing algorithms were correct.

3.2 Results and discussion

Regarding data conversion to the new data structure, the performance results are described in table 1. All the performance tests were made with a Intel Core i7 720QM with a 1.60GHz processor.

Considering that opening text files with sizes of this order of magnitude by loading them on python

Table 1: Conversion times

text file size (MB)	Conversion times (s)	
	raw data	zoom levels
346,8	41	85
435,1	50	104
954,6	91	217
1.021,2	109	234
1.297,3	157	357

Table 2: Load times for .txt and .h5 files

file size (MB)	Load times (s)	
	.txt file	.h5 file
14	0.01	6.35
144	0.04	64.33
347	0.57	349.33
424	0.79	(Memory Error)

might take a long time or even cause a memory error, the presented results (see table 2) are an evidence of the benefits of the developed data structure on data acessing/visualization.

The performance of the visualization tool is independent of the type and size of the signal being visualized as well as of the zoom level on which the user is ”navigating” with the developed tool.

Operations like zooming and panning over long term biosignals, that take several seconds using python visualization methods, are practically instantaneous using the developed tools.

Since the conversion only has to be carried out once, and acessing data from the new structure takes only miliseconds, it is possible to understand the advantages brought by the presented tools.

Figure 4 shows the aspect of the designed visualization tool. Five channels (ECG, Respiration and the three accelerometer components - x, y and z) are visible and one of them (ECG) is expanded. At the bottom of the image, there is an overview window with the entire signal drawn, and a rectangle indicating the current time window selected.

The processing results of the application of the developed MapReduce algorithm for the detection of ECG peaks were compared with the output of the standard algorithms (without parallel processing), in order to analyze the efficiency of the new tools.

4 CONCLUSIONS

Considering standard formats for storage and exchange of biological and physical signals, it is easy to see that the new and innovative developed data structure allows a broader approach to the visualization

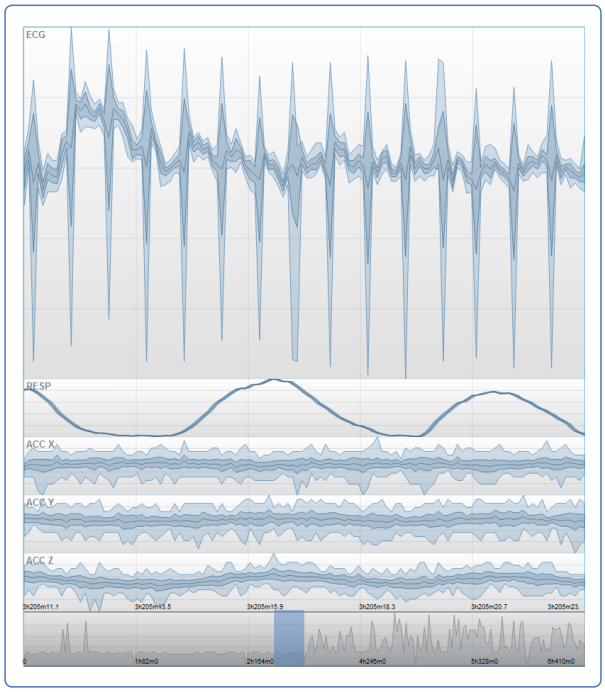

Figure 4: Biosignal visualization tool

and processing of biosignals (particularly for long term biosignals). Besides allowing the user to save the results of the parallel biosignal processing algorithms, saving the raw data from the acquisition and possible important information about the subject or the recorded signals, this format allows a new way of exploring biological data, in a fast and intuitive multilevel visualization of the biosignals. Since the developed visualization tools are compatible with the web environment, they can be used for data sharing in the internet.

5 FUTURE WORK

In future work we aim to create an algorithm that allows processed data visualization, as a way to link the processed data and the signal. We intend to make it possible to visualize at the same time the signal and important processed data, such as the ECG peaks detected, or the EDA events that ocurred.

Other future goal is to develop new processing algorithms adapted to long term biosignals, such as the heart rate variability (HRV), since it’s parameters are of great importance in clinical cases that need long term monitoring, such as neuromuscular diseases.

Regarding parallel processing techniques, some improvements are still necessary, such as an automatic calculation of the indicated number of overlap-

ping samples to be considered for each processing operation.

