Machine Learning in HealthCare Informatics

Kishan Rameshchandra Rama

Report for the course of Introduction to the Research in Electrical and Computer Engineering

Supervisor(s):	Prof. Alexandra Sofia Martins de Carvalho
	Prof. Susana de Almeida Mendes Vinga Martins

Examination Committee

Supervisor: Prof. Susana de Almeida Mendes Vinga Martins
Member of the Committee: Prof. Helena Isabel Aidos Lopes

June 2017

Contents

Chapter 1

Introduction

1.1 Motivation

A new exciting era is emerging in HealthCare that will change completely our understanding of medicine and medical practice. The huge amount of data produced within HealthCare Informatics holds important knowledge to be gained. Machine Learning and data mining techniques are being used in order to extract this knowledge, helping to realize the goals of diagnosting, treating, helping and healing all patients in need of healthcare [1]. The ultimate goal is to improve the quality of healthcare offered to the patients in a personalized way.

Data collected come in many forms and from different sources. Usually this data is stored in Electronic Medical Records (EMRs) that contains large amounts of longitudinal data that tell us the clinical history of the patients. The objective of personalized medicine is to use this information to create individual treatment plans and even predict patient's response to targeted therapies [2]. An interesting way of performing this precision medicine is to use genomic data to create specific gene-based therapies. The mapping of the human genome has permitted a deeper insight on how genetic traits contribute to health and disease of individuals. However, patient's genetics data is only acquired for certain diagnoses and not in every medical center [3]. Moreover, the management and integration of this complex genomic data into EMR systems presents several challenges [4].

To move towards a healthcare system that is personalized to the individuals we need to fight what is called 'imprecision' medicine [5] i.e. often doctors prescribe the most common drugs and treatments to treat their patients for a given disease. Commonly these treatments do not work and this situation could be avoided if a different approach was followed. Hence, in this work we try to use longitudinal data present in EMRs of a huge database from the *Sociedade Portuguesa de Reumatologia* (Reuma.pt [10]) to find similarities between patients. In this way adjustments can be made to improve the efficiency of ongoing treatments.

A promising research area is sequence alignment. In the 1970s an important alignment method was developed by Needleman and Wunsch (NW) [6] to tackle the difficulty of finding similarities between biological sequences such as protein sequences. This approach is used to assess homology between molecular sequences, however, several adaptations have been made to diversify its use in other applications. The temporal information present in EMRs can be used in alignment methods to find, for example, similarities between patients based on their medical histories. In this work, we try to cluster patients from Reuma.pt database based on sequence similarity, by using the Temporal Needleman-Wunsch Algorithm (TNW) [7], a modified NW approach that uses time information between events.

1.2 Objectives

The main objective of this study is to implement and test a sequence alignment algorithm that takes into account the temporal information between events of a sequence. The algorithm is then tested on Reuma.pt dataset with the goal of finding groups of patients with similar temporal characteristic. Several parameters of the algorithm are tuned in order to obtain interpretable results. The main goal is threefold:

1. Provide a short literature review of HealthCare Informatics, introducing longitudinal data, also known as time series, emerging from EMRs.
2. Study the Temporal Needleman-Wunsch algorithm, starting with its time-independent version, the Needleman-Wunsch algorithm.
3. Implement and test the sequence alignment algorithm with the available data.

1.3 Document Outline

The remaining part of this report is organised as follows. Chapter 2 presents theoretical background on important topics for this work. Section 2.1 gives a bigger picture of the field of HealthCare Informatics. In Section 2.2 a brief description of the Reuma.pt dataset is done and finally in Section 2.3 the two sequence alignment methods (NW and TNW) are presented.

In Chapter 3 it is explained how the TNW algorithm was tested on the provided dataset and the results are shown and discussed.

Chapter 4 presents the conclusions and suggestions for future work with a detailed plan of the tasks to perform during the master thesis development.

Chapter 2

Background

2.1 HealthCare Informatics

Healthcare informatics is a multidisciplinary field that combines both healthcare data and machine learning with the objective of finding patterns of interest that can help doctors to detect and treat diseases earlier [8]. This big area of healthcare informatics consists of different tasks such as data acquisition, transmission, storage and retrieval of relevant information. One of the biggest challenges and research being developed in healthcare is in data collection and storage. Traditional manual and largely paper based medical records are shifting to a computerized and standardized manner of data collection which can reduce human errors and facilitate the productivity of medical centers in providing quality healthcare. EMR (electronic medical record) is a digital version of a patient's medical records that store a variety of clinical data ranging from structured data - basic demographics, drug administrations, laboratory data, medical history - to medical images such as brain images and clinical notes written as free text by doctors. These medical records, usually contain large amounts of longitudinal data, characterized by repeated measurements of the same variables during time, that can be in regular time intervals or different. The analysis of longitudinal data can distinguish changes over time within individuals [11].

A great advantage of EMR systems is that patient's information can be easily consulted and shared between hospitals reducing the costs of bureaucracy. The heterogeneity of the data available combined with machine learning techniques can help treat patients in a more personalized way. Moreover, large EMR databases in healthcare can be used to acquire knowledge at a population level to predict, for example, an epidemy. The existence of EMR systems in Portugal is still not a reality in many health centers [9]. However, in this study we used data from Reuma.pt [10] that is an example of a portuguese implementation of an EMR system.

Research developed in Healthcare Informatics is divided into many sub-fields such as the ones presented in Figure 2.1. Each of these sub-fields utilizes different types of data. Bioinformatics uses molecular level data, Image informatics considers tissue level data, Clinical Informatics takes patient level data and Public Health (PH) Informatics uses data gathered from the population [1]. Bioinformatics is a huge area that can even be separated from the Healthcare Informatics, however, research developed in this area is helping to answer questions at different levels of healthcare. The born of Bioinformatics is marked by the Needleman-Wunsch Algorithm that we are going to present in Section 2.3 that was used to study homology between molecular sequences. The huge amount of data that exists in this field, such as gene expression data, increases the importance of developing data mining techniques that are efficient. In the field of Image Informatics the available data is quite different from the other fields, being characterized by large amounts of data. The datasets gathered in this field can be used to train machine learning algorithms to find, for example, tumors in breast MRIs, or to learn how the brain works from brain images. The sub-field of Clinical Informatics works directly with the patients and tries to answer clinical questions, mostly making predictions that can help doctors in making decisions about their patients. Finally PH Informatics is concerned with extracting relevant medical insights through data mining techniques from population data. This data can be collected either from hospitals and government statistics or directly from the population in social media.

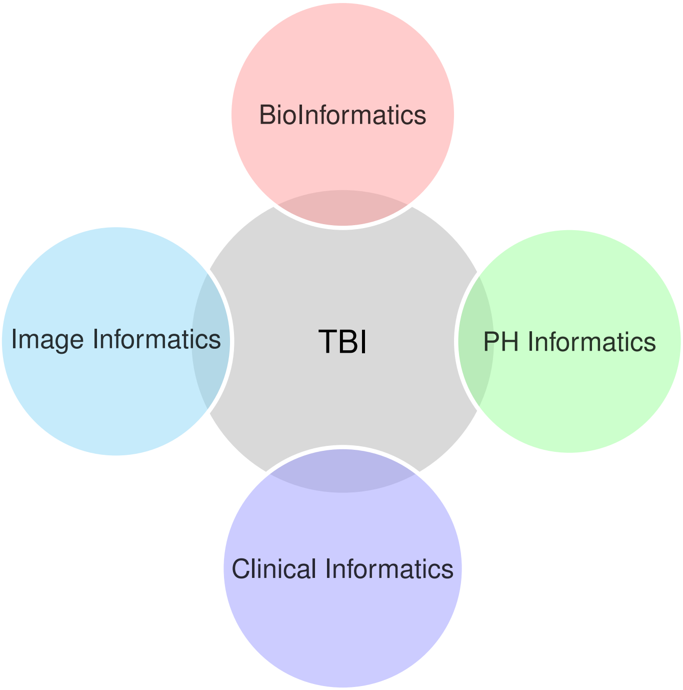


Figure 2.1: Sub-fields of Healthcare Informatics where Translation Bioinformatics (TBI) aggregates all the other fields.

Often the data from these sub-fields overlap, thus data from different levels is used. This overlap leads to Translation BioInformatics (TBI), the future of Healthcare Informatics, that tries to combine data from all the other sub-fields to answers various questions at a clinical level. In this work, we fall into the Clinical Informatics group because of the type of data being used but since we also have some laboratory data it is possible to say that we are working on the TBI field.

