Classifying Tree Structures using Elastic Matching of Sequence Encodings

Angeliki Skoura[1], Iosif Mporas[1], and Vasileios Megalooikonomou[1,2]

[1]Multidimensional Data Analysis and Knowledge Discovery Laboratory

Dept. of Computer Engineering and Informatics, University of Patras

26500 Patras, Greece

[2]Center for Data Analytics and Biomedical Informatics, Temple University

Philadelphia, USA

{skoura, vasilis}@ceid.upatras.gr, imporas@upatras.gr

Abstract: Structures of tree topology are frequently encountered in nature and in a range of scientific domains. In this paper, a multi-step framework is presented to classify tree topologies introducing the idea of elastic matching of their sequence encodings. Initially, representative sequences of the branching topologies are obtained using node labeling and tree traversal schemes. The similarity between tree topologies is then quantified by applying elastic matching techniques. The resulting sequence alignment reveals corresponding node groups providing a better understanding of matching tree topologies. The new similarity approach is explored using various classification algorithms and is applied to a medical dataset outperforming state-of-the-art techniques by at least 6.6% and 3.5% in terms of absolute specificity and accuracy correspondingly.

Keywords: Tree structures, Elastic matching, Breast ductal trees

1. Introduction

Branching or tree topology is a fundamental mechanism of nature which usually arises where there is a reason to maximize the area of contact between a structure and its environment under certain constrains [1]. For example, plant shoot systems maximize the area for photosynthesis and gas exchange under mechanical constraints such as gravity and wind damage [2]. Similarly, branching topology is fundamental to the development and function of many vertebrate organs including lung, kidney, mammary gland and brain [3]. In animal tissues, branching is developed to create a large surface area for exchange between the external environment and internal tissues into a small volume. For example, the branching structure of mammalian lungs enables gas exchange while minimizing the total distance from alveoli, the terminal ends of the respiratory tree, to the trachea. In the case of blood systems, vascular topographical geometry far from being a totally random network has a tendency to conform to physical principles such as minimization of shear stress and work across the vasculature [4]. In geology, the branching structure of river networks is an organized signature of soil erosional mechanics [5]. Furthermore, the concept of tree topology is widely used in sciences in order to represent hierarchical relationships among objects. Presenting data in the form of tree diagrams is an effective and valuable mechanism to organize existing data for a range of disciplines [6]. In biology for example, phylogenetic trees represent the evolutionary relationship between different species or organisms and RNA secondary structures are represented as ordered labeled trees to facilitate their comparison [7]. Additionally, trees are among the most common and well-studied combinatorial structures in computer science. Various kinds of data structures referred to as trees represent ordering relationships amongst a set of values through the use of pointers offering efficient solutions to the frequent operations of node insert, delete, and update. Moreover, ontologies which capture the structure of a domain are represented as trees with terms as tree nodes and the relations between the terms as branches.

Tree matching is important in many applications and refers to quantifying the degree of similarity between two trees and finding alignments among tree nodes. In computational domains, the most common measure for assessing the similarity of two labeled trees is the edit distance metric [8] which computes the cost of transforming one tree based on three edit operations on nodes; insertion, deletion, and relabeling. Based on the edit distance, many tree matching techniques have been proposed and require a model that defines the relabeling cost between nodes and the insertion/deletion cost for nodes which are not matched [9]. Given such a model, the tree-matching problem is to find a lowest-cost mapping between trees. Another method to measure similarity between two trees is the largest common subtree; this approach is useful in chemistry and computational biology where substructures usually represent conserved structural motifs [10]. A similar methodology, the transferable ratio, has been proposed to measure the ability of transforming one tree to another and the method was applied to the analysis of secondary structures derived from RNA species [11].

In the field of medical image analysis, tree matching methods have been proposed in classification studies of anatomical tree structures of the human body to reveal aspects of physiology about the corresponding organs. Regarding the type of descriptive characteristics of tree topology, existing methods can be divided into three categories; index-based, vector-based and matrix-based similarity approaches. The algorithms of the former category usually quantify geometric characteristics or compute dimensionless measures of tree topology. For example, in [12] retinal vessel width or equivalently vessel diameter was suggested as an important parameter in retinal blood flow measurement. Airway morphometry information including human airway diameters were considered for the clinical assessment of bronchoconstrictive diseases such as asthma and the associated evaluation of treatment effectiveness [13]. Tree asymmetry, a numerical index quantifying the asymmetry of a binary tree, was proposed as an effective way to detect early radiological findings in galactograms regarding breast cancer [14]. The approaches of the second category employ encoding techniques to obtain symbolic representations of tree topologies. In the field of neuroscience, Sholl analysis [15], a method for quantifying neuronal dendritic branches relative to distance from the neuronal body, achieved widespread application towards the analysis of dendritic geometry, ramification richness and dendritic branching patterns [16, 17]. Representation schemes using symbolic string representation of branching and text mining techniques were proposed to analyze tree structures appearing in medical images [14, 18]. Finally, regarding matrix-based approaches, the ramification matrix (R-matrix) representation of ductal networks were used in breast modeling for mammography simulation [19]. R-matrices, whose elements represent the probability of branching at various levels of a ramified tree, were analyzed in order to correlate ductal tree anatomy with clinical findings. More recently, a family of graph kernels was introduced to analyze airway tree structure and geometry with respect to diagnosis of chronic obstructive pulmonary disease [20].

In this paper, we describe a tree matching framework which introduces the concept of elastic matching of sequences of tree encodings in order to find nonlinear pairwise matching of nodes between tree topologies. We explore the proposed modular architecture by testing various algorithms for each methodology module; node labeling, tree traversal, elastic sequence matching and classification methods. The main contributions of the paper are:

- a novel framework for computing similarity and performing classification of tree topologies
- a new concept of node mapping between trees, derived from the alignment of the sequence encodings

- exploration of the modules of the proposed framework using three traversal modes, three labeling schemes, two sequence matching techniques and six classifiers
- performance evaluation of the new framework by means of sensitivity, specificity and accuracy and outperforming state-of-the-art tree similarity techniques in the context of a medical application.

To apply the proposed methodological framework, a dataset of breast ductal trees was selected in order to test a clinical hypothesis regarding radiological findings of breast cancer. The analysis of galactograms, medical images which visualize the breast ductal tree, provides insight into the topology of breast ductal network which may be affected by the presence or increased risk of breast cancer [19]. Additionally, using the manual classification of ductal trees by physicians in order to discriminate patients with reported radiological findings regarding breast cancer and normal cases, the new framework is evaluated by means of sensitivity, specificity and accuracy. Considering previously proposed characterization methods which focused on comparing ductal tree encodings [14, 18], unequal length of tree representations has remained a major challenge in comparing tree structures. Our approach addresses this problem enabling the comparison of tree structures of different number of nodes and offers a better understanding of how tree-shape anatomical structures could be compared. The experimental results showed that the proposed methodology outperformed the state-of-the-art methods proposed for the classification of ductal trees. More specifically, the best-performing scheme of the proposed framework outperformed the best-performing state-of-the-art technique [18] by 6.6% and 3.5% in terms of absolute specificity and accuracy correspondingly.

The paper is organized as follows. The next section presents an overview of the proposed framework for classifying tree topologies. In Section 3, the experimental setup is presented in detail. Then, in Section 4 the application of the proposed methodology is demonstrated on the clinical dataset of galactograms and evaluation is provided using comparative results, explanatory figures and examples. Finally, conclusions and suggestions are discussed in Section 5.