6 ACKNOWLEDGEMENTS

This work was partially supported by National Strategic Reference Framework (NSRF-QREN) under project "LUL", "wiCardioResp" and "Do-IT", and Seventh Framework Programme (FP7) program under project ICT4Depression, whose support the authors gratefully acknowledge. The authors also thank PLUX Wireless Biosignals for providing the acquisition system and sensors necessary to this investigation.

REFERENCES

Bronzino, J. (2000). *The biomedical engineering handbook*. Number vol. 1 in The Biomedical Engineering Handbook. CRC Press.

HDF group (2007). HDF5 - HDF group [online] Available at:http://www.hdfgroup.org/HDF5/ [Accessed 5 September 2011].

Kappeler-Setz, C., Schumm, J., Kusserow, M., Arnrich, B., and Trster, G. (2010). Towards long term monitoring of electrodermal activity in daily life. *Potentials*.

Kayyali, H., Weimer, S., Frederick, C., Martin, C., Basa, D., Juguilon, J., and Jugilioni, F. (2008). Remotely attended home monitoring of sleep disorders. *Telemed J E Health*, 14(4):371–4.

Pan, J. and Tompkins, W. (1985). A real-time qrs detection algorithm. *Biomedical Engineering, IEEE Transactions on*, (3):230–236.

Pinto, A., Almeida, J. P., Pinto, S., Pereira, J., Oliveira, A. G., and de Carvalho, M. (2010). Home telemonitoring of non-invasive ventilation decreases healthcare utilisation in a prospective controlled trial of patients with amyotrophic lateral sclerosis. *Journal of Neurology, Neurosurgery & Psychiatry*.

Poh, M.-Z., Loddenkemper, T., Swenson, N., Goyal, S., Madsen, J., and Picard, R. (2010). Continuous monitoring of electrodermal activity during epileptic seizures using a wearable sensor. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 4415 – 4418.

Sousa, J., Palma, S., Silva, H., and Gamboa, H. (2010). aal@ home: a new home care wireless biosignal monitoring tool for ambient assisted living.

wiCardioResp (2011). wiCardioResp [online] Available at: http://wicardioresp.plux.info/ [Accessed 5 September 2011].

\section*{APPENDIX}

Appendix B

Work route

During this work, a variety of tools were used. The presented algorithms were developed using Python [43], C [38] and Javascript [16] programming languages, with the help of Eclipse and Aptana as an integrated development environment. The Python packages used were the numpy [34], scipy [33], matplotlib [19], h5py [17], weave [39] and parallel python [45].

During a research work, as the defined objectives are achieved, new ideas arise, and the problems that emerge while dealing with the purposed tasks lead to new approaches, in order to meet the expected results.

Initially, the idea of this work was to create a new data structure that could provide the researchers or clinicians with tools for a fast and effective biosignals visualization and processing. Since **.hdf5* files were chosen to store the data, and the used acquisition equipment writes data into **.txt* files, a conversion tool had to be developed. This tool should be able to read text files with millions of lines, which is a time consuming task. The first approach was to use python, and read the text file, line by line, converting data to the new file type. However, it was decided to try to improve the performance of the developed algorithm. In order to reduce the time consuming of the developed conversion algorithm, a text file reading function was developed using C programming language. This C code was input in the python code, using python's "weave" package. This alteration allowed a faster data conversion.

The data visualization tool, described in Chapter 4 of this work was also matured from an initial version. The first version of the visualization tool was developed using a python package (matplotlib). This visualization tool met the objectives of this work, however it was not very fast. Looking for a faster biosignal navigation/visualization tool, an alternative was developed. Due to the advantages of using the internet, the new version of the visualization tool was designed in a web environment, using javascript as a data displaying system and python as data server. This update enabled an improvement in the tool's navigation speed. Asking for new data windows using only python is tens of times slower than using Javascript, because the matplotlib redraws all the graphical elements on each time window selection, while with Javascript it was possible to only draw the lines that represent the signals being inspected.

In the beginning of this work, signal processing was thought to be done using non-parallel algorithms. However, it was chosen to take advantage of the parallel processing concept, because the application of non-parallel algorithms would not be feasible. The parallel algorithms were developed taking the `MapReduce` framework as a guiding example.

In the end, the objectives were accomplished, with improved results that exceed the initial expectations. Despite being a non-linear route, since different approaches have been designed and tested until the presented results were obtained, this work produced very useful tools which can prove to be very valuable for long-term biosignals visualization and processing.