2.2 The Reuma.pt Database

Reuma.pt [10] is a Portuguese rheumatology database developed by the Portuguese Society of Rheumatology (SPR) that follows rheumatics patients nationwide, with the goal of monitoring disease progression and assuring treatment efficacy and safety. It includes mostly patients with rheumatoid arthritis (RA), ankylosing spondylitis (AS), psoriatic arhritis (PsA) and juvenile idiopathic arthritis (JIA) being treated with biological therapies or receiving synthetic disease modifying anti-rheumatic drugs (DMARDs). Data has been gathered since 2008 in 21 Rheumatology Departments assigned to the Portuguese National Health Service, 2 Military Hospitals (Lisboa and Porto), 1 public-private institution and 6 private centers. Patient data is being collected in a regular basis , more or less every three/six months.

The majority of patients registered in Reuma.pt have rheumatoid arthritis, an autoimune disease that causes pain and swelling in the wrist and small joints of the hand and feet [12]. Treatments for RA can mitigate these symptoms and provide a better lifestyle to the patients. Traditional therapies consist on using DMARDs but when patient fail to respond they switch to biological therapies. It is crucial to identify the most effective RA treatment early in order to avoid further complications and progression of the disease.

The dataset provided for this study comes from two centers and only contains patients diagnosed with rheumatoid arthritis. In total there is information about 426 patients and 9305 appointments. For each patient several variables are measured and calculated during time such as age, disease duration, gender, medical scores, patient questionnaires, active therapies and others. This data can be divided into static data that does not change over time, and longitudinal data. In Table 2.1 is summarized the type of data that is being collected.

Static data	Longitudinal data
-Demographic and anthropometric data	-Lab tests
-Life style habits	-Tender and swollen joint counts
-Work status	-Patient questionnaires
-Co-morbities	-Functional assessment scores
	-Previous and current therapies
	-Adverse events
	-Reasons for discontinuation

Table 2.1: Static and longitudinal data in Reuma.pt.

For RA patients the most important features are the RA disease activity score (DAS28), health assessment questionnaire (HAQ) (a detailed questionnaire on patient ability of performing daily routines), visual analogue scales(VAS) (patient and physician disease activity and patient reported pain), erythrocyte sedimentation rate (ESR) and C reactive protein(CRP).

In more detail, DAS28 is a very common disease activity score measured in RA developed by Dutch rheumatologists that takes into account 28 joints [13]. To compute this score a specialist has to count the number of swollen and tender joints (out of 28), take blood to measure the erythrocyte sedimentation rate (ESR) or C reactive protein (CRP) and obtain a global assessment of health done by the patient. Therefore the DAS28 score combines all the other features presented before into a formula given by:

$$DAS28 = 0.56.\sqrt{tender28} + 0.28.\sqrt{swollen28} + 0.70.\ln(ESR) + 0.014.VAS, \tag{2.1}$$

where $tender28$ and $swollen28$ is the number of tender and swollen joints, respectively. The DAS28 can also be calculate with the CRP instead of ESR. This formula is given by:

$$DAS28 = 0.56.\sqrt{tender28} + 0.28.\sqrt{swollen28} + 0.36.\ln(CRP + 1) + 0.014.VAS + 0.96 \tag{2.2}$$

Note that there are several variations on how to calculate these scores. For instance instead of using four variables there are formulas that use three variables. The result of this computation is a number between 0 and 10 and provides useful information regarding the current activity of the disease on the patient. A DAS28 above 5.1 indicates a high activity of RA whereas a value lower than 3.2 indicates low activity. It is possible to say that a patient is in remission if a DAS28 value below 2.6 is computed.

Blood tests to measure ESR and CRP help doctors to diagnose and follow the progression of rheumatoid arthritis. The ESR and CRP measures the degree of inflammation in the joints. The difference between the two is that CRP is a more sensitive measure of inflammation than ESR. Higher inflammation of the joints typically means higher values of these tests hence one of the aims of treatment is to reduce the ESR and CRP levels [14]. Finally HAQ is more detailed questionnaire for assessment of RA where patients report the amount of difficulty to perform daily routine activities such as dressing and grooming, arising, eating, walking and many others whereas VAS is a method that tries to measure the pain that a patient is feeling. VAS is usually a line, 100 mm in lenght, with the words NO PAIN on the left and VERY SEVERE PAIN on the right where the patient marks their perception of pain in the current state.

2.3 Sequence alignment methods

For a better comprehension of the Temporal Needleman-Wunsch (TNW) algorithm it is a good approach to understand first the Needleman-Wunsch (NW) algorithm. In the next section the NW algorithm will be presented with an example to show the various steps of the method.

2.3.1 The Needleman-Wunsch Algortihm

The NW algorithm is a global sequence algorithm method commonly used in Bioinformatics to assess similarity between molecular sequences [6]. The alignments that we are looking for are not exact matchings between two sequences but approximate matchings that maximizes letter-to-letter (consider a string sequence) matches. For example consider aligning the sequence **ATGGCGT** with **ATGAGT**. Immediately it is possible to check that a exact matching between these two strings does not exist. However if we allow the existence of gaps, approximate matchings can be obtained such as:

$$\begin{array}{ccccccc} \text{A} & \text{T} & \text{G} & \text{G} & \text{C} & \text{G} & \text{T} \\ \text{A} & \text{T} & \text{G} & \text{-} & \text{A} & \text{G} & \text{T} \end{array} \qquad \begin{array}{ccccccc} \text{A} & \text{T} & \text{G} & \text{G} & \text{C} & \text{G} & \text{T} \\ \text{A} & \text{-} & \text{T} & \text{G} & \text{A} & \text{G} & \text{T} \end{array}$$

This type of matching is very useful in molecular biology to study the structural, functional and evolutionary relationship between the molecular sequences.

The NW method is mathematically proven to find the optimal alignment between two sequences. It is based on dynamic programming where the basic idea is to solve the problem by dividing into smaller problems; solve the smaller problems optimally and then use the sub-solutions to construct an optimal solution for the original problem.

For a pair of sequences, $X = x_1, ..., x_m$ and $Y = y_1, ..., y_n$ where x_i and y_j for $i \in [1, m]$ and $j \in [1, n]$ are the elements of sequences X and Y with size m and n, respectively; the algorithm uses two matrices score H and traceback T that considers all possible pairs of letters from the two sequences to build the alignment. The NW algorithm consists of three steps:

1. Initialisation of the score matrix H:

$$H_{r0} = -rg; H_{0c} = -cg \quad \forall r \in [0, m], c \in [0, n] \tag{2.3}$$

2. Calculation of the scores and filling the traceback matrix. The rest of the score matrix is defined as:

$$H_{i,j} = \max \begin{cases} H_{i-1,j-1} + S(x_i, y_j) \\ H_{i-1,j} - g \\ H_{i,j-1} - g \end{cases}, \quad \forall i \in [1, m], j \in [1, n] \tag{2.4}$$

3. Deducing the alignment from the traceback matrix T.

In the equations 2.3 and 2.4, $S(x_i, y_j)$ is a user pre-defined scoring schema that measures the similarity between sequence elements x_i and y_j while g is a constant gap penalty chosen by the user that allows us to penalize the alignment score whenever a gap is inserted.

Example:

To illustrate the steps of the algorithm let us consider two sequences given by:

$$X = SEND \qquad Y = AND$$

First let us define our scoring schema $S(x, y)$. The simple basic scoring schema can be assumed as, if two elements x_i and y_j of our sequences are equal then the matching score is 1 ($S(x_i, y_j) = 1$) otherwise we have a mismatch and the score is -1 ($S(x_i, y_j) = -1$). However in this fabricated example one combination of elements of the sequences that mismatch have a score -2. In Figure 2.2 a scoring system is defined for all combinations of the letters in both sequences. The gap penalty is assumed as 2.

Now we follow the three steps described above:

1. **Initialisation step**: The score matrix H is initialized with $m + 1$ rows and $n + 1$ columns where m and n are the lengths of sequences X and Y respectively. The extra row and column is given for

	A	D	E	N	S
A	1	-1	-2	-1	-1
D	-1	1	-1	-1	-1
E	-2	-1	1	-1	-1
N	-1	-1	-1	1	-1
S	-1	-1	-1	-1	1

Figure 2.2: Pre-defined scoring schema S.

alignments with gaps. With equation 2.3 we fill the first row and first column (see Figure 2.3). The traceback matrix is initialised according to Figure 2.4.

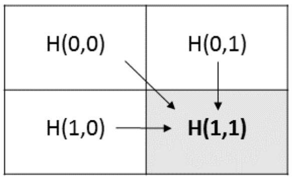
		S	E	N	D
	0	-2	-4	-6	-8
A	-2				
N	-4				
D	-6				

Figure 2.3: Initialisation of score matrix H.

		S	E	N	D
	done	left	left	left	left
A	up				
N	up				
D	up				

Figure 2.4: Initialisation of traceback matrix T.