2. Methodology

2.1 Framework for Tree Structure Classification

The proposed framework consists of a modular architecture for the classification of tree structures to a closed set of target classes $1 \leq k \leq K$. The block diagram of the proposed framework is illustrated in Fig.1. The input to the framework is outlines of tree topologies. The modules of the framework perform independently to each other, thus offering the ability to change or adapt any of them without breaking up the overall framework operation. As can be seen in Fig. 1 the architecture can briefly divided into two phases, namely the training and the test phase.

FIGURE 1

During the training phase a set of I annotated (i.e. of known target class labels k) tree topologies, $\hat{T} = \{T_i\}$, where $1 \leq i \leq I$ is the i-th training tree topology, are used for building a classification model C. In detail, each of the T_i training trees is initially passed through the node labeling block, where every node of the tree is labeled with an identity number according to a node labeling function NL, i.e. $L_i = NL(T_i)$. The corresponding trees with labeled nodes, L_i, are afterwards processed by the tree traversing module, where the tree structure is described by a sequence of labels, S_i, using a traverse function Tr, i.e. $S_i = Tr(L_i)$. The sequence S_i corresponds to the node-path after applying the traverse function Tr for passing through all labeled nodes of the i-th tree. Subsequently, the I estimated sequences S_i, with $1 \leq i \leq I$, are compared against a set of J reference sequences, S_j^{ref}, with $1 \leq j \leq J$, using a sequence matching function D. The set of reference sequences $\mathrm{S^{ref}}$ consists of equal number $\mathrm{n = J/K}$ of tree sequences for each target class k (the tree sequences have been extracted using the same processing steps as those followed for the training tree topologies). In case that n is smaller than the total number of tree sequences of a class k with cardinality $\mathrm{c_k}$, a set $\mathrm{S^{ref,k}}$ consisted of n tree sequences is selected to represent sequences of the $\mathrm{k^{th}}$ class according to the formula:

$$\mathrm{S^{ref,k} = \{S_x : D(S_x, S_y) = max_{x,y} D(S_x, S_y), 1 \leq x, y \leq c_k, x \neq y\}}.$$

The selected n tree sequences have the maximum distance among the class trees and are considered to represent the $\mathrm{k^{th}}$class.

For the i-th training tree topology, the matching function D estimates the matching distance between the sequence S_i and each of the J reference sequences constructing the feature vector $F_i \in \mathbb{R}^J$, shown in Fig. 1. After processing the I training tree topologies, the corresponding feature vectors are used to train a classification model C. The data-mining algorithm used for building the classification model C will model the underlying information of the distance of a tree topology i from each of the reference tree topologies j. Thus, reference tree topologies with high discriminative ability among the target classes will be weighted higher than reference tree topologies with low discriminative ability the corresponding dimension of which will slightly be utilized from the data-mining algorithm.

During the test phase, a tree topology of unknown target class, Y, is processed by the node labeling module and a labeled tree is constructed according to the NL function, i.e. $L_Y = NL(Y)$. From the labeled tree L_Y the corresponding traverse sequence is estimated using the Tr function, i.e. $S_Y = Tr(L_Y)$. At the sequence matching module the S_Y sequence is matched against the same reference sequences, S_j^{ref}, used in the training phase and the test feature vector $F_Y \epsilon \mathbb{R}^J$ is estimated. The decision d of the target class in which the test tree topology belongs to is taken by the classification module using the model C, i.e. $d = f_c(F_Y)$, where $d \in \{1 \leq k \leq K\}$.

The modular architecture of the proposed framework allows the use of different algorithms for the implementation of each of the modules (node labeling, tree traversing, sequence matching and classification), independently from the other ones. Furthermore, the framework can be applied to different sets of reference tree topologies, thus making the proposed architecture applicable to scenarios with different amount of available data annotated with their label class.

2.2 Exploring algorithms of labeling, traversal, sequence matching and classification

The framework for tree structure classification described above was applied and evaluated on a dataset of galactograms. The experiments presented in the following section were performed by evaluating three types of labeling, three types of traversal, two algorithms for elastic matching and six classification algorithms.

Considering the node labeling three functions were used. These are (i) the $\mathrm{NL_{OFF}}$ approach [14] and two new modifications of it, namely (ii) the $\mathrm{NL_{LOG}}$, and (iii) the $\mathrm{NL_{INV}}$ approach, which are proposed here for the first time. According to the $\mathrm{NL_{OFF}}$ approach, the label of the node (i, j) is the numerical value $(2^{\mathrm{i}} + \mathrm{j})$ where i refers to the i-th level, assuming that the root's level is 0 and the level is increased moving downwards, and j refers to the position of the node inside each level, assuming that the leftmost node of every level has $\mathrm{j} = 0$ and j is increased by one for each node (moving rightwards). Considering that using the $\mathrm{NL_{OFF}}$ approach the labels increase exponentially across tree levels (for a tree of L levels, $L \in \mathbb{N}, L > 1$ the labeling range is the interval $[1, 2^L]$), we propose $\mathrm{NL_{LOG}}$ and $\mathrm{NL_{INV}}$ to decrease the range of labels (the labeling range is $[0, L]$ and $[1/2^L, 1]$ correspondingly). The $\mathrm{NL_{LOG}}$ labels are generated by applying logarithm function (base $b = 2$) to $\mathrm{NL_{OFF}}$ labels, while the $\mathrm{NL_{INV}}$

labels are generated by the inverse number of $\mathrm{NL_{OFF}}$ labels. Although in both modifications the labeling range is reduced, using $\mathrm{NL_{LOG}}$ the labels increase across tree levels, whereas, using $\mathrm{NL_{INV}}$ the labels decrease across tree levels. In all labeling approaches, the nipple of the traced ductal tree was considered as the tree root.

Three types of tree traversing were tested. These are (i) the Level Order Traversal ($\mathrm{T_{LO}}$), (ii) the Pre-Order Traversal ($\mathrm{T_{PO}}$) and a modification of post order, namely (iii) the TRiple pre-order Traversal ($\mathrm{T_{TR}}$), which is proposed here for the first time. According to $\mathrm{T_{LO}}$ traversal, every node on a level is visited before going to a lower level following a breadth-first manner. The $\mathrm{T_{PO}}$ traversal is a type of depth-first traversal which starts by visiting the root, traverses the left sub-tree and afterwards the right sub-tree. This procedure is performed recursively. The $\mathrm{T_{TR}}$ approach is derived from the Pre-Order Traversal with the difference that every parent node is visited three times (not only once initially but also after traversing its left and its right sub-tree). Let l(p) denote the label of an internal (non-leaf) node p of a tree. Using this type of traversal, the subsequence included between the two first occurrences of $l(p)$ corresponds to the left sub-tree emerging from the node p whereas the subsequence included between the two last occurrences of $l(p)$ corresponds to the right sub-tree emerging from the node p. The sequence encoding generated by the $\mathrm{T_{LO}}$ and $\mathrm{T_{PO}}$ traversal of tree of N nodes is of length N whereas in the case of $\mathrm{T_{TR}}$ traversal the sequence encoding is of length $2N-1$.

