BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

CLASSIFICATION OF METAGENOMIC SAMPLES USING DIGITAL PROCESSING OF GENOMIC SIGNALS

KLASIFIKACE METAGENOMICKÝCH VZORKŮ S VYUŽITÍM ČÍSLICOVÉHO ZPRACOVÁNÍ GENOMICKÝCH SIGNÁLŮ

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR
AUTOR PRÁCE

Filip Najbr

SUPERVISOR
VEDOUCÍ PRÁCE

Ing. Kristýna Kupková

BRNO 2018

VYSOKÉ UČENÍ
TECHNICKÉ
V BRNĚ TECHN

Bachelor’s thesis

Bachelor’s study field **Biomedical Technology and Bioinformatics**
Department of Biomedical Engineering

Student: Filip Najbr ***ID:*** 175271
Year of study: 3 ***Academic year:*** 2017/18

TITLE OF THESIS:

Classification of metagenomicsamples using digital processing of genomic signals

INSTRUCTION:

1) Prepare a literature review of metagenomics sample classification methods. 2) Study the methods of transforming genomic sequences into numerical form and the features extractable from this signal representation. 3) In a selected programming environment, create a program for the genomic sequence transformation into a signal representation followed by extraction of features for classification. 4) Complete the existing program with machine learning methods allowing automatic classification. 5) Test the resulting program on a sample of real data obtained from public repositories and statistically evaluate the results.

REFERENCE:

[1] DING, Xiao, Fudong CHENG, Changchang CAO a Xiao SUN. DectICO: an alignment-free supervised metagenomic classification method based on feature extraction and dynamic selection. BMC Bioinformatics [online]. 2015, 16(1). DOI: 10.1186/s12859-015-0753-3. ISSN 1471-2105.

[2] CUI, Hongfei a Xuegong ZHANG. Alignment-free supervised classification of metagenomes by recursive SVM. BMC Genomics [online]. 2013, 14(1), 641. DOI: 10.1186/1471-2164-14-641. ISSN 1471-2164.

Assigment deadline: 5.2.2018 ***Submission deadline:***25.5.2018

Head of thesis: Ing. Kristýna Kupková

prof. Ing. Ivo Provazník, Ph.D.
Subject Council chairman

WARNING:

The author of this Bachelor’s Thesis claims that by creating this thesis he/she did not infringe the rights of third persons and the personal and/or property rights of third persons were not subjected to derogatory treatment. The author is fully aware of the legal consequences of an infringement of provisions as per Section 11 and following of Act No 121/2000 Coll. on copyright and rights related to copyright and on amendments to some other laws (the Copyright Act) in the wording of subsequent directives including the possible criminal consequences as resulting from provisions of Part 2, Chapter VI, Article 4 of Criminal Code 40/2009 Coll.

ABSTRAKT

Cílem této práce je využití metod sloužících k číselnému zpracování genomických signálů a následná tvorba programu, který pomocí těchto metod, vytvoří vhodnou numerickou reprezentaci metagenomických vzorků, vyextrahuje z ní vhodé příznaky a pomocí nich rozliší jedince zdravé a jedince s onemocněním diabetes mellitus 2. typu za použití metod strojového učení.

KLÍČOVÁ SLOVA

zpracování signálů, metagenomika, klasifikace, diabetes mellitus 2. stupně

ABSTRACT

The aim of this thesis is the use of methods for numerical processing of genomic signals and the subsequent creation of a program, which by using these methods creates a suitable numerical representation of metagenomic samples, extracts appropriate features and classifies healthy individuals and individuals with type 2 diabetes mellitus using machine learning methods.

KEYWORDS

signal processing, metagenomics, classification, type 2 diabetes

NAJBR, F. *Classification of metagenomic samples using digital processing of genomic signals.* Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, 2017. 45p. Vedoucí práce: Ing. Kristýna Kupková.

DECLARATION

I declare that I have elaborated my bachelor’s thesis on the theme of “Classification of metagenomic samples using digital processing of genomic signals” independently under the supervision of the bachelor’s thesis supervisor and using specialized literature and other information resources, all of which are quoted in the work and listed in the literature at the end of the thesis. As the author of this thesis I furthermore declare that in connection with the creation of this thesis I have not infringed the copyrights of third parties, in particular I did not interfere illegally with foreign authors' personal rights and I am fully aware of the consequences of violation of the provisions of Section 11 et seq. / 2000 Coll., Including the possible criminal law consequences resulting from the provisions of Part Two, Title VI. Part 4 of the Criminal Code No. 40/2009 Coll.

Brno May 25th 2018 ……………………………..

podpis autora

ACKNOWLEDGEMENT

I would like to thank to my supervisor Ing. Kristýna Kupková for the professional help, consultation and all the valuable advice she gave me during the writing of this work.

Brno May 25th 2018 ……………………………..

podpis autora

CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

As far as we know, almost all biological processes are dependant on microbes. Chemical cycles which are responsible for the key elements of life such as carbon, hydrogen, oxygen, nitrogen, phosphorus and sulfur are all at least touched by microorganisms. Today‘s life as it is, is unimaginable without using microbes in fields like pharmacology, medicine, agriculture, food industry etc.. Considered how important these organisms are for us, we propably get to conclusion they need to be thoroughly examined for our benefits. Metagenomics is cross-disciplinary scientific field, that is studying genetic material directly from a sample which can be gained from large scale of different environments e.g. sea, dirt or human gut. With rapid development of next generation sequencing there is immense opportunity to explore whole metagenomes of different individuals and their influence on them.

The aim of this thesis is to find out whether it is possible to classify an individual with type 2 diabetes from healthy one, using data extracted from their gut metagenome. In order to do such classification, digital processing of genomic signals, subsequent analysis of features extracted from these genomic signals and machine learning algorithm is used.

First part of this thesis is dedicated to current methods of classification of metagenomic samples. At the beginning of this section is short description of workflow of metagenome sequencing methods. This description is followed by reasons for using alignment-free methods instead of methods that are alignment-based. In the end of this section there is listed couple of aligment-free classification methods which are described in detail.

Next part covers various methods of digital processing of the genomic signal. It primarily deals with which signal representation is best for the feature extraction. It also mentions some different features that can be used for the classification of samples.

In the next section I briefly describe type 2 diabetes and discuss various reasons why this disease can affect microbial composition in the human gut.

Last theoretical part is about machine learning. In this chapter simple yet effective classification algorithm that is later used in practical part is fully described.

In the end of this thesis I am presenting my solution for classification of metagenomic samples and its result along with brief statistics and disscusion of outcome of this work.

1 CLASSIFICATION OF METAGENOMIC SAMPLES

The very basis of the metagenomic approach is to work with the genetic information of all the organisms present in the sample. Such a procedure gives us many benefits including: detection of the abundance of microorganisms in different environments, analysis of organisms that cannot be cultivated for various reasons or information on the composition and functions of various ecosystems [1], [2].

Thanks to the rapid fall in prices, next-generation sequencing (NGS) is now used, which significantly accelerates the development of metagenomics [3]. This type of analysis involves several essential steps. First, complete DNA is extracted from the entire sample. The DNA is then fragmented into smaller pieces using different techniques (e.g. sonification). Fragmentation is followed by the DNA ligation of the adapter for the final preparation e.g. Ilumina library. These libraries are sequenced by paired-end reads for the largest amplicons coverage. Use of single-end reads is also possible. The reads are then categorized and connected into contigs. At this point, an optional de novo genome assembly is available. In order to do this process genome binning of contigs is done to reconstruct the complete genome and assign them to the closest possible taxonomy. Additionally, a functional analysis can be performed to determine the functions of the appropriate genes [2]. This entire process is shown in Fig 1.

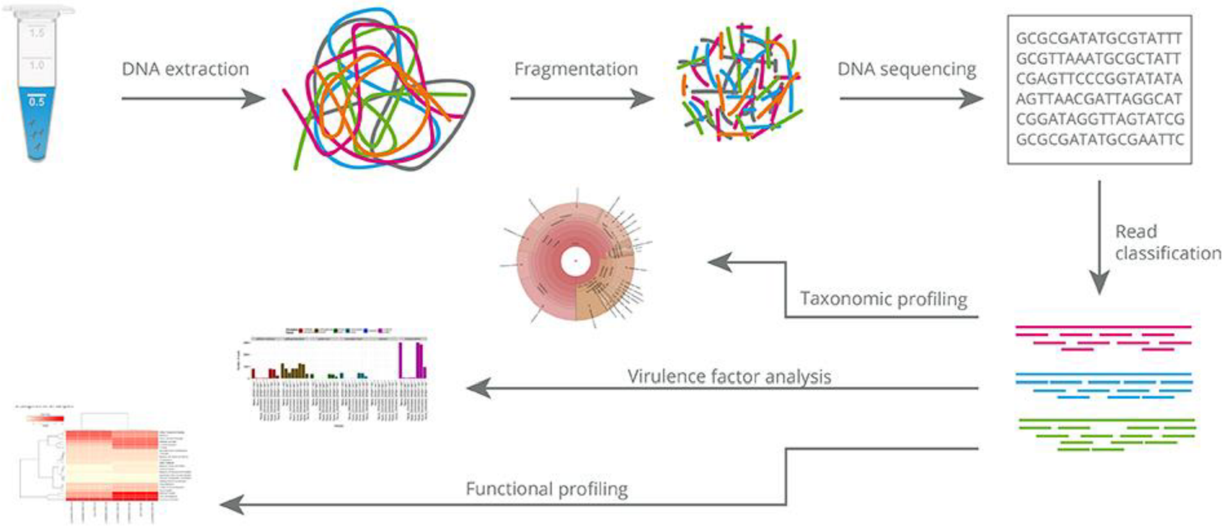


Fig 1. General process of metagenome analysis. Edited and obtained from [2].

In this thesis, metagenomic sample will be classified into two groups, based on whether it is from healthy individual or individual with type 2 diabetes.

1.1 Alignment-based methods

By the end of the 20th century, there has been great progress in the field of bioinformatics, mainly through the development of many alignment algorithms. Not only well-known BLAST algorithm, or multiple sequence aligners such as ClustalW, but also a whole-genome aligners such as BLASTZ or TBD. These tools have played a crucial role in obtaining information about genetic materials [4].

All alignment-based programs, regardless of the alignment process, have a common goal. Look for the similar order of bases or amino acids that are found in 2 or more sequences. Output of this process are series of matches, mismatches or (in case of inserted or deleted base) gaps. Alignment-based programs have their disadvantages, considering the complexity and volume of genetic information in the metagenomic approach [1], [4].

One problem is that programs based on alignment assume a homologous sequence with a number of linearly sequenced sections, known as collinearity. Unfortunately, in a metagenomic sample, this assumption is impaired, for example, by viral genomes that are different due to the high frequency of mutations.

The following disadvantage is the fact that not all the genomes are sequenced yet, which can lead to non-classified sequence or false positive result.

Another major problem is the memory and time consumption of these algorithms, which clearly indicates a limited application to this kind of data. The number of possible alignments for 2 sequences grows very fast with the sequence length according to the formula (1.1) [4]:

$$A = \frac{(2N)!}{(N!)^2} \tag{1.1}$$

where A is a number of possible alignments and N is length of a sequences. Result of aligning 2 sequences with N = 100, we get to 10^{60} possible alignments.

