BIOTECHNO 2020

The Twelfth International Conference on Bioinformatics, Biocomputational Systems and Biotechnologies

ISBN: 978-1-61208-792-4

September 27th – October 1st, 2020

BIOTECHNO 2020 Editors

Birgit Gersbeck-Schierholz, Leibniz Universität Hannover, Germany

BIOTECHNO 2020

Forward

The Twelfth International Conference on Bioinformatics, Biocomputational Systems and Biotechnologies (BIOTECHNO 2020) continued a series of events covering these three main areas: bioinformatics, biomedical technologies, and biocomputing.

Bioinformatics deals with the system-level study of complex interactions in biosystems providing a quantitative systemic approach to understand them and appropriate tool support and concepts to model them. Understanding and modeling biosystems requires simulation of biological behaviors and functions. Bioinformatics itself constitutes a vast area of research and specialization, as many classical domains such as databases, modeling, and regular expressions are used to represent, store, retrieve and process a huge volume of knowledge.

Biotechnology is defined as the industrial use of living organisms or biological techniques developed through basic research. Bio-oriented technologies became very popular in various research topics and industrial market segments. Current human mechanisms seem to offer significant ways for improving theories, algorithms, technologies, products and systems. The focus is driven by fundamentals in approaching and applying biotechnologies in terms of engineering methods, special electronics, and special materials and systems. Borrowing simplicity and performance from the real life, biodevices cover a large spectrum of areas, from sensors, chips, and biometry to computing. One of the chief domains is represented by the biomedical biotechnologies, from instrumentation to monitoring, from simple sensors to integrated systems, including image processing and visualization systems. As the state-of-the-art evolves at fast speed, new biotechnologies and biosystems become available. Their rapid integration in real life becomes a challenge.

Brain-computing, biocomputing, and computation biology and microbiology represent advanced methodologies and mechanisms in approaching and understanding the challenging behavior of life mechanisms. Using bio-ontologies, biosemantics and special processing concepts, progress was achieved in dealing with genomics, biopharmaceutical and molecular intelligence, in the biology and microbiology domains. The area brings a rich spectrum of informatics paradigms, such as epidemic models, pattern classification, graph theory, or stochastic models, to support special biocomputing applications in biomedical, genetics, molecular and cellular biology and microbiology. While progress is achieved with a high speed, challenges must be overcome for large-scale bio-subsystems, special genomics cases, bio-nanotechnologies, drugs, or microbial propagation and immunity.

We take here the opportunity to warmly thank all the members of the BIOTECHNO 2020 technical program committee, as well as all the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to BIOTECHNO 2020. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions. We also thank the members of the BIOTECHNO 2020 organizing committee for their help in handling the logistics of this event.

BIOTECHNO 2020 Chairs

BIOTECHNO 2020 Steering Committee
Birgit Gersbeck-Schierholz, Leibniz Universität Hannover, Germany
Hesham H. Ali, University of Nebraska at Omaha, USA

BIOTECHNO 2020 Publicity Chair
Joseyda Jaqueline More, Universitat Politecnica de Valencia, Spain
Marta Botella-Campos, Universitat Politecnica de Valencia, Spain

BIOTECHNO 2020 Industry/Research Advisory Committee
Steffen Heber, North Carolina State University, USA
Alexandru Floares, SAIA Institute, Romania
Gilles Bernot, University Nice Sophia Antipolis, France
Erliang Zeng,University of Iowa, USA
Y-h. Taguchi, Chuo University, Japan

BIOTECHNO 2020
Committee

BIOTECHNO 2020 Steering Committee
Birgit Gersbeck-Schierholz, Leibniz Universität Hannover, Germany
Hesham H. Ali, University of Nebraska at Omaha, USA

BIOTECHNO 2020 Publicity Chair
Joseyda Jaqueline More, Universitat Politecnica de Valencia, Spain
Marta Botella-Campos, Universitat Politecnica de Valencia, Spain

BIOTECHNO 2020 Industry/Research Advisory Committee
Steffen Heber, North Carolina State University, USA
Alexandru Floares, SAIA Institute, Romania
Gilles Bernot, University Nice Sophia Antipolis, France
Erliang Zeng,University of Iowa, USA
Y-h. Taguchi, Chuo University, Japan

BIOTECHNO 2020 Technical Program Committee
Behrooz Abbaszadeh, University of Ottawa, Canada
Antonino Abbruzzo, University of Palermo, Italy
Jens Allmer, Hochschule Ruhr West - University of Applied Sciences, Germany
Yoseph Bar-Cohen, Electroactive Technologies / NDEAA Lab - Jet Propulsion Laboratory (JPL), USA
Kais Belwafi, King Saud University, Saudi Arabia
Boubaker Ben Ali, University of Bordeaux, France / University of Manouba, Tunisia
Razvan Bocu, Transilvania University of Brasov, Romania
Matthias Chung, Virginia Tech, USA
Peter Clote, Boston College, USA
Santa Di Cataldo, Politecnico di Torino, Italy
Maria Evelina Fantacci, University of Pisa, Italy
Asier Ibeas, Universitat Autònoma de Barcelona, Spain
Valentinas Klevas, Lithuanian Energy Institute, Lithuania
Jan Kubicek, VSB - Technical University of Ostrava, Czech Republic
Antonio LaTorre, Universidad Politécnica de Madrid, Spain
Cedric Lhoussaine, University Lille, France
Chilukuri K. Mohan, Syracuse University, USA
Chen Li, Monash University, Australia
Yiheng Liang, Bridgewater State University, USA
Tatjana Lončar-Turukalo, University of Novi Sad, Serbia
Constantin Paleologu, University Politehnica of Bucharest, Romania
Vincent Rodin, University of Brest, France
Ulrich Rueckert, Bielefeld University, Germany
Thomas Schmid, Universität Leipzig, Germany
Andrew Schumann, University of Information Technology and Management in Rzeszow, Poland
Christine Sinoquet, University of Nantes, France
Moeez M. Subhani, University of Derby, UK
Sophia Tsoka, King's College London, UK

Erliang Zeng,University of Iowa, USA
Haowen Zhang, Georgia Institute of Technology, USA
Qiang Zhu, The University of Michigan - Dearborn, USA

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Biological Digital Signal Processing

Interpretation and Combination

Ionescu Marius
"Politehnica" University of Bucharest
Faculty of Electronics, Telecommunications and Information Technology
Bucharest, Romania
Email: yonescumarius@gmail.com

Abstract— **In this work, we discuss methods, filters and algorithms for processing of biological signals, as well as the interpretation and display of the results. Biological signals can help us diagnose certain diseases, and their combination and interpretation can provide us with relevant information about our health. The discussed problem is related to how biological signals can be processed, combined, interpreted and displayed in order to make accurate diagnoses. The article illustrates a new prototype based on spectroscopic methods which uses near infrared sensors to monitor blood glucose levels. The prototype combines spectroscopic methods with other methods, such as Electrocardiography or Electromyography. The work focuses on light absorbance in matter and on non-invasive blood glucose detection using near infrared technology by colorimetric interpretation of the values transmitted.**

Keywords- Digital signal processing; Haar filter; Butterworth filter; signals combination; signals interpretation; spectroscopic signals.

I. INTRODUCTION

In this paper, we discuss methods and algorithms for processing of biological signals, as well as the interpretation and display of the results using development platforms that enable digital acquisition and processing of biological signals. Digital signal processing is a bio-medical method that can help us make a faster diagnosis and provides more reliable treatment options for patients. Combining medical digital signals from Electrocardiography, Electroencephalography, Electromyography, or spectroscopic near infrared, can help us in monitoring diseases. Biological signals can come from different types of sources: audio, video, electrical, magnetic, etc. The challenge is to understand how these signals have been converted into electrical signals through methods of capturing and using transducers, such as sensors that measure physical and chemical values [1].

The challenge mentioned above is related to the interpretation of signals for the calculation and prognosis of diseases, such as diabetes, cancer or stroke. The prototype proposed in this article is an assembly between an Arduino board and two near infrared sensors for absorption and colorimetry. The signals acquired by Electrocardiography, Electromyography or light absorption sensors through biological tissues are processed and combined by a piece of software. Signal processing is based on the processing of the biological electrical properties of the body, which occur in tissues. Biological signals can be correlated with the mechanical, magnetic or spectroscopic signals, and used in biological analysis and signal processing [3].

Regarding the technique of acquiring biological signals, nowadays, Biological Signal Import Module (BSIM) is often used for acquiring biological signals [23]. It supports the acquisition of analog biological signals (2.5 V) from sensors like a pH electrode or an UV detector. BSIM uses multiple acquisition channels to acquire and interpret data using appropriate software for each channel. Thus, a module that receives signals from more than one electrode may be able to generate data for the analysis of Electrocardiography and Electromyography signals, as well as spectroscopic signals. This analysis is important because it helps with the detection of diseases that a patient may suffer from by processing biological signals.

According to the studies of Lapique Nicolas [1], professor in the Department of Biosystems Science and Engineering, Zurich, Germany, a biological signal processing circuit is based on a biological sensor that controls the activity of individual components using an internal timer. This prevents a sensor circuit from being active when the system is not in use and there is no need for processing biological data transmission. When the system is active, it transmits data via a control signal [1]. However, it is a challenge to combine different biological components to form a complex bio-signal in order to convey as much biological information as possible to the computer software for analysis and processing [21]. Lapique Nicolas [1] explains that biological signals travel differently through an electronic wire, and that, in biology, there is a variety of different signals from proteins to micro ribonucleic acid molecules [2]. A special feature in processing biological signals consists not only in transforming a signal into another, but also in transforming multiple input signals into multiple output signals.

The rest of the paper is structured as follows. In Section II, we talk about processing, interpretation and display of signals, Electrocardiography, Electromyography and near infrared spectroscopy signals. In Sections III and IV, we

 ISBN: 978-1-61208-792-4

present the filters that can be applied over these signals, such as Butterworth and Haar filters, for a more accurate interpretation and acquisition of channel settings. In Section V, we discuss the near infrared signals for glucose and blood analysis. We conclude the paper in Section VI.

II. ELECTROCARDIOGRAPHY AND ELECTROMIOGRAPHY SIGNAL PROCESSING, COMBINING AND DISPLAYING

Electrocardiography signals can be combined with other signals, or information can be extracted from signals coming from different sources, such as Electromyography or near infrared, spectroscopic signals [4].

The acquisition of Electrocardiography signals uses the latest generation of microprocessors. Before being forwarded, the signals are extensively processed. At the moment, the acquisition of Electrocardiography signals is investigated with silver/silver chloride electrodes. The Ag (silver)/AgCl (silver chloride) electrode is used in common Electrocardiography systems and has a maximum offset voltage of ± 300 mV. A ± 0.5 mV desired signal is superimposed on the electrode offset. In addition, the system also takes the noise 50/60 Hz power lines forming common mode signal. The amplitude of power line noise could be very large and must be filtered [5].

Signal processing is a big challenge as the real value of the signal will be in an environment of 0.5 mV offset by 300 mV. Other factors, such as Alternating Current (AC) power interference, Radio-Frequency (RF) interference from surgery equipment, and implanted devices, or rhythm changes and physiological monitoring system, can also have an impact. The main sources of noise in Electrocardiography are:

- Low frequency noise (drift);
- Power line interference (50 Hz or 60);
- Muscular noise (this noise is very difficult to remove because it is in the same region as the real signal. It is usually corrected by software);
- Other interferences (ie., radio frequency noise from other equipment).

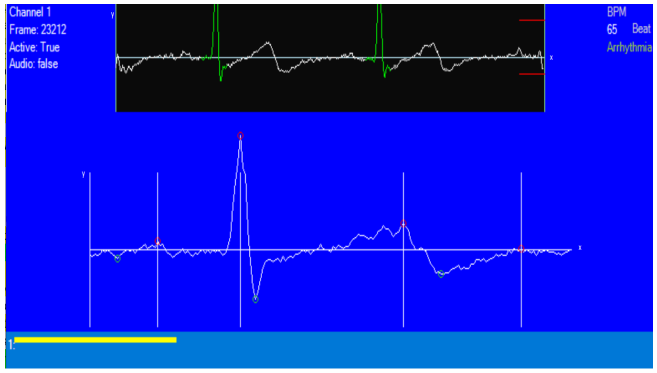


Figure 1. Combined graphical display of a raw ECG and an EMG chart.

This article aims to discuss the acquisition, takeover and processing of Electrocardiography and Electromyography signals, combining and displaying them using software. The goal is to extract as much information as possible from the data that is captured by using a developed board, which uses an advanced microcontroller receiving advanced processing signals from transducers. The signals which came in through the serial computer software are processed and displayed using advanced graphics.

The goal is to extract as much information as possible from the received data and display it in a more comprehensive way in order to be understood by bioengineers, doctors or trained personnel, based on which they can diagnose and predict certain diseases or information about the health of a patient or a pathological case (Figure 1).

III. SIGNALS - HAAR AND BUTTERWORTH FILTERS

For processing and filtering graphics, we used two filters. The Haar filter, which is part of a wavelet family, is used in mathematics for waves. Wavelet analysis is similar to Fourier analysis because it allows a target function to be represented as an orthonormal basis. Using the wavelets for Electrocardiograph representation is quite useful if the sampled signal is continuous and has sudden transitions (Figure 1).

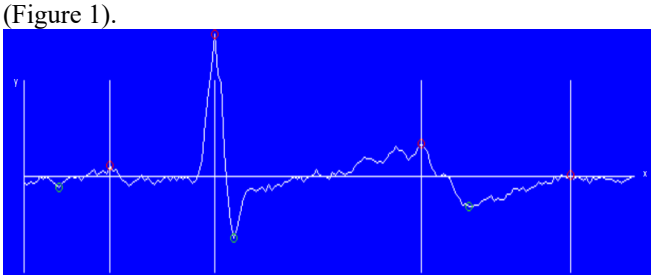


Figure 2. Electrocardiography signal capture.

One advantage of using a Haar filter for Electrocardiography signals graphical representation that it helps us represent any sample time as a continuous function, uniformly, approximated by linear combinations. Thus, this algorithm is extended to those areas where any function of this type can be uniformly approximated by continuous functions [14]. Samples are types of discontinuous functions that can distort the signal according to the formula below, where $\delta_{n,k}$ represent Kronecker delta and $\psi_{n,k}$ represent the real line R [25].

$$\int_{-\infty}^{\infty} 2^{\frac{(n+n_1)}{2}} \Psi\left(2^n t - k\right)\Psi\left(2^n t - k_1\right)dt = \delta_{n,n_1}\delta_{k,k_1} \quad (1)$$

Input sequences, which, in our case, are sampled Electrocardiography signals, are passed through a matrix type Haar by applying the wavelet transform discrete type, a 4x4 matrix:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \quad (2)$$

Butterworth had a reputation for solving "impossible" mathematical problems. At the time, this filter design required a considerable amount of designer experience because of the limitations of the theory in use. The filter has not been used for more than 30 years after its publication [24]. Butterworth has shown that close successive approximations were obtained by increasing the number of screening the correct values. At the time, filter waves generated substantial low-pass filter. Butterworth has shown that a low-pass filter can be designed with a cutoff frequency normalized to 1 radian per second and whose frequency response is:

$$G(\omega) = \frac{1}{\sqrt{1 + \omega^{2n}}} \quad (3)$$

where ω is the angular frequency in radians per second, and n is the number of poles in the filter equal to the number of reactive elements in a passive filter. If $\omega = 1$, the magnitude of this type of filter passband is $1 / \sqrt{2} \approx 0.707$, which is half power or -3 dB. Butterworth filters work only with an even number of poles in his work. He can ignore that these filters can be designed with an odd number of poles. He built his higher order filters, the filters with two poles separated by vacuum tube amplifiers. The frequency response plot of 2, 4, 6, 8 and 10 pole filters is shown as A, B, C, D and E in his original chart.

Butterworth solved the equations of two or four-pole filters, that show how the latter could be in waterfall when they are separated by vacuum tube amplifiers, allowing the construction of higher order filters despite the losses. In 1930, Butterworth used forms of coil with diameter of 1.25 cm and 3 cm long, with plug-in terminals, capacitors and associated resistors contained inside a coil. Coil resistance forms part of the load plate. Two poles were used for each vacuum tube and RC coupling was used for the electric grid of the next tube [7].

The Butterworth filtering algorithm can be transformed with the Haar filter used for Electrocardiography graphics. That can help to sample the Butterworth signal processing, where the algorithm has a defined number of low and high pass Butterworth filters with three poles, and which works on a certain frequency threshold [17]. A band-pass filter can be implemented by applying sequential algorithms to filter high-pass and low-pass [15].

In this sense, we applied algorithms corresponding to impulse response filters, which were designed by applying the bilinear transformation to the transfer functions of the corresponding analog filters [9], resulting in a recursive digital filter with seven real coefficients. So, in this application, we will have Butterworth type filters with the following settings (Figure 3):

- Butterworth_FreqHP - frequency high-pass which has the default 3 dB;
- Butterworth_FreqLP - low-pass frequency is 170 dB default value;
- Butterworth_Level - up crossing that has the default 1;
- Butterworth_PowerHP - high-pass power that has the default 57;

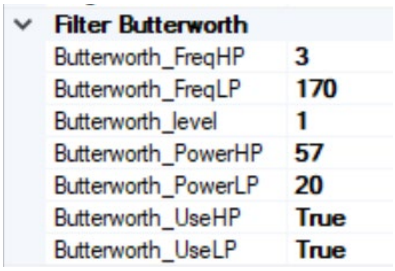


Figure 3. Settings the seven real coefficients for Butterworth filter

- Butterworth_PowerLP - low-pass power that has the default 20;
- Butterworth_UseHP - to enable high-pass, default is true;
- Butterworth_UseLP - to enable low-pass, default is true;

The expressions for filtering coefficients depending on the separation frequency and the sampling period are derived. The transfer function shows a plateau over the passband and a gradual attenuation more apparent at the frequencies above and below the cutoff frequency, with a slope of 60 dB / decade [8].

There is an attenuation of 3 dB frequency cutting and a gradual increase in phase shift frequency at every 10 steps. Low-pass filters show a maximum of 8 % overshoot and high-pass filters down show a maximum overshoot of about 35%. The algorithm to calculate filter coefficients for an arbitrary limit frequency of Electroencephalography may be useful in modern laboratories and for software designers for electrophysiological applications [19].

IV. SETTINGS AND USING CHANNELS FOR ARRHYTHMIA

Current applications for processing biological signals can set and use multiple channels simultaneously, which can receive different signals. Each channel has its settings. For example, the heart rate settings can be used as follows:

- Beat Level High;
- Beat Level High Limit;
- Beat Level Low;
- Beat Level Low Limit;
- Filter Haar;
- Filter Butterworth;

Each channel can sample its own independent set of signals and may apply a set of specified filters. In our case,

the Electrocardiography signal can analyze, filter and display muscle activities in real time from the main sample data transmitted by a transducer.

The application allows diagnosis mode that draws the PQST axis based on filters used at some point. The goal is to save the PQST state at certain time intervals [12].

The signals based on flows and electric excitations of the body, detected and transmitted by electrodes, can display, process and set various diagnoses, prognoses and can interpolate the obtained information, so that the area of diagnostics includes batch jobs related to other regions or functions of the biological body, such as Electrocardiography or Electromyography (Figure 4).