For the elastic matching between tree sequences two methods were evaluated, namely the Dynamic Time Warping (DTW) [21] and the Minimum Variance Matching (MVM) [22]. Let us consider two sequences $X=\{x_1,x_2,\dots,x_N\}$ and $Y=\{y_1,y_2,\dots,y_M\}$ of length $N,M\in\mathbb{N},N\leq M$. The DTW method finds the optimal alignment between the two series under three constrains: boundary, monotonicity and continuity conditions. According to the boundary conditions, the first and the last elements of X and Y are required to be aligned to each other; that is the entire tree sequences are aligned. The monotonicity condition prevents the matching backwards, i.e. if an element in X precedes a second one this should also hold for the corresponding elements in Y, and vice versa. According to the continuity condition no element in X and Y can be omitted. The similarity of these sequences is computed as the distance of the aligned elements of X and Y. The DTW technique suffers from lack of flexibility on end matching points as well as it is sensitive to outliers. In contrast to the DTW method which aligns all elements between the sequences, the MVM method allows skipping elements of the larger sequence when computing the alignment. MVM is used to find the best matching part of the larger sequence Y given sequence X and it guarantees that the whole smaller sequence will be matched. The distance value between two sequences is estimated directly based on the distances of corresponding elements, as in the DTW method.

Fig. 2 - Fig. 4 present examples of matching two trees using the proposed methodology. Let the two trees presented in Fig. 2d. In case of NL$_{OFF}$ labeling and T$_{LO}$ traversal, the corresponding tree sequences are $T_1 = \{1,2,3,4,5,6,7,14,15,28,29,30,31,58,59,60,61,62,63,118,119\}$ and $T_2 = \{1,2,3,6,7,12,13,14,15,24,25,26,27,30,31,48,49,50,51\}$ which are plotted in Fig. 2a. The alignment of the two sequences using the DTW matching scheme is presented in Fig.2b and the corresponding alignment using the MVM scheme is presented in Fig.2c. The interpretation of the node mapping of the two sequences is shown in Fig.2d and Fig.2e for the two matching schemes correspondingly; the aligned nodes between the trees T$_1$ and T$_2$ are presented with the same color. Fig. 3 shows the alignment of two trees using the NL$_{OFF}$ labeling and the T$_{PO}$ traversal mode. In this case the tree sequences are $T_1 = \{1,2,4,5,3,6,7,14,28,29,58,59,118,119,15,30,60,61,31,62,63\}$ and $T_2 = \{1,2,3,6,12,24,48,49,25,50,51,13,26,27,7,14,15,30,31\}$. Fig. 3b-c and Fig. 2d-e present the alignment of the two sequences and the aligned tree topologies using DTW and MVM matching scheme correspondingly. By employing elastic matching techniques on tree encodings, a group of nodes (one or more nodes) of a tree is mapped to a group of nodes of the comparing tree allowing nonlinear mapping between the nodes of the compared trees. Using DTW all nodes of both sequence are aligned, however, using MVM nodes of the larger sequence remain unaligned. In the example of Fig.2e, the nodes $\{118,119\}$ of T_1 are not aligned using MVM; these nodes are colored in gray. In the case of NL$_{OFF}$ labeling and T$_{TR}$ traversal mode, the tree sequences are $T_1 = \{1,2,4,2,5,2,1,3,6,3,7,14,28,14,29,58,29,59,118,59,119,59,29,14,7,15,30,60,30,61,30,15,31,62,31,63,31,15,7,3,1\}$ and $T_2 = \{1,2,1,3,6,12,24,48,24,49,24,12,25,50,25,51,25,12,6,13,26,13,27,13,6,3,7,14,7,15,30,15,31,15,7,3,1\}$. Fig. 4 shows the alignment of the trees T_1 and T_2 using the NL$_{LOG}$ labeling and the T$_{PO}$ traversal mode. The use of T$_{TR}$ traversal mode is proposed here as it results in matching subtrees between the compared trees and this effect is visualized in Fig. 5. The alignment between the subtrees of T_1 and T_2 using DTW and MVM matching schemes is presented in Fig. 5a and Fig 5b correspondingly.

FIGURE 2

FIGURE 3

FIGURE 4

FIGURE 5

After applying the matching algorithms to tree encodings, for each instance (i.e. for each test ductal tree) a feature vector is computed consisting of the distance value between the test tree and each tree of the reference set. These feature vectors are used as input to a classification algorithm to decide the class of the test tree. For the classification stage we employed the following machine learning algorithms, namely the C4.5 decision tree (denoted as J48) [23], the support vector machines (SVM) implemented with the sequential minimal optimization method using polynomial kernel function [24, 25], the IBk k-nearest neighbors algorithm [26], the 3-layer multilayer perception (MLP) neural network [27], the random tree (RTree) and the random forest (RForest) algorithms [28]. For the construction of classification models we relied on the WEKA machine learning software toolkit [28].

3. Experimental Dataset

The experimental data consisted of 77 x-ray galactograms (Fig. 6a) acquired at the Thomas Jefferson University Hospital and the University Hospital of Pennsylvania, USA. Regarding the classes of the dataset, 55 images corresponded to women with No radiological Findings (denoted here as class NF) and 22 to women with Reported radiological Findings (denoted as class RF). Certain preprocessing steps were required to obtain the outlines of the ductal tree structures. At first, the boundaries of ductal trees needed to be traced out of the background of the image. The tree structures were reconstructed by identifying points of branching and resolving potential defects such as anastomoses. Node annotation of ductal trees and tree 2-dimensional reconstruction were performed manually by expert physicians and nipple was considered as the root for all trees of the dataset (Fig. 6b). The average number of tree nodes in NF class was 158.18 whereas the average number of tree nodes in RF class was 179.27.

4. Experimental Results

The proposed framework for tree structure classification described in Section 2 was evaluated under the experimental setup described in Section 3. During evaluation the leave-one-out cross validation protocol was followed in order to ensure no overlap between training and test datasets. For each fold of the cross validation protocol the evaluation was performed for three traversal modes, three labeling schemes, two sequence matching techniques and six classifiers. For evaluation metrics we used the sensitivity (or true positive rate, i.e. the percentage of galactograms classified in RF class, which actually belong in RF class according to the experts/ physicians), the specificity (or true negative rate, i.e. the percentage of ductal trees classified in NF class, which actually belong in NF class according to the experts/physicians) and the accuracy (i.e. the percentage of correctly classified trees). The performance results of the tree structure classification framework in terms of sensitivity, specificity and accuracy are shown in Tables 1, 2 and 3 respectively. The best performing setups of the framework are indicated in bold.

TABLE 1

TABLE 2

TABLE 3

As can be seen in the Tables, among tested *node labeling* techniques, the LOG method outperformed the other two labeling schemes in almost all cases regardless of traversal mode, matching technique and classification model. Moreover, NL_{INV} and NL_{OFF} methods achieved similar high performance for the classifiers IBk, MLP, RTree and RForest. The use of logarithms of the labels rather than the original values of NL_{OFF} labeling reduced the resulting values of sequence distance of the matching step which represented the inputs of classifiers. Both inversion and logarithmization of the NL_{OFF} labels resulted in a shorter range of distances. However, the conversion of labels using a logarithmic scale (NL_{LOG}) provided a good modification of the NL_{OFF} scheme which assigns labels of exponential size given the number of tree levels.

Regarding *tree traversal* modes, the T_{LO} method achieved better results in most cases compared to T_{PO} and T_{TR} traversing (for IBk, SMO, MLP, RTree and RForest) especially when combined with IBk or SMO. Comparing the two basic types of traversal, the T_{LO} encoding resulted in pairwise matching of nodes of similar tree levels (Fig. 2), whereas the T_{PO} encoding included matching a node with a sub-tree between the compared trees (Fig. 3). Note that by aligning nodes of similar tree levels, the corresponding pairwise distance had smaller values compared to aligning nodes to subtrees, which is typical when using the T_{PO} method. The general superiority of the T_{LO} method indicated that aligning similar tree levels resulted in a more effective alignment.