2. **Matrix H fill step**: The remaining matrix H positions are filled by row or column starting at $H_{1,1}$ by using equation 2.4. At each position i, j the score is computed taking into account the values at three other positions at the immediately adjacent northwest diagonal, up and left cells. In Figure 2.5 (left side) is presented a pictorial representation of this computation for the first position $(1, 1)$.



$$v_{diag} = H_{0,0} + S(A, S) = 0 + (-1) = -1$$

$$v_{up} = H_{0,1} - g = -2 - 2 = -4$$

$$v_{left} = H_{1,0} - g = -2 - 2 = -4$$

Figure 2.5: Pictorial representation on computation of $H_{1,1}$.

The value of $H_{1,1}$ is the maximum of the three values v_{diag}, v_{up} and v_{left} presented in Figure 2.5 (right side). Hence $H_{1,1} = v_{diag} = -1$ and the corresponding traceback matrix cell is filled with 'diag' (Figure 2.6). If instead of v_{diag} the maximum value was v_{up} or v_{left} then the corresponding traceback matrix positions would be 'up' and 'left' respectively.

		S	E	N	D
	0	-2	-4	-6	-8
A	-2	**-1**			
N	-4				
D	-6				

		S	E	N	D
	done	left	left	left	left
A	up	**diag**			
N	up				
D	up				

Figure 2.6: Filling $H_{1,1}$ (left) and $T_{1,1}$ (right).

The same procedure is followed to compute all the other cells. The final score S and traceback T matrices are presented in Figure 2.7. Note that the final score of the alignment is given by the last cell to be filled, i.e in this example the final score is $H_{3,4} = -1$.

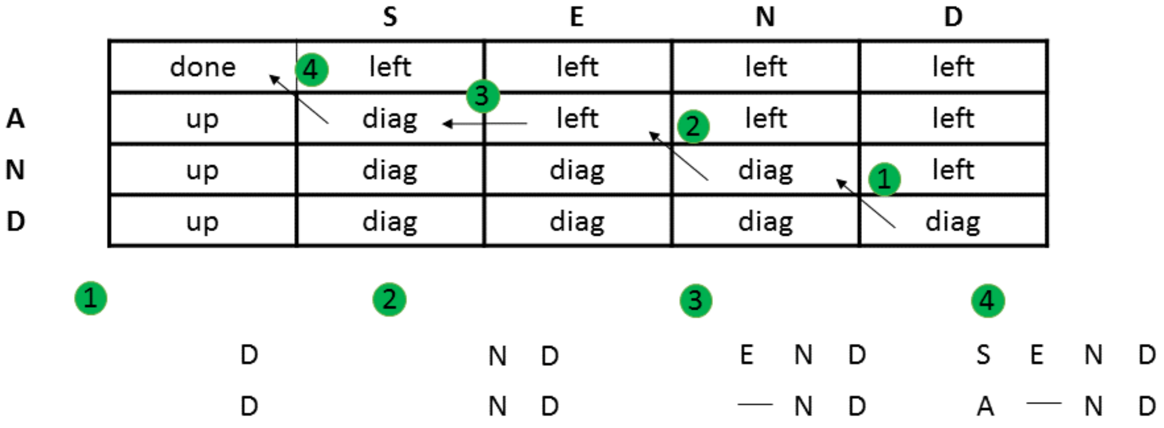
3. **Traceback step**: The alignment between the two sequences can now be deduced from the traceback matrix. The traceback procedure begins with the last position to be filled (bottom right cell)

		S	E	N	D
	0	-2	-4	-6	-8
A	-2	1	-3	-5	-7
N	-4	-3	-2	-2	-4
D	-6	-5	-4	-3	-1

		S	E	N	D
	done	left	left	left	left
A	up	diag	left	left	left
N	up	diag	diag	diag	left
D	up	diag	diag	diag	diag

Figure 2.7: Final score and traceback matrices in the left and right sides, respectively.

and ends at the first matrix position (top left cell - "done"). At each position we move according to the value stored in it, i.e the three possible moves are diagonal (northwest direction), up and left. In Figure 2.8 is shown the traceback that was performed on the traceback matrix with the alignments being done at each move numbered from 1 to 4.



		S	E	N	D
	done	left	left	left	left
A	up	diag	left	left	left
N	up	diag	diag	diag	left
D	up	diag	diag	diag	diag

Figure 2.8: Traceback procedure using traceback matrix.

In the first move we look at the value in the bottom right cell that is "diag", which means that the pair "DD" of letters corresponding to the two sequences are aligned. Then we move diagonally from the position (3,4) to the position (2,3). The latter cell also stores the value "diag" hence the same procedure applies here and the the pair "NN" is aligned. In the third step the cell (1,2) has the value "left". This means that a gap is introduced in the left sequence, i.e the letter "E" from sequence "SEND" is aligned with a gap and the letter "A" from sequence "AND" will look for other alignments while we traceback. Finnaly in step 4 we encounter again the value "diag" at cell (1,1) and letter "A" is aligned with "S" concluding the alignment procedure.

In summary, the values stored in the traceback matrix indicate the alignment:

- diag - the letters from two sequences are aligned
- left - a gap is introduced in the left sequence
- up - a gap is introduce in the top sequence.

2.3.2 The Temporal Needleman-Wunsch Algorithm

The TNW algorithm is a modified version of the NW method that incorporates the transition times between elements of a sequence. Temporal alignment methods are vastly used for time-series alignment in areas such as speech recognition where we try to match signals that have different speed have the same meaning. However these methods do not impose a penalty for missing events neither use the relative time between them.

So first, we need to create these temporal sequences as a pre-processing step. Given two consecutive events A and B, and the transition time t between them, then there are two possible encodings:

- Suffix - encoded (SE): A.t , B.0
- Prefix- ecoded (PE): 0.A , t.B

Memoryless version

The goal of this method is to assess similarity between two sequences by considering the appropriate transition times of events in the two sequences. Consider two sequences: $X = x_1.t_{x_1}, x_2.t_{x_2}, ..., x_m.0$ and $Y = y_1.t_{y_1}, y_2.t_{y_2}, ..., y_n.0$, where t_{x_i} and t_{y_j} are the transition times for elements x_i and y_j, respectively, for $i \in [1, m-1]$ and $j \in [1, n-1]$. The simplest method is to use the transition times of the sequence elements that are being compared in computing the score matrix H. In this memoryless version the H matrix is computed using equations 2.3 and 2.5 :

$$H_{i,j} = \max \begin{cases} H_{i-1,j-1} + S(x_i, y_j) - f(t_{x_i}, t_{y_j}) \\ H_{i-1,j} + g & , \quad \forall i \in [1, m], j \in [1, n] \\ H_{i,j-1} + g \end{cases} \tag{2.5}$$

The only difference between equations 2.4 and 2.5 is the introduction of a term $f(t_{x_i}, t_{y_j})$ that is a user defined temporal penalty function. The aim of this function is to reduce the similarity $S(x_i, y_j)$ by an amount that depends on t_{x_i} and t_{y_j}. In [7] the following temporal penalty function was used:

$$f(t_{x_i}, t_{y_j}) = T_p \frac{|t_{x_i} - t_{y_j}|}{\max(t_{x_i}, t_{y_j})}, \tag{2.6}$$

where T_p is some constant factor that will impose the maximum penalty on $S(x_i, x_j)$. This penalty function computes a percentage discrepancy between times t_i and t_j of two events x_i and x_j that are compared. If the events being compared have the same transition times associated to both of them that means that they are similar and the penalty function will be zero. But if the transition times are different then a penalty that depends on the times will be imposed.

The memoryless version presents some limitations such as ignoring the transition times for events that align to gaps in score computations , and the choice of encoding affects the overall score and alignment of sequences. These limitations can be better understood with the example in Figure 2.9. The top row of the figure shows the sequences (suffix-encoded): $X = A.t_1, B.t_2, C.t_3, D.0$ and $Y = A.t_4, D.0$

where $t_4 = t_1 + t_2 + t_3$. In the middle row the alignment and score obtained with the memoryless method is shown for the SE sequences described above where the temporal penalty function applied for the matched A-pair is $-f(t_1, t_4)$ and zero for the matched D-pair. However if PE sequences were used then the temporal penalty function would have been zero for the matched A-pair and $-f(t_3, t_4)$ for the matched D-pair. Hence the choice of encoding affects the overall score of the alignment and it might change the result of the alignment.

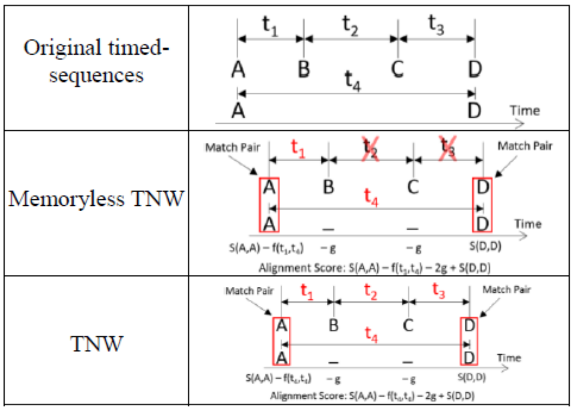


Figure 2.9: Alignment of two sequences (top row) and score computation based on the memoryless version (middle row) and ideal TNW algorithm (bottom row) [7].