This problem can be partially resolved by dynamic programming that allows to find optimal alignment without saving all possible solutions, but it is too computationally difficult and time-consuming. Therefore, we are looking for faster and more efficient solution, which can alignment-free methods provide [1], [4], [5].

1.2 Alignment-free methods

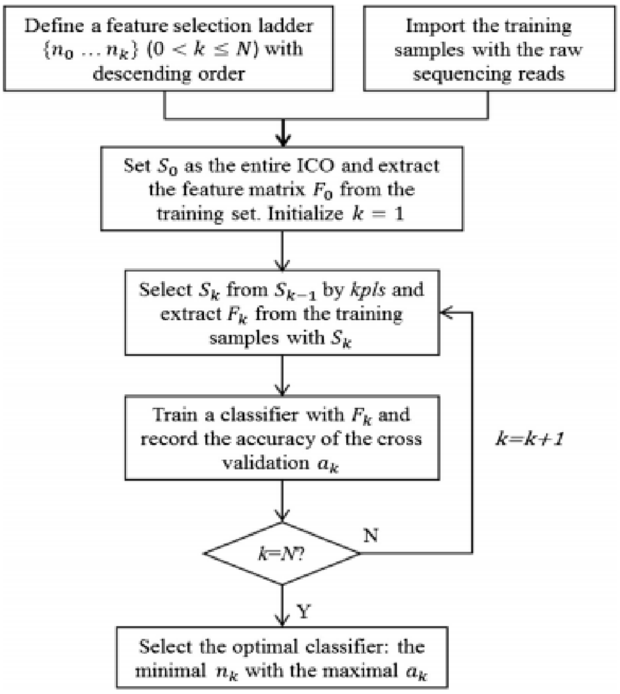
Alignment-free procedures can be defined as any method of quantifying a sequence similarity that does not use alignment, or whose output is not aligned at any step of the algorithm. These approaches are therefore suitable for comparing metagenomic samples as they are not dependent on dynamic programming and are significantly easier to compute. In addition, they do not affect recombination and mutation processes in viral genomes, so they are usable in cases where alignment-based procedures fail [4], [7]. The disadvantage of these methods is that they do not allow to identify the functional elements of the sequences [1], [6].

However, considering that samples sequenced so far is estimated to be only 10^{-20} % of the total DNA on Earth [7], alignment-free methods are providing fast solution of analyzing and obtaining information directly from raw NGS data. There is over a 100 different algorithms with this kind of approach [7]. In order to elucidate the procedures of at least some of them, the following two sections are available.

1.2.1 DectICO

Within all of the different techniques, ICO method was focusing on rather rare feature which is intrinsic correlation of oligonucleotides (ICO). It turned out to be effective as it was able to obtain better differences between genomic sequences, compared to other sequence-composition based feature methods. However, this algorithm seems to present inaccurate results due to high-dimensionality of the feature set. This kind of feature set also increased computational complexity. This inaccurate classification led to rebuilding original algorithm into a new one [1].

DectICO is supervised algorithm that classifies metagenomic samples through dynamic selection of optimal ICO feature set using kernel partial least squares. Workflow of DectICO method is shown in Fig 2[1].



Define a feature selection ladder $\{n_0 \dots n_k\}$ $(0 < k \leq N)$ with descending order
Import the training samples with the raw sequencing reads
Set S_0 as the entire ICO and extract the feature matrix F_0 from the training set. Initialize $k = 1$
Select S_k from S_{k-1} by *kpls* and extract F_k from the training samples with S_k
Train a classifier with F_k and record the accuracy of the cross validation a_k
k=k+1
k=N?
N
Y
Select the optimal classifier: the minimal n_k with the maximal a_k

Fig 2. DectICO algorithm process chart. Edited and obtained from [1].

Results obtained by this "enhanced" ICO algorithm were more accurate when longer oligonucliotides were used, which can be considered as an improvement of original algorithm [1].

1.2.2 Recursive SVM

Algorithms of supportive vectors belong to machine learning methods that are capable of separation non-linearly bounded data using a linear function. The basic principle is based on transforming the original two-dimensional space into the multi-dimensional one, where we are able to separate individual classes linearly. Using the algorithm, we are looking for hyperplane that optimally distributes training data. The data closest to the hyperplane are called supportive vectors [8], [9].

Recursive SVM (R-SVM), however, is algorithm modified to perform selection of the feature, while building the classifier in a recursive way through multiple steps that are following a given descendant ladder. There is, of course risk of overfitting. To minimalize this risk, the basic linear kernel is used in the SVM. This keeps the least model complexity for cases of small sample size, while high feature dimension. Each time feature is selected, SVM is first applied by R-SVM on all available features. Whole process is driven by decision function, which is very similar to basic artificial neuron function:

$$g(x) = \text{sgn}\{\sum_{i=1}^{n} a_i y_i (x_i . x) + b\}, \qquad (1.2)$$

where n is the number of samples in the training set, x is the feature vector of a test sample, x_i is the vector of training sample i and $y_i \in \{-1, 1\}$ is the corresponding class label. The parameters α_i's and b are trained from the training dataset by maximizing the separation margin and minimizing the prediction error on training data. The sum $\sum_{i=1}^{n} a_i y_i x_i$ is considered as the weight vector of the features. It can be also described as a contribution of the feature in the trained classifier. As a next step features are ranked by their differences between two classes, which are weighted by trained weights in the SVM. Top features are selected for the classification and feature selection at the next level [6].

This method was tested on metagenome dataset with high accuracy. It can be therefore presented as representative method for supervised classification [6].

2 DIGITAL PROCESSING OF DNA SIGNAL

Deoxyribonucleic acid (DNA) is made out of 4 basic building blocks (also called bases or nucleotides): adenine, guanine, thymine and cytosine. When DNA is processed and sequenced, desired output is string of letters in text format that basically shows us how is the DNA composed. This character representation of DNA is established by International Union of Pure and Applied Chemistry (IUPAC), and is summarized in a Tab 1 [10], [11]:

Character	**Explanation**
A	**A**denine
G	**G**uanine
T	**T**hymine
C	**C**ytosine
R	pu**R**ine (G or A)
Y	p**Y**rimidine (C or T)
M	a**M**ino (A or C)
K	**K**eto (G or T)
S	**S**trong interaction (G or C)
W	**W**eak interaction (A or T)

Tab 1. IUPAC Nucleotide code characters, edited and obtained from [10].

As mentioned in previous chapter, alignment-free classification methods are based on quantifying a sequence, so it can be approached as digital signal. Numerical representations serve this purpose [11].

2.1 Numerical representations

To process genomic signal as a digital one, conversion of character string into numbers is obviously needed. Throughout years of development in bioinformatics, many methods were created. [11] In this part of thesis is presented only small portion of them.

2.1.1 Voss Numerical Mapping

Amongst the popular techniques of DNA numerical representation is Voss Mapping method. It turns genomic sequence into 4 binary indicator arrays $x_A(n)$, $x_T(n)$, $x_C(n)$, $x_G(n)$. Each of this array is filled with zeroes and ones, depending on position of chosen nucleotide. Zero in position n where chosen nucleotide is not present, one in the opposite situation [11]. For random sequence: ATTGCA, Tab 2 explains the described process best:

n	1	2	3	4	5	6
Sequence	A	T	T	G	C	A
$x_A(n)$:	1	0	0	0	0	1
$x_C(n)$:	0	0	0	0	1	0
$x_G(n)$:	0	0	0	1	0	0
$x_T(n)$:	0	1	1	0	0	0

Tab 2. Example of indicator arrays for given sequence.

This method indicates purely the frequencies of the bases. It works very well for spectral analysis of DNA, which gives us information about coding region of the sequence. Unfortunately, this representation is 4 dimensional, because each base is represented by four dimensional vector [11]. That makes it impossible visualize without using Fourier transformation and therefore not very suitable for metagenomic classification [12].

2.1.2 DNA Walk

As the name of this method suggest, it uses so called "walker" to visualize change of course in DNA sequence. Walker is cumulative variable which changes along the sequence. There are several types of DNA Walk. There are 2 basic types presented here. First is 2-dimensional, the other is 1-dimensional. Both are described in detail in following sections [11].

2.1.2.1 2D DNA Walk

In two-dimensional DNA walk method walker is represented by a complex number that could be defined as:

$$W = 0 + 0j \tag{2.1}$$

before algorithm goes through sequence. Both real and imaginary value of W cumulates the values of x(n), where n is the position in the sequence and x is changing according to base present at position n:

$$\begin{aligned} x(n) &= 1 && \text{if A} \\ x(n) &= -1 && \text{if G} \\ x(n) &= j && \text{if T} \\ x(n) &= -j && \text{if C} \end{aligned} \tag{2.2}$$

This type of DNA walk can sufficiently map course of the DNA sequence, without losing any biological information. We can even notice feature like repetitive DNA behaviour(in the staircase like part of the graph) in Fig 3 [11]:

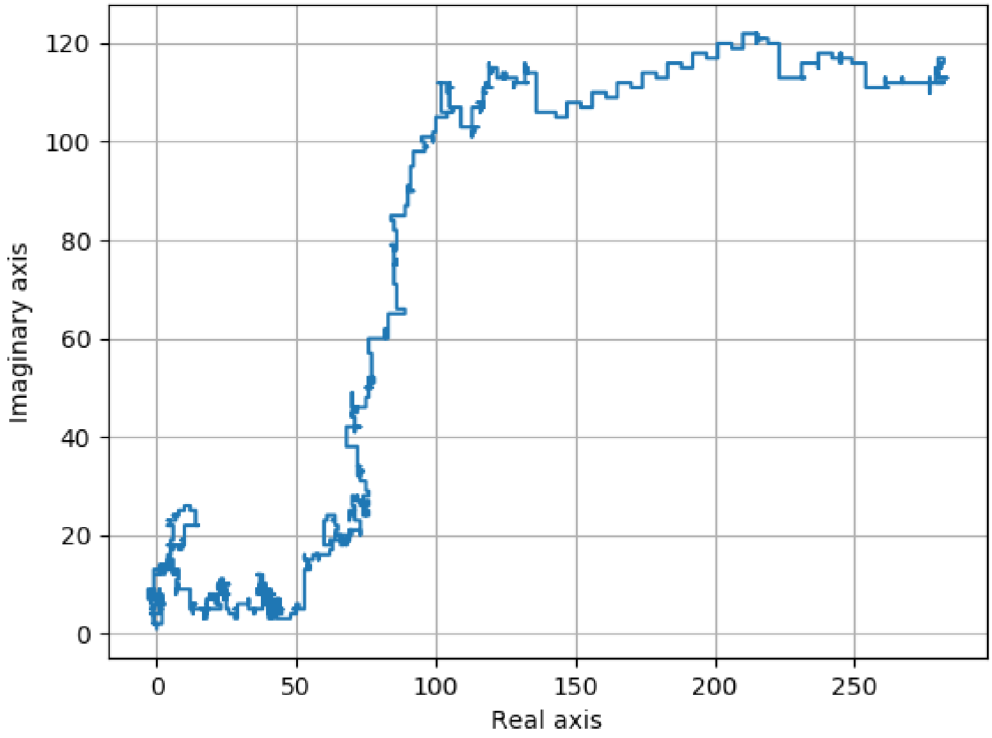


Fig 3. 2D DNA Walk of *Helicobacter pylori* [AE001439] (4066 to 5435).

Same as with Voss mapping this method provides us with multidimensional visualization, meaning that this method is not suitable for feature extraction and further analysis [12].