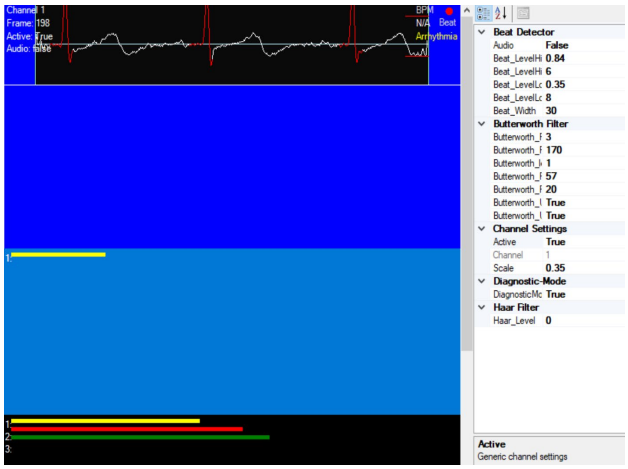


Figure 4. Settings channels of application processing signals

Arrhythmia (Figure 5) is a problem with the rate or rhythm of the heartbeat. During an arrhythmia, the heart can beat too fast, too slowly, or with an irregular rhythm. If the heartbeat is below 60 beats per minute, the condition is called bradycardia and if it is over 100 beats per minute, the condition is called tachycardia.

The arrhythmia algorithm calculation is based on data from the sampling difference every 6 beats. If a heartbeat is detected at each 6 beats, an anomaly is detected and the software will send alerts (Figure 6).

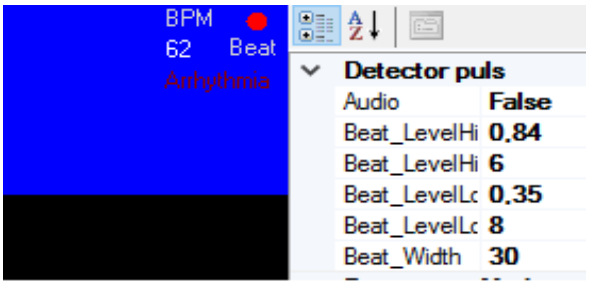


Figure 5. Arrhythmia detector

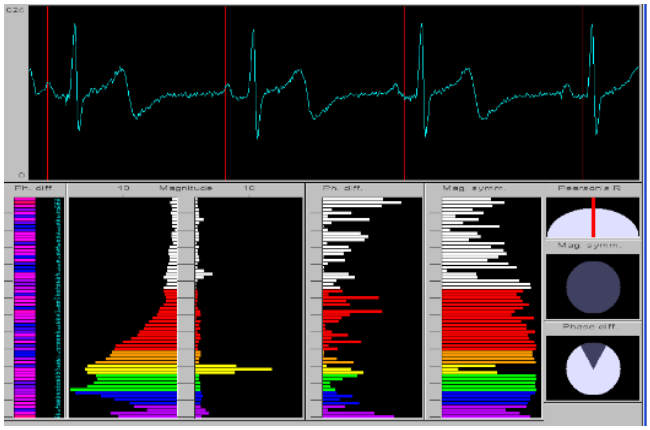


Figure 6. Acquisition data Electrocardiography and Electromyography

The proposed prototype starts with Electrocardiography signals, which can be interpreted and displayed by the software. To this prototype were attached sensors, such as near infrared spectroscopic sensors, which measure the absorption of light through biological tissue. In this sense, an algorithm has been developed to calculate light absorption through the three layers of the skin [10], light absorption through blood, plasma and formed elements (erythrocytes, leukocytes and thrombocytes, as well as nutrients). In this article, we focus only on the absorption of light through glucose, as a biological signal correlated with Electrocardiography signals.

V. NEAR INFRARED SPECTROSCOPY SIGNAL IN MEASURING GLUCOSE

A. Introduction

One of the modern methods proposed in this paper at the development phase and research is non-invasive glucose detection in correlation with Electrocardiography signals. The prototype has two hardware modules that are interconnected through different interfaces. Electrocardiography helps spectroscopic measurement (in this case, glucose) by providing the moment of maximum blood flow through the body.

The invasive colorimetric method produces a chemical reaction activated by an enzyme, not oxidizing glucose. The process itself consists of placing a drop of blood on a test strip. The blood glucose will react with a chemical reagent (bromine or chlorine) that changes color. This modification of blood color is measured and interpreted as a blood glucose level. In this regard, the proposed prototype in this paper brings two general problems in detecting the glucose level in the blood [22]:

- Non-invasive blood glucose detection using near infrared technology by colorimetric interpretation of the values transmitted by a sensor;
- Absorption of light in matter;

 ISBN: 978-1-61208-792-4

The research proposes an interface that analyzes and calculates the colorimetric values transmitted by a development board through two near infrared sensors: one for light absorption and one for colorimetry. The blood glucose measurement algorithm is based on the absorption of the amount of monochromatic light that passes through the tissues to the capillaries that contain blood [10]. The blood glucose concentration is measured based on the amount of monochromatic light absorption through tissues [14]. The error rate can go up to 20% due to the light that has to pass through to the capillary veins. Also, the spectral bandwidth may have large errors in this sense. The paper proposes an interpretation algorithm based on the RGB interpretation of the data coming from the near infrared sensor, and only an interpretation of the monochromatic absorption. RGB data is interpreted based on tint and saturation to obtain the average blood glucose level [22].

Classical blood glucose measurement devices are based on colorimetry and measure the amount of light absorbed by matter [20].This way, the glucose concentration is measured by the detection of the luminous intensity passing through a blood sample, which contains the serum and chemical reagent products [6].

The measurement procedure is similar to urine analysis, where a urine sample passes through the yellow light and absorbs the blue and green lights [21].

B. Prototype

The proposed prototype uses light that passes through matter, for example, through our finger, so that the light can reach the capillaries. Tests were performed on a physiological serum that was mixed with 100g of glucose [18]. The near infrared sensor passes through the sample, being helped by the auxiliary light, and returns the RGB values in order to determine the amount of glucose in the sample. To analyze and measure blood glucose, some standard measures should be taken (Table I).

TABLE I. STANDARD ABSORBANCE

1.	Wavelength (400 ~ 800 nm)
2.	The standard amount of glucose used in the test
3.	Incubation time
4.	Standard sample quantity
5.	Volume of the reagent
6.	Limit of Absorbance
7.	RGB color standard

The principle of measurement is based on the uniform dispersion of light through matter to capillary. This is facilitated by the auxiliary light that allows the near infrared sensor to collect better values of the RGB in the blood. To measure the density of blood glucose, the data of a polynomial mathematical function is utilized [21].

Maximum glucose absorption is detected between 260 nm and 270 nm and the one of xyloses is from 245 nm to 255 nm. At 270 nm, absorption of xyloses is only half of the glucose. The 6 channels of the sensor detect the light absorbance at a given wavelength (R = 610nm; S = 680nm; T = 730nm; U = 760nm; V = 810nm; W = 860nm) on which blood glucose can be detected. We know that normal blood glucose is between 4.4 millimole and 6.7 millimole per liter (ie., between 0.8 and 1.2 grams per liter) taken at no more than 6.7 millimols per liter (1.2 grams per liter) two hours after having a meal [22].

People who do not have diabetes should have a value below 6.9 mmol / L (0.25 g / L) and those with diabetes have a value between 5.0-7.2 mmol / 0.9-1.3 g / L) before meals and less than 10 mmol / L (1.8 g / L) after meals, according to [22].

The application of near infrared spectroscopy on the human body is based on the fact that absorption of near infrared light from human body tissues contains important information about changes in hemoglobin concentration, which is very important for the detection of glucose in tissues. When a certain area of the brain is activated, it detects that the volume of blood in the area is changing rapidly [13]. Near infrared spectroscopy technology can be used as a rapid monitoring tool for cases of intracranial hemorrhage by placing the scanner on the head [11]. When it is internal bleeding from a stroke, the blood can be concentrated in a single location where the near infrared light will be more absorbed than in other locations [22].

The prototype proposed is based on optical spectroscopy that quantifies the level of glucose in human blood based on several near infrared sensors. The proposed prototype device has not yet been tested on the real human body to determine the level of blood glucose.

A module that contains more sensors stays on top and passes through the finger that sits on a device. The bottom module is equipped with an internal module to retrieve the signals transmitted by the sensors [22]. The process involves inserting the near infrared light beam into the test samples (tissues) and detecting the amount of light passing through these samples. Near infrared transmission spectroscopy is practiced on the fingertips or ear lobes, while for the forearms and cheeks, reflexive spectroscopy is not used due to the fact that the near infrared does not have the same penetration power. When near infrared light passes through a tissue, glucose is detected when the tissue absorption rate is very low. It should be noted that near infrared spectroscopy is renowned for its simple concept and its applications. This technique can be used to monitor the water content in the blood that can be avoided by selecting a specific infrared range.

The prototype (Figure 7) is an assembly between an Arduino board and two near infrared sensors of absorption and colorimetry. The signals are taken and processed by software resulting in a combination of Electrocardiography, Electromyography and light absorption through biological tissues. AS7263 is the near infrared version of the spectral sensor capable of measuring 610, 680, 730, 760, 810 and

 ISBN: 978-1-61208-792-4

860 nm of light, each with a maximum detection error of 20nm. The 6 light channels have the following wavelengths: R = 610nm; S = 680nm; T = 730nm; U = 760 nm; V = 810nm; W = 860nm.

TABLE II. PROTOTYPES RESULTS ACQUISITION

Glucose/nm	R-610	S-680	T-730	U-760	V-810	W-860
200 mg Gl	235.25	84.75	27.79	16.67	20.77	15.45
400 mg Gl	135.25	78.75	20.79	18.91	22.77	17.45
600 mg Gl	94.36	30.99	15.0	9.95	10.89	8.14
800 mg Gl	78.42	67.15	62.52	57.73	58.40	48.85
1000 mg Gl	65.42	30.99	57.52	55.22	49.30	43.15

For glucose measurement in vitro, a high glucose solution (100 mMol) was used and the near infrared spectra were measured. The software application receives the results from channels and tempF signals from a module based on an Arduino device to which a near infrared sensor module has been attached. Two aqueous glucose solutions were prepared in advance for in vitro testing. An initial solution of ~ 100 mMol and one ~200 mMol were prepared. Several readings were performed for each concentration. Finally, an average of each reference set was taken (Figure 7). If the glucose concentration (mMol) increases (reading), the output voltage increases.

Figure 7. Hardware prototype

VI. CONCLUSION AND FUTURE WORK

Channels analysis receiving multiple signals can be very useful in the diagnosis and prognosis of many diseases, especially when the channels contain rich information that can be extracted from Electrocardiography, Electromyography, Electroencephalography, near infrared spectroscopy etc. In this case, the Electrocardiography signal can be processed and correlated with other signals using specially developed algorithms to analyze and display more useful information in a comprehensive way. In the future, the program will try to implement periodograms, correlograms, and other signal analysis tools to display a rich variety of information that may be useful for bioengineering and medical staff [22].

The application aims to implement in the future:

- an advanced graphical display for Electrocardiography and Electromyography that can correlate with and illustrate muscles activity;
- an advanced graphic display of Electrocardiography arrhythmia;
- real time display of possible diseases based on Electrocardiography signals;
- attaching new sensors, which makes possible the display a heartbeat correlation with muscles activity in real-time;
- saving and creating a database of transmitted and processed values in real time to create a history.

ACKNOWLEDGEMENTS

This work has been funded by the Operational Program Human Capital of the European Funds Ministry through Financial Agreement 51675/09.07.2019, SMIS code 125125.

REFERENCES

[1] N. Lapique and Y. Benenson, “Digital switching in a biosensor circuit via programmable timing of gene availability”, *Nature Chemical Biology*, 14 October 2014, pp 1020–1027.

[2] L. Prochazka, B. Angelici, B. Häfliger and Y. Benenson, “Highly modular bow-tie gene circuits with programmable dynamic behavior”, *Nature Communications*, 14 October 2014, pp 1-12.

[3] B. Widrow et al., “Adaptive noise cancelling: Principles and applications”, *Proc. IEEE*, vol. 63, 1975, pp. 1692-1716.

[4] A. Bharadwaj and U. Kamath, “Techniques for accurate ECG signal processing”, Cypress Semiconductor Corp., February 2011, pp 1-7.

[5] G. Bianchi and R. Sorrentino, “Electronic filter simulation & design”, *McGraw-Hill Professional*, 2007, pp. 17–20.

[6] J. Smith, M. Jr. Jones and L. Houghton, Future of health insurance. N Engl. J. Med. 965, 1999, pp. 325–329

[7] S. Updike and G. Hicks, “The enzyme electrode”, Nature, Vol. 214, 1967, pp. 986–988.

[8] A. Caduff, M. Talary and P. Zakharov, “Cutaneous blood perfusion, as a perturbing factor for noninvasive glucose monitoring”, Diabetes Technol. Ther., vol. 12, 2010, pp. 1–9.

[9] G. Dongmin, D. Zhang, L. Zhanga and L. Guangming, “Non-invasive bloodglucose monitoring for diabetics by means of breath signal analysis”, Sensors and Actuators B: Chemical, vol. 173, October 2012, pp. 106–113.

[10] A. N. Bashkatov, "Optical properties of human skin, subcutaneous and mucous tissues in the wavelength range from 400 to 2000 nm", Journal of Physics D: Applied Physics, vol. 38, 2005, pp. 2543-2555.

[11] K. Kong, "Multiphoton microscopy in life sciences", Journal of Microscopy, vol. 200-2, 2000, pp.83-104.

[12] Jurgen C. de Graaff, "Influence of Repetitive Finger Puncturing on Skin Perfusion and Capillary Blood Analysis in Patients with Diabetes Mellitus", DutchHeartFoundation,1999, pp 1-12.

[13] L. Florea and D. Diamond, “Advances in wearable chemical sensor design for monitoring biological fluids,” Sensors Actuators B Chem., vol. 211, 2015, pp. 403–418.

[14] J. M. McMillin, “Clinical methods: The history, physical, and laboratory examinations,” Blood Glucose, 3rd ed., Boston, MA, USA: Butterworth, 1990, ch. 141.

[15] Glucosemeters4u.com, “Glucometers comparison,” 2015. [Online]. Available: http://www.glucosemeters4u.com/. [Accessed 9, 2020]

[16] C.-F. So, K.-S. Choi, T. K. S. Wong and J. W. Y. Chung, “Recent advances in noninvasive glucose monitoring,” Med. Devices Evidence Res., vol. 5, June 2012, pp. 45–52.

[17] A. Tura, S. Sbrignadello, D. Cianciavicchia, G. Pacini, and P. Ravazzani, “A low frequency electromagnetic sensor for indirect measurement of glucose concentration: In vitro experiments in different conductive solutions,” Sensors, vol. 10, no. 6, 2010, pp. 5346–5358.

[18] D M Nathan et. al., Diabetes Control and Complications Trial Research Group, ‘‘The effect of intensive treatment of diabetes on the

 ISBN: 978-1-61208-792-4

development and progression of long-term complications in insulin-dependent diabetes mellitus'', N. Engl. J. Med. 329, 977, 1993.
[19] L. H. Xu, Z. F. Liu, I. Yakovlev, M. Y. Tretyakov and R. M. Lees, Infrared Phys. Technol. 45, March 2004, pp. 31.
[20] C. S. Sunandana, Physical applications of photoacoustic spectroscopy [Online]. Available: http://onlinelibrary. wiley.com/doi/ I O. I 1050 102/abstract, February 15, 2006, 10 02/pssa.22.
[21] M. Ionescu, "Measuring and detecting blood glucose by methods non-invasive", pp 1-7, Ecai 2018 - International Conference – 10th Edition Electronics, Computers and Artificial Intelligence 28 - 30 June, 2018, Iasi, Romania, pp 5-6.
[22] M. Ionescu and P. Sever, "Algorithms of Absorbance and Colorimeter for Measuring Blood Glucose", pp 3-6, Atee 2019 - The 11th International Symposium On Advanced Topics In Electrical Engineering March 28-30, 2019, Bucharest, Romania, 1999, pp 5-6.
[23] BioLogic Signal Import Module, [Online] www.bio-rad.com/webroot/web/pdf/lsr/literature/4006229.pdf [Accessed 9, 2020]
[24] Wikipedia, Butterworth filter, [Online] en.wikipedia.org/wiki/Butterworth_filter
[25] Wikipedia, Haar filter, [Online] https://en.wikipedia.org/wiki/Haar_wavelet

 ISBN: 978-1-61208-792-4

A Versatile Combination of Classifiers for Protein Function Prediction

Haneen Altartouri

Institute for Neural Computation
Ruhr-University Bochum, Germany
Email: haneen.altartouri@ini.rub.de

Tobias Glasmachers

Institute for Neural Computation
Ruhr-University Bochum, Germany
Email: tobias.glasmachers@ini.rub.de

***Abstract*—Protein classification problems can be addressed with a wide range of machine learning methods. Top performance is achieved with a variety of methods, and the best method depends on the data set under study. Therefore, a minimal requirement for a general proceeding is to consider multiple classifiers and to tune their hyperparameters. Further highly task-specific performance gains can be achieved through additional measures like feature selection, which is particularly important for high-dimensional descriptors, or with separate classifiers for different clusters. In this paper, we design a versatile classifier with the aim to combine all of the above options, but with robust defaults and fallback options. We demonstrate systematic performance improvements across a wide range of protein prediction problems.**

Keywords–protein classification; feature selection; clustering.

I. INTRODUCTION

In recent years, an increasing number of protein sequences has been extracted through high-throughput sequencing technologies. As a consequence, identifying functions of these sequences became one of the most interesting and challenging topics in bioinformatics [1]. Different computational approaches to predict the functions of protein sequences in an efficient way have been explored.

In most approaches, prediction of protein functions is based on supervised classification algorithms, which construct a learning model determining the relation between the protein sequences and their functions. The trained model can help in predicting the function of the new sequences. For many protein sequence datasets, the predictive accuracy achievable this way is not fully satisfactory. Classification is often easy if the discriminative features are homogeneous for the whole data set. For heterogeneous datasets, we should, therefore, find homogeneous regions and address them with separate classifiers. In particular, tuning feature vectors and hyperparameters specifically for each region can improve the overall performance of the prediction.

Clustering can be used for obtaining homogeneous regions inside a dataset. Clustering is a class of unsupervised learning methods, which group similar data based on their properties without depending on labels. Sequences within a cluster are more similar to each other than sequences in other clusters. In this research, clustering is applied prior to classification to construct meaningful homogeneous sub-datasets in order to improve the performance of the classification.

Several researchers explored combinations of classification and clustering in different applications, such as disease diagnosis [2], text classification [3][4], and network traffic classification [5]. Their results show that this combination can improve predictive performance in many cases. In the field of protein problems, clustering has been used for many years to group proteins into families [6]–[8]. However, the effect of using clustering algorithms to reduce the heterogeneity of the protein datasets in order to improve the performance of supervised prediction of the function of proteins has not yet been studied. In this study, we close this research gap.

Clustering can be used to improve the performance of classification either by reducing the feature representation [9][10], or by extracting structural information form the data. In the second case, clustering is used to discover a structure in the training examples. Some approaches use the clustering information by expanding the feature vectors with new attributes extracted from clusters. For example, Kyriakopoulou et al. [3] have enhanced the text classification performance for the spam detection problem by grouping the training data into clusters and then each cluster contributes one meta-feature to the feature space of the training and testing data. Finally, they used a Support Vector Machine (SVM) classifier [11] to classify the expanded data containing the original features and meta features. Their experimental results demonstrate that the inclusion of meta features improves the classification accuracy. Xiao et al. [2] constructed a clustering-based attribute selection measure from the clustering step. This attribute called hybrid information gain ratio takes, into consideration the class label and the cluster of the sample. They trained a C4.5 decision tree based on this ratio. Their results show that using the new attribute improved the performance of classification for healthcare and disease diagnoses problems.