The objective with the TNW algorithm is to use the total transition time between the match-pairs instead of just considering the times associated with each sequence elements. Using the total transition time between match-pairs would give a temporal penalty function $-f(t_1 + t_2 + t_3) = -f(t_4, t_4) = 0$ as shown in the third row of Figure 2.9. The full TNW approach changes the temporal penalty function from using the transition times of the match-pair alone, -$f(t_{x_i}, t_{y_j})$ to a penalty function $-f(t_{s1}, t_{s2})$ where t_{s1} and t_{s2} are total transition times between matched event pairs associated with the first and second sequence, respectively.

Full TNW version

The Full TNW algorithm can compute the temporal penalties with the total transition time between consecutive match pairs discussed before, by using PE sequences and two auxiliary matrices TR and TC that *accumulate* these transition times for events that align with gaps associated with the first X and second sequence Y, respectively. The score matrix H is now calculated using:

$$H_{i,j} = \max \begin{Bmatrix} H_{i-1,j-1} + S(x_i, y_j) - f(t_{x_i} + TR_{i-1,j-1}, t_{y_j} + TC_{i-1,j-1}) \\ \\ H_{i-1,j} + g \\ \\ H_{i,j-1} + g \end{Bmatrix}, \quad \forall i \in [1, m], j \in [1, n] \tag{2.7}$$

where TR and TC are given by:

$$TR_{i,j} = \begin{cases} 0, & \text{if } T(i,j) = \text{'diag'} \\ TR_{i,j-1} & \text{if } T(i,j) = \text{'left'} \\ TR_{i-1,j} + t_{xi} & \text{if } T(i,j) = \text{'up'} \end{cases} \tag{2.8}$$

$$TC_{i,j} = \begin{cases} 0, & \text{if } T(i,j) = \text{'diag'} \\ TC_{i,j-1} + T_{yj} & \text{if } T(i,j) = \text{'left'} \\ Tc_{i-1,j} & \text{if } T(i,j) = \text{'up'} \end{cases} \tag{2.9}$$

As an example, the results obtained by applying the TNW algorithm in the example of Figure 2.9 are shown in Figure 2.10 and Figure 2.11 where the resulting H and T matrices ,and the additional matrices TR and TC are presented, respectively. In this example the pre-defined scoring schema S consisted on scoring 1 matching events and -1.1 for mismatches. The gap penalty was defined as 0.5. These scores were chosen in order reflect preference for aligning events with gaps instead of aligning mismatch events.

First let us understand the matrices TR and TC. As already explained these matrices store values that represent the accumulated transition times for events that are aligned with gaps in the first sequence X and second sequence Y, respectively. This notion is clearly visible in Figure 2.11 where in the left side (TR matrix) it is possible to see an accumulation of transition times (t_1, t_2, t_3) of sequence X as the row i increases and the same happens in the right side (TC matrix) but now column wise with only t_4 being accumulated. For example the cell position $T_{2,0}$ indicates 'up' which means that there is no alignment; in this case $TR_{2,0} = TR_{1,0} + t_{x_2} = 0 + t_1 = t1$ and $TC_{2,0} = TC_{1,0} = 0$. Because event B aligned to a gap we stored this transition time to use in a future calculation when some events align. In the cases where we found a match, we add the diagonal TR and TC values to the transition times t_{x_i} and t_{y_j} of the events being compared. In the matrices positions (4,2) a match is discovered hence the temporal penalty is computed using $TR_{3,1} + t_{x_4} = t_1 + t_2 + t_3$ and $TC_{3,1} + t_{y_1} = 0 + t_4 = t_4$ which gives the function $-f(t_1 + t_2 + t_3, t_4)$ that is zero. Since we found a match pair the values for the corresponding cell positions of TR and TC are reset to zero. In this manner subsequent events that are aligned will use time information from the last pair of events that were matched.

		0.A	**t4.D**
	0	-0.5	-1
0.A	-0.5	1	0.5
t1.B	-1	0.5	0
t2.C	-1.5	0	-0.5
t3.D	-2	-0.5	1

		0.A	**t4.D**
	done	left	left
0.A	up	diag	left
t1.B	up	up	up
t2.C	up	up	left
t3.D	up	up	diag

Figure 2.10: H (left) and T (right) matrices computed for the example of Figure 2.9.

In conclusion, the Full TNW algorithm is a temporal sequence alignment method that can be used for pattern discovery and matching sequences where the timing information between events is relevant. The authors that proposed this method tested it for autonomous chemotherapy-protocol recognition

		0.A	**t4.D**
	0	0	0
0.A	0	0	0
t1.B	t1	t1	t1
t2.C	t1+t2	t1+t2	t1+t2
t3.D	t1+t2+t3	t1+t2+t3	0

		0.A	**t4.D**
	0	0	t4
0.A	0	0	t4
t1.B	0	0	t4
t2.C	0	0	t4
t3.D	0	0	0

Figure 2.11: TR (left) and TC (right) matrices computed for the example of Figure 2.9.

where patients treatment histories are aligned with standard recommended protocols to understand which protocol a patient is following. It is noted in the paper that the algorithm has to be tested on other healthcare datasets.

Chapter 3

Results

In this chapter two experiments are set to study the application of the TNW algorithm with the reuma.pt data and the results are presented. The first experiment consists on performing alignment with sequences created with the patient treatment history whereas the second experiment is related to studying the progression of an important feature discussed before, the DAS28, between the patients. With these experiments we try to find similarities between patients based on their treatment or DAS28 history. The main steps of the proposed approach are shown in Figure 3.1. The initial step consists on taking the raw data and pre-process it to obtain temporal sequences. Then on the second step sequence alignment is performed between all pairs of patients and a distance matrix is obtained. Finally Hierarchical Clustering is performed with the distance matrix obtained previously. The last step Clinical Interpretation was not studied yet.

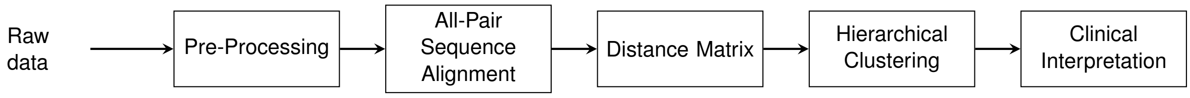


Figure 3.1: Diagram of the proposed approach.

Note that the proposed approach is followed independently for the two experiments. To avoid repetition we just present a detailed explanation of the main steps for the first experiment and point the main differences whenever necessary when explaining the second experiment. The following sections will discuss the proposed approach.

In order to perform these experiments, code was written in python that implemented all the steps. These scripts can be found at `https://github.com/KishanRama/TNW-Reuma.pt`.

3.1 Data Pre-Processing

As explained in Section 2.2 the available data consists of observations of several variables during time. In the first experiment we will look only at the variables *id_doente*,*n_bio_corrente* and *n_dias_duracao* where *id_doente* is the unique reuma.pt id of the patient, *n_bio_corrente* is the number of the current biological medicine and *n_dias_duracao* is the duration of the medication. A fraction of this data is presented in

Table 3.1 where each row indicates an observation, i.e an appointment of a patient. Is it possible to verify that there are several rows that repeat which mean that in different appointments the therapy did not change. Also there are some columns without information that we eliminate in this pre-processing step. The main pre-processing steps are:

1. Eliminate repeated [*id_doente n_bio_corrente*] rows.

2. Eliminate rows with NA values. These rows do not contain necessary information to create temporal sequences.

3. Convert numerical values of *n_bio_corrente* into strings in alphabetical order, i.e *n_bio_corrente* = 1 → A, *n_bio_corrente* = 2 → B and so on. This is done in order to make a distinguish between the transition times that are numerical (*n_dias_duracao*) and the events (*n_bio_corrente*).

4. Create the prefix-encoded temporal sequences

id_doente	**n_bio_corrente**	**n_dias_duracao**
33496	2	156
33496	2	156
33496	2	156
33496	2	156
⋮	⋮	⋮
33496	4	1696
33496	4	1696
33499	2	148
33499	2	148
33499	2	148
33499	2	148
33499	2	NA
33499	2	NA
33499	2	NA
33499	2	NA
33502	0	NA
33502	1	120
33502	1	120
⋮	⋮	⋮

Table 3.1: Part of data used to create temporal sequences.

For illustration, the temporal sequences obtained with the data presented in Table 3.1 are presented in Table 3.2. The event Z marks the end of a sequence hence the drug that the patient is taking is given by the previous event in the sequence. Once a patient begins a biological therapy he never stops taking these medicines. Note that the time information associated with a biological at each row is used as the prefix of the next biological that appears in the data when building the temporal sequences.