2.1.2.2 1D DNA Walk

In this case we have only one dimensional walker, which works similarly, but goes only up (+1) when pyrimidine (C or T) occurs in sequence or down (-1) in the case of purine (A or G). This type of walk therefore illustrates relative content of purine and pyrimidine and is shown in Fig 4[11]:

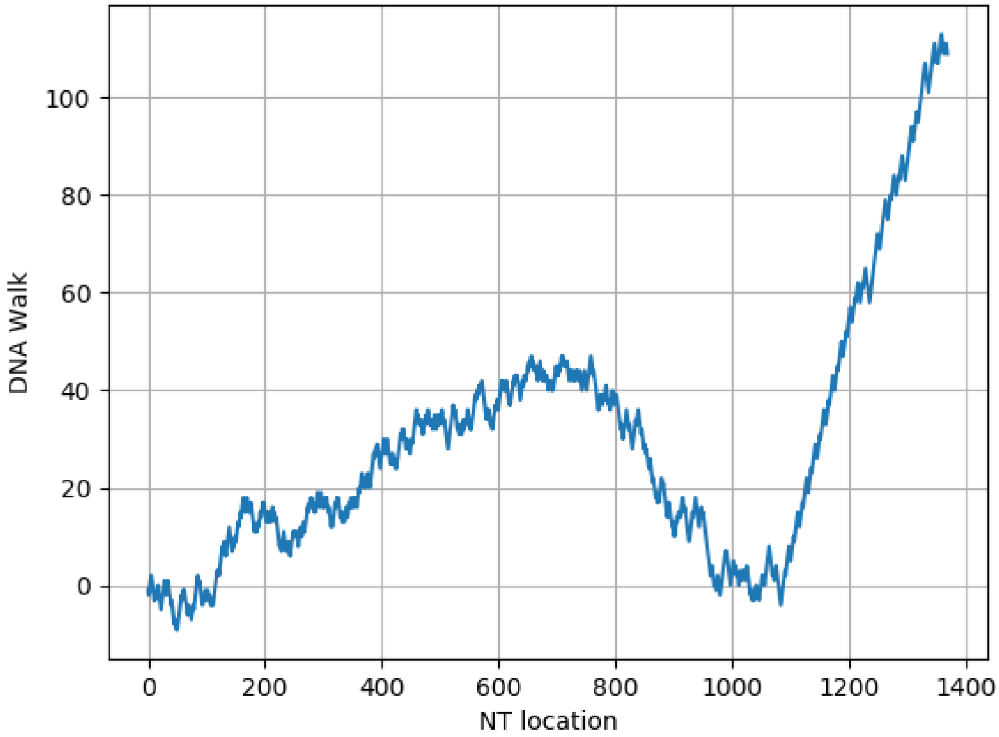


Fig 4. 1D DNA walk of *H. pylori* [AE001439] (4066 to 5435).

Although this representation is one dimensional and allows us easy feature extraction, it is not suitable because half of the biological information is lost, because of mapping only purines and pyrimidines, not individual bases [12].

2.1.3 Phase representation

This method of representing genomic sequence is based on visualizing phase of a complex number. Each of the bases is represented by complex number as Fig 5 shows:

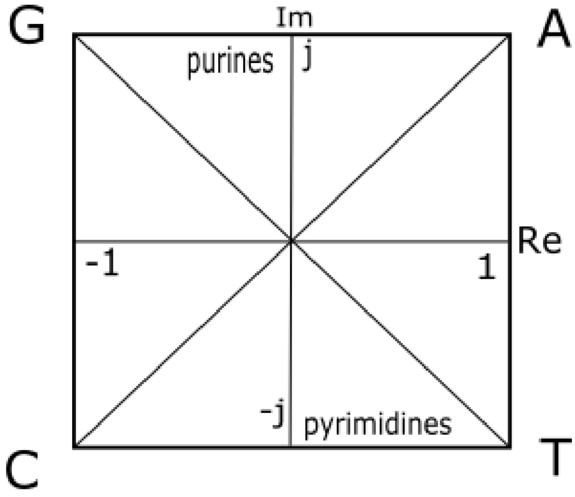


Fig 5. Complex representation of nucleotides. Edited and obtained from [11].

This allows us to assign specific angle (phase) for each base, so it can be distinguished without loss of biological information. That is possible through periodicity of a phase (it is not changed when adding or subtracting multiple of 2π). Various signal representations can be visualized through this method [13].

2.1.3.1 Phase

As mentioned above, each base of the sequence is represented by phase of a complex number. Specific values are:

$$A = \frac{1}{4}\pi \qquad T = -\frac{1}{4}\pi \qquad G = \frac{3}{4}\pi \qquad C = -\frac{3}{4}\pi \tag{2.3}$$

Assembling these values into an array, where each position corresponds to the position of the nucleotide leads to simple phase signal representation shown in Fig 6 and Fig 7:

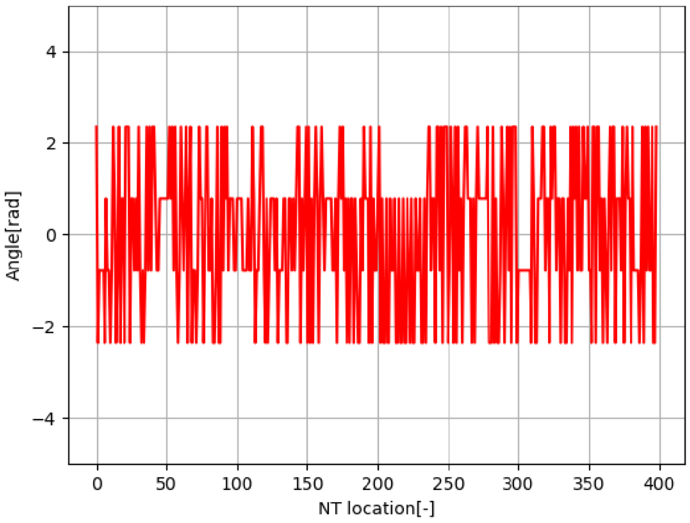


Fig 6. Phase representation of *Escherichia coli* str. K-12 substr. MG1655, bases 1-400

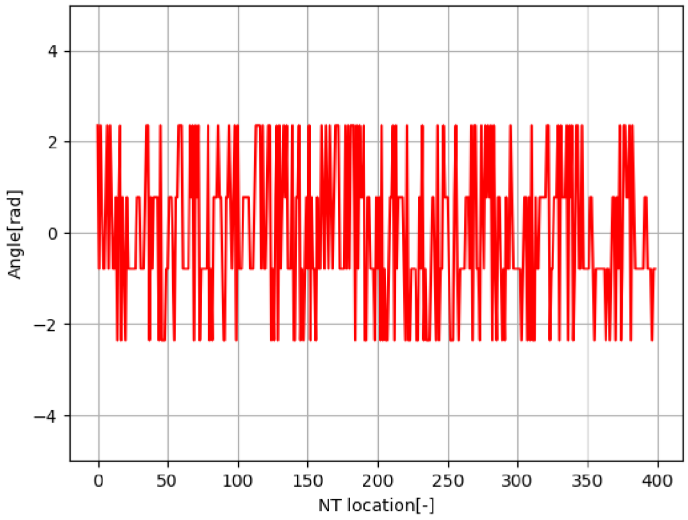


Fig 7. Phase representation of *H. pylori*, bases 1-400

Phase provides us simple yet good representation of genomic signal. It is suitable for feature extraction, because it is one dimension signal, with all biological information preserved [12], [13].

2.1.3.2 Cumulated phase

Is defined as a sum of the phases of the whole sequence from the first element to given position. That can be also represented by equation (2.4) [13]:

$$P_c = \frac{\pi}{4}\left[3\left(f_G - f_C\right) + \left(f_A - f_T\right)\right] \tag{2.4}$$

where P_c is cumulated phase, and f_n is sum of specific nucleotide (where n can be G, C, A or T) at a given position.

This method is suitable for feature extraction as well, because it possesses all the important properties of the previous method. Furthermore, it is visually possible to distinguish each organisms signal from one another [12], as it is presented in the next Fig 8:

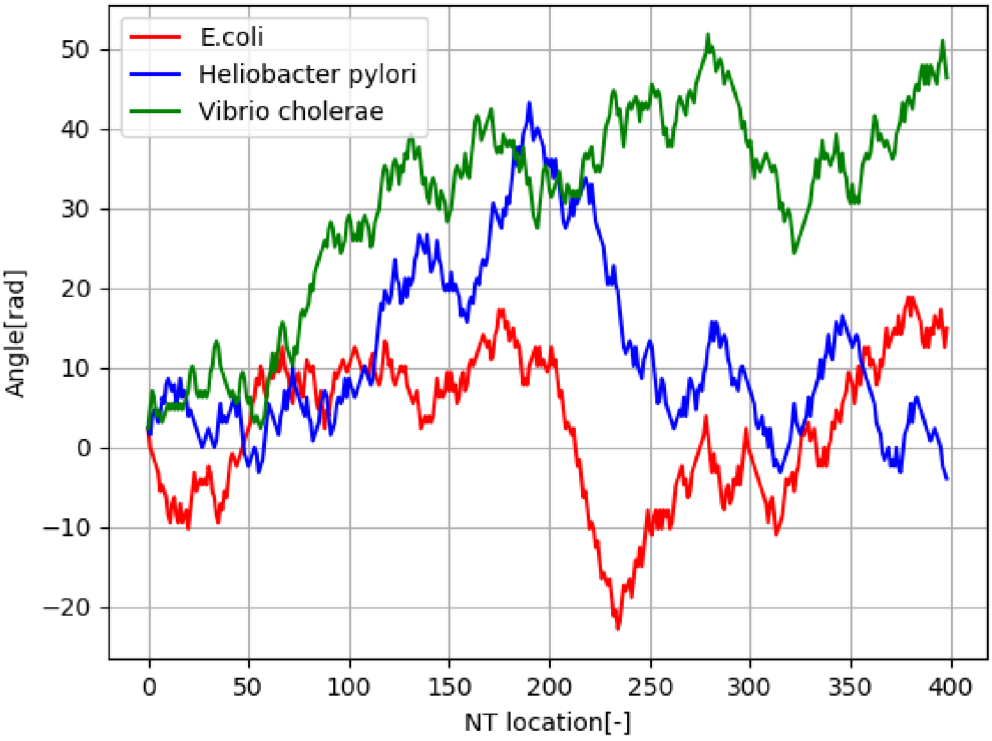


Fig 8. Cumulated phase of different organisms

2.1.3.3 Unwrapped phase

This representation is, as the previous two representations, suitable for feature extraction. It is different though, because it does not provide us information about nucleotide composition of the DNA. Instead it provides us information about relative frequency of transitions between nucleotides [13].