The most commonly used approach to combine clustering with classification depends on breaking down a complex classification problem into simpler problems using clustering, then training a single classifier on each cluster. Rajamohamed et al. [12] applied k-means and rough k-means to group credit card churn samples into clusters. Then, they divided each cluster into testing and training data to apply a classifier within each cluster. Different classifiers were tested and the results showed that combining the rough k-means with SVMs improved the classification performance compared to using a single classifier. Gaddam et al. [13] combined k-means clustering and the ID3 decision tree learning methods for classifying anomalous and normal activities in a network, an active electronic circuit, and a mechanical mass beam system. In their work, the dataset is divided into k subsets based on the

 ISBN: 978-1-61208-792-4

similarity, then the ID3 classifier is trained on each cluster. The results showed that this hybrid achieved better performance than a single global classifier. Fradkin [14] applied clustering within classes to artificially increase the number of classes, then a multi-class classifier was trained to distinguish between the clusters. The results showed that clustering within classes can improve the classification in many cases.

In this work, we aim at designing a general approach for improving the prediction of protein sequences. To this end, we reduce the heterogeneity of the dataset (if it exists) by constructing meaningful homogeneous regions (sub-datasets), and then handling each sub-dataset separately as a small problem inside a large complex dataset. This allows us to train different classifiers in each sub-dataset, and importantly, to select features and tune the hyper-parameters of these classifiers separately. We also introduce an option to return back to the classifier trained on the whole complex dataset in case of weakness, as proposed in [14]. This is an important mechanism that greatly stabilizes the results and that avoids over-fitting to small clusters. We analyse the features inside each sub-datasets and apply feature reduction to select locally significant features to help in distinguishing sequences that belong to different classes and improve the sub-dataset classifier, which implies improving the overall performance. In contrast, existing hybrid models apply reduction only *before* the clustering to select globally significant features [15][16].

We tested two methods for features reduction, and different classifiers were trained and their hyperparameters tuned. We evaluated the effect of the proposed approach on six protein function prediction problems. Our results show that the proposed approach improves the performance of the prediction in most cases, without degrading performance in other cases.

The remainder of this paper is organized as follows: the next section describes the proposed approach in detail. Section III briefly introduces the benchmarks of this study. In section IV, we present the experimental results and discuss our findings. In section V, we close with conclusions from our work.

II. The proposed Approach

We propose a versatile approach for the classification of protein functions consisting of the following steps:

1) encode variable-length protein sequences with a fixed-length descriptor or feature vector,
2) cluster the dataset into sub-datasets,
3) apply feature reduction inside each sub-dataset
4) train multiple classifiers for each sub-dataset, and
5) decide for each subset which classifier to use.

The proposed approach is summarised in Figure 1, where the number of sub-datasets is 2.

We go through these steps one by one.

A. Representing Protein Sequences

We aim to represent protein sequences in a form that can be easily handled by machine learning algorithms. The main challenge is that sequences can have different lengths. For many learning machines, we need to encode these sequences into a fixed-length descriptor that extracts the relevant features. In this study, we rely on Chou's Pseudo Amino Acid Composition (PseAAC) descriptors for protein sequence encoding [17]. It has been demonstrated that PseAAC descriptors are extremely effective features for protein problems [18]–[22]. PseAAC represents amino acid frequencies and, in addition, it preserves most of the sequence-order information [23][24]. A protein sequence is represented by 20 + λ numerical features. The first 20 features are the occurrence frequencies of the 20 amino acids. The remaining λ descriptors encode the sequence order. For a detailed description of PseAAC, we refer to [17][25].

PseAAC depends on Physico-Chemical Properties (PCPs) of the amino acids to represent the sequence. A PCP is a (scalar) physical or chemical feature of the amino acid. In this work, we used two sets of PCPs. The first set is rather small and it consists of three PCPs used in Chou's work [17]: hydrophobicity, hydrophilicity, and side chain mass. The other set is more rich: it contains fifty non-redundant PCPs of amino acids proposed by Georgiev [26], such as: normalized relative frequency of double bend , pK (-COOH) , relative mutability and flexibility parameter for two rigid neighbors .

B. Clustering the Dataset into Sub-datasets

The second step in this approach is clustering the dataset (D) into sub-datasets (SDs). Clustering is a process of grouping the samples into meaningful clusters, with the aim to identify groups of homologous protein sequences. This way, we break down a complex protein prediction problem into a set of simpler problems. To keep the approach manageable, we applied only one clustering algorithm, which is k-means [27].

K-means is a partition clustering algorithm, where each sample belongs to a unique cluster. It is widely used in bioinformatics [28][29] because it is simple, easy to implement, and reasonably fast. For details on k-means, we refer to [27]. To apply k-means, we need to select the number of clusters (k), which is a hyperparameter of the method. Optimal clustering requires dataset-specific tuning of k [30] and, since there is no method that is guaranteed to find the optimal value for k (determining the right number of clusters is still an open problem in clustering research), we apply an array of pre-defined values for k and study its effect on the overall performance.

C. Reducing Feature Vector Dimensionality

After clustering, we apply feature reduction inside each sub-dataset in order to optimally separate sequences that belong to different classes. Yang et al. [22] showed that applying feature reduction on PseAAC features can improve the performance of protein classification.

Feature reduction is an important step before applying machine learning algorithms if some of these features are irrelevant or redundant [31], and possibly add noise. These redundant and irreverent features do not contribute to the accuracy of a predictive model and sometimes even reduce its performance. Then, removing these features can improve the accuracy of the model, or decrease the size of the feature space without affecting the prediction accuracy [31]. To study the effect of reducing the feature vector, we have tested two reduction techniques: the Recursive Feature Elimination (RFE) algorithm as a feature selection technique, and Principal Component Analysis (PCA) as a feature extraction technique. For more details about these algorithms, please see [32] and [33], respectively.

 ISBN: 978-1-61208-792-4

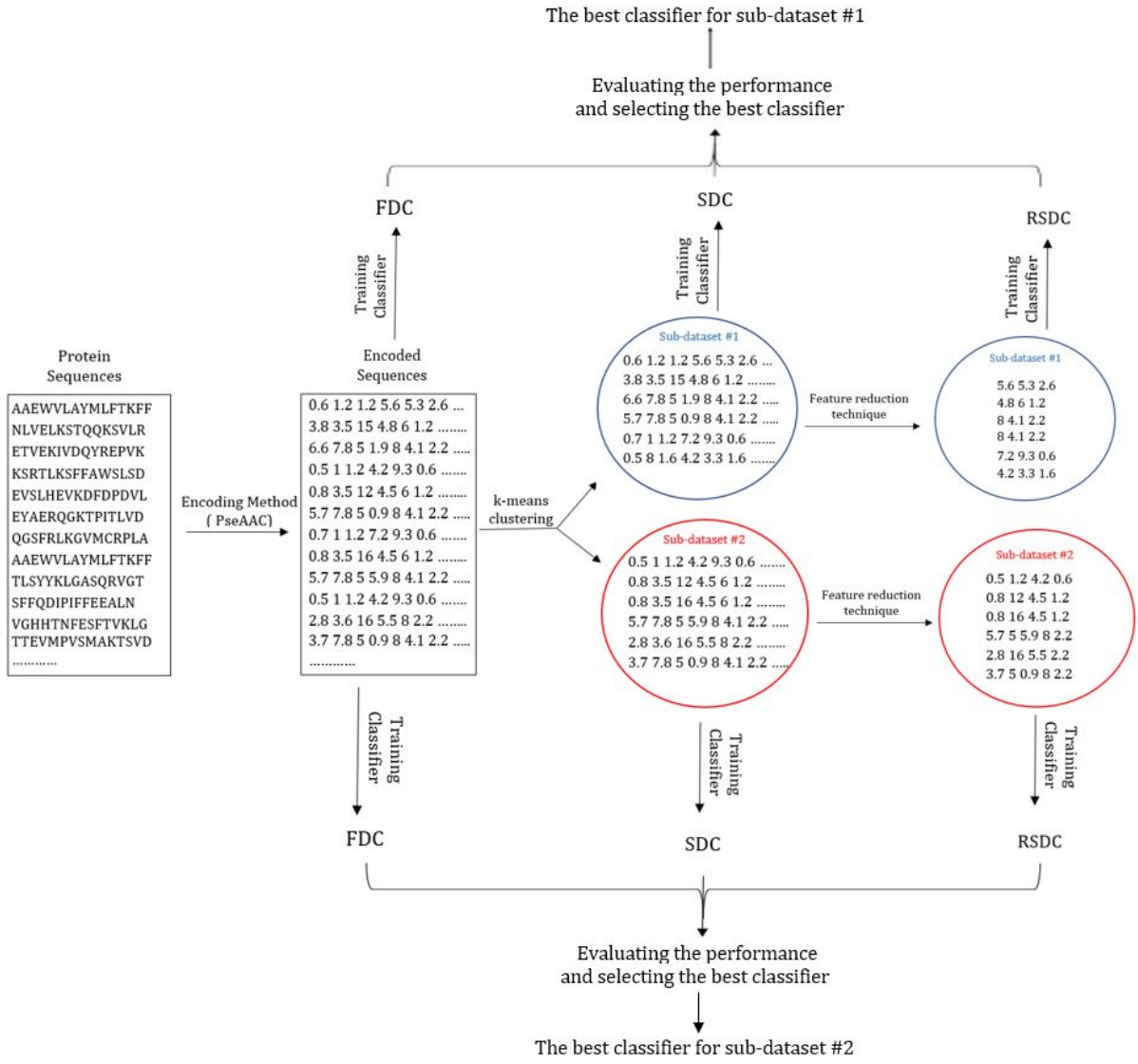


Figure 1. The proposed Approach.

D. Training Classifiers on the Sub-datasets

The central step of this approach is to train a classifier. We apply a set of standard supervised learning algorithms that build a predictive model based on the training data. The model is used to classify new samples (testing data).

After dividing the protein dataset into k simpler problems (sub-datasets) in the clustering step, we train one classifier per sub-dataset, so each classifier focuses on classifying the proteins in a specific region. This step includes hyperparameter tuning, where the tuning procedure employed depends on the classifier at hand.

For some sub-datasets, we cannot train a reliable classifier because there is not enough data. In such a case, we need to use another classifier to process data within this region. Therefore, we resort to the classifier trained on the whole dataset as proposed by [14], which we refer to as the "Full Dataset Classifier" (FDC).

From the previous steps, we have sub-datasets with the full features, and a view on these sub-datsets with reduced features. Therefore, for each sub-dataset, we train two classifers: one using the full feature set, called "Sub-dataset Classifier" (SDC), and the other using the reduced feature set, called "Reduced Sub-dataset Classifier" (RSDC).

E. Classifier Selection

For each sub-dataset, we have up to three classifiers available: FDC, SDC, and RSDC. We select one of them into our predictive model. To this end, we estimate the performance of all three classifiers by means of cross-validation, restricted to the sub-dataset. We select the classifier with the highest AUC (area under the receiver operator characteristic curve). The classifier with the highest AUC is then responsible for classifying all data in the corresponding cluster.

This last step proves to be crucial for achieving high predictive accuracy. Our intuition on the classifier selection step is as follows. Our basic hypothesis is that protein datasets can consist of meaningful sub-datasets, where different feature sets are discriminative in each subset. That logic leads straight

to the construction of the RSDC classifiers. However, using RSDC in all cases does not work well. On the one hand, this is the case if there is no pronounced subset structure in the dataset under study. On the other hand, some clusters represent harder classification problems than others. In the hard clusters, RSDCs suffer from the reduced amount of data. Then, feature reduction and even model training can be unreliable and subject to a high risk of over-fitting, while the global classifier makes better use of surrounding points. Therefore, it is important to provide the robust fallback options of reverting to the SDC or even to the FDC.

When a new sequence is incoming, we measure its similarity with the centroids of the sub-datasets and assign it to the nearest centroid. Then, we apply the chosen classifier based on the previous cases. We want to stress that the decisions in all five design stages, namely how to represent sequences, how many clusters to use, which learning machine to apply and how to set its hyperparameters, which features to reduce, and which classifier to pick for each cluster, are all made based on cross-validation, so that we end up with a single hybrid classifier.

III. Benchmark Datasets

To evaluate the performance of the proposed approach, we have used six protein datasets. Table 1 summarize all these datasets. The first three datasets contain long sequences and the last three contain peptide sequences. All datasets are binary classification problems. We have split each dataset into a training and a testing set.

IV. Experiments and Results

The main aim of our experiments is to demonstrate the benefit of our approach as compared to the FDC, which is a natural baseline. We used the following parameter settings in our experiments. The PseAAC encoding, as described in Section II-A consists of two parts. The weight of the features representing the sequence order was set to $w = 1/2$. The length of the shortest sequence was set to $\lambda = 7$ for peptides and to $\lambda = 30$ for long protein sequences. These settings result in 27 and 50 PreAAC features, respectively.

Since the distribution of the sequences differs from one dataset to another, we have to tune the number of sub-datsets (k) for each dataset. For small datasets (Caspase-3, DNA-binding, and Antioxidant proteins), k is selected from a range of 2 to 7 with a step size of 1, and for the other datasets k is selected from a range of 5 to 30 with a step size of 5.

We have tuned the hyper-parameters for the FDC, SDC, and RSDC classifiers using 5-fold cross-validation repeated 3 times, and we have applied an inner cross-validation for RFE to assess generalization on an independent dataset and avoid over-fitting [40]. Cross-validation was also used for estimating the quality of the classifiers.

To evaluate the performance of the classifiers, we depend on sensitivity (SN), specificity (SP), and Matthew's Correlation Coefficient (MCC) [41]. We also use the Receiver Operating Characteristics (ROC) curve. This curve illustrates the achievable trade-offs between true positive rate and false positive rate. The quality of the ROC curve is measured by computing the area under the ROC curve (AUC) [42]. All values displayed in this research are computed on independent test sets.

A. Selecting the Best Classifier for the Proposed Approach

In our experimental study, we considered the following types of classifiers: SVM [43], Random Forest (RF) [44], Artificial Neural Network (ANN) [45], and eXtreme Gradient Boosting (xGBoost) [46]. We trained FDCs with all four learning machines with protein sequences represented by PseAAC descriptors using 50 PCPs [26] and using 3 PCPs [17]. Figure 2 shows the ROC curves and AUC values of all four classifiers on the six datasets.

For some datasets, there are significant differences in classifier performance and, in some cases, the number of PCPs makes a difference. It can be deduced from Figure 2 that different classifiers work well for different problems, a fact we account for with our approach. SVMs are a solid choice for most datasets using 50 PCPs, while RF is the best choice when using 3 PCPs. Therefore, we restrict ourselves to SVM and RF classifiers. In the following, we present two sets of experiments. First, we assess the effect of splitting the dataset into sub-datasets by comparing SDCs and FDC, then we investigate the effect of feature learning inside each cluster by comparing SDCs and RSDCs.

B. Impact of Training Multiple Classifiers

In order to study the effect of training separate classifiers in some regions within the dataset, we run several experiments varying k (number of sub-datasets). Table II shows a comparison between using FDC only (baseline), and SDCs with the option to use FDC for weak SDCs based on AUC values, as described in Section II-B. The results in the table represent the best cross-validation performance over k and the machine-specific hyperparameters. We observed that, in the most cases, applying SDCs with option to resort to the FDC in a per-cluster manner improves the performance over using FDC only, except for the AMP dataset, where the improvement is very small for both SVM and RF. For three datasets, the effect of using multiple classifiers on the overall AUC is small (1% improvement), while for RNA-binding, Antioxidant, and MHCII datasets, we have achieved a respectable improvement of 3% in the AUC values. For the other metrics, we achieved significant improvements.

The results clearly indicate that classification based on SDCs with the fallback option to the FDC consistently improves over the FDC baseline.

The best result is obtained for the RNA-binding dataset: 3% and 12% improvement for the AUC and MCC values, respectively, by grouping the dataset into 5 sub-datsets and using SVM-SDCs inside 3 groups, while the other 2 groups depend on the SVM-FDC. On the other hand, the best result obtained with an RF is 1% improvement in both AUC and MCC where $k = 5$. For MHCII peptides, we obtained the best result by grouping the dataset into 15 clusters, but using SDCs for 3 groups only. This result indicates that, in some cases, significant improvements are achievable by handling only a few sensitive regions with specific SDCs. Furthermore, it is worth noting that, for most datasets, SVM achieved better improvement than RF inside SDs, except for the Antioxidant dataset.

C. Impact of Reducing the Features Inside Sub-datasets

As detailed in Section II-C, the proposed approach allows to reduce the features separately for each sub-dataset. Figure 3

 ISBN: 978-1-61208-792-4

TABLE I. DATASETS USED FOR THE APPROACH EVALUATION

Dataset	# of Positives	# of Negatives	sequences length
DNA-binding proteins [34]	523 binding proteins	543 non binding proteins	50 - 1323 amino acids
Antioxidant proteins [35]	250 antioxidant	1547 non-antioxidant	31 - 1463 amino acids
RNA-binding proteins [36]	2780 binding proteins	7077 non binding proteins	50 - 8799 amino acids
Antimicrobial peptides (AMP) [37]	869 AMPs	2405 non-AMPs	8 - 103 amino acids
Caspase 3 human substrates [38]	247 cleaved peptides	247 non-cleaved peptides	14 amino acids
Major Histocompa. Complex II (MHCII) [39]	3510 binding peptides	1656 non-binding peptides	9 - 37 amino acids

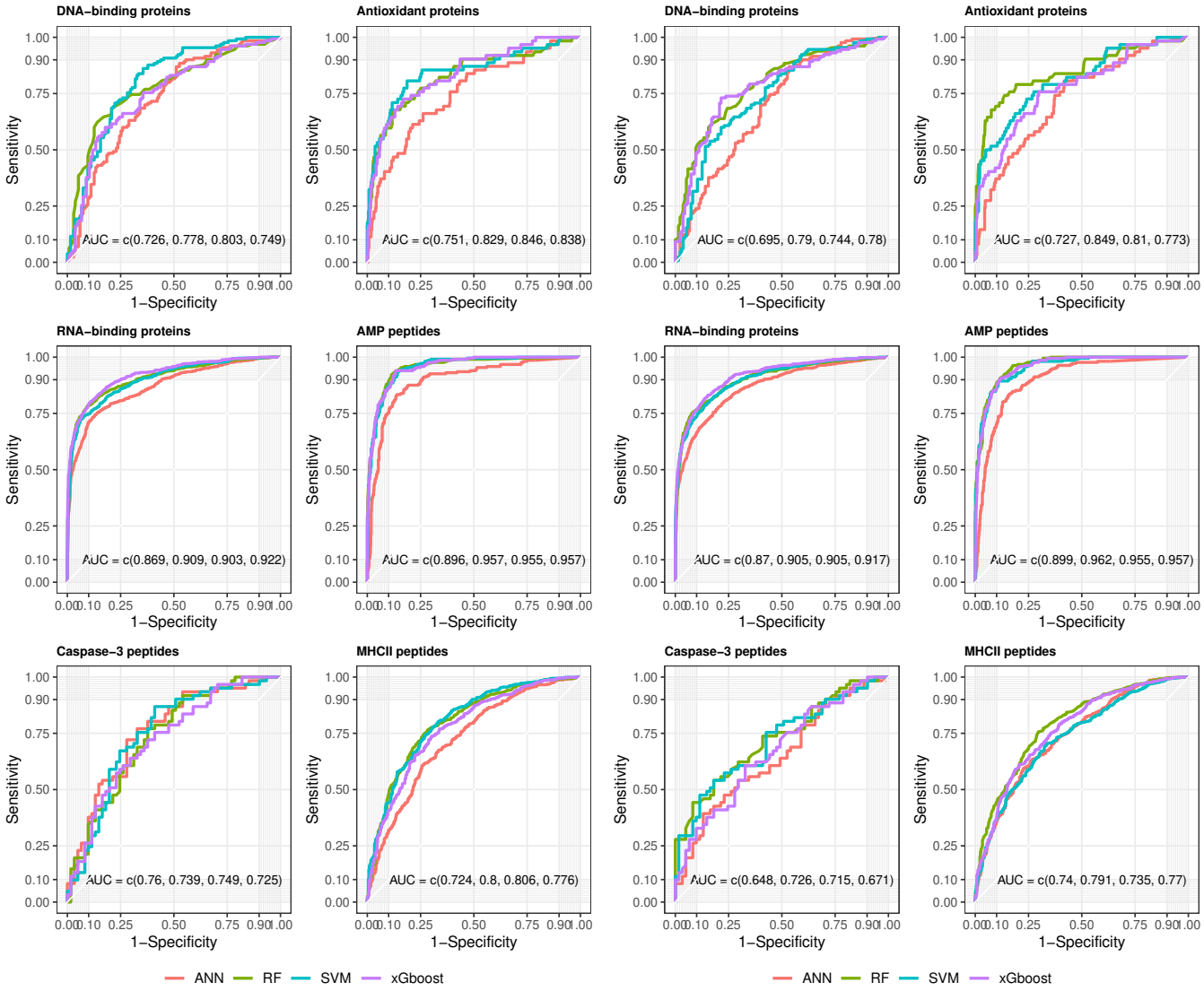


Figure 2. ROC curves for 4 different classifiers: ANN, RF, SVM, and xGBoost (with AUC values in parentheses), using 50 PCPs (columns 1 and 2) and 3 PCPs (columns 3 and 4).

illustrates that different feature sets can be discriminative in different sub-datasets. It shows relative feature importance for two sub-datasets of MHCII at $k = 20$. The importance of the features differs not only between the two sub-datasets, but also from the full dataset. Therefore, applying feature reduction on a per-cluster basis has the potential to improve overall performance.