In this data pre-processing step temporal sequences were created that tell us the drug administration history of each patient. Perfoming alignment of these sequences help to understand which type of therapies are more predominant between the patients or which ones are more effective in dealing with the disease in hand.

id_doente	PE Temporal Sequence
33496	0.B,156...D,1696.Z
33499	0.B,148.Z
33502	0.A,120...Z
⋮	⋮

Table 3.2: PE Temporal Sequences for each patient.

3.2 Sequence Alignment

After creating the temporal sequences in the pre-processing step, it is possible to perform alignment between all patient pairs using the TNW algorithm. The information of all the alignments can be summarized into a similarity matrix like the one represented in Figure 3.2. In this matrix scores at a position (i, j) represents the score of the alignment of a id_doente_i with a id_doente_j. The diagonal has zero values of scores because alignment of a patient sequence with himself gives a perfect alignment but ideally since this is a similarity matrix these values should be very high. However, the diagonal values are not going to be used in the clustering step and note that the matrix is symmetri, hence, we only need to compute the upper triangle part of the matrix.

	id_doente_1	id_doente_2	···	id_doente_i	···	id_doente_N
id_doente_1	0	score12	···	score1i	···	score1N
id_doente_2	score21	0	···	score2i	···	score2N
⋮	⋮	⋮	⋱	⋮	⋮	⋮
id_doente_i	scorei1	scorei2	···	0	···	scoreiN
⋮	⋮	⋮	⋮	⋮	⋱	⋮
id_doente_N	scoreN1	scoreN2	···	scoreNi	···	0

Figure 3.2: Similarity matrix.

Tuning the parameters

One of the main difficulties when using the TNW algorithm is that the best alignments are obtained given some user-defined parameters. Choosing appropriate parameters of the TNW depends on the application. In this work, the initial experiment started by using the same parameters as used in the TNW algorithm paper [7]. A simple scoring schema was used where pairs that match get a score of 1 and mismatches get a score of -1.1, and a gap penalty of 0.5. The lower value for mismatches reflects the preference for aligning events with gaps instead of aligning mismatch events. Note that in this work the scoring schema could be modified because the events of our sequences are medicines and a scoring schema could be defined that reflects similarities between them.

The user-defined temporal penalty function that was used is the same as presented in Equation 2.6 which requires a user defined value for T_p. A value of T_p = 0.25 does not generally change the alignment of sequences when computed with and without the temporal distances. The main objective is to obtain scores that are different due to time information.

During the development of thesis different variations of these parameters have to be tested. However, with the application in hand there is limited amount of variations that we can perform on the parameters and that is coherent with the alignments wanted in this work. For instance, by keeping the same scoring schema, if we increase too much the value of T_p and keep a constant small value for the gap penalty then we will get no alignment at all between the sequences. Another example is, if we set the constant gap very high, then events that are not the same are going to be aligned. As already stated, different scoring schemas, that incorporates clinical knowledge, also have to be tested.

This is to show that there are variations that do not make sense and hence during the thesis work a more detailed investigation has to be made on how to perform tests with a limited amount of tuning on the parameters.

3.3 Hierarchical Clustering of patients

The sequence alignment step provides a similarity matrix that can be used in a clustering algorithm. In this work we used hierarchical clustering to obtain patient clusters. Hierarchical clustering algorithms are divided into two categories:

- **Agglomerative** (bottom-up approach) - Each object (in this case patients) starts from the bottom as a single cluster and go up through merging the two closest objects. This method is iterated until all objects fall into a single group.
- **Divisive** (top-down approach) - This approach is the opposite of the agglomerative one, here we start by assuming one single cluster containing all objects and then we split the group into two recursively until each group consists of a single object.

The results of hierarchical clustering are usually presented in a dendrogram, allowing to visualize a tree showing the order and distances of merges during the hierarchical clustering. The agglomerative clustering was the method used in this study. The main steps of this approach are:

1. Start with a collection C of n single clusters: each cluster c_i contains one object x_i.
2. Repeat until one cluster is left:
   (a) use a distance matrix find a pair of clusters that are closest to each other: $\min_{i,j} D(c_i, c_j)$
   (b) merge the cluster c_i with c_j to form a new cluster c_{i+j}
   (c) remove c_i and c_j from the collection C and add $c_{i,j}$ to C

This approach requires a distance metric over clusters that measure the cluster distance $D(c_i, c_j)$. Several metrics can be used; in our experiments four metrics, presented in Figure 3.3, were used. Single link computes the distance between closest elements in clusters while complete link considers the farthest elements in the clusters to be merged. Average link takes the average of all pairwise distances, being less affected by outliers, and the centroids metric calculates the distance between centroids of two clusters.

Before using the agglomerative clustering algorithm we need to convert the similarity matrix obtained in the previous step into a distance matrix. The algorithm requires a distance matrix that contains

Single link:

$$D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$$

Complete link:

$$D(c_1, c_2) = \max_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$$

Average link:

$$D(c_1, c_2) = \frac{1}{|c_1|} \frac{1}{|c_2|} \sum_{x_1 \in c_2} \sum_{x_2 \in c_2} D(x_1, x_2)$$

Centroids:

$$D(c_1, c_2) = D\left(\left(\frac{1}{|c_1|} \sum_{x \in c_1} x\right), \left(\frac{1}{|c_2|} \sum_{x \in c_2} x\right)\right)$$

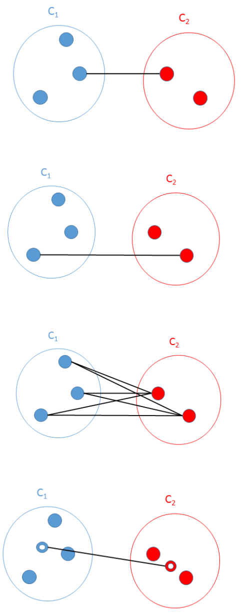


Figure 3.3: Cluster distance measures.

values greater than or equal to zero. The similarity matrix obtained with the alignments could contain negative values, hence, we need to perform a conversion. First of all, when we merge clusters we want to merge the ones that are more similar. Since the clustering algorithm merge the closest objects we need to convert our similarity matrix into a dissimilarity matrix in order to the highest scores that indicate the most similar become the lowest scores. Then a shift is made to make all the values positive. In Figure 3.4 a visualization of this conversion is shown. The transformation consists on negating all the similarity scores in order to obtain dissimilarity scores and then adding the absolute value of the minimum of dissimilarity scores to obtain the distance scores used in the clustering.

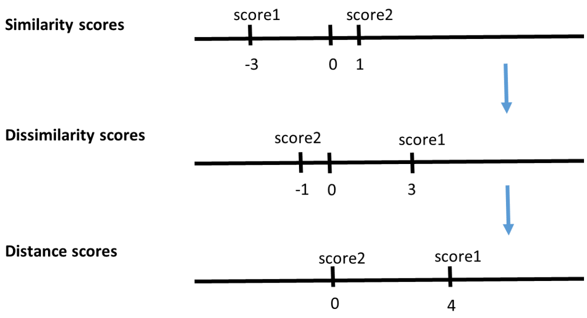


Figure 3.4: Conversion of the similarity scores into distance scores to use in agglomerative clustering.

After the conversion is made and the distance matrix values are computed then the agglomerative clustering is easily performed with the *python* function linkage [15]. Let us first present the dendrogram of a hierarchical clustering that used the complete linkage method (Figure 3.5). In the horizontal axis we can see the indices of the patients and in the vertical axis the distances between clusters. Starting from a patient/cluster at the bottom we can see a vertical line up to a horizontal line. The height, where horizontal lines appear, corresponds to the distance between this patient/cluster and the patient/cluster on the other end of the horizontal line. In summary, horizontal lines mean cluster merges and vertical lines tell us about which patients or clusters were part of the merge forming a new cluster.

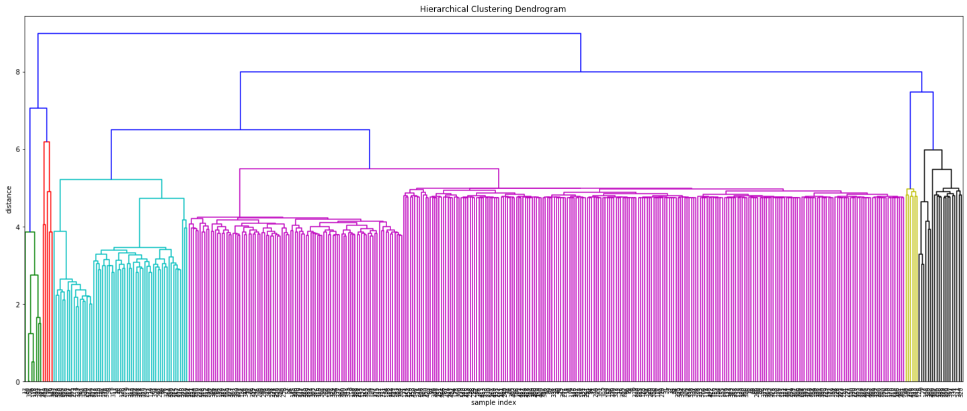


Figure 3.5: Dendrogram of complete link hierarchical clustering with the TNW parameters fixed in Section 3.2 ($g = 0.5$).