Again, each base has its corresponding phase, but the algorithm is driven by transitions between bases. These transitions are described as positive(A→G, G→C, C→T, T→A), where phase is corrected by its increase by π/2. Or negative (A→T, T→C, C→G, G→A), where correction of phase is done by its decrease by π/2. Other possible transitions are neutral [13]. Fig 9 shows results of this method:

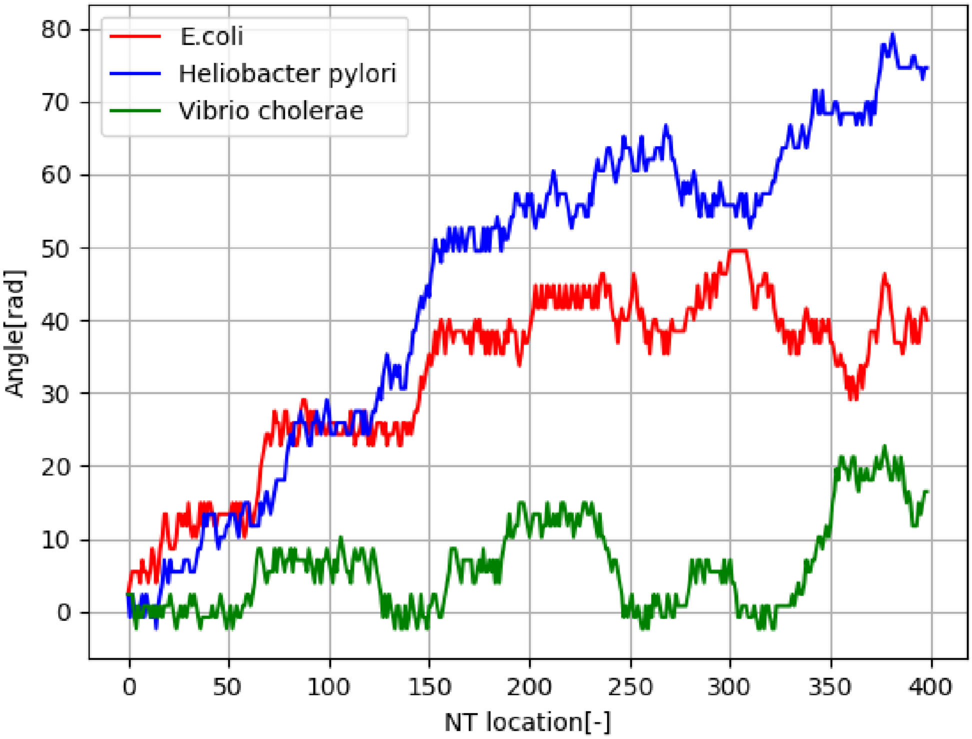


Fig 9. Unwrapped phase of different organisms

All of the phase representations are one dimensional, and preserve enough biological information to work with them in further analysis. Although it may seem from Fig 8 and Fig 9 that cumulated phase and unwrapped phase are enough to classify metagenomic data, it is not that simple. Metagenomic samples are large in the volume of data. Because of that, it is necessary to get simple information from larger amount of data. For this purpose, feature extraction is used.

2.2 Feature extraction

As mentioned in the previous chapter, feature extraction is essential step for obtaining information needed for classification of metagenomic samples. Assuming that there is a signal from which features can be extracted, there are plenty possible features that can be obtained. The following section describes those that seem appropriate for the classification and further analysis [12]. The signals from which the features are extracted are the already mentioned phase representations from the previous chapter.

2.2.1 Standard deviation

To understand the concept of standard deviation, it is necessary to define the magnitude on which it is based on. The magnitude is called variation. Variation indicates how statistic values differ from the average but in second power. Desired information is therefore squared and that complicates its interpretation. In order to get back to the original units, we need to square root the value of variance thus obtaining standard deviation. Variance is defined by equation (2.5):

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \qquad (2.5)$$

where n is a number of values in the sample, x_i is specific value out of set of measured values and $\bar{x}$ is mean value of the set of measured values. Standard deviation is then defined like this [14]:

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2} \qquad (2.6)$$

Extracting standard deviation out of phase, cumulated phase and unwrapped phase of different organisms can lead to interesting results which are shown in Fig 10.

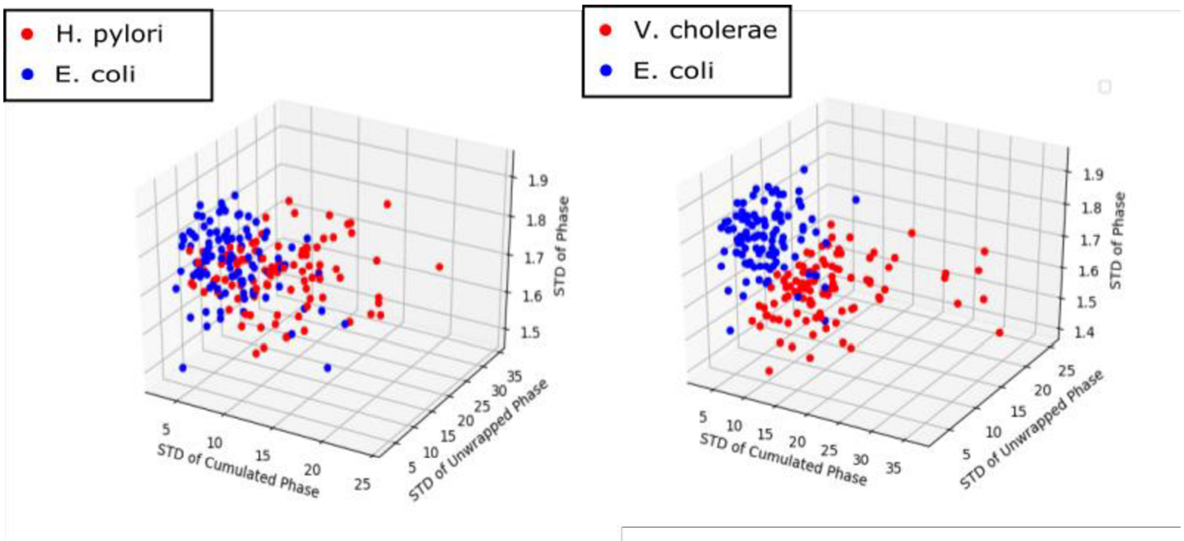


Fig 10. Standard deviations of *E.Coli*(blue), *H.pylori*(red - left) and *Vibrio cholerae*(red – right)

In both cases extraction of standard deviations is used from 100 DNA reads, that are 100 bases long. In Fig. 10 is possible to see clearly that 2 different clusters are formed, which are visually distinguishable. This can be there for used for classifying 2 different organisms. Problem might occur while classifying more organisms from metagenomic sample (clusters would overlap). However, goal of this work is to classify metagenomic samples into two groups of healthy and diseased individuals.

2.2.2 Hjorth descriptors

These features are originally used for analyzing EEG signals. However, signals of phases carry similar properties, like non-stationarity, so idea to use Hjorth descriptors as extracted feature is appropriate. This approach also significantly reduces computational time. There are three Hjorth descriptors and they are defined by following equations:

$$A = Activity = \sigma_0^2 \tag{2.7}$$

$$M = Mobility = \frac{\sigma_1}{\sigma_0} \tag{2.8}$$

$$C = Complexity = \frac{\frac{\sigma_2}{\sigma_1}}{\frac{\sigma_1}{\sigma_0}} \tag{2.9}$$

where σ_0^2 is variance of genomic signal, σ_1 and σ_2 are standard deviations of the first and second derivatives of the signal [12], [15].

Visualizing Hjorth descriptors leads to results which are shown in Fig 11.

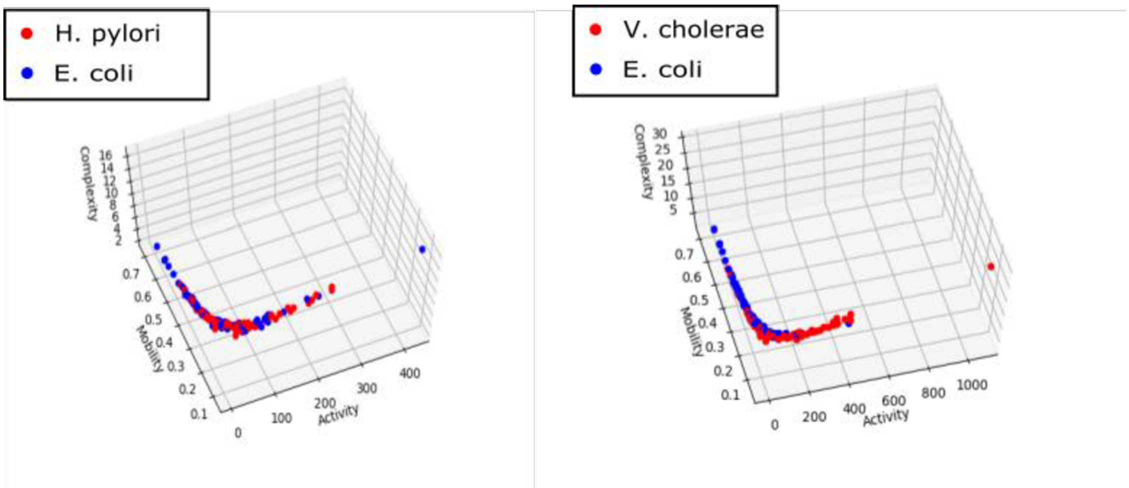


Fig 11. Extracted Hjorth descriptors from cumulated phase of *E.coli*(blue), *H. pylori*(red - left) and *V. cholerae* (red - right).

As Fig 11 and Fig 12 suggest this method might be usable for classification. However same problem as with standard deviation occurs. Classifying more organisms in metagenomic sample could lead to cluster overlap. Machine-learning algorithms or cluster analysis might at least partially solve that problem [12]. For classification into two groups, this method as well as standard deviation seem to be usable. However you can also see in Fig 11. Extracted Hjorth descriptors from cumulated phase of *E.coli*(blue), *H. pylori*(red - left) and *V. cholerae* (red - right). that there is a slight flexion of the data, which is not appropriate for classification. It is caused by usage of cumulative phase and therefore, it will be better to use standard phase to extract Hjorth descriptors as seen in Fig 12.

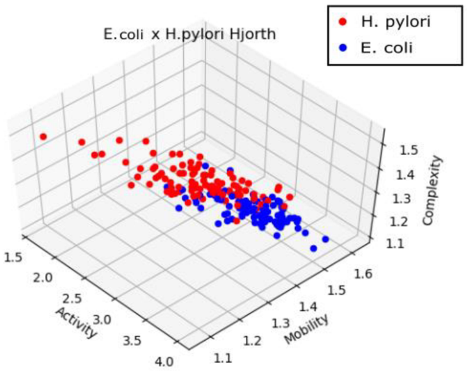


Fig 12. ***E. coli*** **X** ***H. pylori*** **Hjorth representation from standart phase**

Classifying taxonomies of different microorganisms is not the aim of this work, but it is important to show that, these features are able to distinguish one organism from another because the underlying cause of connection between metagenomics data and T2D probably lies in difference of gut microbiota of healthy and sick individuals, as next chapter clarifies.

With program that can extract features from signal representations, it is possible to step in the main area of this thesis which is classification of metagenomic samples in order to find out whether it is possible to distinguish individuals with type 2 diabetes (T2D) from healthy individuals based on their gut metagenome.

3 TYPE 2 DIABETES

Diabetes mellitus is metabolic disorder, which causes inability to process glucose in organism in physiological conditions due to relative or absolute insulin resistance. In this type of diabetes, the lack of insulin is relative, meaning that pancreas is producing enough insulin, but tissues in body aren't responding to it, in other words they are resistant. In later stages pancreatic β cells (cells that are responsible for producing insulin), may be depleted which leads to an absolute insulin deficiency [16]. Some of the risk factors for this disease are presented in a Tab 3 on the next page.