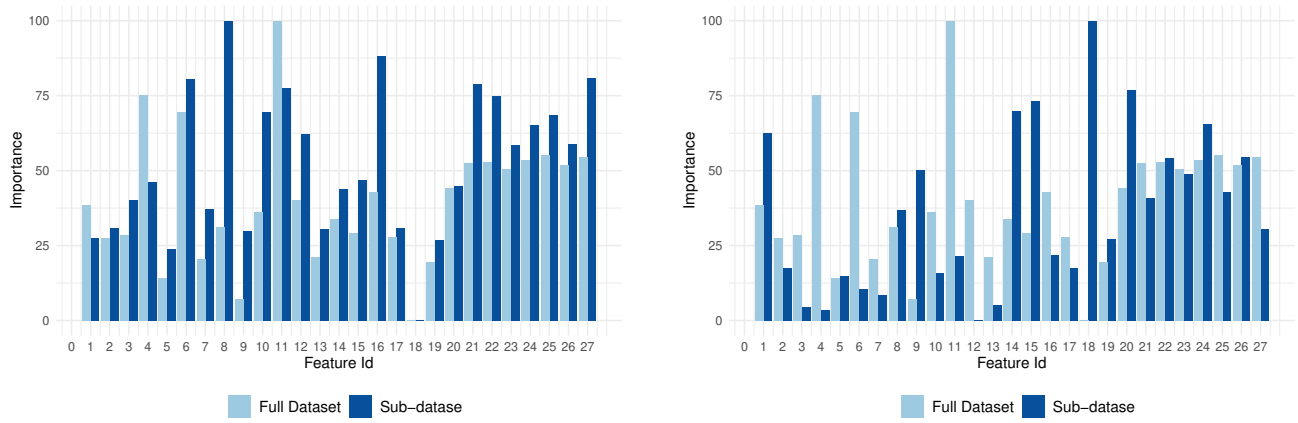
We ran two sets of experiments to study the effect of reducing the features inside the sub-datasets. In the first case, we select the best set of features that maximize the performance of prediction using RFE and, in the second case, we apply PCA to extract a new descriptor to represent the data by selecting principal components that cover at least 95% of the total variance. Like for SDCs, if the resulting RSDC turned out to be unreliable, then we returned back to FDC or SDC based on the cross-validated AUC scores.

Table II shows the best improvement of RSDCs with options of reverting to SDCs or FDC, compared to SDCs with option of reverting to FDC. The results show that RFE outperforms PCA in reducing the features inside the sub-

 ISBN: 978-1-61208-792-4

TABLE II. COMPARISON BETWEEN USING ONLY FDC, SDCs WITH OPTION TO USE FDC IN THE WEAK CLUSTERS, AND RSDCs WITH OPTIONS TO USE FDC OR SDCs IN THE WEAK CLUSTERS FOR 6 BENCHMARKS.

Method	SVM						RF					
	k	AUC	SEN	SPE	MCC	(FDC,SDC, RSDC(algo.))	k	AUC	SEN	SPE	MCC	(FDC,SDC, RSDC(algo.))
DNA-binding proteins												
- FDC only (the baseline)	-	0.8033	0.7769	0.6963	0.4744	-	-	0.7899	0.6692	0.7556	0.4266	-
- SDCs with reverting option to FDC	4	0.8197	0.7769	0.763	0.5398	(2,2,-)	3	0.7859	0.6923	0.7778	0.4721	(2,1,-)
- RSDC with reverting option to FDC or SDCs (RFE, PCA)	2	0.8433	0.8154	0.7481	0.5643	(1,0,1(RFE))	2	0.8348	0.7308	0.8074	0.5401	(0,0,2(PCA))
Antioxidiant proteins												
- FDC only (the baseline)	-	0.8405	0.68	0.8987	0.5193	-	-	0.8493	0.7419	0.8627	0.5032	-
- SDCs with reverting option to FDC	4	0.8591	0.7333	0.8966	0.5545	(2,2,-)	4	0.8706	0.7097	0.9275	0.5991	(3,1,-)
- RSDC with reverting option to FDC or SDCs (RFE, PCA)	4	0.8681	0.7333	0.9009	0.5625	(2,1,1(RFE))	-	no improvement achieved				-
RNA-binding proteins												
- FDC only (the baseline)	-	0.903	0.6331	0.9582	0.6548	-	-	0.9053	0.636	0.9661	0.6727	-
- SDCs with reverting option to FDC	5	0.9301	0.7942	0.9588	0.7788	(2,3,-)	5	0.9136	0.659	0.9644	0.687	(4,1,-)
- RSDC with reverting option to FDC or SDCs (RFE, PCA)	5	0.9412	0.8187	0.961	0.801	(2,0,3(RFE))	10	0.9344	0.705	0.9638	0.721	(5,0,5(RFE))
AMP peptides												
- FDC only (the baseline)	-	0.9552	0.765	0.9418	0.7247	-	-	0.9624	0.7926	0.9484	0.7574	-
- SDCs with reverting option to FDC	5	0.9634	0.788	0.9434	0.7451	(3,2,-)	25	0.9619	0.7926	0.9567	0.7724	(21,4,-)
- RSDC with reverting option to FDC or SDCs (RFE, PCA)	30	0.956	0.8203	0.9401	0.7638	(23,3,4(RFE))	5	0.9741	0.8295	0.9551	0.7967	(2,0,3(RFE))
Caspase 3 peptides												
- FDC only (the baseline)	-	0.7487	0.623	0.7541	0.3803	-	-	0.7263	0.7377	0.5246	0.2685	-
- SDCs with reverting option to FDC	2	0.7474	0.6393	0.7705	0.4134	(1,1,-)	6	0.7417	0.7377	0.5574	0.3	(5,1,-)
- RSDC with reverting option to FDC or SDCs (RFE, PCA)	2	0.7565	0.7541	0.7869	0.5413	(0,0,2(RFE))	-	no improvement achieved				-
MHCII peptides												
- FDC only (the baseline)	-	0.8034	0.7605	0.6981	0.4396	-	-	0.7909	0.7571	0.7029	0.4401	-
- SDCs with reverting option to FDC	15	0.8371	0.7765	0.7488	0.5022	(12,3,-)	30	0.8042	0.7537	0.715	0.4472	(19,11,-)
- RSDC with reverting option to FDC or SDCs (RFE, PCA)	15	0.843	0.7879	0.7536	0.5192	(10,2,3(RFE))	20	0.8475	0.7697	0.7754	0.5182	(8,2,10(RFE))



(a) Sub-dataset #1, where $k = 20$

(b) Sub-dataset #2, where $k = 20$

Figure 3. Comparison between the importance of features inside the full dataset and inside sub-datsets using Gini importance [44]

datasets for most of the cases except for the DNA-binding dataset, where training an RF on PCs improves the overall performance.

Although using the proposed approach can improve the overall performance either using SVM or RF as classifiers inside the sub-datasets, we have achieved the highest performance using the SVM classifier in most cases, except for AMP peptides.

For the DNA-binding dataset, we achieved an improvement of 3% for both AUC and MCC by grouping the dataset into 2 sub-datasets with SVM-RFE for one subset, while the other set uses the FDC. In effect, we enhance the performance by 4% for AUC and 9% for MCC compared to the FDC baseline. On the other hand, for $k = 2$ and using PCA and RF on these two sub-datasets, we achieved 5% and 4% improvement for AUC and MCC compared to the SDC without feature reduction, which corresponds to improvements of 5% and 12% for AUC and MCC compared to the FDC baseline.

For the RNA-binding dataset, we have improved the MCC by 3% using SVM-RFE with a very small improvement in AUC, while we achieved 2% and 4% improvement for AUC and MCC using RF-RFE.

For Antioxidant, we have achieved only 1% improvement for both AUC and MCC using SVM-RFE compared to using

SDCs without features selection, and no improvement using RF-RFE and RF-PCA. This indicates that, unsurprisingly, feature reduction does not help for all problems, and we can just depend on using RF with SDCs.

As mentioned in the previous section, SDCs barely improve over the FDC on the AMP dataset. In contrast, splitting the dataset into 5 clusters and applying RF-RFE inside of three of them improved the baseline by about 4% for MCC, with a very small improvement in AUC.

For Caspase 3, we did not improve the overall AUC of the classifier, but we achieved a significant improvement in the MCC value (about 13% improvement) using SVM-RFE compared to use the SDCs with FDC option. On the other hand, for the MHCII dataset, if we depend on the SVM as a classifier algorithm inside the sub-datasets, feature reduction did not pay off, since the improvement was very small (about 1%). However, in order to achieve similar results with an RF, we need to group the dataset into 20 sub-datasets and apply RF-RFE inside 10 of them.

Going beyond predictive performance, we also analyzed the role of the features selected within the clusters. In most cases, RFE shows that the frequencies of amino acids play an important role in classifying the sequences inside the clusters, while the sequence order has a higher impact on classifying the full dataset. Figure 4 illustrates the rank of the optimal set of features for 9 sub-sets of MHCII out of $k = 20$, compared to the full dataset using RF-RFE. For datasets containing long protein sequences, RFE shows that the optimal sets of features for clusters contain only a bit more than 50% of all available descriptors, and most of theses descriptors represent amino acid frequencies.

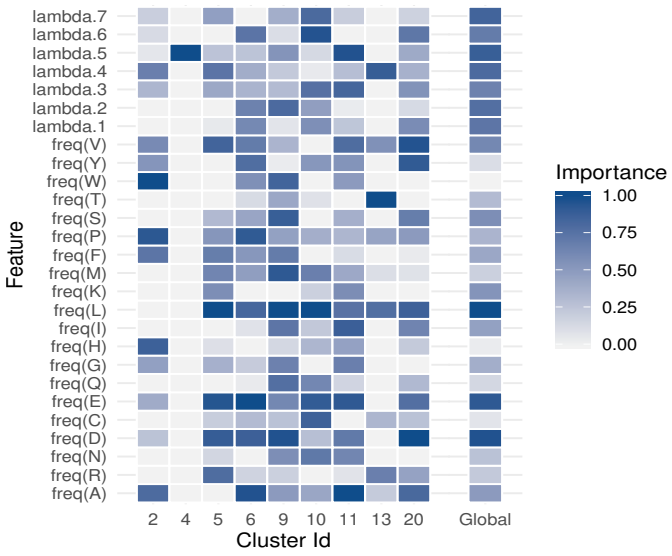


Figure 4. The importance of features based on RF-RFE

V. CONCLUSION

We have studied the effect of exploiting homogeneous sub-datasets inside protein sequence data by training multiple classifiers on sub-datsets. The proposed approach handles each sub-dataset as a separate classification problem that requires tuning the hyper-parameters and finding the best features separately. More hyperparameter choices on smaller datasets can potentially give rise to over-fitting. Therefore, it is imperative for robust performance to allow the classifiers to revert to classifiers trained on all features, and even on the full dataset, as fallback options.

In this study, we have evaluated the performance of SVM and RF classifiers inside the sub-datsets, and RFE and PCA are tested as a reduction feature algorithms. SVM and SVM-RFE achieved good performance for most datasets. The performance of the proposed approach depends on the number of sub-datasets, the encoding method, and for each cluster the classifier with its hyperparameters and the feature reduction method applied. We find that, for different datasets, the best performance is achieved with different approaches. Our approach is sufficiently versatile to account for this finding.

The results indicate that the proposed approach improved the overall performance of function prediction of protein sequences in most cases. Hence, they indicate that many protein sequence datasets suffer from heterogeneity.

REFERENCES

[1] R. Saidi, M. Maddouri, and E. Mephu Nguifo, “Protein sequences classification by means of feature extraction with substitution matrices,” BMC Bioinformatics, vol. 11, 2010, p. 175.

[2] J. Xiao, Y. Tian, L. Xie, and J. Huang, “A hybrid classification framework based on clustering,” IEEE Transactions on Industrial Informatics, 2019, pp. 1–1.

[3] A. Kyriakopoulou and T. Kalamboukis, “Combining clustering with classification for spam detection in social bookmarking systems,” RSDC, 2008.

[4] A. Thomas and M. Resmipriya, “An efficient text classification scheme using clustering,” Procedia Technology, vol. 24, 2016, pp. 1220–1225.

[5] J. Erman, M. Arlitt, and A. Mahanti, “Traffic classification using clustering algorithms,” Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data, MineNet’06, vol. 2006, 2006, pp. 281–286.

[6] A. Krause, J. Stoye, and M. Vingron, “Large scale hierarchical clustering of protein sequences,” BMC Bioinformatics, vol. 6, 2005, p. 15.

[7] W. bang Chen and C. Zhang, “A hybrid framework for protein sequence clustering and classification using signature motif information,” Integrated Computer-Aided Engineering, vol. 16, 2009, pp. 353–365.

[8] L. Szilágyi, L. Medvés, and S. M. Szilágyi, “A modified Markov clustering approach to unsupervised classification of protein sequences,” Neurocomputing, vol. 73, 2010, pp. 2332–2345.

[9] S. Chormunge and S. Jena, “Correlation based feature selection with clustering for high dimensional data,” Journal of Electrical Systems and Information Technology, vol. 5, 2018, pp. 542–549.

[10] X. Zhu et al., “A new unsupervised feature selection algorithm using similarity-based feature clustering,” Computational Intelligence, vol. 35, 2018.

[11] C. Cortes and V. Vapnik, “Support-vector networks,” Machine Learning, vol. 20, no. 3, 1995, pp. 273–297.

[12] R. Rajamohamed and J. Manokaran, “Improved credit card churn prediction based on rough clustering and supervised learning techniques,” Cluster Computing, vol. 21, 2018, pp. 1–13.

[13] S. Gaddam, V. Phoha, and K. Balagani, “K-means+id3: A novel method for supervised anomaly detection by cascading k-means clustering and id3 decision tree learning methods,” Knowledge and Data Engineering, IEEE Transactions on, vol. 19, 2007, pp. 345–354.

[14] D. Fradkin, “Within-class and unsupervised clustering improve accuracy and extract local structure for supervised classification,” PhD thesis, Rutgers, The State University of New Jersey, 2006.

[15] Y. K. Alapati and K. Sindhu, “Combining clustering with classification: A technique to improve classification accuracy,” International Journal of Computer Science Engineering (IJCSE), vol. 5, 2016, p. 3.

 ISBN: 978-1-61208-792-4

[16] H. Malik and J. Kender, "Classification by pattern-based hierarchical clustering." From Local Patterns to Global Models Workshop (ECML/PKDD 2008), Antwerp, Belgium, 2008.

[17] K.-C. Chou, "Prediction of protein cellular attributes using pseudo-amino acid composition," Proteins, vol. 43, 2001, pp. 246–55.

[18] Y. Fang, Y. Guo, Y. Feng, and M. Li, "Predicting dna-binding proteins: Approached from chou's pseudo amino acid composition and other specific sequence features," Amino Acids, vol. 34, 2008, pp. 103–9.

[19] D. Georgiou, T. Karakasidis, J. Nieto, and A. Torres-Iglesias, "Use of fuzzy clustering technique and matrices to classify amino acids and its impact to chou's pseudo amino acid composition," Journal of Theoretical Biology, vol. 257, 2008, pp. 17–26.

[20] P. Wang et al., "Prediction of antimicrobial peptides based on sequence alignment and feature selection methods," PloS One, vol. 6, 2011, p. e18476.

[21] N. Xiaohui et al., "Using the concept of chou's pseudo amino acid composition to predict protein solubility: An approach with entropies in information theory," Journal of Theoretical Biology, vol. 332, 2013, pp. 392—396.

[22] R. Yang, C. Zhang, L. Zhang, and R. Gao, "A two-step feature selection method to predict cancerlectins by multiview features and synthetic minority oversampling technique," BioMed Research International, vol. 2018, 2018, pp. 1–10.

[23] K.-C. Chou and H.-B. Shen, "Euk-mploc: A fusion classifier for large-scale eukaryotic protein subcellular location prediction by incorporating multiple sites," Journal of Proteome Research, vol. 6, 2007, pp. 1728–1734.

[24] ——, "Chou, k.c. & shen, h.b. review: recent progresses in protein subcellular location prediction. anal. biochem. 370, 1-16," Analytical Biochemistry, vol. 370, 2007, pp. 1–16.

[25] K.-C. Chou, "Pseudo amino acid composition and its applications in bioinformatics, proteomics and system biology," Current Proteomics - CURR PROTEOMICS, vol. 6, 2009, pp. 262–274.

[26] A. Georgiev, "Interpretable numerical descriptors of amino acid space," Journal of Computational Biology, vol. 16, no. 5, 2009, pp. 703–723.

[27] J. A. Hartigan and M. A. Wong, "A k-means clustering algorithm," Applied Statistics, vol. 28, no. 1, 1979, pp. 100–108.

[28] T. Chappell, S. Geva, and J. Hogan, "K-means clustering of biological sequences," 22nd Australasian Document Computing Symposium, 2017, pp. 1–4.

[29] A. Bustamam, H. Tasman, N. Yuniarti, Frisca, and I. Mursidah, "Application of k-means clustering algorithm in grouping the dna sequences of hepatitis b virus (hbv)," AIP Conference Proceedings, vol. 1862, 2017, p. 030134.

[30] F. Shahnaz, M. Berry, V. Pauca, and R. Plemmons, "Document clustering using nonnegative matrix factorization," Information Processing & Management, vol. 42, 2006, pp. 373–386.

[31] E. Alpaydin, Introduction to Machine Learning, 2nd ed. The Adaptive Computation and Machine Learning series, Massachusetts Institute of Technology, 2010.

[32] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," Machine Learning, vol. 46, 2002, pp. 389–422.

[33] I. Jolliffe, Principal Component Analysis, 2nd ed. Springer, New York, 2002.

[34] S. Chowdhury, S. Shatabda, and I. A. Dehzangi, "idnaprot-es: Identification of dna-binding proteins using evolutionary and structural features," Scientific Reports, vol. 7, 2017.

[35] P.-M. Feng, H. Lin, and W. Chen, "Identification of antioxidants from sequence information using naïve bayes," Computational and Mathematical Methods in Medicine, vol. 2013, 2013, p. 567529.

[36] X. Zhang and S. Liu, "Rbppred: predicting rna-binding proteins from sequence using svm," Bioinformatics (Oxford, England), vol. 33, 2016, p. 854–862.