A visual analysis can be done on this dendrogram to find clusters. Usually, we are looking for huge distance jumps/gaps (in the vertical axis) that indicate that something merged at this level should not have been merged. This is very interesting because it tell us that the groups being merged probably do not belong to the same cluster. In Figure 3.5 there are some distance jumps but they are not that big compared to the other merges. However, if we draw a cut-off line at around 6,5 and count the intersection of the vertical lines with this cut-off line we can found 6 clusters. There is no definitive answer on how to draw this cut-off line and in this work a visual analysis is made of the dendrograms. In development of thesis a more detailed study on this aspect has to be made.

One of the main difficulties with this work is in choosing a correct distance metric. This choice can be application dependent and in our case we did not have any insight that could help us in choosing a linkage method. A good approach is to check the Cophenetic Correlation Coefficient of our clustering. This coefficient is a measure of how faithful the hierarchical clustering preserved the original pairwise distance between the elements and it is defined by:

$$c = \frac{\sum_{i<j}(D(x_i,x_j)-d)(Z(x_i,x_j)-z)}{\sqrt{\sum_{i<j}(D(x_i,x_j)-d)^2 \sum_{i<j}(Z(x_i,x_j)-z)^2}}, \tag{3.1}$$

where $D(x_i, x_j)$ represents the distances between the original elements, $Z(x_i, x_j)$ the cophenetic distances between two elements x_i and x_j that is given by the height in the dendrogram at which they were merged. The value of d and z represents the average of these original pairwise distances and the cophenetic distances, respectively. The closer the value of c is to 1, the better the clustering preserves the original distances.

3.4 Experiment 1 - Biological therapy sequences

As explained previously, the first experiment consists on using treatment sequences of the patients to perform alignment. The main steps were already explained in the previous chapters for this case. Here, we present the tests that were performed and discuss the results.

The tests consisted on performing the hierarchical clustering with the 4 metrics already discussed and calculating the cophenetic distances to analyse which metric preserves better the distances between the patients. Also the gap penalty of the TNW algorithm that was fixed with 0.5 in Section 3.2 was changed to 0. Setting the gap penalty to 0 and not changing the scoring schema allows us to obtain non-negative alignment scores. In this way the conversion presented in Figure 3.4 is not need. The only conversion we made for this case is to invert the similarity scores to obtain dissimilarity scores. These two variations of the gap values allows us to test two ways of transforming a similarity matrix into a distance matrix. In Table 3.3 the results of these tests are presented. Note that in the last column with the gap penalty set to 0 we normalized the scores by the number of matches obtained with an alignment. The reason for this is that the scores of the alignments increases with the number of matches when $g = 0$. Hence, if we align two huge sequence where the number of matches is bigger than the number of matches obtained with a perfect alignment between two smaller sequences, we are mislead to think that the huge sequences are more similar then the smaller ones. A normalization ensures a correct comparison where a value of 1 indicates perfect alignment and 0 no alignment at all. Since all the sequences end with the event Z as explained in Section 3.1 at least one match will be encountered.

Linkage Method	**Cophenetic Correlation Coefficient**		
	g = 0.5	g = 0	g = 0 Normalized
Single	0.648	0.411	0.374
Complete	0.813	0.830	0.357
Average	0.869	0.867	0.575
Centroids	0.632	0.817	0.729

Table 3.3: Cophenetic Correlation Coefficient values for experiment 1 with 4 linkage methods and varying the gap penalty of TNW algorithm.

The best cophenetic correlation coefficient values were obtained using the complete and average link with gap penalty set to 0 and 0.5. In general it was verified that with normalized scores the values become lower.

Let us analyse visually the dendrograms obtained with the best cophenetic values. In Figure 3.5 we already presented the dendrogram for the complete linkage method with $g = 0.5$. Despite absence of huge jumps we can infer the existence of 6 clusters of patients. In Figure 3.6, again with the same value of gap, the dendrogram when using average link is presented. By analysing the dendrogram it is hard to define a cut-off line: a cut-off line at around 5.5 would yield 6 clusters while at a height of 6 we would obtain 3 clusters. Again there are no huge jumps detected and hence it is hard to assess if it is possible to divide our data into different clusters.

The results obtained with $g = 0$ are quite interesting since bigger jumps and more defined clusters

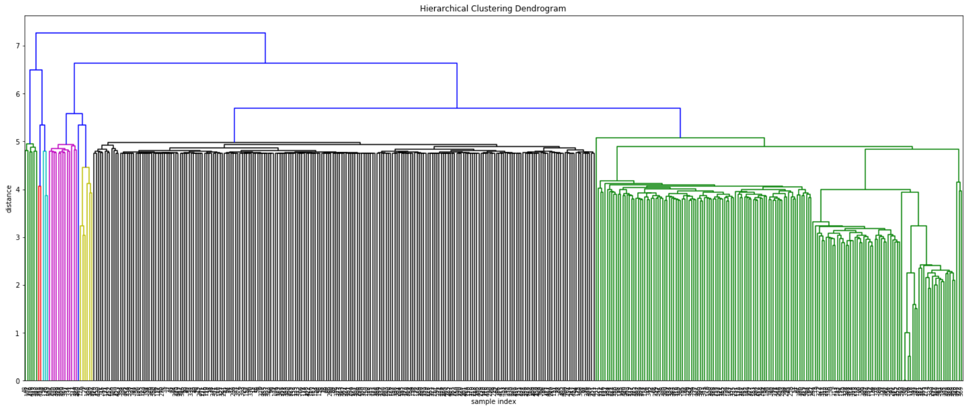


Figure 3.6: Dendrogram of average link hierarchical clustering with the TNW parameters fixed in Section 3.2 ($g = 0.5$).

are observed. However, the interpretation of this results is somehow more difficult. In Figures 3.7 and 3.8 the dendrograms for the complete link and average link are presented, respectively. These are the ones that presented better cophenetic correlations values. Regarding the first dendrogram it is possible to see clearly four clusters where three of them are small clusters and the remaining is a big cluster containing most of the patients. To understand better what type of patients were grouped, a detailed analysis of two of the smallest clusters was done. The smallest cluster, with two patients, revealed that the corresponding patient sequences were exactly the same but with different time intervals between the events. The other cluster, with eight elements, contained equal patients sequences but two of them were different . These two elements, from the eight, that were different had sequences where just part of it matched with the sequences of the other elements of the cluster.

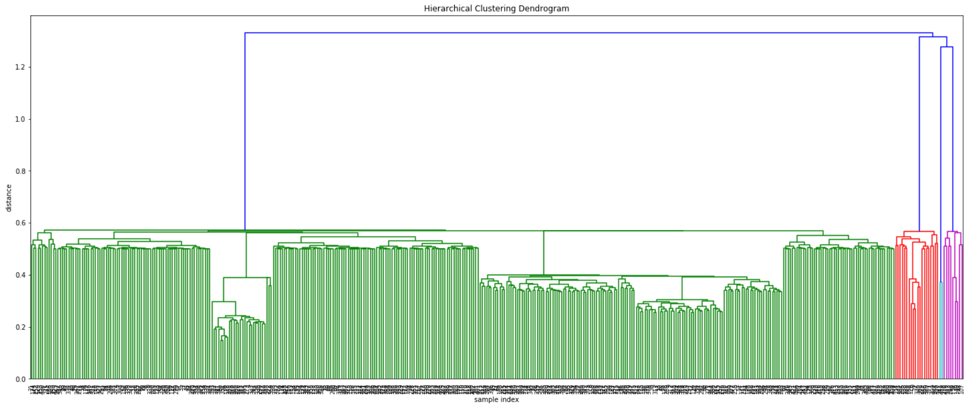


Figure 3.7: Dendrogram of complete link hierarchical clustering with $g = 0$.

The dendrogram with the average link method also presents huge gaps and the same observations can be made as in the previous case. There are four well defined clusters where three of them are considerable smaller compared to the bigger one. Again the smallest clusters were analysed in detail, to observe the patients sequences in these groups. The clusters contained sequences very similar to each other (same events). However, several sequences in these clusters were not exactly equal to the others but where just some part of it matched with the other sequences. In general, it is possible to conclude that the smallest clusters contained patients similar in terms of events (biological medicines taken by the patients).