Regarding *sequence matching*, the MVM algorithm outperformed the DTW scheme for almost all cases apart from the cases of MLP and Random Tree. The omission of continuity condition which is the main difference between DTW and MVM schemes resulted in higher classification rates for almost all evaluation metrics indicating that focusing on a smaller subset of tree nodes provides more effective tree alignment. The fact that all elements, including outliers, participate in the correspondence optimized by DTW often leads to an incorrect correspondence of other sequence elements [22]. However, using MVM outliers are omitted and the correspondence computed is not corrupted.

Comparing among the tested *classifiers*, J48 and IBk achieved the best results averaged over all methodology variants (J48: {Sens, Spec, Acc} = {66.7%, 71.5%, 70.9%}, IBk: {Sens, Spec, Acc} = {69.7%, 71.4%, 70.6%}). Moreover, the IBk classifier presented the minimum difference between the rates of Sensitivity and Specificity, indicating stability in detecting both true positive and true negative cases. Among the rest of the classification algorithms, PolyKernel presented high results when combined with BFE traversal and MVM technique regardless the labeling method.

Although the best results of RForest and RTree were similar concerning sensitivity, specificity was favored in the case of RForest. Thus, the overall accuracy of RForest is higher than this of RTree. Moreover, MLP is well suited for NL_{LOG} labeling regardless the traversal mode.

The highest accuracy, which is the most valuable evaluation measure in the application of classification of ductal trees according to physicians [19] was achieved by the J48 decision tree ({Sens, Spec, Acc} = {86.4%, 90.9%, 88.6%}) when combined with NL_{LOG} labeling, NL_{TRI} encoding and MVM matching (let denote this methodology variant as LOG_TRI_MVM_J48).

For comparison, ductal tree classification techniques reported in literature such classification using text mining of Prüfer encodings and depth-first sting encoding [18], NL_{OFF} labeling and text mining and classification and tree asymmetry index [14] and geometrical tree features in boosting frameworks [29] were used. For comparison the best results of these methodologies were considered. The proposed LOG_TRI_MVM_J48 approach outperformed state-of-the-art characterization techniques based on Prüfer encoding and text mining by 6.6% and 3.5% in terms of absolute Specificity and Accuracy correspondingly. Moreover, compared to Prüfer encoding scheme which aligns nodes using their parents' labels, the proposed approach of elastic matching enabled a more effective comparison of tree sequences of equal length. Compared to the NL_{OFF} labeling and text mining technique [29], the Sensitivity of the proposed LOG_TRI_MVM_J48 scheme is reduced by

0.7%, however, the Specificity is enhanced by 15.5%. Additionally, the proposed matching methodology provides a more interpretable framework for mapping tree topologies.

Regarding the proposed labeling techniques and compared to the $\mathrm{NL_{OFF}}$ approach whose labeling range is $[1, 2^L]$, the $\mathrm{NL_{LOG}}$ and $\mathrm{NL_{INV}}$ techniques are used to decrease the labeling range which becomes $[0, \mathrm{L}]$ and $[1/2^{\mathrm{L}}, 1]$ correspondingly for the two methods, where L is the number of tree levels, $L > 1$. Moreover, the labels generated by the $\mathrm{NL_{OFF}}$ technique are natural numbers but the labels generated by the modifications of it are real numbers (positive real numbers except the root's label in $\mathrm{NL_{LOG}}$ scheme which equals zero). Both inversion function and logarithmic function are 1-1 functions and naming conflicts may occur only as a result of storage limitations regarding the variables that represent the labels' values. In the experiments, the labels of the $\mathrm{NL_{LOG}}$ and $\mathrm{NL_{INV}}$ techniques are saved in double-precision floating-point format which occupies 64 bits in the computer memory) and gives 15–17 significant decimal digits precision [31]. As the maximum number of levels, that the trees of the dataset used in the experiments had, was 21, using the $\mathrm{NL_{LOG}}$ technique did not result in naming conflicts (for example, the floating-point numeric value of $log_2(2^{21})$ was not equal to the floating-point numeric value of $log_2(2^{21} - 1)$). Similarly, the $\mathrm{NL_{INV}}$ technique did not result in naming conflicts. That is, although rounding of labels is performed due to saving a real number using a finite number of decimal digits, there were no naming conflicts for the dataset used.

In general, the labeling schemes affect the inputs of the classification models since the feature vectors consist of distances between sequences. The distance between two sequences X and Y (either using the DTW or the MVM matching technique) is the Euclidean distance of the aligned elements of X and Y. The distance range between any two elements $\mathrm{x_1} \in \mathrm{X}$ and $\mathrm{y_1} \in \mathrm{Y}$ when using the $\mathrm{NL_{OFF}}$ approach is $\mathrm{D_{OFF}} \in [0, 2^{\mathrm{L}} - 1]$. The distance ranges when using the $\mathrm{NL_{LOG}}$ and $\mathrm{NL_{INV}}$ labeling schemes are $\mathrm{D_{LOG}} \in [0, \mathrm{L}]$ and $\mathrm{D_{INV}} \in [0{,}1 - 1/2^{\mathrm{L}}]$ correspondingly. In order to perform a rough comparative study of the three labeling schemes, the averaged results of Sensitivity, Specificity and Accuracy over all traversal modes, matching schemes and classifiers are presented correspondingly: $\mathrm{NL_{OFF}}$ = {65.01, 72.70, 68.91}, $\mathrm{NL_{LOG}}$ = {66.19, 73.01, 69.62}, $\mathrm{NL_{INV}}$ = {63.13, 69.94, 66.54}. The $\mathrm{NL_{INV}}$ presented the worst performance among labeling schemes regardless of the evaluation metric. This result might be attributed to the fact that the distance range $\mathrm{D_{INV}}$ is the shortest among the distance ranges of all tested labeling schemes, meaning that the elements of the feature vectors are not effectively differentiated. Comparing $\mathrm{NL_{LOG}}$ and $\mathrm{NL_{OFF}}$ labeling schemes, the $\mathrm{NL_{LOG}}$ method outperformed the $\mathrm{NL_{OFF}}$ technique indicating that reducing the initial distance range from $\mathrm{D_{OFF}} \in [0, 2^{\mathrm{L}} - 1]$ to the shorter range $\mathrm{D_{LOG}} \in [0, \mathrm{L}]$ is more effective. As many trees of the dataset are unbalanced, $\mathrm{NL_{OFF}}$ labels which increase exponentially regarding the number of tree levels L result in distances (i.e. feature ranges) of large variation. In the literature [32] it is known that the performance of classifiers is improved when using features of similar ranges. Therefore, we were motivated to apply binary logarithmization on the $\mathrm{NL_{OFF}}$ labels that offers shorter feature ranges.

Finally, the $\mathrm{T_{TR}}$ approach offers larger tree sequence representations which in turn result in larger distance ranges compared to the other tested traversal methods. Larger distance ranges result in degradation of the classification performance [32]. Therefore, comparing the combination of the $\mathrm{T_{TR}}$ method and the $\mathrm{NL_{OFF}}$ labeling scheme to the combination of the other tested traversal methods and the $\mathrm{NL_{OFF}}$ labeling scheme, we observed that the first approach presented inferior performance. However, when combining the $\mathrm{T_{TR}}$ method with the $\mathrm{NL_{LOG}}$ labeling scheme, we observed that in several cases (LOG_TRI_MVM_J48 variant and LOG_TRI_DTW_MLP variant) the resulting classifiers presented better performance than the alternative traversal methods. This could be attributed to the nature of the logarithmic function which reduces wide-ranging quantities to smaller scopes.