[37] W.-Z. Lin and D. Xu, "Imbalanced multi-label learning for identifying antimicrobial peptides and their functional types," Bioinformatics, vol. 32, 2016, p. btw560.

[38] M. Ayyash, H. Tamimi, and Y. Ashhab, "Developing a powerful in silico tool for the discovery of novel caspase-3 substrates: a preliminary screening of the human proteome," BMC Bioinformatics, 2012.

[39] M. Nielsen and O. Lund, "Nn-align. an artificial neural network-based alignment algorithm for mhc class ii peptide binding prediction," BMC Bioinformatics, vol. 10, 2009, p. 296.

[40] K. Ron and H. J. George, "Wrappers for feature subset selection," Artificial Intelligence, vol. 97, 1997, pp. 273–324.

[41] B. W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," Biochimica et Biophysica Acta (BBA) - Protein Structure, vol. 405, no. 2, 1975, pp. 442–451.

[42] T. Fawcett, "An introduction to roc analysis," Pattern Recognition Letters, vol. 27, 2006, p. 861–874.

[43] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, 1995, pp. 273–297.

[44] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, 2001, pp. 5–32.

[45] M. A. Nielsen, Neural Networks and Deep Learning. Determination Press, 2015.

[46] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 785–794.

 ISBN: 978-1-61208-792-4

Accelerating Charged Single α-helix Detection on FPGA

Sam Khozama, Zoltán Nagy and Zoltán Gáspári

Pázmány Péter Catholic University
Faculty of Information Technology and Bionics
Budapest, Hungary
Email: {khozama.sam, nagy.zoltan, gaspari.zoltan}@itk.ppke.hu

***Abstract*—Novel biological sequences are determined at an extreme pace producing a huge amount of data each day. Implementing and speeding up the bioinformatics algorithms, which need very fast and accurate results, is the main advantage of reconfigurable architectures like Field-Programmable Gate Array (FPGA), in addition to the low precision input data for these kinds of algorithms, which can be stored in 2-5 bits. Detecting structural motifs such as Charged Single α-Helixes (CSAH) is a computationally intensive task which can be accelerated by using FPGA. The goal of our research is to further improve the processing speed of our algorithm called FT_CHARGE to analyze large databases in a reasonable time. Using the largest state-of-the-art FPGA devices, either the number of processing units or the parallelism inside the processing units can be increased. In this paper, we provide details and compare the two design approaches in terms of speed, implementation and accuracy. We propose a new architecture that can perform search for CSAH 32 times faster compared to our previous FPGA implementation.**

Keywords–FPGA; FT_CHARGE; CSAH; Charged Single α-helix; Hardware acceleration.

I. INTRODUCTION

The Charged Single Alpha-Helix (CSAH or simply SAH) is a unique structural motif in proteins [1]. It has been experimentally characterized only in a small number of proteins, therefore, its recognition by prediction methods is of high importance. Generally, more accurate algorithms tend to be slow, thus, efforts are concentrated to speed up these methods to make them applicable to large sequence sets [2]. Here, we present the speedup of FT_CHARGE, one of the earliest methods based on Fourier transformation. We have previously implemented an FPGA-based version of this algorithm [3] ,which is further improved here. Our novel implementation offers higher computational speed to allow processing of very large protein sequence sets, such as full genomes, or even metagenomic samples, within hours.

The goal of using FPGA is to accelerate different sequence searches and sequence matching in bioinformatics algorithms such as the Smith-Waterman (SW) algorithm. The parallel nature of FPGA, with its huge number of fine-grained blocks (configurable logic blocks) provides a convenient architecture for the additional implementation of bioinformatics algorithm; this way, FPGA and the bioinformatics algorithms have parallelism at a fundamental level [4]. So, a large amount of small and simple functional units can be implemented [5]. The Arithmetic Logic Unit (ALU), which is responsible for performing instructions in conventional computers, which is limited by the fact that it can only perform one instruction at a time [6]. Our task is to implement in parallel any function or circuit that meets the requirements of an application rather than sequentially, to achieve optimal performance.

II. FT_CHARGE ALGORITHM

Both SCAN4CSAH [1] and FT_CHARGE are used to detect CSAHs motifs and these two algorithms use completely different computational methods for analyzing protein sequences conceptually. Because the FT CHARGE algorithm is a very computationally intensive algorithm, it is suitable to accelerate it on FPGA. The standard FAST-All (FASTA) format is the input format for both algorithms. The analysis of biopolymer sequences utilizes Fourier transformation regularly. To implement the FT_CHARGE algorithm, we downloaded the database to the host computer and pre-processed it by using only 2-bit encoding per sequence element. Charges are assigned as follows: -1 for Asp (Aspartic acid) and Glu (Glutamic acid) , +0.5 for His (Histidine), +1 for Arg (Arginine) and Lys (Lysine) and zero for any other amino acid residue.[1]. This encoding is done by the host computer. The host computer sends this encoded data to the FPGA, where it is processed, and we receive back only the filtered results. Consequently, only the candidate sequences expected to have α-helix will be sent back to the host computer. As shown in Figure 1, the four consecutive steps of the algorithm are (1) the Charge Correlation Computation, (2) the Fast Fourier Transform (FFT) Computation, (3) the Maximum Finder and (4) Extreme Value Distribution Computation. The Charge Correlation function is defined by the following function:

$$R(k,n) = \sum_{i=k}^{k+m-n} c(i)c(i+n) \quad (1)$$

where $c(i)$ is the charge assigned to the i^{th} amino acid, m is the length of the window, $1 \leq k \leq l-m$ is the starting position of the current window and l is the length of the sequence. In our case, windows of 32 or 64 elements are examined during the analysis of the full sequences.

The output of the Charge Correlation Calculation block is connected to the FFT block. The role of the Maximum Finder block is to find the maximum amplitude and its frequency in the FFT spectra, which are used to fit an Extreme Value Distribution (EVD). At last, determining the threshold for signaling a charged single α-helix motif is done by using the fitted distribution.

 ISBN: 978-1-61208-792-4

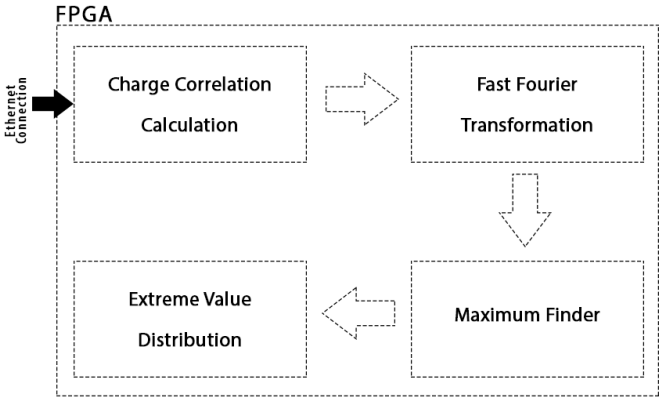


Figure 1. Block diagram of the system implemented on FPGA.

TABLE I. AREA REQUIREMENTS OF OUR PREVIOUSLY IMPLEMENTED SYSTEM [3]

	CLB LUTs	CLB Flip-Flops	BRAM	DSP Slices
Complete system	44662	53122	90	157
ZCU102	274080	548160	912	2520
Estimation	6	10	10	16

III. FPGA IMPLEMENTATION

FPGA manufacturers offer very efficient signal processing libraries to utilize in highly-intensive computation applications. In our previous work [4], we implemented the FT_CHARGE algorithm on a small FPGA. It used the FFT IP core from Xilinx CoreGenerator and the main goal was optimizing the computation of the charge correlation function to feed the FFT core efficiently. The computation time for the whole UniProt TREMBLE database [7] took nearly 24 hours. The stages were computed in parallel and the system was running on 100 MHz clock-frequency. The 32 and 64 element version required a new 2bit wide information sequence element in every 32^{nd} and 64^{th} clock-cycle. The memory bandwidth requirement was 6.25 Mbit/s and 3.125 Mbit/s in the case of 32 and 64 element windows, respectively.

The first way to improve our previous implementation is simply to replicate the system-blocks as much as we can, to fit into a larger ZYNQ device [8]. The FPGA resource usage of our previously implemented system and the estimated number of processing blocks on Xilinx ZCU102 board equipped with a larger FPGA compared to our previous study are shown in Table I.

In this paper, we have the opportunity to increase the performance nearly 6 times, as the limiting factor is the number of Lookup tables (LUTs). In light of this, we should take into consideration what does it mean to have such a huge number of hardware blocks on FPGA. Our previous computing unit connected to the main memory of the board via 6 input and 3 output Advanced Extensible Interfaces (AXI) [9]. These 9 interfaces should be replicated 6 times here. We can say that the area required by the AXI interconnect blocks to connect all the 54 AXI interfaces of these 6 units on the ZYNQ board to the memory interface will use a significant portion of the device. Thus, we could accelerate the implementation, but it would still require a lot of time because of a larger database.

A more efficient way to extend our previous solution is to implement several units in parallel, using a larger Xilinx ZYNQ board [10] [11]. Therefore, we went further to utilize some of the recent advantages of the High-Level Synthesis (HLS). Our suggested solution is to change the output of the Charge Correlation block in order to compute and send all 32 or 64 elements to FFT in each clock-cycle and also to replace the previous FFT module with more parallel processing units. This will definitely help us to improve computing performance in the system.

We proposed a new FFT implementation based on the Cooley-Tukey algorithm [12] that accepts 32 or 64 elements in each clock-cycle. The multiplication processes inside the FFT butterfly diagram normally use floating-point numbers, but this requires a very large area. In addition, we decreased the number of bits and saved a large amount of space because of using fixed-point numbers. The accuracy of the new solution is almost the same with the previous one. The basic idea is to improve the speed of computation by parallelizing the algorithm.

The system has implemented on the ZCU102 board. There are 2 options when preparing the input-output interfaces: (1) either load the input sequence by sequence or (2) load many sequences into a large buffer. Before processing each sequence, we need to load some special parameters to the control registers of the FT_CHARGE processing block. These parameters include the address of the sequence, the buffer where the result should be saved, and the length of the sequence. Sending these parameters take time since they should be written through the relatively slow AXI lite interface. According to the co-simulation results, sending one parameter takes around 5-6 clock-cycles, so, setting all the needed parameters will require a long time (in case of short sequences, it will be comparable with computation time of the whole sequence).

Here, our improvement consists of working on larger buffers, where one buffer contains several sequences concatenated one after each other and, instead of giving the length of one sequence, we give an array of lengths to the FT_CHARGE algorithm. For example, if we have 4 MB buffer, we can place several sequences in this large buffer and the cumulative length of these sequences is in the order of 16 million. In 1 byte, we can store 4 sequence elements because 2 bits can encode 1 element from the sequence. In this case, we have a task, which may last for 16 million clock cycles and, when we start the operation of this block, we need nearly 100 clock-cycles to set up the system with the required parameters. After that initial delay, the processing unit can work for a long time, consequently, the utilization of the hardware will be greatly increased.

IV. FAST FOURIER TRANSFORMATION BLOCK

For efficient computation of the spectra of the charge correlated window, the FFT can be used. A well known computing method of the FFT is the Cooley-Tukey Algorithm [12]. The Discrete Fourier Transform (DFT) of a signal is defined as follows:

$$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{-2\pi i}{N} nk} \quad (2)$$

where x_n is an element of the input vector and k is an integer ranging from 0 to $N-1$, where N is the size of the transform.

 ISBN: 978-1-61208-792-4

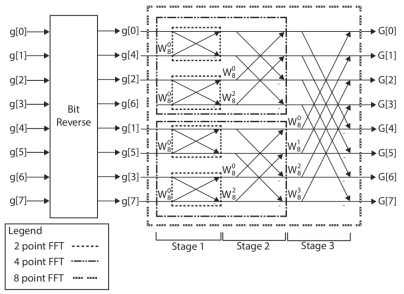


Figure 2. Cooley-Tukey algorithm: An 8 point FFT built recursively.

We can rewrite the computation of X_k as:

$$X_k = E_k + e^{\frac{-2\pi i}{N}k} O_k \quad (3)$$

$$X_{k+N/2} = E_k - e^{\frac{-2\pi i}{N}k} O_k \quad (4)$$

A block used to compute these equations is called butterfly, which is a small DFT. The Butterflies can be connected systematically to build larger FFT blocks, as shown in Figure 2 for an 8 input FFT. The number of stages is $log_2 N$ and $N/2$ Butterflies are required in each stage. As a result, the number of operations during the Fourier Transformation computation is reduced to $\mathcal{O}(n \log n)$ from $\mathcal{O}(n^2)$. Also, this structure provides a convenient way to implement parallel FFT by using $N \times log_2 N$ Butterflies.

Our previously implemented system used Xilinx FFT IP, which does the transformation serially. So, 32 or 64 clock-cycles are required to load the samples. The whole computation is done in 32 or 64 clock-cycles. The previous system has one processing the pipeline for 32 elements window and two processing pipelines for the 64 elements window. Therefore, the time required for 32 and 64 element window computations is roughly the same, if we have a large group of sequences.

In this paper, we modified the charge correlation part to provide these data elements in parallel, which means computation speed could be increased 32 times. So, by implementing FFT and Charge Correlation computation in parallel 32 times, a speedup can be expected because all the 32 or 64 outputs of the Charge Correlation block can be computed in one clock-cycle and shifted to the FFT. After some delay, the FFT block will provide all the 32 or 64 transformed results in one clock-cycle. The Maximum Finder should also compute the maximum value of the 32 or 64 element window in one clock cycle. The Extreme Value distribution block is not modified because it works on the maximum value of the FFT spectra and its position, just like in the previous implementation.

Our system's expectations are that each pipeline will require one input stream and one output stream. Therefore, 4 AXI interfaces are required all together. If these blocks are replicated 2 or 3 times, we still need only 8 or 12 memory ports, which is more manageable than the 54 ports of the previous serial implementation.

For easier testing, we use the MATLAB code and C code from the previous implementation. In this work, create a complete test bench in Vivado HLS [10] to load a valid sequence data to the system, process it and check whether the result is good or not.

TABLE II. AREA REQUIREMENTS OF THE CURRENT SYSTEM WITH DIFFERENT NUMBER REPRESENTATIONS

	CLB LUTs	CLB Flip-Flops	CLBs	Block RAM	DSPs
ZCU102	274 080	548 160	599 550	912	2 520
Fixed-point 18bit	44 043	42 344	8 155	13.5	668
Fixed-point 23bit	51 572	50 378	9 417	13.5	768
Floating-point	225 157	321 469	34 092	13.5	2 146

TABLE III. RUNTIME PROCESSING OF THE SPROT DATABASE (SECONDS), FPGA AND SOFTWARE IMPLEMENTATIONS

	with communication	without communication
Previous System	275.857	-
Current System		
Fixed-point	20.218	2.24159
Floating-point	23.1604	5.15008
AMD Ryzen 5 3400G	223.52	-
INTEL Core i5-4590	371.731	-

Table II shows three different representations that could be used in the current system. The original used an 18 bit word length inside FFT. The second one used 23 and 24 bits in the cases of 32 and 64 element windows. These designs were running on a 250MHz clock-frequency and were processing one window in each clock-cycle. The last one was a single-precision floating-point version.

By choosing the fixed-point 18 bit representation, the limiting factor will be the number of Digital Signal Processor (DSP) slices. In this case only, 30% of the resources are used and we can replicate the system-blocks three times.

Using single-precision floating-point numbers requires 6 times more DSP slices, because one floating-point multiply add (MADD) unit requires 4 DSP slices for the multiplier and 2 DSP slices for the adder, compared to a single DSP slice in the 18 and 23 bit cases. To fit the floating-point version into the ZCU102 board, the Initiation Interval of the FFT block is increased to two. In this case, the FFT is computed in two clock-cycles and the floating-point multiply-add units are shared between two operations, effectively halving the DSP slice requirement of the circuit. The FPGA resource utilization in this case, is over 90%, which might cause timing issues during implementation. Therefore, the clock-frequency of the design is reduced to 150MHz for stable operation. With the floating-point representation, the more accurate results, the more DSP slices are needed. The computation speed of the floating point solution is reduced to 75 million windows/s due to the slower 150MHz clock frequency and the two clock cycle processing time of the FFT. The computation times of the different solutions for the UniProt SPROT database [7] are summarized in Table III.

V. SYSTEM TESTING

After finished testing all the methods, which represented the functionality of the system in a synthesizable form, we started preparing the Xilinx environment to execute our implementation on a real circuit. On the host PC, we started preprocessing the database of amino acid sequences. The data is downloaded from the UniProt website [7] in FASTA format and converted to charges. We are using a ZCU102 board

 ISBN: 978-1-61208-792-4

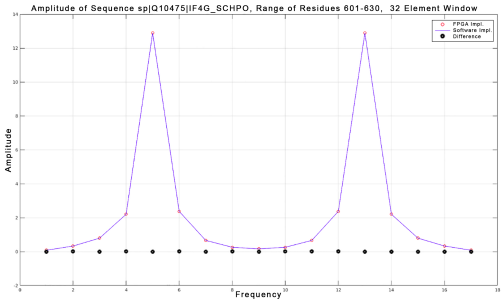


Figure 3. Amplitude of FFT, 32 elements window computation.

from Xilinx, which, in turn, has ARM Central Processing Unit cores. We are utilizing them for running Linux to handle the communication between the host computer and the ZCU102 board over gigabit Ethernet. After accepting the data, the ARM cores are sending it to the FPGA with Direct Memory Access (DMA). Our reference solutions for the FT_CHARGE algorithm are computed using MATLAB and C++. The result of these two were compared to the FPGA output. The comparison procedure included the number of hits, which represent the candidates windows expected to contain α-helix. Vivado Software Development Kit (SDK) was used for developing this software. While the circuit was running, we experienced a few cases where the sequences were detected on the software side but not in the hardware one. Examination of the undetected sequences showed two types of errors; in case of the first error type, the amplitude was close to the amplitude threshold (7.0). Some elements were not found because the maximum amplitude computed on the FPGA was smaller than the threshold due to the rounding error of the fixed-point computation. The other error has been noticed after plotting the FFT amplitudes in MATLAB. The representation of the amplitude in case of the 32 element window is plotted in Figure 3. A similar behaviour can be observed in the case of some 64 element windows too.

The figure shows that the difference can not be determined by the human eye, because the results of the software and the FPGA computation are correct to at least three or four decimal digits. The question raised here is: why are these sequences recognized only in software?

After examining the numbers in greater depth, we determined that we have two nearly equal peaks for half of the sequences in the software. In the hardware, we have a difference in the third or fourth decimal value, as mentioned before and, because of the rounding error of this very tinny difference, the output of the hardware is changed and no longer in sync with the software. We can easily overcome this problem by increasing the width of the registers storing the partial results in the FFT block by adding more fractional bits. Unfortunately this solution requires more resources because the multipliers in the DSP slices are working on 25 bit and and 18 bit signed inputs. Therefore two DSP slices required, instead of one, when on one of the inputs of the multiplication is in the $(26 - 49)$ bit range.

The consequences of the differences in numerical precision are to be investigated on large biological sequence sets to determine whether they cause any negative consequences for SAH detection. The double peaks, observed for some sequences, likely come from regular larger repeating units in the protein sequences. This issue and its relevance are also under investigation.

VI. Conclusion

In this paper, we improved the previously proposed FPGA based system for speeding up the α-helix detection algorithm by replicating the main three blocks as much as possible to fit a larger FPGA board. In addition, implementing these processing units in parallel enables fast search on larger protein databases and runs the whole system at a speed 30 times higher than the previous implementation. We have finished modifying the code of the whole system to be fitted with FPGA specifications represented in the 2-bit representation for the Charge Correlation Calculation module and also the transition from floating-point to fixed-point during the FFT module. This is required to use FPGA resources more efficiently.