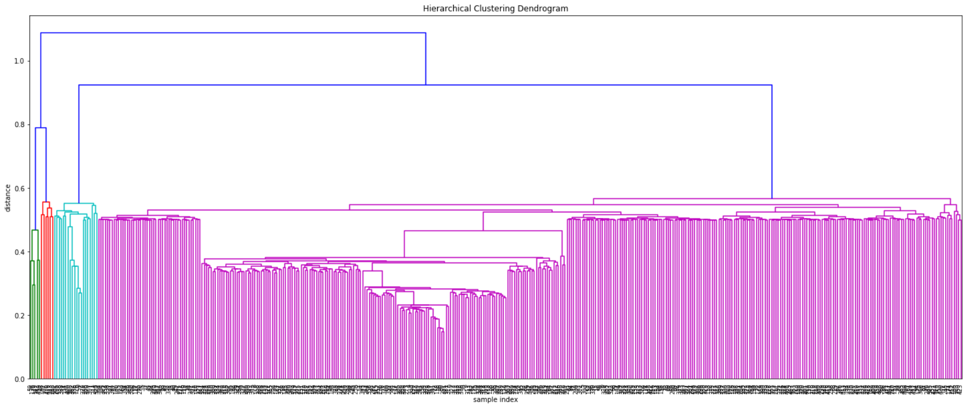


Figure 3.8: Dendrogram of average link hierarchical clustering with $g = 0$.

Finally, the results of tests with the normalized scores and $g = 0$ are not as good as the others and no inferences can be made. In Figure 3.9 the result of the complete link clustering is presented. Since we normalized the scores and they range between 0 and 1 the distances between the sequences do not differ too much which is why it is not possible to see jumps in the dendrogram. The remaining results are not presented since all of them are more or less similar to Figure 3.9 where it is very difficult to assess the existence of clusters.

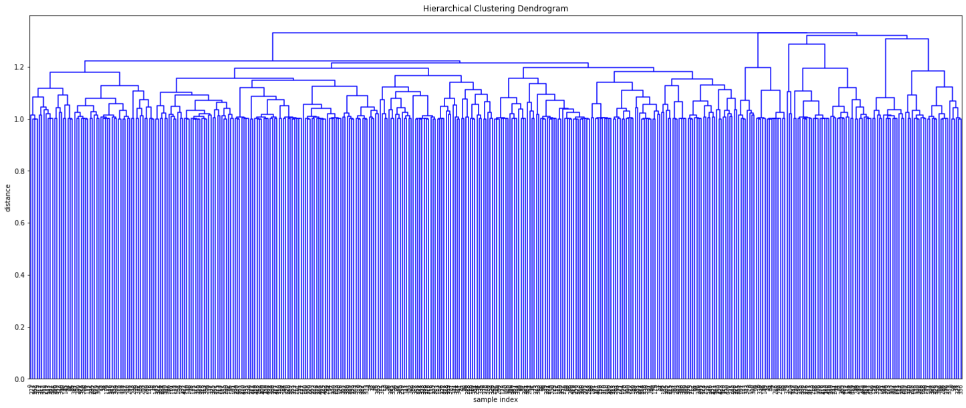


Figure 3.9: Dendrogram of complet link hierarchical clustering with $g = 0$ and normalized scores.

With this experiment it is possible to arrive at several conclusions. The best linkage methods when doing a manual analysis of the dendrogram and computing the cophenetic values are the complete and average linkage methods. Actually, this observation makes sense; when comparing two clusters a good metric to assess similarity between them it is to compare the most dissimilars patients (farthest elements) in both clusters (complete link). Also taking the average of all distances between two clusters is a good approach because it minimizes the effect of less similar patients, i.e outliers when comparing clusters. Another interesting observation was the fact that with $g = 0$ more interpretable results were obtained than with $g = 0$ and normalized scores. Despite when using only $g = 0$ the scores obtained with the alignments are not the most correct ones in the sense of global similarity of sequences the results show well defined clusters. An interpretation for using only $g = 0$ is that higher scores obtained in the alignments simply mean that huge parts of sequences were matched.

3.5 Experiment 2 - DAS28 sequences

This experiment uses the clinical history of the measurements of DAS28 to create sequences and perform alignment. Previously it was referred the importance of DAS28 in understanding the progression and current state of rheumatoid arthritis in patients. The main difference in implementing this experiment compared to the previous one is on the pre-processing step, presented in Section 3.1. It is somehow similar to when we created the 'biological' sequences but here we will look at the *das28_4v* variable i.e the DAS28 measurement that takes four variables in his computation. The other variable that we will look into is the date of the appointment *dt_consulta* that will be used to compute relative interval times between the measurements taken at each appointment. In Figure 3.4 a resume of this information is presented.with the raw data that will be pre-processed.

id_doente	**dt_consulta**	**das28_4v**
33496	03/03/2008	7.984
33496	17/03/2008	5.748
33496	21/04/2008	3.535
33496	09/06/2008	NA
33496	05/11/2008	NA
33496	04/12/2008	4.347
⋮	⋮	⋮
33499	14/03/2011	NA
33502	12/11/2007	4.36
33502	29/11/2007	5.688
33502	12/12/2007	4.85
.	.	.

Table 3.4: Part of data used to create DAS28 temporal sequences.

The main pre-processing steps are:

1. Eliminate repeated [*id_doente das28_4v*] rows.

2. Eliminate rows with NA values.

3. Convert the values of *das28_4v* into four discrete events corresponding to the activity of the disease. Values above 5.1 indicate high level of activity and are indicated by letter D and values below 2.6 represent remission are indicated by letter A. Low activity and medium activity of the disease are represented by letters B and C and correspond to the intervals [2.6 3.2[and [3.2 5,1], respectively.

4. For each row, expect for the first row for each patient, that will be zero, the relative time interval is given by the difference between the current row and the previous one. These time intervals are associated with the corresponding DAS28 events already created in the previous step.

5. Create the prefix-encoded temporal sequences

After creating the temporal sequences and computing the alignments between all patient pairs the same tests were conducted as in the previous experiment. However, in this case for $g = 0$ with normalized scores a modification was made. Before, the “biological sequences” had all a common event Z that allowed the existence of at least one match when comparing two sequences but in this experiment absence of matches were observed hence normalized scores would not be possible to compute. To overcome this difficulty we artificially introduce an event Z with a corresponding positive time associated to it for all sequences. In Table 3.5 the results of the tests are presented.

Linkage Method	**Cophenetic Correlation Values**		
	g=0.5	g=0	g=0 with Z Normalized
Single	0.176	0.278	0.07
Complete	0.499	0.139	0.273
Average	0.580	0.170	0.458
Centroids	0.450	0.186	0.445

Table 3.5: Cophenetic Correlation Coefficient values for experiment 2 with 4 linkage methods and varying the gap penalty of TNW algorithm.

In general, the cophenetic correlation values computed in these experiment are quite low compared to the previous one. Again by doing a manual analysis the most interesting results were obtained with $g = 0.5$ for complete link and $g = 0$ for average link that are presented in Figures 3.10 and 3.11, respectively.

It is clearly visible the existence of 5 clusters in Figure 3.10 if we draw a cut-off line at a distance of 45.

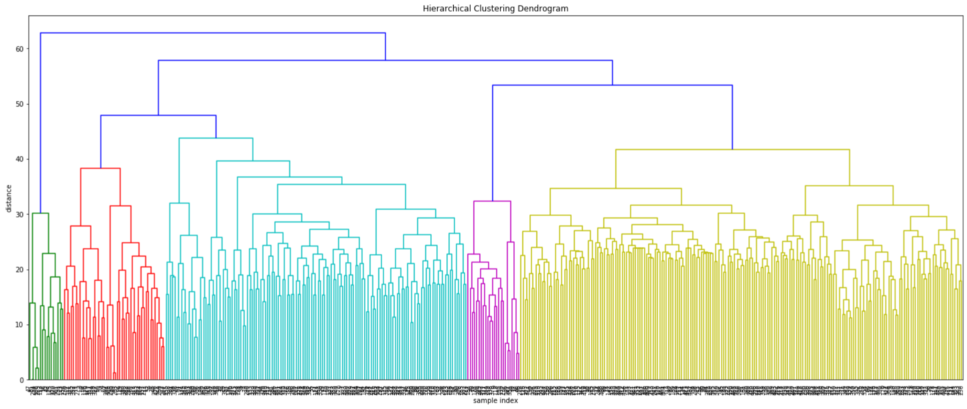


Figure 3.10: Dendrogram of complete link hierarchical clustering with $g = 0.5$.

When varying the gap penalty to 0 higher jumps were detected on the dendrograms obtained with the several linkage methods. In Figure 3.11 we decided to present the average link where 3 clusters are observed.