5. Conclusion

In this paper, a multistep framework for matching and classifying tree topologies is presented. The key contribution of this work is the idea of matching tree topologies based on tree encoding methods and elastic sequence matching techniques. Our approach enables comparing tree topologies of different number of nodes or tree depth and quantifies their similarity. Furthermore, the resulted node alignments between the compared trees introduce a new concept of matching tree topologies. For application and evaluation purposes, a medical dataset was employed and classification experiments of breast ductal trees were performed regarding reported galactographic findings of breast cancer. The application of the methodology resulted in outperforming state-of-the-art approaches of characterization of tree structures in medical images via the analysis of sequence representations. Our future research plans include exploring new sequence encoding methods to represent tree topologies and applying the proposed methodology to other types of tree structures.

6. Acknowledgement

The authors are grateful to P. Bakic and A. Maidment from the Hospital of the University of Pennsylvania for providing anonymized galactograms and medical expertise. This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Pro-gram "Education and Lifelong Learning" of the National Strategic Reference Frame-work (NSRF) - Research Funding Programs: Heracleitus II and Thales, Investing in knowledge society through the European Social Fund.

7. References

[1] J. A. Davies, Branching Morphogenesis, Landes Bioscience, New York, (2006).

[2] J. Read, A. Stokes, Plant biomechanics in an ecological context, Am. J. Botany, vol. 93 (10) (2006), pp. 1546-1565.

[3] D. Iber, D. Menshykau, The control of branching morphogenesis, Open Biology, 3 (9) (2013).

[4] M. Zamir, J. Medeiros, Arterial branching in monkey and man, J. Gen. Physiology, 77 (1982), pp. 353–360.

[5] J. T. Perron, P. W. Richardson, K. L. Ferrier, The root of branching river networks, Nature, 492 (7427) (2012), pp. 100-103.

[6] S. Knipe, Data Trees as a Means of Presenting Complex Data Analysis, Open Journal of Knowledge Management, 7 (2013).

[7] B. A. Shapiro, K. Zhang, Comparing multiple RNA secondary structures using tree comparisons, Computer applications in the biosciences, 6 (4) (1990), pp. 309-318.

[8] K.C. Tai, The tree-to-tree correction problem, J. Assoc. Comput. Machinery, 26 (1979), pp. 422-433.

[9] P. Bille, A survey on tree edit distance and related problems, Theoretical Computer Science, vol. 337 (1-3), (2005), pp. 217-239.

[10] Y. Takahashi, Y. Satoh, H. Suzuki, S. Sasaki, Recoginition of largest common structural fragment among a variety of chemical structures, Analytical Sciences, 3 (1987), pp. 23-28.

[11] P. Kilpelainen and H. Mannila, Ordered and Unordered Tree Inclusion, Journal of SIAM Journal on Computing archive, 24 (2) (1995), pp. 340-356

[12] H. C. Chen, V. Patel, J. Wiek, S. M. Rassam, E. M. Kohner, Vessel diameter changes during the cardiac cycle, Eye, 8 (1994), pp. 97-103.

[13] T. A. Lewis, Y. Tzeng, E. L. McKinstry, A. C. Tooker, K. Hong, Y. Sun, J. Mansour, Z. Handler, M. S. Albert, Quantification of airway diameters and 3D airway tree rendering from dynamic hyperpolarized 3He magnetic resonance imaging, Magnetic Resonance in Medicine, 53 (2) (2005), pp. 474-478.

[14] A. Skoura, M. Barnathan, V. Megalooikonomou, Classification of ductal tree structures in galactograms, In: 6th IEEE Int. Symposium on Biomedical Imaging 2009 (ISBI '09), pp. 1015-1018.

[15] D.A. Sholl, Dendritic organization in the neurons of the visual and motor cortices of the cat, J Anatomy, 87 (1953), pp. 387-406.

[16] F. Caserta, W. D. Eldred, E. Fernandez, R. E. Hausman, L. R. Stanford, S. V. Bulderev, S. Schwarzer, H. E. Stanley, Determination of fractal dimension of physiologically characterized neurons in two and three dimensions, Journal of Neuroscience Methods, 56 (2) (1995), pp. 133-144.

[17] D. Ristanović, N. T. Milošević, V. Stulić, Application of modified Sholl analysis to neuronal dendritic arborization of the cat spinal cord, Journal of Neuroscience Methods, 158 (2), (2006), pp. 212-218.

[18] V. Megalooikonomou, M. Barnathan, D. Kontos, P.R. Bakic, A.D. Maidment, A Representation and Classification Scheme for Tree-like Structures in Medical Images: Analyzing the Branching Pattern of Ductal Trees in X-ray Galactograms, IEEE Trans. on Medical Imaging, 28 (4) (2009), pp. 487-493.

[19] P. R. Bakic, M. Albert, A. D. Maidment, Classification of galactograms with ramification matrices: preliminary results, Academic Radiology, 10 (2003), pp. 198-204.

[20] A. Feragen, J. Petersen, D. Grimm, A. Dirksen, J. H. Pedersen, K. Borgwardt, M. Bruijne, Geometric Tree Kernels: Classification of COPD from Airway Tree Geometry, In: 23rd Information Processing in Medical Imaging, Lecture Notes in Computer Science Volume, 7917 (2013), pp 171-183.

[21] E. Keogh, C. A. Ratanamahatana, Exact indexing of dynamic time warping, Knowledge and Information Systems, 7 (3) (2005), pp. 358-386.

[22] L. J. Latecki , V. Megalooikonomou , Q. Wang , R. Lakaemper , C. A. Ratanamahatana, E. Keogh, Partial Elastic Matching of Time Series, In: 5th IEEE International Conference on Data Mining 2005 (ICDM'05), pp. 701-704.

[23] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers (1993).

[24] J. Platt, Fast Training of Support Vector Machines using Sequential Minimal Optimization, Advances in Kernel Methods - Support Vector Learning (1998).

[25] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, K. R. K. Murthy, Improvements to Platt's SMO Algorithm for SVM Classifier Design, Neural Computation, 13 (3) (2001), pp. 637-649.

[26] D. Aha, D. Kibler, Instance-based learning algorithms, Machine Learning, 6 (1991), pp. 37-66.

[27] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning Internal Representations by Error Propagation, Parallel distributed processing: Explorations in the microstructure of cognition, 1, MIT Press, (1986).

[28] L. Breiman, Random Forests, Machine Learning. 45(1) (2001), pp. 5-32.

[29] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The WEKA Data Mining Software: An Update; SIGKDD Explorations, 11 (1) (2009).

[30] A. Skoura, T. Nuzhnaya, P. R. Bakic, V. Megalooikonomou, Classifying ductal trees using structural features and ensemble learning techniques, In: 14th Conference on Engineering Applications of Neural Networks 2013 (EANN '13), pp. 146-155.

[31] W. Kahan , Lecture Notes on the Status of IEEE Standard 754 for Binary Floating-Point Arithmetic, The Institute of Electrical and Electronics Engineers, New York, USA, (1987).

[32] A.B. A. Graf, A. J. Smola, S. Borer, Classification in a Normalized Feature Space Using Support Vector Machines, IEEE Trans. on Neural Networks, 14 (3), (2003)

Table 1. Comparison of several tree traversal methods, labeling schemes and classifiers in terms of sensitivity.