On one hand, we were able to significantly reduce the area required to implement the circuit on FPGA by using fixed-point representation inside the FFT module. On the other hand computing performance is increased by a factor of three, while the accuracy of the results is similar to the accuracy of the floating-point solution. We have also tested this code, implemented on FPGA with real sequence data, and we obtained the same results as with the previous version on MATLAB. So, the error is in an acceptable range and the hardware version is working properly.

References

[1] D. Süveges, Z. Gáspári, G. Tóth, and L. Nyitray, "Charged single α-helix: a versatile protein structural motif," Proteins, vol. 74, 2009, pp. 905–916, doi: 10.1002/prot.22183.

[2] D. Simm and M. Kollmar, "A command-line tool for predicting stable single α-helices (SAH-domains), and the SAH-domain distribution across eukaryotes." PLoS one, vol. 13, no. 2, 2018, p. e0191924, doi: 10.1371/journal.pone.0191924.

[3] Á. Kovács, D. Dudola, L. Nyitray, G. Tóth, Z. Nagy, and Z. Gáspári, "Detection of single α-helices in large protein sequence sets using hardware acceleration," Journal of structural biology, vol. 204, no. 1, 2018, pp. 109–116, doi: 10.1016/j.jsb.2018.06.005.

[4] Z. Nagy, Z. Gáspári, and A. Kovács, "Accelerating a charged single α-helix search algorithm in protein sequences using FPGA," in CNNA 2016; 15th International Workshop on Cellular Nanoscale Networks and their Applications. VDE, 2016, pp. 1–2.

[5] D. G. Bailey, Design for embedded image processing on FPGAs. John Wiley & Sons, 2011.

[6] D. Abramson, A. de Silva, M. Randall, and A. Posutla, "Parallel special purpose architectures for high speed optimisation," in Proceedings of the Second Australasian Conference on Parallel and Real Time Systems, 1995, pp. 13–20.

[7] "Uniprot," URL: https://www.uniprot.org/.

[8] F. Albu et al., "Implementation of (Normalised) RLS lattice on Virtex," in International Conference on Field Programmable Logic and Applications. Springer, 2001, pp. 91–100, doi: 10.1007/3-540-44687-7_10.

[9] Xilinx, AXI4-Stream, Infrastructure IP Suite v3.0, LogiCORE IP Product Guide, Xilinx, December 2018.

[10] ——, Introduction to FPGA Design with Vivado High-Level Synthesis, UG998, Xilinx, Jan. 2019.

[11] ——, ZCU102 Evaluation Board User Guide UG1182 (v1.6), Xilinx, June 2019.

[12] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. of Computation, vol. 19, 1965, pp. 297–301, doi.org/10.1090/S0025-5718-1965-0178586-1.

 ISBN: 978-1-61208-792-4

Cancer Classification through a Hybrid Machine Learning Approach

Elmira Amiri Souri, Sophia Tsoka

Department of Informatics
Faculty of Natural and Mathematical Sciences
King's College London
London, WC2B 4BG, United Kingdom
Email: elmira.amiri@kcl.ac.uk, sophia.tsoka@kcl.ac.uk

***Abstract*—Understanding the underlying principles of cancer is a key endeavour in biomedical data mining. Although machine learning methods have been successful in discriminating normal from cancerous tissue with good accuracy, understanding of progression and formation of cancer across various cancer types is still restricted. Since cancer is a complex disease, being able to identify subgroups and investigate them separately may help in increasing the depth of our knowledge in terms of driver genes and oncogenic pathways. Moreover, as genes never act in isolation, methods that focus on single genes individually may be less efficient in uncovering key underlying molecular interactions. Algorithms that are capable of discovering the effect of combinations of genes have the potential to pave the way for extracting a new class of gene signatures that are neither mutated nor expressed differently, but rather act as mediators in forming oncogenic pathways. Here, we present a hybrid machine learning model to find cancer subgroups and an associated set of marker genes. In the proposed model, *autoencoders* are used to create a rich compressed set of features to identify cancer subgroups. Then, a two-step algorithm is developed based on information theory and regression analysis to find a set of discriminatory genes for each selected group for different types of cancer. This analysis is conducted based on the combined expression of genes to discover a new subset of genes associated with cancer. We show that we can still predict cancer accurately by decreasing the number of genes from thousands to tens for each subgroup. Pathway enrichment analysis is performed to find important pathways associated with a specific cancer type. The model is extensively analysed on datasets across nine cancer types and links between cancers are studied based on common gene signatures.**

Keywords–Machine Learning; Disease Classification; Clustering; Cancer Prediction.

I. INTRODUCTION

Cancer is a major cause of reduction in quality of life, with about 18.1 million new cases and 9.6 million cancer deaths noted recently (2018 [1]). Early detection of cancer can significantly improve prognosis, therefore, understanding the biology of cancer especially with regards to early detection is vital. Traditionally, clinical features such as age, tumor size, and cancer stage have been used to assist the prognosis of cancer, however these are only useful in late stage diagnosis and may not aid prediction [2].

High throughput technologies, such as microarray gene expression profiling and next-generation sequencing have produced an enormous amount of data which can be used to dissect cancer more accurately [3]. Early detection necessitates understanding the mechanism of cancer development via relevant associated and biological pathways. However, heterogeneity of tissues and genetics of patients prevent the identification of robust biomarkers [4] and the high dimensionality of expression data renders the selection of relevant genes in different types of cancer difficult [5]. Finally, as genes do not act in isolation and their combined effects lead to a variety of resultant phenotypes, the complexity of biomarker signatures increases [6].

Recently, machine learning and deep learning methods have resulted in advancement in the capability of prediction in many research fields with big and complex data, with notable applications in cancer research [7]–[10]. Deep learning methods have illustrated excellent potential in handling large and complex datasets and, together with the availablity of appropriate cancer profiling datasets [11], enable applications that can divulge key biomarker genes and pathways for disease types and increase our understanding of the mechanistic basis of cancer [8].

Identifying subgroups of similar pattern facilitates understanding of disease formation and progression. Once cancer subgroups are extracted, feature selection can be used for knowledge discovery through identification of key gene signatures [12]–[14]. Typically, methodologies rely on differentially expressed genes (for example, use of SAM [15], RVM [16] and SMVar [17]). However, these methods only focus on single genes and do not reflect the fact that genes work in functional groups. Additionally, there are genes contributing to cancer which may not be differentially expressed but may rather act as mediators in oncogenic pathways within a cancer network, establishing the connections between genes that are mutated or transcriptionally altered. Related work includes the work by Ghanat Bari et. al [9] that employ many concurrent Support Vector Machine models to derive a new class of cancer-related genes (named Class II genes) that are neither mutated nor differentially expressed, but proposed to act as potential key mediators in creating networks of cancer.

This work reports the development of a pipeline where the first stage involves application of an autoencoder, an unsupervised deep learning-based model, to compress high-dimensional gene expression data. Then, clustering is performed on the compressed gene expression data to discover different cancer subgroups, then each is assigned into two main classes called, *pure* and *mixed* based on the relevant sample

 ISBN: 978-1-61208-792-4

label. Tumours which are very different from normal tissues form the *pure* groups, while tumours that are similar to normal samples fit into the *mixed* subgroups (mix of normal samples and tumours) . In the second stage, for each of these subgroups, a subset of gene biomarkers is selected through unsupervised (for *pure* subgroups) or supervised (for *mixed* subgroups) algorithms. The supervised method is a two-step algorithm based on information theory and regression analysis. Figure 1 shows the proposed framework. This approach was implemented for each of nine cancer types and it was shown that the derived gene markers are efficient in disease prediction. To highlight key cancer mechanistic details, pathway enrichment analysis was also applied and the network between different cancer based on common biomarkets was investigated. In Section II, the materials and methods applied in this paper are reviewed. Section III presents the results of the framework. Section IV concludes the paper and goes over the future work.

II. Materials and Methods

Gene expression data corresponding to nine cancer types were obtained from Gene Expression Omnibus (GEO) [18] for Affymetrix Human Genome U133 Plus 2.0 platform [9]. A total of 6957 cancer and 1850 normal tissue samples were collected. Table I shows the list of cancer data used in this paper. For pathway analysis, 188 KEGG [19] pathways were downloaded from GSEA, Broad Institute [20]. Raw Affymetrix data were normalised through Robust Multichip Average (RMA) [21] through the R BioConductor `rma` function [22]. Probes were mapped to genes by the Affymetrix Human Genome U133 Plus 2.0 Array annotation data (chip HG-U133_Plus_2) using the R Bioconductor annotation package `hgu133plus2.db` [23]. In cases of multiple probes mapping to the same gene, the average value of these probes is taken.

Datasets were split into training (90%) and external validation (test) set (10%) and on the training dataset all metrics were calculated through a 10-fold cross validation scheme, repeated 5 times. The training and test datasets are scaled by `StandardScaler`. To balance data, we applied Synthetic Minority Over-sampling Technique [24] using `SMOTE` function from imbalanced-learn 0.4.2 Python package to the training dataset to prevent overfitting on one class. Since the number of features (genes) is much larger than the number of samples, we should avoid to decrease the number of sample for balancing the data, therefore, oversampling is performed on training data.

To compress the expression of genes to the smallest set, autoencoder [25] was used. It is implemented using a multilayer neural network with a hidden layer in the middle and consists of two parts of encoding ($\phi : \chi \rightarrow F$) and decoding ($\psi : F \rightarrow \chi$). The loss function is defined in a way that the output is reconstructed from the input. Autoencoder is implemented by using Tensorflow 1.12.0 with three hidden layers and `Tanh` activation. Then, to identify groups of patients with similar gene expression patterns, several clustering algorithms were implemented (e.g., k-means, Spectral Clustering, Gaussian Mixture Models) in Scikit-learn 0.21.2 with default hyperparameters. As the successful method depends on the actual structure of the dataset [26], we found that for the size and nature of our data, k-means performed well (for an extensive study of clustering algorithms on large datasets, see [27]). For the implementation of clustering algorithm `MiniBatchKMeans` function with random initializations number= 3, batch size= 100, and reassignment ratio= 0.01 was used. The best number of clusters is selected by silhouette index [28]. After clustering, each sample is assigned to one of the modules; Modules with samples of the same label will be considered *pure*, whereas clusters with mixed labels (normal and tumor) will be identified as *mixed*.

$$C_i = \begin{cases} pure & \text{if } n_i^t/n_i^n < \alpha \\ & \text{or } n_i^n/n_i^t < \alpha, \\ mixed & \text{otherwise.} \end{cases}$$

where C_i is the i^{th} cluster, n_i^t and n_i^n are the number of tumor and normal samples in cluster i respectively, and α is a threshold set to 0.1.

The next stage involved finding a subset of biomarkers that can best characterise samples in each cluster. For *pure* clusters, since the label of all samples is the same, unsupervised feature selection was used, whereas in the case of *mixed* clusters, supervised feature selection was applied. Specifically, in the *pure* cluster, Principle Component Analysis (PCA) was applied to compress gene expression features and the overall contribution of each gene forming the principle components calculated by applying an inverse transform of the PCA to an identity matrix to observe which features had the highest contribution. For the implementation of the first step of our feature selection algorithm, `SelectKBest` function with `mutual_info_classif` score function and for the second step `LassoCV` were used. Similarly, to perform PCA, we used `PCA`. In the case of *mixed* clusters, a two-step feature selection algorithm called *BestLasso* was implemented based on combination of information theory and LASSO (Least Absolute Shrinkage and Selection Operator) [29]. In BestLasso algorithm, first a subset of the highest contributing features is chosen by estimating the mutual information [30] of every feature with the labels, then Lasso was used to select the best set of features. The main reason for performing this two-step process is because gene expression data has high dimensionality and performing Lasso on all data becomes prohibitively slow and complex. Algorithm 1 shows this procedure.

Differentially Expressed (DE) genes were selected by calculating t-test in R `Limma 3.26.9` package. The p-value was adjusted by the moderated t-test for multiple testing by `BH-adjusted` (Benjamini-Hochberg method). We used `topTable` function from limma with log-fold-change `(logFC)` $> |2|$. [-23pt]

Once features were selected, classification was performed by learning a model on the selected features to predict

TABLE I. The list of cancer data used in this paper

Cancer	# of samples	# of tumor samples	# of normal samples
Breast	2113	1984	129
Ovary	954	839	115
Colon	1765	1557	208
Prostate	389	299	90
Skin	621	357	264
Liver	588	279	309
Pancreatic	259	178	81
Kidney	1031	589	442
Lung	1087	875	212
Total	**8807**	**6957**	**1850**

 ISBN: 978-1-61208-792-4

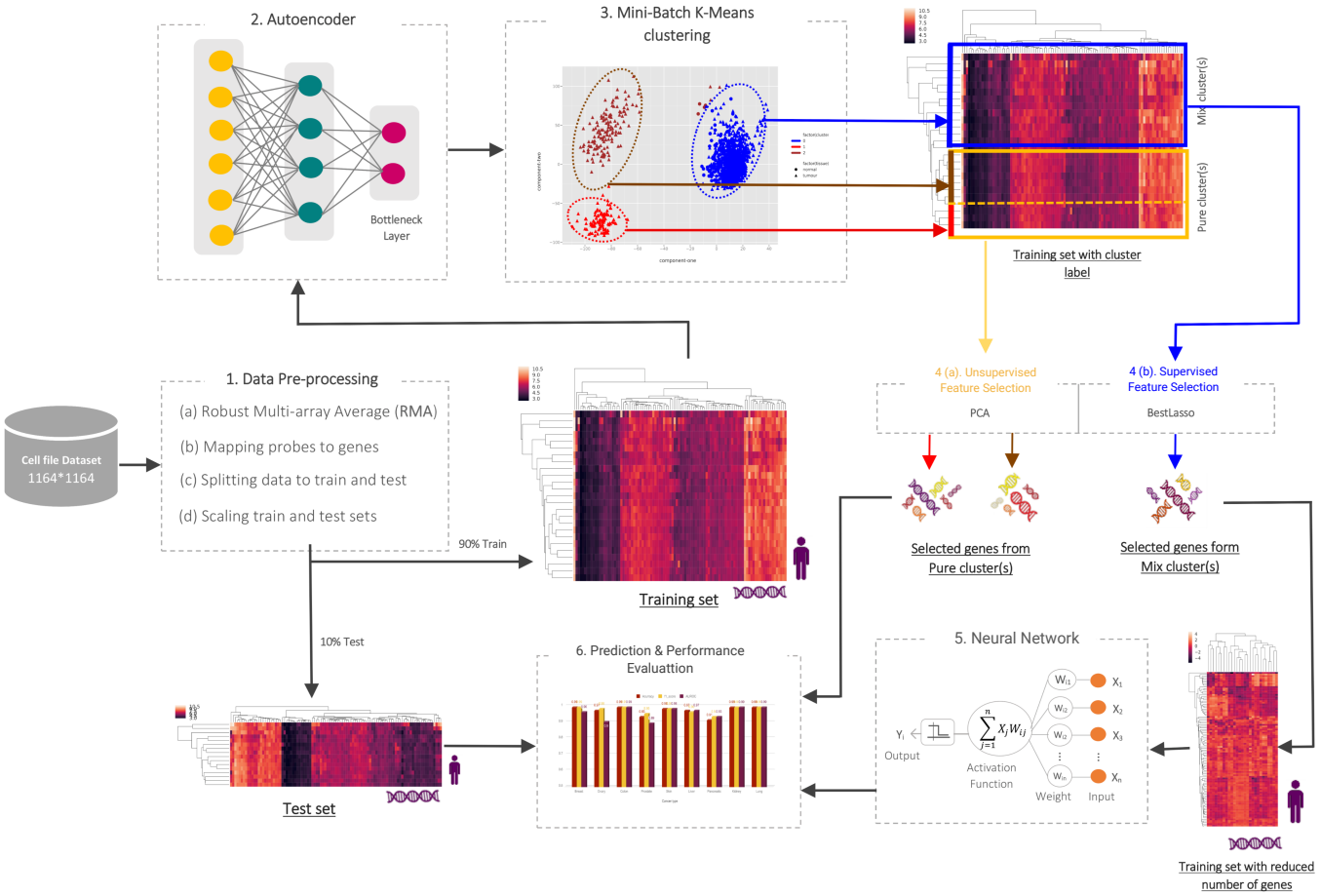


Figure 1. Overview of the proposed model to infer cancer subtypes and gene signatures (data here relate to the breast cancer dataset). 1.Preprocessing data. 2. An Autoencoder applied to compress the high-dimensional set of genes 3. The samples are grouped into different subgroups based on the encoded features. The cluster labels are added to each sample 4(a). All the genes of the subgroups that have same tissue types (*pure* clusters) are fed to PCA to select their highest contributing genes. 4(b). All the genes of the subgroups with *mixed* tissue types along with their labels are then input into a feature selection algorithm, BestLasso. 5. The selected features are used in a neural network to learn a model able to predict their labels. 6. The whole model is evaluated on the test set 5 times 10-fold cross-validation.

Algorithm 1 BestLasso algorithm. $X_{N\times m}$ is the gene expression data of N samples and m genes. Y is the label for each sample (0 and 1). k is the number of desired features from Mutual Information algorithm. $\mu(f_i)$ is the probability density for sampling f_i.

procedure BESTLASSO($X_{N\times m}, Y, k$)
  $feature\_scores \leftarrow [\,]$
  for $i = 1 \rightarrow m$ **do**
    $f_i \leftarrow X[:, i]$
    $s_{ia} \leftarrow \sum_{Y} \int \mu(Y, f_i) \log \mu(f_i|Y) df_i$
    $s_{ib} \leftarrow \int \mu(f_i) \log \mu(f_i) df_i$
    $s_i \leftarrow (s_{ia} - s_{ib})$
    $feature\_scores[i] \leftarrow s_i$
  $X'_{N\times k} \leftarrow$ Top k features with highest $feature\_scores$
  return $Lasso(X', Y)$

tissue type (normal or tumor) in *mixed* clusters as samples in *pure* clusters. Different techniques were evaluated, including Logistic Regression, Support Vector Machine and Random Forrest, and neural networks in Scikit-learn 0.21.2 with default hyperparameters. Among them neural networks had roughly better average performance on 9 cancer datasets. Then, neural networks hayperparameters were tuned using `GridSearchCV` and a model with an input layer, two hidden layers (30 and 5 nodes, 'relu' activation function) and an output layer with 'sigmoid' activation function was chosen. Five times 10-fold cross-validation was done on the training data and the test set data was used for evaluation of the classification procedure through accuracy, F1 score, and area under ROC curve (AUROC) metrics.

III. RESULTS AND DISCUSSION

Cancer is a heterogeneous disease with different histopathological and molecular subtypes, each with different diagnosis and therapies [31]. The goal of this work is

 ISBN: 978-1-61208-792-4

to propose a way to find these subtypes by maximising the intra-group and minimising inter-group similarity [32]. Sample stratification is difficult when relying only on clinical data, therefore the use of gene expression facilitates more meaningful pattern extraction and sample stratification. To achieve this target, machine learning clustering methods can identify groups of patients with similar gene expression patterns. In this section, we present the results of our method in selecting *pure* and *mixed* subgroups on nine different cancer types. We then introduce the genes related to cancer for each subgroup and report the high performance of the model using these genes. Lastly, we conduct pathway enrichment analysis and show that some of these results can provide validation through existing relevant literature.

In order to discover subgroups, a key step in clustering is determining the optimal number of clusters. A common method to perform this is by evaluating clusters using cluster validity indices, where samples are closely linked within the same cluster and are well-separated from members of other clusters [28]. To identify the optimal number of cancer subgroups, the average Silhouette score of 5 different number of clusters was computed. For breast cancer for example, the best way of subgrouping data is with three clusters (Figure 1 section 3). One of these subgroups is a *mix* of normal and tumor tissues, while the other two contain only tumor samples i.e., *pure* cluster. For all cancers, Table II shows the average Silhouette scores for each type respectively. In most cases, clustering has been able to separate the samples well using the compressed set of genes.