Risk factor	Relative risk
Age >= 45 years	5-6x
Obesity: BMI >= 30kg/m^2	4-5x
Overweight: BMI >= 25, < 30 30kg/m^2	2-3x
Hypertension	2-3x
Hyperlipidemia	4x
Family history	
One 1st degree relative or two 2nd degree relatives	2-3x
Two 1st degree relatives or one 1st degree and two 2nd degree relatives	5-6x
Genetic variant carrier	
Heterozygous	1,1 - 1,4x
Homozygous	Up to 10x

Tab 3. Relative risk associated with type 2 diabetes risk factors, taken from [17].

3.1 Diagnostics & metagenome classification

Conventional diagnosis of T2D is based on the presence of hyperglycemia. It is done by determining body response to insulin. T2D is diagnosed when postprandial increase in blood glucose is found (impaired ability of β-cells to respond to increased plasma glucose) [16].

Several studies [17], [18], 0, [19], have successfully shown that human gut metagenome is connected with a presence of this disease. Qin et al. [18] even pointed out that gathered metagenomic data from healthy and diseased individuals differ. Dysbiosis (state where balance of normal microbiota in human gut is disturbed) was found in diseased individuals. However, mentioned dysbiosis was only moderate, and in addition to that there was no other factor that would differentiate healthy and diseased individuals. Nevertheless, the study found that it is possible to differentiate these individuals through metagenomic samples [18].

This fact is crucial for the objective of this thesis which is distinguishing healthy individuals from T2D patients purely from the metagenomic data using digital signal processing. Methods described in previous chapters will be used on the data gathered to conduct mentioned study. The following chapter describes a simple machine learning algorithm that will be useful for classifying healthy and diseased individuals.

4 MACHINE LEARNING

Machine learning is nowadays a widely used tool. Voice recognition, web search engines, internet translators, smart cars, predictive financial software, and other modern day conveniences are now built on or working with machine learning algorithms. It is, therefore, undoubtedly a field that is already a major part of everyday life, and will increasingly influence us in the future.

Technically, it is the ability of the machine to learn to solve a problem for which it is not directly programmed. Such a problem can be, for example: sorting statistical data, prediction of behavior in particular system, optimalization of computational algorithm or already mentioned classification of different data. This chapter briefly describes machine learning algorithm, which is subsequently used in the practical part to classify metagenomic data.

4.1 Artificial neuron

The first comparison of the neuron cell and the process computer element was presented by Warren McCulloch and Walter Pitts in 1943. The element that represented the artificial neuron was practically a simple function with weighted inputs, a given threshold, and a binary output. This simple model, however, quite accurately describes the properties of a real neuron. Fig 13 is illustrating similarities between proposed model and real neuron.

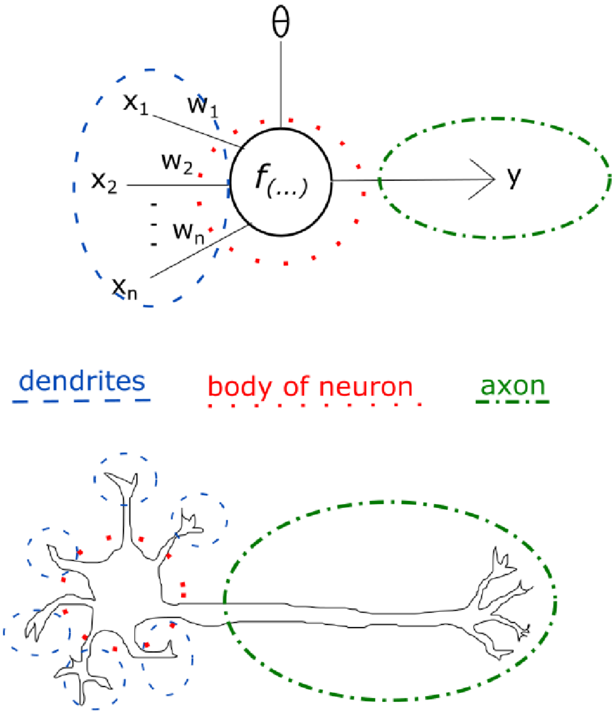


Fig 13. Figure of similarities between neuron cell and artificial neuron

Neuron is a sub-unit of a much more complex neural network, an organ that is responsible for processes throughout the body – the brain. Any action that is done by a human being is initiated by electric signal that enters through dendrites into the body of neuron, where it is accumulated. If the signal is large enough, in other words, exceeds a certain threshold, body of neuron sends an output impulse that is transmitted through the axon to perform the given action [21].

The artificial neuron works in the same way and is described by the equation:

$$y = f\left[\sum_{i=1}^{N} w_i x_i - \vartheta\right] \tag{4.1}$$

It has N possible inputs ($x_1, x_2, \ldots, x_N$) whose importance is given by weights ($w_1, w_2, \ldots, w_N$). Value of ϑ then defines the threshold necessary for so-called neuron activation. The body of the neuron is represented by the chosen function. For the classification of data into 2 groups as in this case, the unit step or sign function (due to binary output) is the most appropriate. These functions are shown on Fig 14:

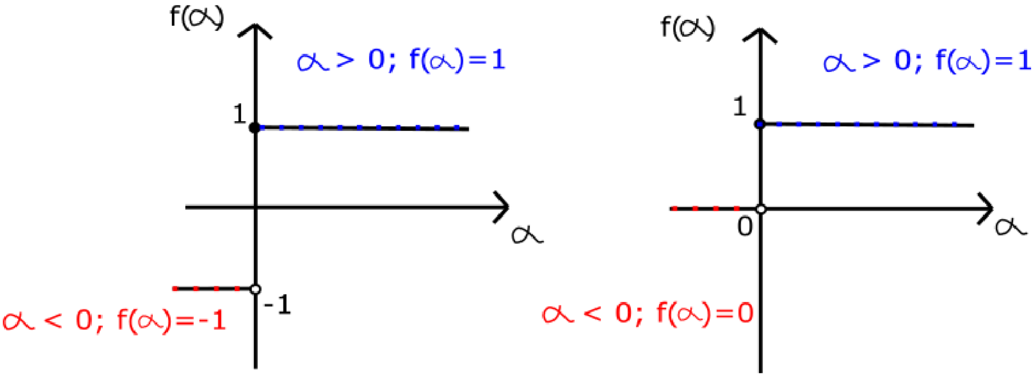


Fig 14. Sign function (left) and unit step function (right)

One artificial neuron with 2-dimensional inputs thus defines the boundary line that we obtain by simply modifying the equation (4.1) getting:

$$\begin{gathered} y = w_1 x_1 + w_2 x_2 - \vartheta = 0 \\ x_2 = -\frac{w_1}{w_2} x_1 + \frac{\vartheta}{w_2}, \end{gathered} \tag{4.2}$$

where $-\frac{w_1}{w_2}$ is a slope of a line and $\frac{\vartheta}{w_2}$ is representing its shift along axis x_2 [21].

Such boundary line is a good tool for classifying 2 different sets of data on a plane as displayed on Fig 15, where everything above is classified as group A, all under the line as group B.

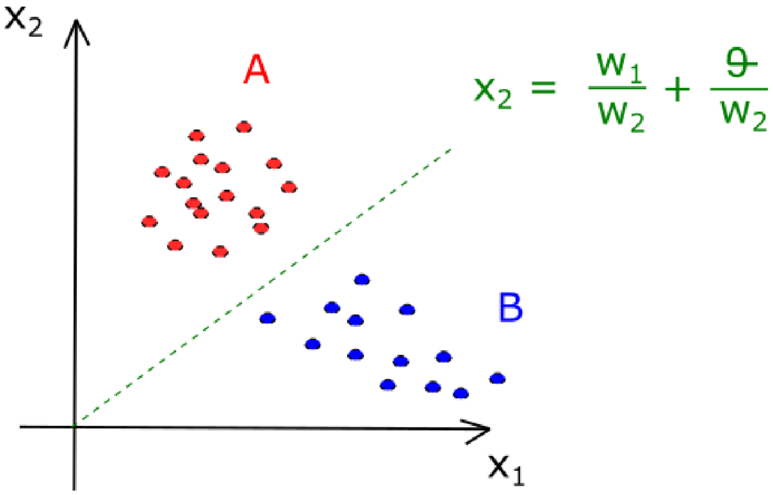


Fig 15. Classification of 2 random sets of data by artificial neuron

For classification of data in 3-dimensional space, neuron must be fed with 3-dimensional input and weights. Other thing needed is an algorithm that teaches the computer to construct a line that best differentiates these two groups of features. Such algorithm is described in following part of this chapter.

4.2 Supervised learning and δ-rule

Supervised learning is the process of finding best values for weights. In order to have neuron capable of successful classification of new input data, training on already classified data is mandatory. Learning is then iteration process, where we adapt weights after each evaluation of input and output data. Training data must be arranged at random. One iteration of this process, where every piece of training data was sent into the neuron, is called epoch. In the end of each epoch, recapitulation is done to check how many of the input data neuron successfully classified (in other words how many times the weights were changed). From this recapitulation is then concluded if the training process continues or is done [21].

One of the many ways to adapt weights of neuron used as simple classifier is so-called δ-rule. This way adapts the weights through following equation:

$$\overline{w}(t+1)=\overline{w}(t)+\mu\left[d(t)-y(t)\right]\overline{x}(t) \qquad (4.3)$$

where correction of weights is dependent on the value of deviation of output y from desired output d and on learning rate μ and input values x. Learning rate is set in range of 0 to 1 and its value affects how much the weights will change in next iteration [21]. Whole algorithm is briefly described in Fig 16 below:

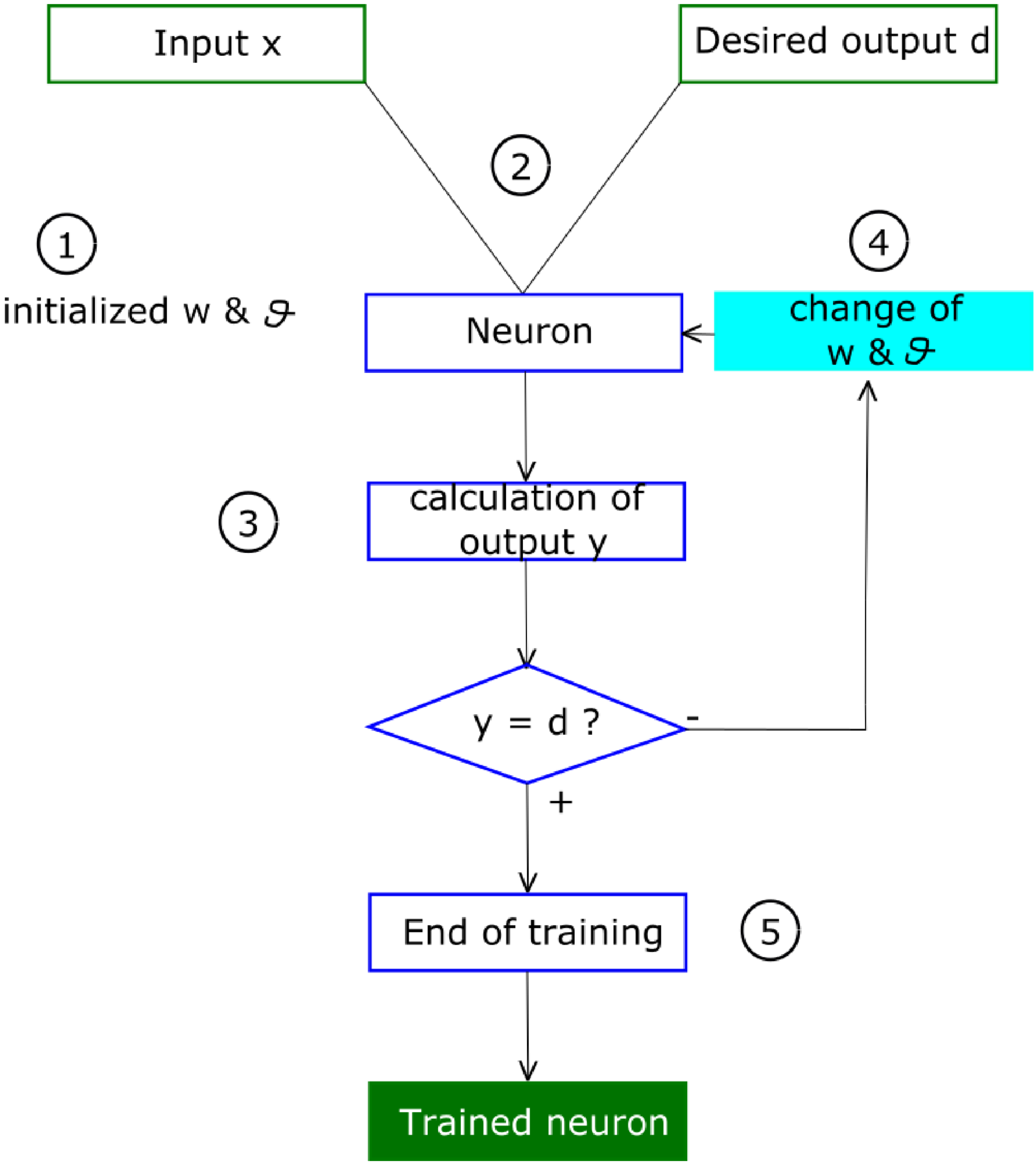


Fig 16. Flowchart of δ-rule algorithm.

This is last theoretical part of this work that needed to be covered in order to describe the whole process of practical part of this thesis. Testing the real dataset and summarizing the result of the tests is introduced in following chapter.

5 RESULTS

In order to test the actual datasets with mentioned methods, Python (version 3.6.5) program was implemented using following libraries: numpy, math, Biopython, matplotlib, mpl_toolkits.mplot3d, xlrd and xlswriter. All information and documentation about the libraries is freely accessible on official websites. Commented source codes of each part of the program are available in supplementary data.

Program itself is divided into 2 working blocks that are described in the flowcharts in Fig 17:

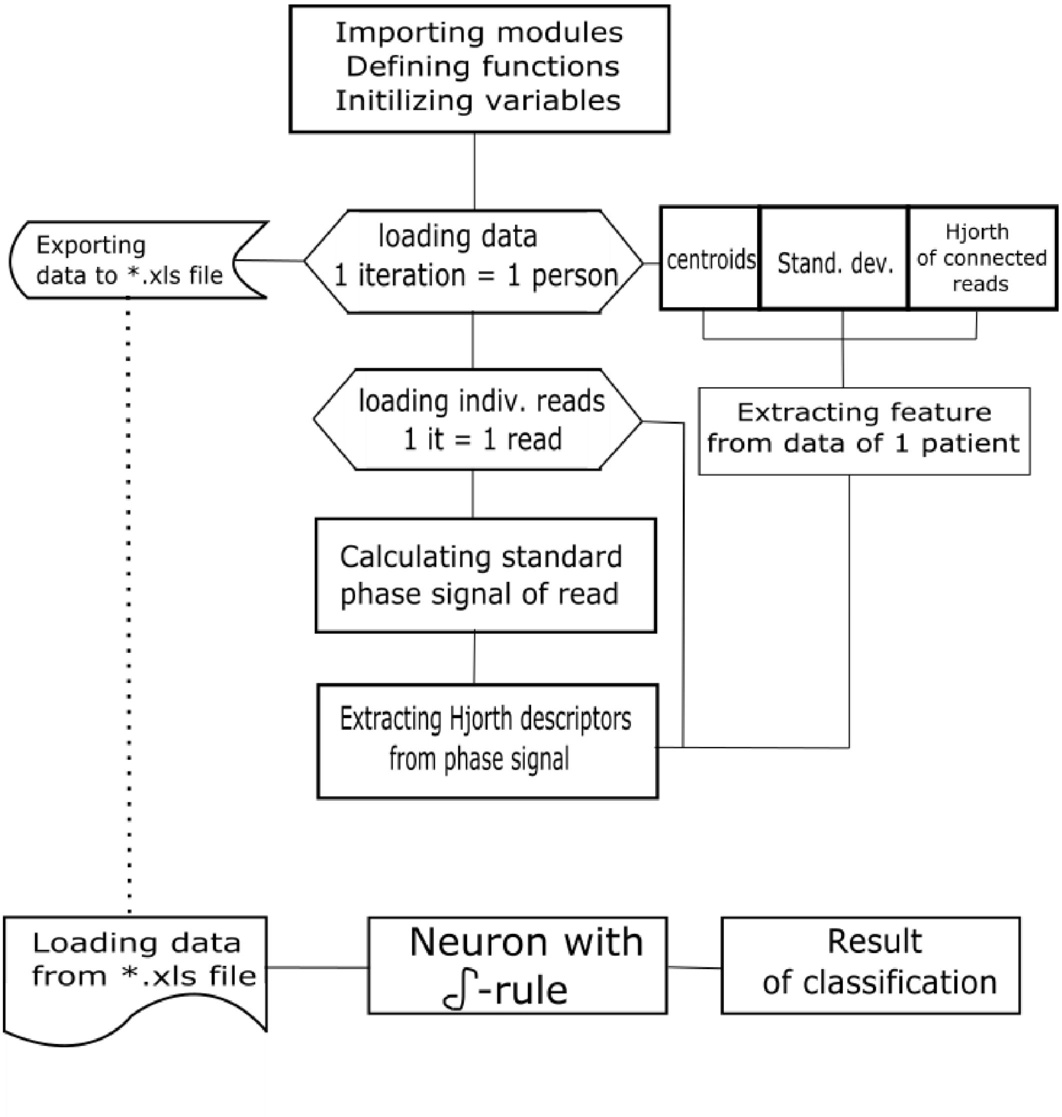

Importing modules
Defining functions
Initilizing variables
Exporting data to *.xls file
loading data
1 iteration = 1 person
centroids
Stand. dev.
Hjorth of connected reads
loading indiv. reads
1 it = 1 read
Extracting feature from data of 1 patient
Calculating standard phase signal of read
Extracting Hjorth descriptors from phase signal
Loading data from *.xls file
Neuron with ∫-rule
Result of classification

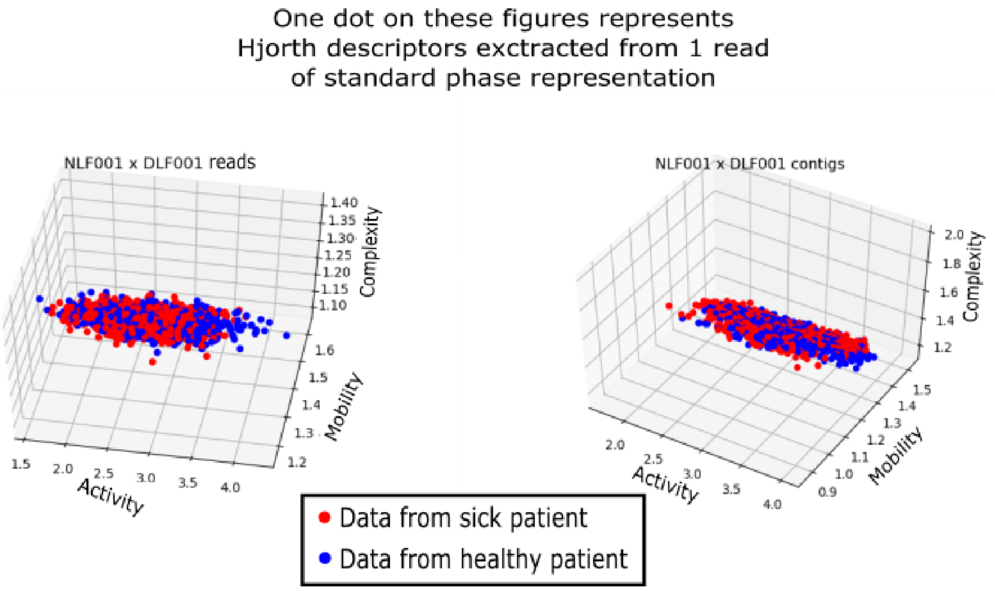
Fig 17. Flowchart of implemented program

5.1 Used datasets

For testing implemented classification algorithm, data offered by study [18] from its first stage were used. It is dataset containing sequenced gut metagenomes of 145 Chinese individuals (age from 14 to 75 years), from which 71 suffer from T2D and 74 are healthy control group, further information gathered about those patients can be found in supplementary data.

Testing the whole dataset that mentioned study offered would be very time consuming and would brought other problems with data mining, processing and also implementing much advanced structure, since dataset from whole metagenome of only 1 person is approximately 3 – 5 gigabytes big.

First decision to make was to choose between parts of contigs of whole data or just individual reads downloaded directly from SRA database. To decide which part will be better for purposes of this thesis couple of reads and parts of contigs was tested and plotted through mpl_toolkits.mplot3d module which Python offers. In terms of resulted plots, very similar results are achieved in both contigs and read as seen in Fig 18:



One dot on these figures represents
Hjorth descriptors exctracted from 1 read
of standard phase representation

Fig 18 Comparison of contig and read plots of metagenomic data
Red dots – patient DLF001, blue dots – patient NLF001

Reads have a constant length of 148 bp, but their disadvantage is that some bases are not identified and it is important to take this into account when implementing the program. Portions of contig are corrected so this is not the problem. On the other hand, parts of contigs are different in size without any alteration, and each further processing result will therefore carry a different amount of information. For this particular reason, the tests presented in the following parts of this work are performed on 1,000 random reads per 1 patient, which were downloaded directly from the SRA database using fastq-dump program. Two groups of files were downloaded, as mentioned in the first paragraph of this section. These files are also available in supplementary data.