Sensitivity (%)													
Traversal	**Labeling**	**J48**		**Poly Kernel**		**IBk**		**MLP**		**RTree**		**RForest**	
		DTW	**MVM**	**DTW**	**MVM**	**DTW**	**MVM**	**DTW**	**MVM**	**DTW**	**MVM**	**DTW**	**MVM**
T_{LO}	NL_{OFF}	68.12	**81.82**	61.00	**86.36**	63.64	77.27	71.00	72.73	63.64	**81.82**	54.55	**81.82**
	NL_{INV}	77.27	68.18	**81.82**	68.18	77.27	77.27	72.73	68.18	59.09	68.18	77.27	68.18
	NL_{LOG}	63.64	77.27	**90.91**	**81.82**	**81.82**	72.73	**81.82**	77.27	**81.82**	68.18	**81.82**	77.27
T_{PO}	NL_{OFF}	59.09	72.73	63.00	50.00	68.18	72.73	57.00	50.00	59.09	72.73	59.09	68.18
	NL_{INV}	63.64	72.73	59.09	22.73	45.45	72.73	59.09	54.55	50.00	45.45	50.00	54.55
	NL_{LOG}	**86.36**	77.27	68.18	27.27	77.27	63.64	77.27	54.55	72.73	59.09	77.27	63.64
T_{TR}	NL_{OFF}	36.36	59.09	63.00	45.45	63.64	63.64	71.00	63.64	72.73	68.18	59.09	59.09
	NL_{INV}	63.64	63.64	63.64	27.27	63.64	68.18	77.27	54.55	**81.82**	63.64	68.18	63.64
	NL_{LOG}	63.64	**86.36**	54.55	36.36	72.73	72.73	77.27	63.64	68.18	72.73	59.09	68.18

Table 2. Comparison of several tree traversal methods, labeling schemes and classifiers in terms of specificity.

Specificity (%)													
Traversal	**Labeling**	**J48**		**Poly Kernel**		**IBk**		**MLP**		**RTree**		**RForest**	
		DTW	**MVM**	**DTW**	**MVM**	**DTW**	**MVM**	**DTW**	**MVM**	**DTW**	**MVM**	**DTW**	**MVM**
T_{LO}	NL_{OFF}	78.32	**86.36**	71.00	72.73	50.00	**86.36**	79.00	77.27	54.55	77.27	77.27	**86.36**
	NL_{INV}	59.09	72.73	63.64	**90.91**	**81.82**	**86.36**	**81.82**	77.27	68.18	72.73	72.73	**86.36**
	NL_{LOG}	**86.36**	63.64	59.09	**86.36**	**81.82**	**81.82**	**81.82**	72.73	**81.82**	**81.82**	77.27	**81.82**
T_{PO}	NL_{OFF}	50.00	**81.82**	76.00	**95.45**	45.45	54.55	59.00	59.09	72.73	**81.82**	68.18	**86.36**
	NL_{INV}	**86.36**	54.55	50.00	54.55	63.64	63.64	54.55	72.73	45.45	63.64	50.00	59.09
	NL_{LOG}	**86.36**	63.64	63.64	72.73	72.73	72.73	**81.82**	72.73	77.27	54.55	**86.36**	63.64
T_{TR}	NL_{OFF}	**81.82**	68.18	76.00	**95.45**	59.09	68.18	69.00	**81.82**	72.73	68.18	77.27	72.73
	NL_{INV}	68.18	59.09	**86.36**	63.64	77.27	77.27	72.73	68.18	77.27	63.64	**90.91**	**81.82**
	NL_{LOG}	77.27	**90.91**	**86.36**	**86.36**	**81.82**	**81.82**	**86.36**	68.18	68.18	63.64	**81.82**	72.73

Table 3. Comparison of several tree traversal methods, labeling schemes and classifiers in terms of accuracy.

Accuracy (%)													
Traversal	**Labeling**	**J48**		**Poly Kernel**		**IBk**		**MLP**		**RTree**		**RForest**	
		DTW	**MVM**	**DTW**	**MVM**	**DTW**	**MVM**	**DTW**	**MVM**	**DTW**	**MVM**	**DTW**	**MVM**
T_{LO}	NL_{OFF}	73.22	**84.09**	66.00	79.55	56.82	**81.82**	75.00	75.00	59.09	79.55	65.91	**84.09**
	NL_{INV}	68.18	70.45	72.73	79.55	79.55	**81.82**	77.72	72.73	63.64	70.45	75.00	77.27
	NL_{LOG}	75.00	70.45	75.00	**84.09**	**81.82**	77.27	**81.82**	75.00	**81.82**	75.00	79.55	79.55
T_{PO}	NL_{OFF}	54.55	77.27	71.00	72.73	56.82	63.64	58.00	54.55	65.91	77.27	63.64	77.27
	NL_{INV}	75.00	66.64	54.55	38.64	54.55	68.18	56.82	63.64	47.73	54.55	50.00	56.82
	NL_{LOG}	**86.36**	70.45	65.91	50.00	75.00	68.18	79.55	63.64	75.00	56.82	**81.82**	63.64
T_{TR}	NL_{OFF}	59.09	63.64	70.00	70.45	61.36	65.91	70.00	72.73	72.73	68.18	68.18	65.91
	NL_{INV}	65.91	61.36	75.00	45.45	70.45	72.73	75.00	61.36	79.55	63.64	79.55	72.73
	NL_{LOG}	70.45	**88.64**	70.45	61.36	77.27	77.27	**81.82**	65.91	68.18	68.18	70.45	70.45

Table 4. Performance of proposed and state-of-the-art frameworks for classification of tree structures.

Methodology		**Sensitivity (%)**	**Specificity (%)**	**Accuracy (%)**
Proposed methodology	*LOG_TRI_MVM_J48 variant*	86.36	90.91	88.64
Classification using sequence encoding and text mining [18]	*(i) Prüfer encoding & text mining*	86.11	84.33	85.16
	(ii) DF labeling & text mining	79.51	72.46	75.89
Classification using asymmetry index [14]	*(i) OFF labeling & text mining*	87.03	75.41	81.22
	(ii) Tree asymmetry	90.23	80.75	85.34
Classification using geometrical features [30]	*(i) Real AdaBoost*	87.36	69.67	76.34
	(ii) Gentle AdaBoost	82.73	67.36	74.45
	(iii) Modest AdaBoost	69.05	53.25	60.12

Fig. 1. Block diagram of the proposed framework for classification of tree structures.