TABLE II. DIFFERENT SIZES OF CLUSTERS AND THEIR AVERAGE SILHOUETTE SCORE

Cancer \ Number of Clusters	2	3	4	5	6
Breast	0.80	**0.82**	0.60	0.55	0.49
Ovary	0.68	**0.73**	0.46	0.47	0.36
Colon	**0.73**	0.55	0.42	0.41	0.38
Prostate	0.59	**0.70**	0.55	0.60	0.54
Skin	0.53	0.56	**0.61**	0.58	0.51
Liver	0.41	**0.52**	0.50	0.39	0.39
Pancreatic	0.55	**0.62**	0.58	0.48	0.42
Kidney	**0.69**	0.50	0.48	0.44	0.41
Lung	**0.51**	0.44	0.34	0.33	0.35

After the optimal subgroups and the type of clusters (*pure* or *mixed*) are identified, gene signature subset selection was performed to extract useful information in each subgroup and reduce dimensionality (out of more than 22,000 genes). Since each cluster represents a different cancer subgroup, studying the selected genes in each cluster individually will lead to the identification of relevant gene signatures. One of the common methods of ranking genes associated to cancer is by selecting genes expressed differently in tumor and normal tissue using statistical methods. Selecting only Differentially Expressed (DE) genes results in genes being considered individually, regardless of their inter-relationships. As traits and phenotypes are caused by interactions of groups of genes [33], here we use a powerful machine learning strategy that can test for different combination of genes sets as means for deriving robust cancer biomarkers that have the ability of predicting cancer with high accuracy. Table III contains the list cancer subgroups and the number of their gene signatures. For example, breast cancer consist of two *pure* and one *mixed* subgroups with different number biomarkers selected in each. A list of the biomarkers for each cancer subgroup is given in Table IV.

TABLE III. LIST OF OPTIMAL SUBGROUP TYPES AND NUMBER OF GENE SIGNATURES IN EACH OF THEM

Cancer	Cluster Types (# of Gene Signature)			
Breast	Pure (28)	Pure (36)	Mixed(70)	
Ovary	Pure (41)	Mixed (90)	Mixed(19)	
Colon	Pure (normal)	Mixed (82)		
Prostate	Mixed (29)	Mixed(53)	Mixed(25)	
Skin	Pure (normal)	Pure (49)	Pure(38)	Mixed(28)
Liver	Mixed (34)	Mixed (27)	Mixed (45)	
Pancreatic	Pure (27)	Mixed (26)	Mixed (9)	
Kidney	Pure (normal)	Mixed (59)		
Lung	Pure (30)	Mixed (56)		

Moreover, methods that rely on just differentially expressed genes ignore mediator genes which are contributing to cancer but may not mutated. Recently, methods that aim to delineate such genes active in connecting oncogenic pathways are reported [9]. From all the gene signatures selected by our framework, some of them are differentially expressed and some are not, which may indicate mediator genes. As an example, mediator genes in breast cancer found by our model are as follows: *ABCA8*, *ARCN1* [34], *ARHGAP20* [35], *ATP5B* [36], *CA4* [37], *CLDN5* [38], *DCTN2*, *FAM13A*, *GLYAT*, *GRIP2*, *GSTM5* [39], *H3F3A*, *HIST1H3I*, *KIF23* [40], *NUP210* [41], *RAB7A* [42], *RPL7A* [42], *RPLP0*, *RPS12*, *SIN3A* [43], *SPTBN1*, *TUBA1C* [44].

Since there are multiple gene signatures common between each cancer, a network of cancers can be outlined. Figure 2 shows this network comprised of all the chosen gene signatures by our model colored based on the 9 different cancer types. Each cancer has their own gene signature while some of them share specific genes, as indicated in the figure. Our analysis showed that *ABCA8* is a hub gene shared between four cancers and known to be involved in multiple cancers in literature [45]–[47]. Another interesting observation is the many common gene signatures between breast and lung cancers: *CA4* [37], *FIGF* [48] , *LDB2* [49], *GPIHBP1* [50], *COL10A1* [51], *SLC19A3* [52], *LYVE1* [53], *IGSF10* [54] , *MYZAP* [55], *SPTBN1*, *ADH1B* [56], *ABCA6* [57], *PIR-FIGF*. Almost all of them are also reported as being associated with lung and breast cancer. It is note that lung is the most likely tissue for cancer metastasis from breast [58] [59].

The results of the prediction of the proposed model on the test set are presented in Table V. The model is performing with higher than 90% accuracy and F1-score in all cases which means that the set of selected genes are capable of accurately distinguishing between cancer and normal tissues. The two cancers with the lowest accuracy are Prostate and Pancreatic cancers, for which the lowest number of samples was available. It is noted that the model may improve upon availability of a larger data size for these cancers.

Once all important genes are selected and validated by our method, we can gain further insight through pathway enrichment analysis for each subgroup. To this end, the number of selected gene signatures in each pathway is determined and normalised by the total number of genes in the pathway, the counts therefore serving to demonstrate the importance of the pathway in the cancer subgroup. Some key pathways are already known as pathways associated with cancer and some of them have not been studied specifically yet and can be aimed for further research. Full list of the most important pathways

 ISBN: 978-1-61208-792-4

TABLE IV. LIST OF BIOMARKERS FOR EACH SUBGROUP

Cancer Subtype	Biomarkers
breast (mixed)	ABCA6, ABCA8, ADAMTS5, ADH1B, ADH1C, ALDH1L1, ANXA1, ARHGAP20, ARID5B, ATOH8, C2orf40, CA4, CD300LG, CEP55, CLDN5, CLEC3B, CNRIP1, COL10A1, COL11A1, COPG2IT1, DPT, FAM13A, FIGF GINS1, GJB2, GLYAT, GPIHBP1, GSTM5, HELLS, HSPB2, HSPB7, IGFBP6, IGSF10, INHBA, ITIH5, KIF14, KIF23, KLF15, LDB2, LINC01614, LRRN4CL, LYVE1, MAMDC2, MATN2, MME, MYZAP, NPR1, NUP210, PAFAH1B3, PAMR1, PGM5, PLAC9, PLIN1, RGN, RRM2, SBK1, SCARA5, SCN4B, SIK2, SLC19A3, SMC4, SPATS2, SPTBN1, TMEM246, TNMD, TNXA, TRIM59, TSHZ2, UHRF1, VIT
breast (pure)	SNHG7, LOC283674, RPS6KA2-AS1, C9orf50, RPL13A, TSPAN16, FLJ31713, RPS12, CFAP100, LINC00967, RPL7A, LOC101928602, LOC100288123, XKR7, HPR, LOC101929738, LOC101929144, FCAR, ACTG1, ZCCHC13, ARCN1, RPLP0, LOC101929680, TUBA1A, TUBA1C, ATP5B, TMEM203, SNORA74A
breast (pure)	HIST1H3I, RFFL, LOC100505716, GRIP2, SLC6A17, LOC645513, RBM26, NENF, C5orf51, APMAP, MLLT10, DHRS7, HDGFL1, IL10RB-AS1, LDLRAD4-AS1, SIN3A, PRDM2, LOC100506858, FKSG29, DANT2, LOC105370977, CACNG6, RAB7A, TMEM161B-AS1, LRRC43, EMC7, DCTN2, USF3, H3F3A, TRAFD1, LOC84843, MTPN, LINC00641, REST, TH2LCRR, RNF152
colon (mixed)	ABCA8, ABCG2, ADAMDEC1, ADH1C, ADTRP, AJUBA, AMPD1, APPL2, BEST4, C15orf48, C2orf88, CA1, CA2, CA7, CDH3, CDKN2B, CEMIP, CHGA, CHP2, CLCA4, CLDN1, CLDN23, COL11A1, CSE1L, CWH43, DHRS11, DUSP14, EDN3, ENTPD5, ETHE1, FKBP1A-SDCBP2, FLJ36848, FOXQ1, FUCA1, GCG, GPAT3, GPD1L, GTF2IRD1, GUCA2B, HAPLN1, HIGD1A, HILPDA, HPGD, INHBA, ITM2C, KIAA1549, KLF4, KRT80, LIFR, LINC00675, LPAR1, LRRC19, MOGAT2, MRGBP, MTHFD1L, NAAA, NFE2L3, NR3C2, P2RX4, PDCD4, PLCL2, PLP1, PRDX6, PYY, SCARA5, SLC25A34, SLC51B, SLC6A6, SLC7A5, SMPDL3A, SNTB1, SPPL2A, SST, TEAD4, TMCC3, TPH1, TRIB3, TSPAN1, TSPAN7, UGDH, VSTM2A, ZG16
kidney (mixed)	ABAT, ACOX2, ALAD, APEH, AQP2, ASS1, ATP6V0D2, CA9, CALB1, CAPN3, CLCNKB, CLDN10, COL23A1, CRYAA , CTSH, DCXR, EFCAB3 , EGLN3, ENPP6, ERP27, FBP1, FBXO16 , FOXI1, FXYD4, GATA3, GGH, HRG, HS6ST2, IGFBP3, IRX1, KCNJ1, KLHL13, KLHL14, KNG1, LARS2, LINC00887, LOC100130278, LOC101928574, LOC102723468, MT1G, NOL3, NPHS2, OAT, PTH1R, RDH11, RGS1, S100A2, SCNN1G, SERPINA5, SLC12A1, SLC25A5, TFAP2B, TMEM213, TMEM30B, TMEM52B, TMPRSS2, TNFAIP6, VIM, ZNF395
liver (ixed)	ADAMTS13, ANXA3, BEX1, BIRC5, BMP5, CFP, CNDP1, COL15A1, CYB5D1, CYP2C8, DACH1, DBH, DCUN1D3, DPF3, F9, GPM6A, HHIP, HSPB1, ITLN1, KAZN, KIAA0907, LCAT, LHX2, LINC01296, MAP2K1, MT1G, MYOM2, NSUN5, NSUN5P1, OLFML2B, PLAC8, PLVAP, POGZ, PROM1, PTH1R, SLC16A5, SLC46A3, SLC5A1, SLCO4C1, SNX27, STAB2, TARBP1, TCF21, THY1, WDR66
liver(mixed)	ADGRG7, ADK, ANGPTL3, BLOC1S1-RDH5, C1orf168, CAP2, CENPF, COL25A1, DGAT2, EPS8L3, ESR1, FREM2, KCNJ16, LAMC1, MT1H, NAPSB, PAMR1, PEG3-AS1, PLCB1, PPM1H, RANBP3L, RPS6KA6, SESTD1, SHC1, SSR2, STEAP3, TREH
lung (mixed)	ABCC3, ADCY4, ADRA1A, AGR2, AKAP2, AMOTL1, ARHGAP6, ASPRV1, BVES, CA4, CCBE1, CDH5, COL10A1, DACH1, FGD5, FGFR4, FOXF1, FUT2, GCNT3, GPRC5A, GRK5, HABP2, HSH2D, IGSF10, KDELR3, LIN7A, MAGI2-AS3, MUC20, MYCT1, NCKAP5, P2RY1, PAK1, PEAR1, PHF2, PPM1F, PROM2, RASIP1, RHBDL2, SDC1, SEMA6A, SGCG, SH3GL2, SH3GL3, SLC19A3, SLC39A11, SOX17, SPINK1, SPOCK2, SPTBN1, STARD13, TAL1, TGFBR3, TMPRSS4, TSPAN18, WFDC2, ZBED2
lung (pure)	LDB2, LYVE1, SDPR, ABCA6, FAM150B, ARHGAP6, RHOJ, AGER, ADH1B, EMCN, GPIHBP1, MMRN1, GRK5, GPM6A, MYZAP, ABCA8, S1PR1, FIGF, ASPA, ANGPTL1, NME1, GRIA1, CA4, EDNRB, PTPRB, SCN7A, TCF21, PCAT19, TEK, FHL1
ovary (pure)	P4HB, NOP10, SRP9, FTL, CDC37, MIF, NBPF10, CHMP2A, ARF1, COPZ1, MRPL37, NDUFAB1, SCAND1, RHOA, PGRMC1, XRN2, PSMC3, POLR2E, EIF4A1, DDOST, SPCS2 , GNB2, TUBA1C, ABHD17A, PRPF31, NDUFA3, PCBP1, RPS27, OST4, OAZ1, APEX1, UBC, RNF181, JTB, TMEM258, RPS5, MRPL34, HSPA8, H3F3A, CHCHD2, LSM7
ovary(mixed)	ABCA8, ABHD11, ABHD17C, ADGRD1, ANKRD29, AOX1, AP1M2, ARHGAP8, ARMCX5-GPRASP2, ARX, ASS1, ATP10D, BAMBI, BDH2, C14orf37, C1orf186, CACNB2, CD24, CELF2, CHD7, CLDN4, CLDN7, CNIH3, CNRIP1, CP, CPED1, CSGALNACT1, CXXC5, CXorf57, DFNA5, ECM2, EPCAM, FAM153A, FAM153A , FLRT2, GHR, GNG11, GPRASP1, GRHL2, HAND2-AS1, HOXC6, IDH2, KCNT2, KLHL14, KPNA5, L3MBTL3, LEMD2, LIN7A, LOC728392, MAF, ME1, MECOM, MUC1, MUM1L1, NBEA, NDNF, NR3C2, OLFML1, PEG3, PID1, PLCL2, POLR3GL, PPM1K, PPP4R4, PRSS35, RNASE4, RPL36A, SERP2, SIGLEC11, SLC30A4, SLC34A2, SLC44A2, SNCA, SORT1, SPINT1, STON2, SYTL1, TCEAL2, TCEAL3, TCEAL7, TES, TFPI, TLE4, TMEM139, TMEM150C, TRIM68, TRPC1, TSPAN5, WFDC2, WHAMMP2
ovary (mixed)	ARID4B, CASP2, FMN2, GS1-259H13.2, HIST1H3I, HPS3, KLHL24, NCOA2, NICN1, PCED1B, PLXND1, PPIAP21, PSMG3-AS1, RAB4A, SHROOM2, TOPBP1, TUBB4B, VSIG1, ZDHHC20
pancreatic (mixed)	FBXO25, HOXC6, NRG4, PDIA2, PRR11, RPL14, SLC25A13, SND1, TNFAIP1
pancreatic (mixed)	AFAP1-AS1, ARMC9, CALU, CLDN1, CLDN4, CST1, CTTN, HIST2H2AA3, HOXB7, HOXC6, KRT18, LOC340340, MAMDC2, MROH6, MSLN, NAT14, NME1-NME2, PKM, PLXNB2, PYGB, RPL23A, SDC1, SDC4, SLPI, TTTY5, VGLL4
pancreatic (pure)	CPEB2, SNHG10, LOC642862, COQ10B, AFF4, HIST1H4B, C6orf106, ARID5B, CDK9, LOC100129112, CCDC117, BOLA2, NOCT, POLR2A, PRDM2, ZFX, C16orf72, B4GALT1, GATAD2A, ATXN2L, LOC101926943, AHNAK, CCNK, RAB7A, CDR1, MTPN, ZNF460
prostate (mixed)	ACSS2, CDKN2A, CPSF7, ENTPD3, FADS1, FAP, IGSF1, KANK4, LINC00328, LINC00869, LOC100996741, LOC158863, LOC441666, NETO2, NFAT5, PCSK5, RNF24, SALL3, SMIM10L2A, SPPL3, ST3GAL5, TMCO3, TMEM241, ZNF595, ZNF93
prostate (mixed)	ACOX2, AMACR, CFC1, COL9A1, CYP4B1, EFS, FHL2, FLRT3, FOXQ1, HADHB, LSAMP, MME, MSMB, MSMO1, NEFH, NPM1, PCAT4, RBBP7, SMIM5, WIF1
prostate (mixed)	ADCY4, ADORA2A, ADRB1, ARHGEF15, ARRDC2, ATHL1, ATP7A, BCKDHB, C10orf10, C1R, CADM3, CHST7, CLEC14A, CLIC2, CNBD2, DMBT1, DOCK9, FAM193B, FECH, FES, FHL5, GIMAP1, IGFBP5, IL15RA, KCNMB2, LCLAT1, LGR4, LINC01503, LOC100507291, LOC100996583, LOC10537679, LOC286071, LYPLA1, MAP3K3, MFAP3, MS4A14, MSC-AS1, NPR2, PDGFRA, PDLIM1, PNPLA4, PPP6R1, PSMA5, SHC1, SLC39A9, TIE1, TMEM218, TMEM255B, TMOD1, TRIP10, TSC22D1, TTR, UGP2
skin (pure)	SMAD1, SLC46A2, CCDC186, NIPAL1, DENND4C, XG, NET1, MYO6, HLF, ATP8B1, THRB, FOXN3, BCL11B, GIPC2, RAPGEFL1, ABHD5, LNX1, CEBPG, MAF, LRBA, LOC284023, RORA, TMTC3, CCDC6, TTC39B, GLTP, DENND2C, MPZL3, F3, PPM1L, ABLIM1, ELOVL4, FBXW7, TUFT1, GAN, ACVR2A, ELL3, LOC101927164
skin (pure)	TBC1D8, LOC102724593, THRA, TMEM262, SIPA1L3, MMP19, XGY2, RPARP-AS1, LOC100132319, SPAG8, ELMSAN1, ESRG, SPIDR, CYP4Z1, PCNT, ADIRF-AS1, LOC101928988, IL17RE, NUDT17, CCDC153, SAPCD1-AS1, LOC283713, EEF1D, LIPH, YPEL2, CDR1, MIR4697HG, DCST2, RPRML, LOC105369671, UBE2NL, SLC9A3R2, AGAP11, ANKRD19P, CENPT, TYSND1, AP1G2, RRN3P3, HSPA1B, LOC101928595, LOC105375061, LOC105379661, SKIDA1, ACTA2-AS1, LOC102723600, GATB, RNF31, FOXH1, CYP21A1P
skin (mixed)	ADAM12, APOBEC3C, ARPC1B, ATL1, BCL2A1, BMP2, BTC, CLDN23, CLDN8, CP, EPB41L4B, FCMR, HN1, HPGDS, IFI27, IGFL1, ITPA, LOC105378074, MIR503HG, MSH5-SAPCD1, NDC80, OASL, PIK3CD, PRDM6, RTP4, SGCG, SLC8A1, TMEM206

 ISBN: 978-1-61208-792-4

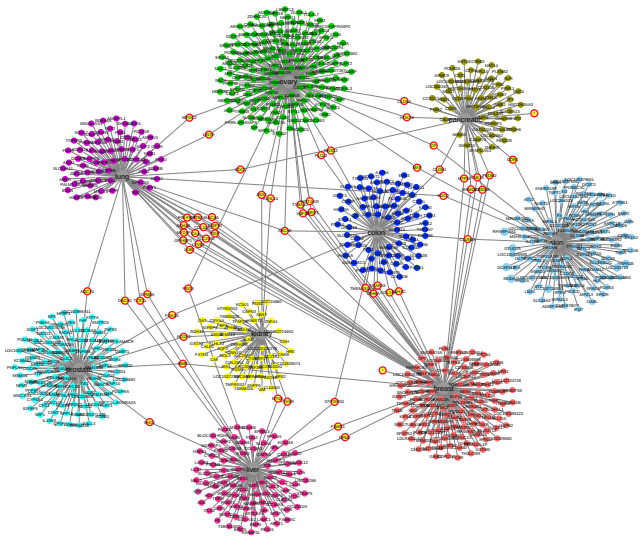


Figure 2. Gene signatures selected by our framework coloured according to cancer type. The common genes connecting more than one cancer are shown in orange.