5.2 Used feature extraction

The next step to solve the problem of classifying downloaded metagenomic samples is to select the correct feature extraction. From the amount of data shown in Fig 19:

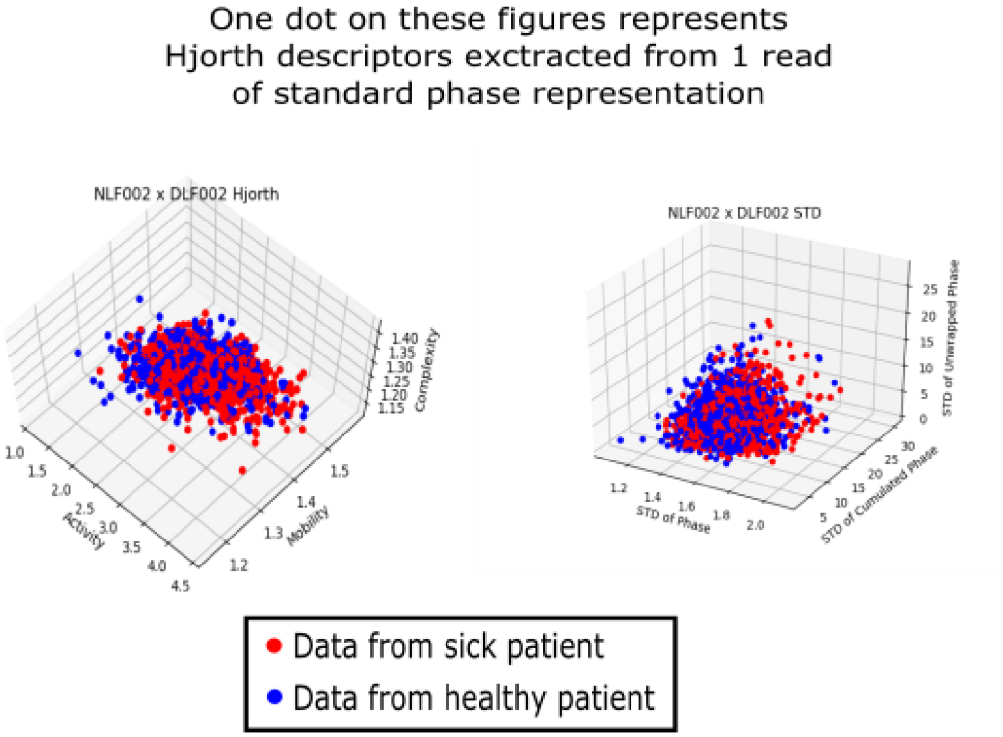


Fig 19. Example of different feature extraction from metagenomic data. Red dots – DLF002, blue dots –NLF002.

it can easily be inferred that classification by a mere neuron would not be possible in this case. The biggest problem would occur if this approach would be tested on whole datasets since they come out with different lengths, making the number of features different for each patient. One cloud, which represents the patient in any of the graphs shown, should be modified/simplified to get the same amount of features for all patients and at the same time it should keep as much original information as possible.

The procedure chosen to test the algorithm proposed for this work was extracting another feature from these clusters of data, keeping all of the information of the 1 patients cloud in 3 coordinates. Please note that from now on, every described feature was extracted was from Hjorth descriptors cloud, since they were proved to be calculated faster than standard deviations of different phase representations.

Following subchapters are presenting several options of follow-up feature extraction.

5.2.1 Centroids

Centroid is nothing else but the calculated center of a given cluster of data. Calculation of centroid coordinates is done according to equation 5.1:

$$x_c = \frac{\sum_{i=1}^{N} x_i}{N}; y_c = \frac{\sum_{i=1}^{N} y_i}{N}; z_c = \frac{\sum_{i=1}^{N} z_i}{N} \tag{5.1}$$

where x_i (Activity), y_i (Mobility) and z_i (Complexity) are coordinates of all points in the cluster and N is number of elements in the cluster. Fig 20 shows how the centroid layout is plotted from each patient's Hjorth parameter clusters.

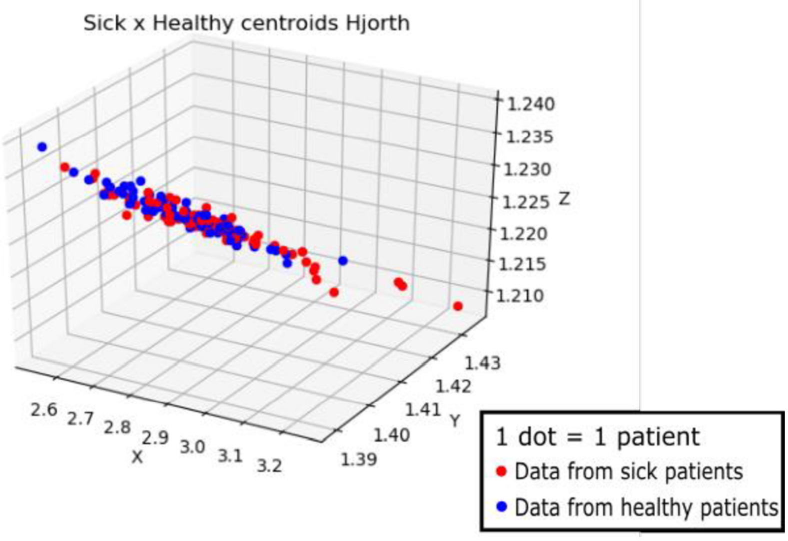


Fig 20. Centroids extracted from Hjorth descriptors of each patient.

5.2.2 Standard deviation

This approach is very similar to the previous one except that the feature extracted from the cluster of Hjorth descriptors is a standard deviation instead of centroid. On Fig 21 its plotted layout can be seen.

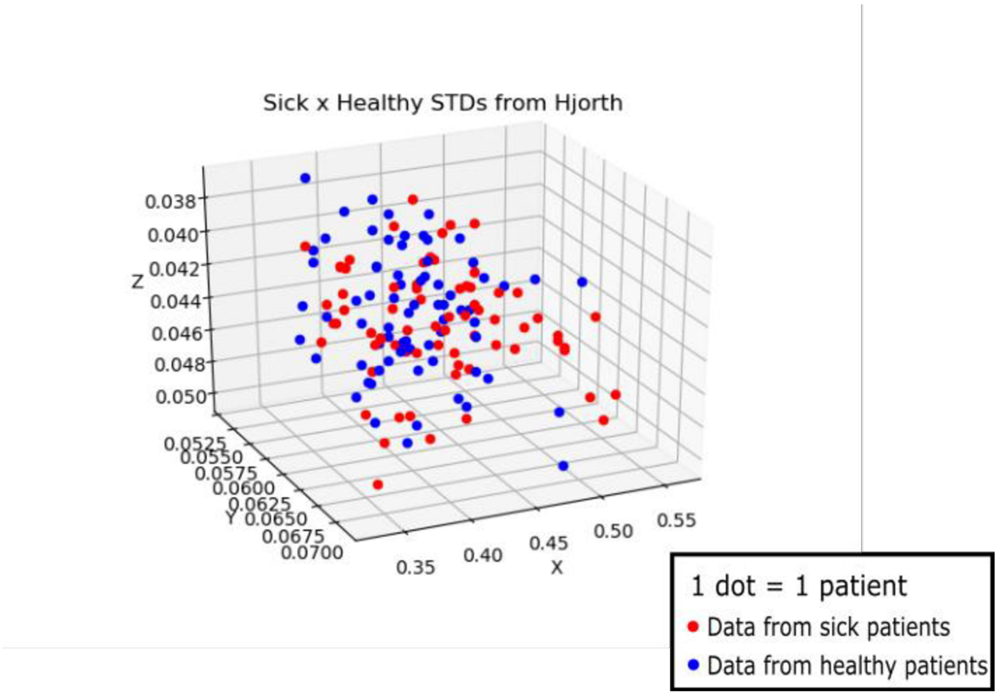


Fig 21. Standard deviations extracted from Hjorth descriptors.

5.2.3 Hjorth descriptors of connected reads

The last approach is somewhat different from the previous two. In this case, we do not calculate Hjorth descriptors from each read separately, but 1 Hjorth descriptor from all reads linked together from one patient getting immediately 3 coordinates in 3D space for each patient. Plotted layout of this feature is in Fig 22:

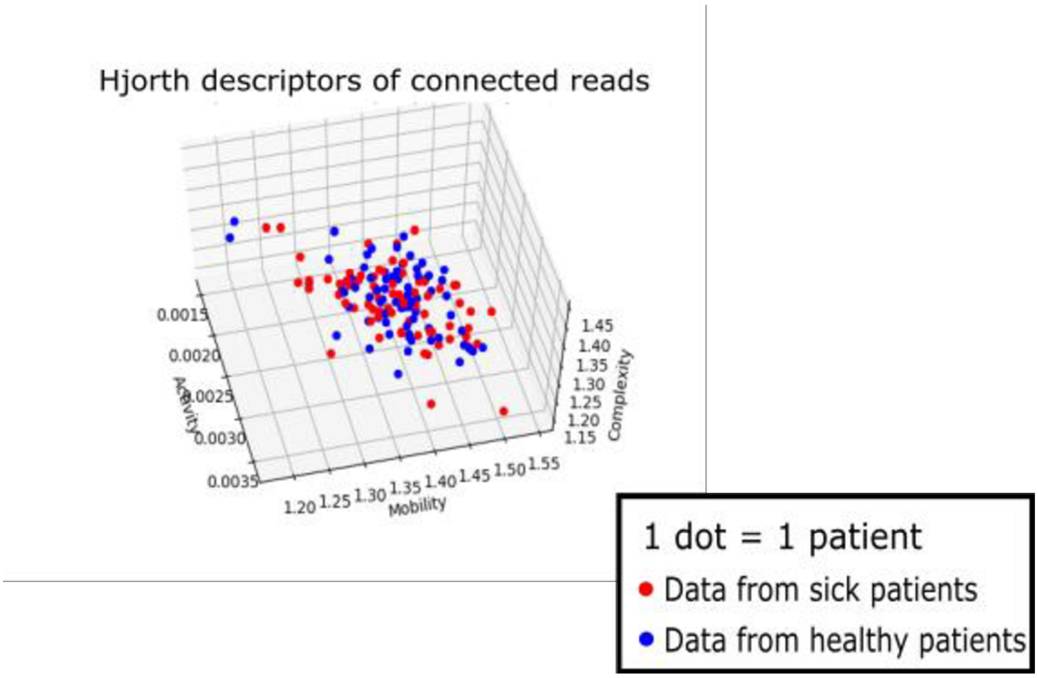


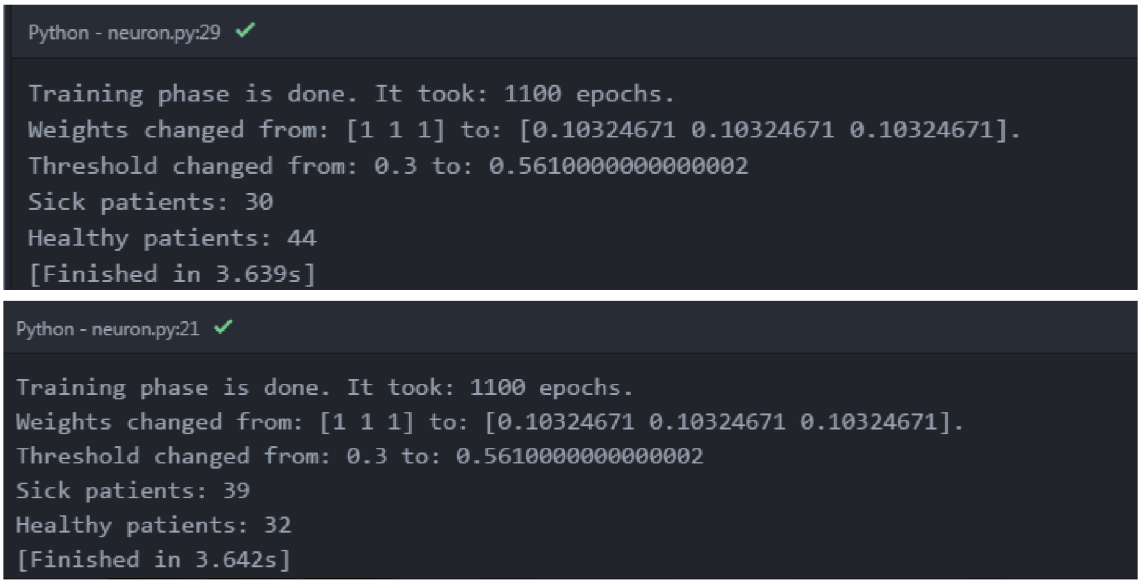
Fig 22. Hjorth descriptors extracted from connected reads of each patient.