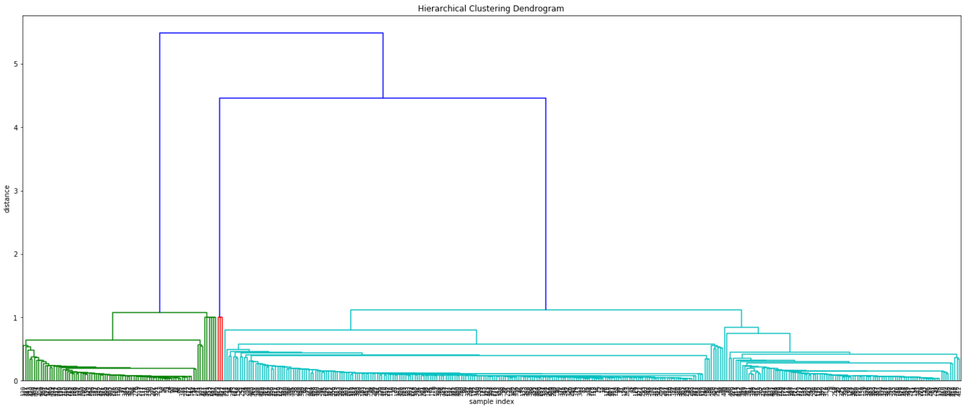


Figure 3.11: Dendrogram of average link hierarchical clustering with $g = 0$.

In order to understand the results obtained in both experiments it is very important interpret them in a clinical way to check if the results make sense or can tell us some relevant information. With these two experiments it was possible to test the variation of a parameter of the TNW algorithm combined with several metrics used in the clustering algorithm. Moreover, two ways of transforming a similarity matrix to distance matrix were assessed which were depended on the choice of the parameter g.

Chapter 4

Conclusions

4.1 Achievements

Finding similarities between patients in healthcare is an important task to achieve the goal of personalized medicine. This study briefly discussed the field of HealthCare Informatics and implemented a temporal alignment method that was tested with longitudinal data from the reuma.pt database. By using the scores obtained with the alignments a hierarchical clustering method was used to obtain several clusters of patients. A visual analysis and a comparison of the resulting dendrograms by the cophenetic correlation coefficient lead to the conclusion that the best distance metrics to use in the clustering method were complete and average link. Also when using a gap penalty of zero in the TNW algorithm more defined clusters were observed. The main difficulty found in this work was on interpretation of the resulting clusters and on the choice of the user-defined parameters of the TNW algorithm.

4.2 Future Work Plan

The Master Thesis work will consist on understanding better how to use correctly the TNW algorithm with the application in hand. More specifically, the choice of the parameters of the algorithm and the resulting alignments have to be further studied and understood. Also different temporal penalty functions have to tested. Another important task to be developed is the transformation of similarity matrix to a distance matrix before the application of a clustering method. In summary, the main aspects to be worked in the thesis are:

- Tuning the parameters - Find the best combinations of parameters. Study different possibilities for the scoring schema based on relevant clinical information (similarities between drugs). Develop new temporal penalty functions.

- Clustering methods - Test more distance metrics with the hierarchical clustering method implemented here. Study and test other clustering approaches that can have as an input a distance matrix.

- Validation of the clustering step - To complement the cophenetic correlation coefficient study other ways of validating the results.

- Clinical interpretation - Understand the results obtained in the clustering step to help treat patients in a more personalized way.

In Figure 4.1 a Gantt Chart is shown with the proposed plan to follow during the Master Thesis development. The expected time and deadlines to accomplish the main tasks described above are presented in the chart. Notice that the first task is Literature Review. It is important to read more about state-of-the-art sequence alignment methods that can inspire new approaches in this work. By taking into account the official delivery date of the thesis in 18th of April 2018, the proposed date to finish the thesis is 10th of March 2018. In this way, there is time for unexpected delays and for reviewing all the work.

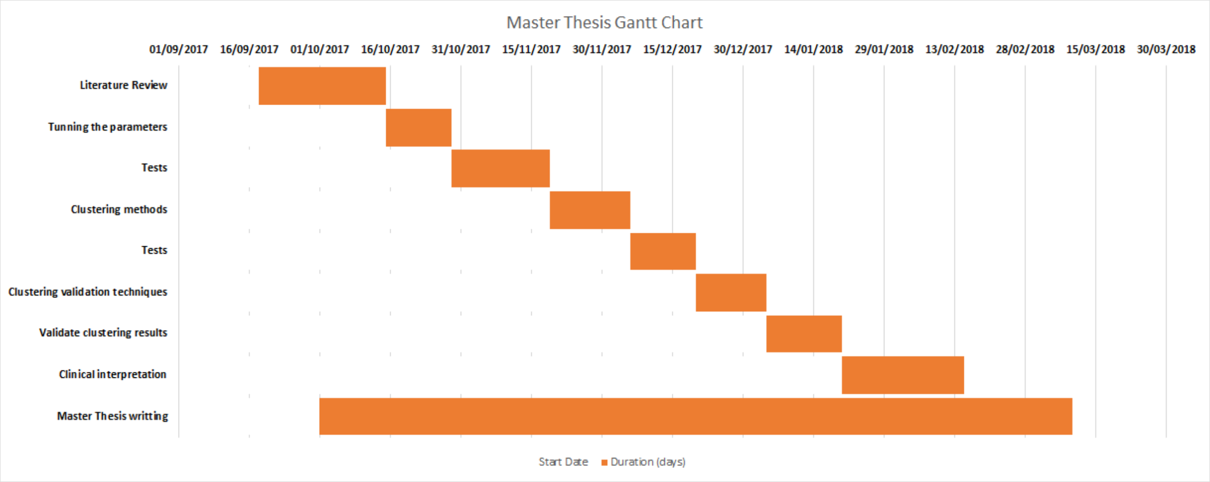


Figure 4.1: Master Thesis Gantt Chart.

Bibliography

[1] M. Herland, T. M. Khoshgoftarr, and R. Wald, “A review of data mining using big data in healthcare informatics,” *Journal of Big Data*, Jun 2014. doi:10.1186/2196-1115-1-2.

[2] J. Davis, E. Lantz, and D. P. et al., “Machine learning for personalized medicine: Will this drug give me a heart attack,” 2008. Proceedings of International Conference on Machine Learning (ICML).

[3] G. H. Fernald, E. Capriotti, and R. D. et al., “Bioinformatics challenges for personalized medicine,” *Bionformatics*, vol. 27, pp. 1741–1748, May 2011.

[4] A. N. Kho, L. V. Rasmussen, and J. J. C. et al., “Practical challenges in integrating genomic data into the electronich health record,” *Genetics in medicine: official journal of the American College of Medical Genetics*, vol. 15, pp. 774–781, Sep 2013.

[5] C. Polanco, “Precision medicine and portable embedded system devices [Letter],” *Nature*, vol. 520, pp. 609–611, Apr 2015.

[6] S. B. Needleman and C. D. Wunsch, “A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins,” *Journal of Molecular Biology*, vol. 48, pp. 443–453, 1970.

[7] H. Syed and A. K. Das, “Temporal Needleman-Wunsch,” *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Dec. 2015. doi:10.1109/DSAA.2015.7344785.

[8] S. Dua, U. R. Acharya, and P. Dua, *Machine Learning in Healthcare Informatics*, ch. 1. Springer, 2014.

[9] J. Duarte, C. F. Portela, A. Abelha, J. Machado, and M. F. Santos, “Electronic Health Record in Dermatology service.” Springer, Berlin, Heidelberg, 2011. In: Cruz-Cunha M.M., Varajão J., Powell P., Martinho R. (eds) ENTERprise Information Systems. CENTERIS 2011. Communications in Computer and Information Science, vol 221.

[10] H. Canhão, A. Faustino, and F. M. et al., “Reuma.pt - The Rheumatic Diseases Portuguese Register,” *Acta Reumatologica Portuguesa*, vol. 36(1), pp. 45–56, Jan. 2011.

[11] P. J. Diggle, P. Heagerty, and K.-Y. L. et al., *Analysis of Longitudinal Data*, ch. 1. Oxford University Press, second ed., 2013.

[12] A. C. of Rheumatology Committee on Communications and Marketing, “Rheumatoid Arthritis,” 2017. Retrivied from https://www.rheumatology.org/i-am-a/patient-caregiver/diseases-conditions/rheumatoid-arthritis.

[13] “Das28 - Introduction.” Retrivied from http://www.das-score.nl/das28/en/introduction-menu.html.

[14] NRAS, “Laboratory tests used in the diagnosis and monitoring of rheumatoid arthritis,” 2003. Retrivied from http://www.nras.org.uk/laboratory-tests-used-in-the-diagnosis-and-monitoring-of-rheumatoid-arthritis.

[15] T. S. community, “scipy.cluster.hierarchy.linkage,” 2014. https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.cluster.hierarchy.linkage.html.