TABLE V. THE ACCURACY, F1 SCORE, AUROC OF OUR MODEL ON 9 CANCER TYPES

Cancer	Acurracy	$f1_{score}$	AUROC
Breast	0.99	0.99	0.96
Ovary	0.97	0.98	0.90
Colon	0.99	0.99	0.99
Prostate	0.93	0.98	0.89
Skin	0.98	0.98	0.98
Liver	0.97	0.96	0.97
Pancreatic	0.91	0.93	0.93
Kidney	0.99	0.99	0.99
Lung	0.99	0.99	0.99

among KEGG pathways for each subgroup summarized in Table VI. Moreover, from our analysis, there are several pathways repeated in multiple cancer subgroups and all these pathways are cited in the literature as associated to cancer. Table VII shows key pathways identified through the KEGG.

IV. CONCLUSION AND FUTURE WORK

With the advent of massively parallel profiling of genes and their products, as well as improved machine learning technologies to handle large and heterogeneous datasets, enhancing analyses for cancer is possible. In this work, we present a hybrid machine learning computational procedure that includes analysis of datasets from multiple cancer types, integration of supervised and unsupervised learning procedures in the same computational framework and the use of autoencoder step that can effectively compress the high dimensionality of the gene expression profiles to discovering cancer subgoups.

Using this approach we identified a set of genes involved in cancer, some of them being recently reported in literature. As another means of validation, we were able to perform classification with very high accuracy using these biomarkers on the test set. In addition to being able to accurately predict cancer, our goal was to increase understanding of the underlying mechanisms by performing analysis on the selected genes and their pathways. Therefore, a network was created to show common biomarker genes among different types of cancer, that can reveal relationships between cancer types, e.g., breast and lung, as previously noted. Additionally, pathway enrichment analysis on our data identified the most important KEGG pathways, with some of them known to have a role in cancer formation and progression. Finally, differentially expressed genes were computed and compared with the selected genes to identify a new set of genes that are believed to act as mediators. The suggested pipeline for subgrouping cancer represents a novel contribution towards analysing transcriptomic cancer tissue data and aiding the development of sophisticated machine learning methods for big, complex and noisy data. In future work, clinical aspects of each subgroup can be taken into consideration by including

TABLE VI. LIST OF IMPORTANT KEGG PATHWAYS FOR EACH SUBGROUPS OF ALL THE NINE CANCERS.

Cancer Subgroup	**KEGG Pathway**
breast (mixed)	RENIN_ANGIOTENSIN_SYSTEM ONE_CARBON_POOL_BY_FOLATE
breast (pure)	PATHOGENIC_ESCHERICHIA_COLI_INFECTION
breast (pure)	VASOPRESSIN_REGULATED_WATER_REABSORPTION
colon (mixed)	NITROGEN_METABOLISM
kidney (mixed)	VALINE_LEUCINE_AND_ISOLEUCINE BIOSYNTHESIS_FOLATE_BIOSYNTHESIS
liver (mixed)	DORSO_VENTRAL_AXIS_FORMATION
liver (mixed)	BETA_ALANINE_METABOLISM
liver (mixed)	RETINOL_METABOLISM
lung (mixed)	GLYCOSPHINGOLIPID_BIOSYNTHESIS_GLOBO_SERIES
lung (pure)	ABC_TRANSPORTERS
ovary (mixed)	GLYCOSAMINOGLYCAN_BIOSYNTHESIS _CHONDROITIN_SULFATE
ovary (mixed)	DORSO_VENTRAL_AXIS_FORMATION
ovary (pure)	PROTEIN_EXPORT
pancreatic(mixed)	PATHOGENIC_ESCHERICHIA_COLI_INFECTION
pancreatic (mixed)	ERBB_SIGNALING_PATHWAY
pancreatic (pure)	GLYCOSAMINOGLYCAN_BIOSYNTHESIS _KERATAN_SULFATE
prostate (mixed)	PRIMARY_BILE_ACID_BIOSYNTHESIS
prostate (mixed)	GLYCOSAMINOGLYCAN_BIOSYNTHESIS _CHONDROITIN_SULFATE
prostate (mixed)	GLYCOSPHINGOLIPID_BIOSYNTHESIS_GANGLIO _SERIES
skin (pure)	THYROID_CANCER
skin (pure)	RNA_POLYMERASE
skin (mixed)	ARRHYTHMOGENIC_RIGHT_VENTRICULAR _CARDIOMYOPATHY_ARVC

TABLE VII. LIST OF IMPORTANT KEGG PATHWAYS IN NINE CANCERS

Pathway	**Reference(s)**
PURINE_METABOLISM	Purines play a critical role in cell proliferation and their broken metabolism has recently been recognized to be related to cancer progression [60]
PATHWAYS_IN_CANCER	KEGG has identified a pathway which is related to cancer [19]
LEUKOCYTE_TRANSENDOTHELIAL_MIGRATION	Leukocytes cells are exploited by tumour cells for extravasation [61]
PYRIMIDINE_METABOLISM	Edwards et al. [62] have extensively studied human skin cutaneous melanoma (SKCM) and found pyrimidine metabolism as a major pathway in its progression.
MAPK_SIGNALING_PATHWAY	The role of mitogen-activated protein kinase (MAPK) pathways in cancer is studied in [63]. Changes in MAPK pathways can mainly affect Ras and B-Raf in extracellular signal-regulated kinase pathway.
FOCAL_ADHESION	Focal adhesion kinase (FAK) plays an important role in tumor progression and metastasis because it is in charge of cancer cell signalling, cell proliferation, cell survival and cell migration [64].
NEUROACTIVE_LIGAND_RECEPTOR_INTERACTION	He et al. [65] studied the gene expression in prostate cancer and found the neuroactive ligand-receptor interaction as one of the enriched pathways.

 ISBN: 978-1-61208-792-4

them as relevant features, using them as prediction outcomes or validating biomarkers against them (e.g., use of survival data for validation). The desired outcome will be to enhance accurate cancer diagnosis, while also paving the way for evaluating therapeutic interventions.

REFERENCES

[1] R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics, 2019," CA: a cancer journal for clinicians, vol. 69, no. 1, 2019, pp. 7–34.

[2] C. A. Borrebaeck, "Precision diagnostics: moving towards protein biomarker signatures of clinical utility in cancer," Nature Reviews Cancer, vol. 17, no. 3, 2017, p. 199.

[3] L. De Cecco, P. Bossi, L. Locati, S. Canevari, and L. Licitra, "Comprehensive gene expression meta-analysis of head and neck squamous cell carcinoma microarray data defines a robust survival predictor," Annals of oncology, vol. 25, no. 8, 2014, pp. 1628–1635.

[4] S. Turajlic, A. Sottoriva, T. Graham, and C. Swanton, "Resolving genetic heterogeneity in cancer," Nature Reviews Genetics, vol. 20, no. 7, 2019, pp. 404–416.

[5] J. Li et al., "Identification of high-quality cancer prognostic markers and metastasis network modules," Nature Communications, vol. 1, Jul 2010, pp. 34 EP –, article.

[6] S. Gao et al., "Identification and construction of combinatory cancer hallmark–based gene signature sets to predict recurrence and chemotherapy benefit in stage ii colorectal cancer," JAMA oncology, vol. 2, no. 1, 2016, pp. 37–45.

[7] A. Penson et al., "Development of genome-derived tumor type prediction to inform clinical cancer care," JAMA oncology, vol. 6, no. 1, 2020, pp. 84–91.

[8] A. Rahimi and M. Gonen, "Discriminating early- and late-stage cancers using multiple kernel learning on gene sets," Bioinformatics, vol. 34, no. 13, 2018, pp. i412–i421.

[9] M. Ghanat Bari, C. Y. Ung, C. Zhang, S. Zhu, and H. Li, "Machine learning-assisted network inference approach to identify a new class of genes that coordinate the functionality of cancer networks," Sci Rep, vol. 7, no. 1, Aug 2017, pp. 6993–6993.

[10] K. Chaudhary, O. B. Poirion, L. Lu, and L. X. Garmire, "Deep learning–based multi-omics integration robustly predicts survival in liver cancer," Clinical Cancer Research, vol. 24, no. 6, 2018, pp. 1248–1259.

[11] Y. Xiao, J. Wu, Z. Lin, and X. Zhao, "A deep learning-based multi-model ensemble method for cancer prediction," Computer Methods and Programs in Biomedicine, vol. 153, 2018, pp. 1 – 9.

[12] S.-B. Cho and H.-H. Won, "Machine learning in dna microarray analysis for cancer classification," in Proceedings of the First Asia-Pacific Bioinformatics Conference on Bioinformatics 2003 - Volume 19, ser. APBC '03. Darlinghurst, Australia: Australian Computer Society, Inc., 2003, pp. 189–198.

[13] Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," Bioinformatics, vol. 23, no. 19, 2007, pp. 2507–2517.

[14] J. Li and E. Wang, "A multiple survival screening algorithm (mss) for identifying high-quality cancer prognostic markers," Feb 2011.

[15] V. G. Tusher, R. Tibshirani, and G. Chu, "Significance analysis of microarrays applied to the ionizing radiation response," Proceedings of the National Academy of Sciences, vol. 98, no. 9, 2001, pp. 5116–5121.

[16] G. W. Wright and R. M. Simon, "A random variance model for detection of differential gene expression in small microarray experiments," Bioinformatics, vol. 19, no. 18, 2003, pp. 2448–2455.

[17] F. Jaffrezic, G. Marot, S. Degrelle, I. Hue, and J.-L. Foulley, "A structural mixed model for variances in differential gene expression studies," vol. 89, no. 1, 2007, pp. 19–25.

[18] R. Edgar, M. Domrachev, and A. E. Lash, "Gene expression omnibus: Ncbi gene expression and hybridization array data repository," Nucleic acids research, vol. 30, no. 1, 2002, pp. 207–210.

[19] M. Kanehisa and S. Goto, "Kegg: kyoto encyclopedia of genes and genomes," Nucleic Acids Res, vol. 28, no. 1, Jan 2000, pp. 27–30, 10592173[pmid].

[20] A. Subramanian et al., "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles," Proceedings of the National Academy of Sciences, vol. 102, no. 43, 2005, pp. 15 545–15 550. [Online]. Available: https://www.pnas.org/content/102/43/15545

[21] R. Irizarry et al., "Exploration, normalization, and summaries of high density oligonucleotide array probe level data," Biostatistics, vol. 4, no. 2, 2003, pp. 249–264.

[22] L. Gautier, L. Cope, B. M. Bolstad, and R. A. Irizarry, "affy—analysis of affymetrix genechip data at the probe level," Bioinformatics, vol. 20, no. 3, 2004, pp. 307–315.

[23] M. Carlson, hgu133plus2.db: Affymetrix Human Genome U133 Plus 2.0 Array annotation data (chip hgu133plus2), r package version 3.2.2.

[24] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," J. Artif. Int. Res., vol. 16, no. 1, Jun. 2002, pp. 321–357.

[25] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, vol. 313, no. 5786, 2006, pp. 504–507.

[26] P. D'haeseleer, "How does gene expression cluster work?" Nature biotechnology, vol. 23, 01 2006, pp. 1499–501.

[27] M. C. P. de Souto, I. G. Costa, D. S. A. de Araujo, T. B. Ludermir, and A. Schliep, "Clustering cancer gene expression data: a comparative study." BMC Bioinformatics, vol. 9, 2008.

[28] P. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," J. Comput. Appl. Math., vol. 20, no. 1, Nov. 1987, pp. 53–65.

[29] V. Fonti, "Feature selection using lasso," Research paper in business analytics, 2017.

[30] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," Phys. Rev. E, vol. 69, Jun 2004, p. 066138.

[31] Z. Liu, X.-S. Zhang, and S. Zhang, "Breast tumor subgroups reveal diverse clinical prognostic power," Sci Rep, vol. 4, Feb 2014, pp. 4002–4002.

[32] D. Jiang, C. Tang, and A. Zhang, "Cluster analysis for gene expression data: A survey," IEEE Trans. on Knowl. and Data Eng., vol. 16, no. 11, Nov. 2004, pp. 1370–1386.

[33] D. Moore, The Dependent Gene: The Fallacy of "Nature vs. Nurture", 01 2002.

[34] J. T.-H. Chang, F. Wang, W. Chapin, and R. S. Huang, "Identification of micrornas as breast cancer prognosis markers through the cancer genome atlas," PLoS One, vol. 11, no. 12, Dec 2016, pp. e0 168 284–e0 168 284, 27959953[pmid].

[35] D. Oliver et al., "Identification of novel cancer therapeutic targets using a designed and pooled shrna library screen," Sci Rep, vol. 7, Feb 2017, pp. 43 023–43 023, 28223711[pmid].

[36] J. Cuezva et al., "The bioenergetic signature of cancer," Cancer Research, vol. 62, no. 22, 2002, pp. 6674–6681.

[37] M. Su et al., "The anti-angiogenic effect and novel mechanisms of action of combretastatin a-4," Sci Rep, vol. 6, Jun 2016, pp. 28 139–28 139, 27338725[pmid].

[38] R. Akizuki, S. Shimobaba, T. Matsunaga, S. Endo, and A. Ikari, "Claudin-5, -7, and -18 suppress proliferation mediated by inhibition of phosphorylation of akt in human lung squamous cell carcinoma," Biochimica et Biophysica Acta (BBA) - Molecular Cell Research, vol. 1864, no. 2, 2017, pp. 293 – 302.

[39] Y. Ke-Da et al., "Genetic variants in gstm3 gene within gstm4-gstm2-gstm1-gstm5-gstm3 cluster influence breast cancer susceptibility depending on gstm1," Breast Cancer Research and Treatment, vol. 121, 2009, pp. 485–496.

[40] J. Zou et al., "Kinesin family deregulation coordinated by bromodomain protein ancca and histone methyltransferase mll for breast cancer cell growth, survival, and tamoxifen resistance," Mol Cancer Res, vol. 12, no. 4, Apr 2014, pp. 539–549, 24391143[pmid].

[41] A. Ruhul et al., "Nuclear pore complex protein, nup210 is a novel mediator of metastasis in breast cancer," NIH Research festival, 2018.

[42] J. Xie et al., "Knockdown of rab7a suppresses the proliferation, migration and xenograft tumor growth of breast cancer cells," Bioscience Reports, 2018.

 ISBN: 978-1-61208-792-4

[43] K. Watanabe et al., "A novel somatic mutation of sin3a detected in breast cancer by whole-exome sequencing enhances cell proliferation through era expression," Scientific Reports, vol. 8, no. 1, 2018, p. 16000.

[44] Y. Wang, H. Xu, B. Zhu, Z. Qiu, and Z. Lin, "Systematic identification of the key candidate genes in breast cancer stroma," Cell Mol Biol Lett, vol. 23, Sep 2018, pp. 44–44, 30237810[pmid].

[45] H. A. M. Sakil, M. Stantic, J. Wolfsberger, S. E. Brage, J. Hansson, and M. T. Wilhelm, "Dnp73 regulates the expression of the multidrug-resistance genes abcb1 and abcb5 in breast cancer and melanoma cells - a short report," Cell Oncol (Dordr), vol. 40, no. 6, 2017, pp. 631–638, 28677036[pmid].

[46] X. Liu et al., "Discovery of microarray-identified genes associated with ovarian cancer progression," International journal of oncology, vol. 46, 04 2015.

[47] K. Xu, J. Cui, V. Olman, Q. Yang, D. Puett, and Y. Xu, "A comparative analysis of gene-expression data of multiple cancer types," PLoS One, vol. 5, no. 10, Oct 2010, pp. e13 696–e13 696, 21060876[pmid].

[48] E. Bailey et al., "Pulmonary vasculopathy associated with figf gene mutation," Am J Pathol, vol. 187, no. 1, Jan 2017, pp. 25–32, 27846380[pmid].

[49] F. Zhang et al., "Identification of key transcription factors associated with lung squamous cell carcinoma," Med Sci Monit, vol. 23, Jan 2017, pp. 172–206, 28081052[pmid].

[50] W. B. Kinlaw, P. W. Baures, L. E. Lupien, W. L. Davis, and N. B. Kuemmerle, "Fatty acids and breast cancer: Make them on site or have them delivered," J Cell Physiol, vol. 231, no. 10, Oct 2016, pp. 2128–2141, 26844415[pmid].

[51] F. Andriani et al., "Diagnostic role of circulating extracellular matrix-related proteins in non-small cell lung cancer," BMC Cancer, vol. 18, no. 1, Sep 2018, pp. 899–899, pMC6145327[pmcid].

[52] I. Cheuk et al., "Association of ep2 receptor and slc19a3 in regulating breast cancer metastasis," Am J Cancer Res, vol. 5, no. 11, Oct 2015, pp. 3389–3399, 26807319[pmid].

[53] O. Kowalczuk, J. Laudanski, W. Laudanski, W. E. Niklinska, M. Kozlowski, and J. Niklinski, "Lymphatics-associated genes are downregulated at transcription level in non-small cell lung cancer," Oncol Lett, vol. 15, no. 5, May 2018, pp. 6752–6762, 29849784[pmid].

[54] M. Bashir, S. Damineni, G. Mukherjee, and P. Kondaiah, "Activin-a signaling promotes epithelial-mesenchymal transition, invasion, and metastatic growth of breast cancer," Npj Breast Cancer, vol. 1, Aug 2015, pp. 15 007 EP –, article.

[55] H. Thomsen et al., "Inbreeding and homozygosity in breast cancer survival," Scientific Reports, vol. 5, Nov 2015, pp. 16 467 EP –, article.

[56] C. McCarty et al., "Alcohol, genetics and risk of breast cancer in the prostate, lung, colorectal and ovarian (plco) cancer screening trial," Breast Cancer Res Treat, vol. 133, no. 2, Jun 2012, pp. 785–792, 22331481[pmid].

[57] D. Mohelnikova et al., "The role of ABC transporters in progression and clinical outcome of colorectal cancer," Mutagenesis, vol. 27, no. 2, 03 2012, pp. 187–196.

[58] B. Weigelt, J. L. Peterse, and L. J. van't Veer, "Breast cancer metastasis: markers and models," Nature Reviews Cancer, vol. 5, Aug 2005, pp. 591 EP –, review Article.

[59] H. Kennecke et al., "Metastatic behavior of breast cancer subtypes," Journal of Clinical Oncology, vol. 28, no. 20, 2010, pp. 3271–3277, pMID: 20498394.

[60] J. Yin, W. Ren, X. Huang, J. Deng, T. Li, and Y. Yin, "Potential mechanisms connecting purine metabolism and cancer therapy," Front Immunol, vol. 9, Jul 2018, pp. 1697–1697, 30105018[pmid].

[61] C. Strell and F. Entschladen, "Extravasation of leukocytes in comparison to tumor cells," Cell Commun Signal, vol. 6, Dec 2008, pp. 10–10, 19055814[pmid].

[62] L. Edwards, R. Gupta, and F. V. Filipp, "Hypermutation of dpyd deregulates pyrimidine metabolism and promotes malignant progression," Mol Cancer Res, vol. 14, no. 2, Feb 2016, pp. 196–206, 26609109[pmid].

[63] A. S. Dhillon, S. Hagan, O. Rath, and W. Kolch, "Map kinase signalling pathways in cancer," Oncogene, vol. 26, May 2007, pp. 3279 EP –, review.

[64] G. W. McLean, N. O. Carragher, E. Avizienyte, J. Evans, V. G. Brunton, and M. C. Frame, "The role of focal-adhesion kinase in cancer – a new therapeutic opportunity," Nature Reviews Cancer, vol. 5, Jul 2005, pp. 505 EP –, review Article.

[65] Z. He, F. Tang, Z. Lu, Y. Huang, H. Lei, Z. Li, and G. Zeng, "Analysis of differentially expressed genes, clinical value and biological pathways in prostate cancer," Am J Transl Res, vol. 10, no. 5, May 2018, pp. 1444–1456.

 ISBN: 978-1-61208-792-4