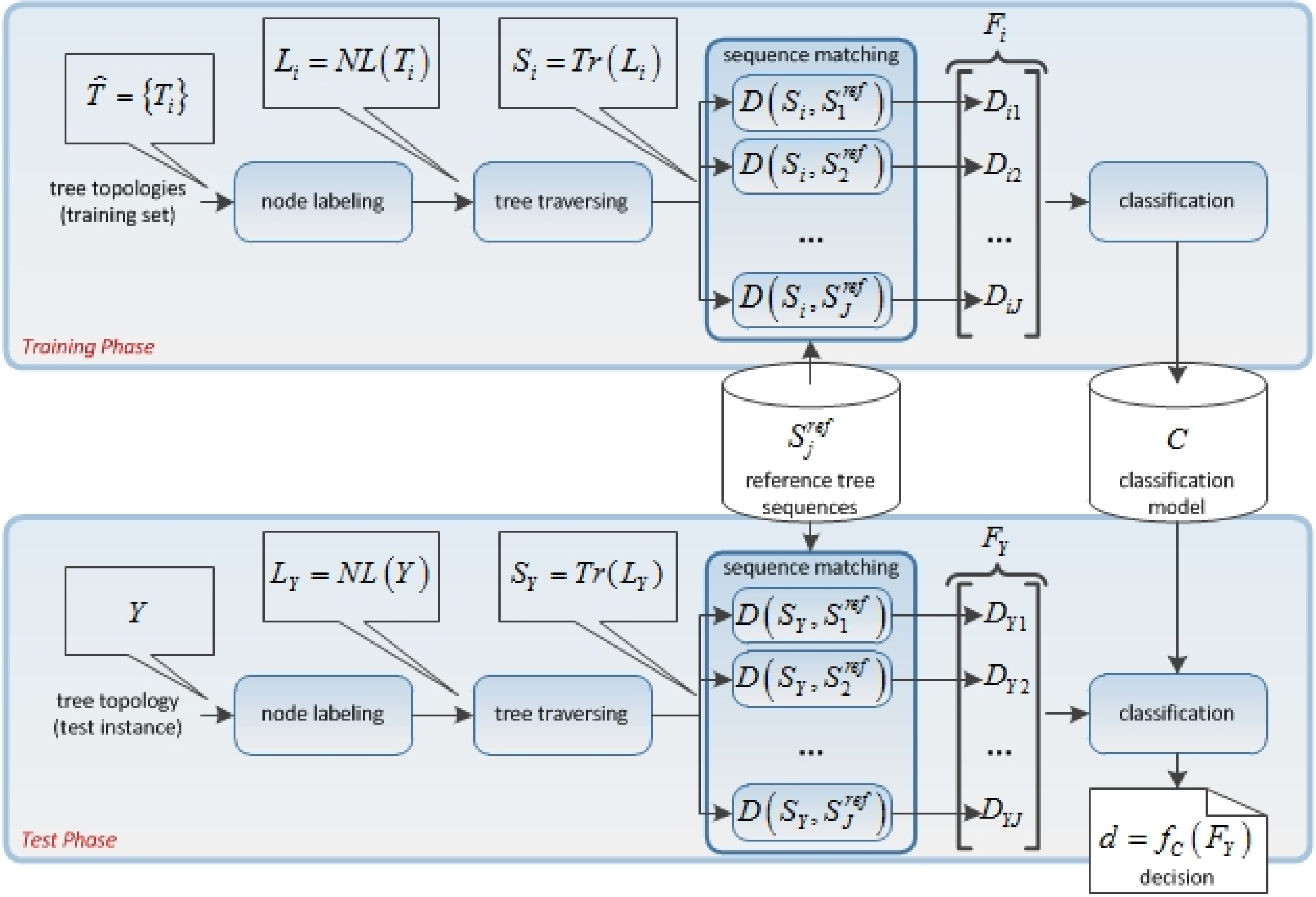


Fig. 2. Sequence representation, alignment and node alignment of the tree sequences T_1 and T_2 (colored in blue and red correspondingly). The NL_{OFF} labeling scheme and the T_{LO} traversal method were employed for tree representation (a), elastic matching was performed applying DTW and MVM techniques (b, c correspondingly). The aligned nodes for the two matching techniques (DTW and MVM) are colored likewise in (d) and (e) correspondingly.

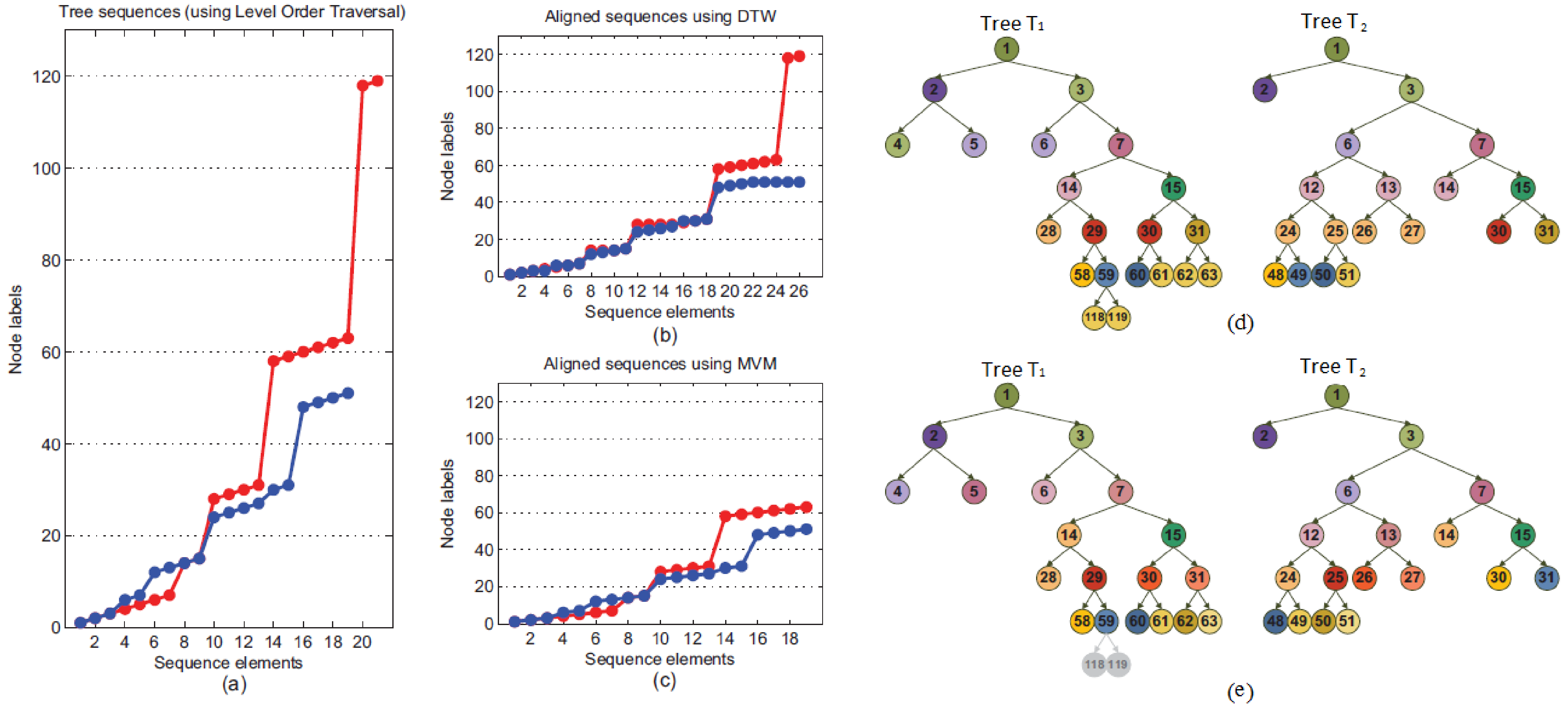


Fig. 3. Sequence representation, alignment and node alignment of the tree sequences T_1 and T_2 (colored in blue and red correspondingly). The NL_{OFF} labeling scheme and the T_{PO} traversal method were employed for tree representation (a), elastic matching was performed applying DTW and MVM techniques (b, c correspondingly). The aligned nodes for the two matching techniques (DTW and MVM) are colored likewise in (d) and (e) correspondingly.