All of these features were extracted from the data mentioned in the previous section using the featureExtractionScript.py, which can also be found in the supplementary data along with exported features in *.xls files.

These features were classified with neuron using δ-rule. Results are presented and discussed in the following section.

5.3 Classification outputs & statistics

The final features extracted from the data have been submitted to the neuron. The initial parameters were estimated to: $w = [1, 1, 1]$, $\vartheta = 0{,}3$, $\mu = 0{,}001$. Training group was designed from actual features specifically for each group of feature extraction. For example: from extracted standard deviation, data from 3 sick and 2 healthy individuals were taken and very slightly changed (6th decimal spot) to make a training group for neuron. With this setup classification process started. Example of its output can be seen on Fig 23Fig 23:



```
Training phase is done. It took: 1100 epochs.
Weights changed from: [1 1 1] to: [0.10324671 0.10324671 0.10324671].
Threshold changed from: 0.3 to: 0.5610000000000002
Sick patients: 30
Healthy patients: 44
[Finished in 3.639s]
```

```
Training phase is done. It took: 1100 epochs.
Weights changed from: [1 1 1] to: [0.10324671 0.10324671 0.10324671].
Threshold changed from: 0.3 to: 0.5610000000000002
Sick patients: 39
Healthy patients: 32
[Finished in 3.642s]
```

Fig 23. Output of classification of healthy control group(up) and sick individuals(down) represented by Hjorth descriptors of connected reads.

Rest of the outputs is well-documented in supplementary data. All them are summarized in statistical Tab 4:

Centroids

TP: 29	FP: 21
FN: 42	TN: 53
Specificity:	71,622%
Sensitivity:	40,845%
Accuracy:	56,552%

Standard Deviations

TP: 44	FP: 40
FN: 27	TN: 34
Specificity:	45,946%
Sensitivity:	61,972%
Accuracy:	53,793%

Hjorth of c.r.

TP: 39	FP: 30
FN: 32	TN: 44
Specificity:	59,459%
Sensitivity:	54,930%
Accuracy:	57,241%

Tab 4. Statistical evaluation of accuracy, efficiency and results of implemented algorithms.

5.4 Discussion

Result of this work are two scripts programmed in the Python (3.6.5). One is used for feature extraction from metagenomic data and the other is used for classification of patients with T2D. Both of these programs are usable with minor adjustments (for example: changing variable with number of patients, or number of reads, etc.) on samples of any formats supported by Biopython library.

Here, mentioned programs were used on parts of data provided by the study [18] and reached the results summarized in the Tab 4.

At first glance, we can claim that the classification of any of the three methods was unsuccessful. None of these approaches exceeded the threshold of 60% and therefore, this method cannot be declared successful on the basis of the tests done in this thesis. Out of all three approaches, Hjorth descriptors from connected reads seem to be most accurate one, with balanced specificity and sensitivity. Nevertheless, its accuracy is only around 57% which is not enough.

The key to success in this case could be definitely the test of the entire dataset, not just its parts as it is done in this work. Another possible upgrade of this method would be the use of a better machine learning algorithm (e.g. artificial neural network using deep learning to analyze the whole datasets).

6 CONCLUSION

This thesis describes different methods of classification of metagenomic samples and how they can be done. Such methods can be used in different manners of biological exploration. This work particularly targets the problem of classifying patients with T2D.

First chapter describes current methods and sums up all basic information needed to dive into digital processing of DNA signal. It also briefly describes some machine learning algorithms that were previously tested and proven useful in classification of metagenomics samples. Here however I present different approach and machine learning algorithm and this chapter is therefore, to present and introduce the reader to other approaches that showed some good results.

Digital processing of DNA signal is described in second chapter. There is fair amount of information about ways of converting character strings to digital signal. It focuses mainly on signal representations that are used on the real-data tests in chapter 5. Phase representations were proven very useful for metagenomic data in many cases and are essential in case of this work too. It also describes the other but not less important part of DNA signal processing which is extracting features important for classification of a real data.

Third chapter was written to summarize important information about the condition of type 2 diabetes. It mainly points out why should be the goal of this work possible.

The last theoretical chapter is a brief introduction to machine learning algorithm used in this thesis.

Practical part of this work describes how implemented program works and on which data it was tested. Each approach that was used to test the mentioned datasets is fully described. Final test on real data was not successful in case of this work. Three different features were extracted and used to classify without providing any satisfying result. In the following discussion I summarize the results of practical part of this thesis and contemplate how better results could be achieved. I think approaches presented in this thesis might have some potential but they must be definitely used tested on the larger sets of data.

BIBLIOGRAPHY

[1] *DING, Xiao, Fudong CHENG, Changchang CAO a Xiao SUN. DectICO: an alignment-free supervised metagenomic classification method based on feature extraction and dynamic selection. BMC Bioinformatics [online]. 2015, 16(1). DOI: 10.1186/s12859-015-0753-3. ISSN 1471-2105.*

[2] *Metagenome analysis. GATC Biotech [online]. London: GATC Biotech, 2017 [Accessed 27 December 2017].* Available from*:* https://www.gatc-biotech.com/en/expertise/genomics/metagenome-analysis.html

[3] THOMAS, Torsten, Jack GILBERT a Folker MEYER. *Metagenomics - a guide from sampling to data analysis. Microbial Informatics and Experimentation. 2012,* ***2****(1), 1-1. DOI: 10.1186/2042-5783-2-3. ISSN 2042-5783.*

[4] ZIELEZINSKI, Andrzej, Susana VINGA, Jonas ALMEIDA a Wojciech M. KARLOWSKI. *Alignment-free sequence comparison: benefits, applications, and tools. Genome Biology. 2017,* ***18****(1), 1-5. DOI: 10.1186/s13059-017-1319-7. ISSN 1474-760x.*

[5] *Editorial: Microbiology by numbers. Nature*. 2011, **2011**(Vol. 9), 1. DOI: DOI:10.1038/nrmicro2644. Available from: https://www.nature.com/articles/nrmicro2644

[6] *CUI, Hongfei a Xuegong ZHANG. Alignment-free supervised classification of metagenomes by recursive SVM. BMC Genomics [online]. 2013, 14(1), 641. DOI: 10.1186/1471-2164-14-641. ISSN 1471-2164*

[7] VINGA, S. *Editorial: Alignment-free methods in computational biology*. *Briefings in Bioinformatics*. 2014, **15**(3), 341-342. DOI: 10.1093/bib/bbu005. ISSN 1467-5463.

[8] *Matematická biologie* [online]. Brno: Masarykova Unverzita, 2015 *[Accessed 27 December 2017]*. Available from: http://portal.matematickabiologie.cz/index.php

[9] KŘÍŽENECKÁ, T. Automatic sleep scoring using polysomnographic data. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, 2017, 70 p, 32-32 Supervisor: Ing. Marina Ronzhina, Ph.D.

[10] Molecular biology review: Nucleotide base codes. *National Center for Biotechnology Information Search database* [online]. USA: National Center for Biotechnology Information, U.S. National Library of Medicine, 1994 *[Accessed 27 December 2017].* Available from: https://www.ncbi.nlm.nih.gov/Class/MLACourse/Modules/MolBioReview/iupac_nt_abbreviations.html

[11] ABO-ZAHHAD, Mohammed, Sabah M. AHMED a Shimaa A. ABD-ELRAHMAN. Genomic Analysis and Classification of Exon and Intron Sequences Using DNA Numerical Mapping Techniques. *International Journal of Information Technology and Computer Science*. 2012, **4**(8), 22-36. DOI: 10.5815/ijitcs.2012.08.03. ISSN 20749007.

[12] KUPKOVÁ, K. Methods for fast sequence comparison and identification in metagenomic data. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, 2016. 74 p., 10 p. supplements. Master’s thesis. Master’s thesis supervisor: Mgr. Ing. Karel Sedlář.

[13] CRISTEA, P.D. Phase analysis of DNA genomic signals. *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS '03*. IEEE, 2003, **2003**(1), V-25-V-28. DOI: 10.1109/ISCAS.2003.1206163. ISBN 0-7803-7761-3.

[14] Skripta Matematika 3. FAJMON, Břetislav, Irena HLAVIČKOVÁ a Michal NOVÁK. *Matematika 3*. 1. Brno: Ústav matematiky FEKT VUT, 2014, s. 137.

[15] KOZUMPLÍK, Jiří. *Analýza biologických signálů: Elektroencefalogram (EEG)* [online]. Brno, 2016 *[Accessed 28 December 2017]*. Available from: https://moodle.vutbr.cz/mod/folder/view.php?id=122421.
Prezentace. Brno university of technology.

[16] *Diabetes mellitus 2. typu (endokrinologie)* [online]. c2017 *[Accessed 27 December 2017]*. Available from: http://www.wikiskripta.eu/index.php?title=Diabetes_mellitus_2._typu_(endokrinologie)&oldid=393847

[17] JOHANSEN TABER, Katherine a Barry DICKINSON. *Genomic-based tools for the risk assessment, management, and prevention of type 2 diabetes*. *The Application of Clinical Genetics*. 2015, **2014**(8), 1-. DOI: 10.2147/TACG.S75583. ISSN 1178-704x.

[18] QIN, Junjie, Yingrui LI, Zhiming CAI, et al. *A metagenome-wide association study of gut microbiota in type 2 diabetes. Nature*. 2012, **490**(7418), 55-60. DOI: 10.1038/nature11450. ISSN 0028-0836.

[19] EVERARD, Amandine a Patrice D. CANI. Diabetes, obesity and gut microbiota. *Elsevier*. 2013, **27**(1), 73-83. DOI: 10.1016/j.bpg.2013.03.007. ISSN 15216918. Available from: http://linkinghub.elsevier.com/retrieve/pii/S1521691813000619

[20] ZHANG, Yong a Heping ZHANG. Microbiota associated with type 2 diabetes and its related complications. *Food Science and Human Wellness*. 2013, **2**(3-4), 167-172. DOI: 10.1016/j.fshw.2013.09.002. ISSN 22134530. Available from: http://linkinghub.elsevier.com/retrieve/pii/S2213453013000451

[21] KOZUMPLÍK, Jiří a Ivo PROVAZNÍK. Úvod do neuronových sítí. *Umělá inteligence v medicíně*. 1. Brno: Ústav biomedicínského inženýrství, FEKT VUT, 2007, 72 - 79.

LIST OF ABBREVIATIONS

NGS – Next-generation sequencing
DNA – Deoxyribonucleic acid
BLAST – Basic Local Alignment Search Tool
TBD – Track Before Detect
ICO – Intristic Correlation of Oligonucleotides
SVM – Supportive Vectors Machine
R-SVM – Recursive Supportive Vectors Machine
IUPAC – International Union of Pure and Applied Chemistry
EEG – Electroencephalogram
NT – Nucleotide
BP – Base pairs
STD – Standard deviation
TP – True positive
FP – False positive
TN – True negative
FN – False negative
IT – Iteration

LIST OF SUPPLEMENTARY DATA

1) CD containing 2 python scripts (one for feature extraction, other for classification), scripts outputs, all of the used patient data with their description, resulting statistics and pdf version of bachelor's thesis.