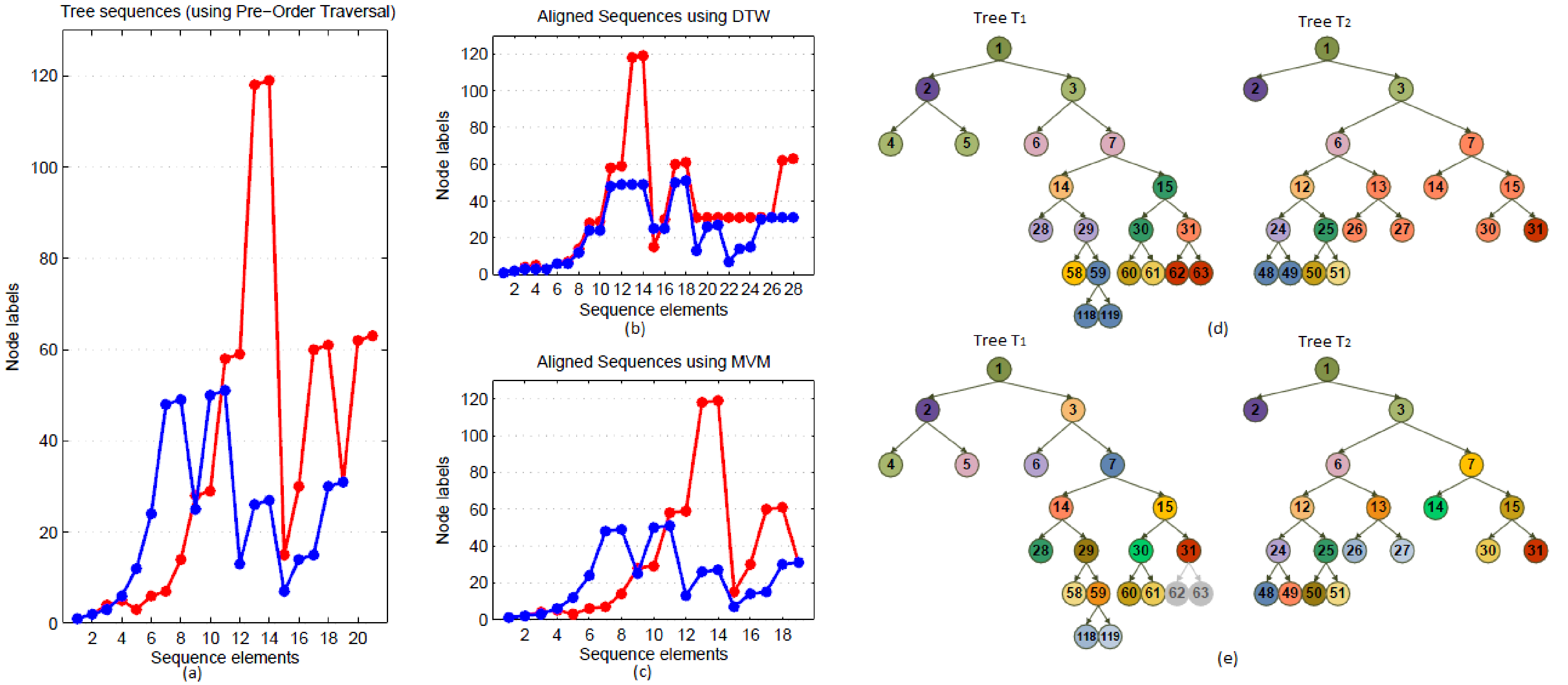


Fig. 4. Sequence representation, alignment and node alignment of the tree sequences T_1 and T_2 (colored in blue and red correspondingly). The NL_{LOG} labeling scheme and the T_{PO} traversal method were employed for tree representation (a), elastic matching was performed applying DTW and MVM techniques (b, c correspondingly). The aligned nodes for the two matching techniques (DTW and MVM) are colored likewise in (d) and (e) correspondingly.

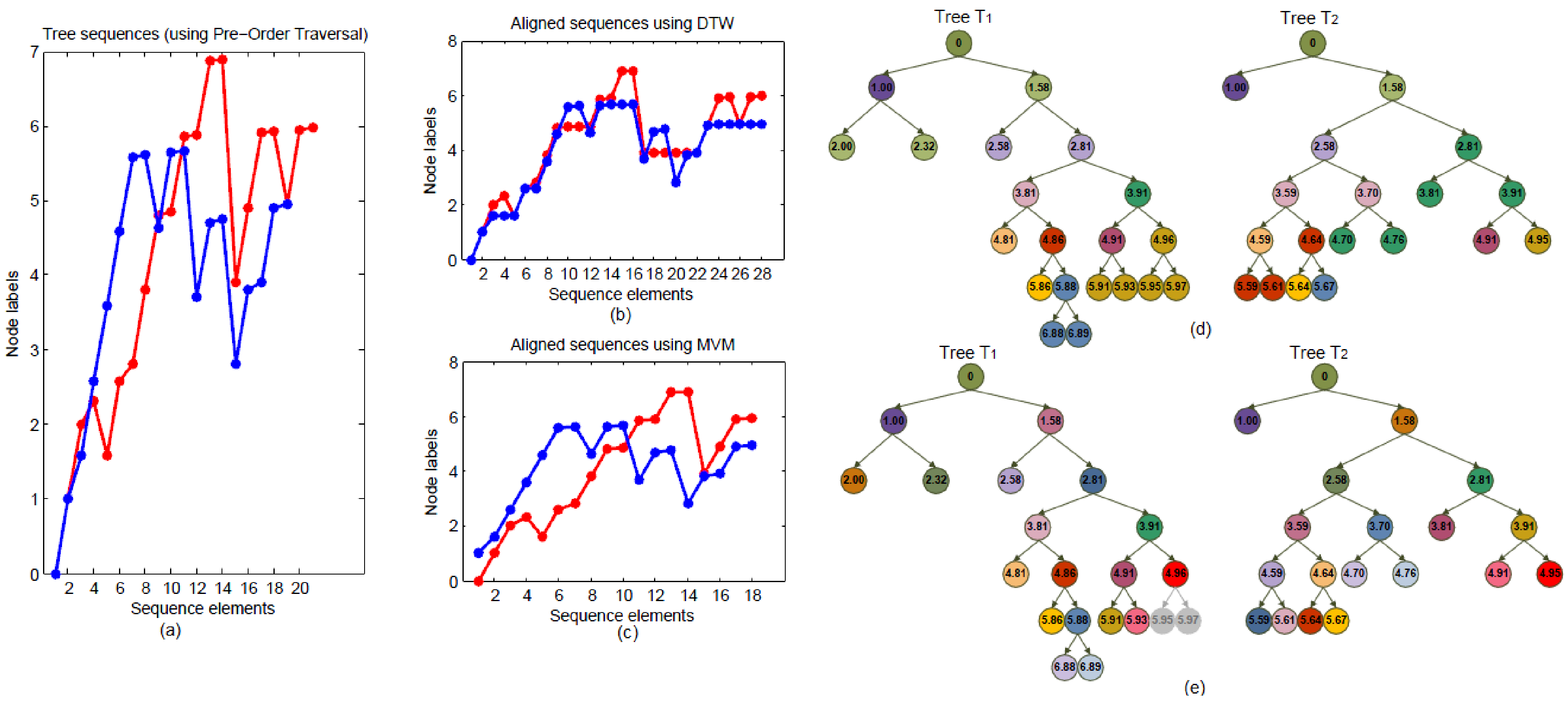


Fig. 5. Exploring the traversal mode TTR. Alignment of subtrees of the trees T1 and T2 using the TTR traversal mode, the NLOFF labeling and two different elastic matching methodologies; DTW (up) and MVM (down).

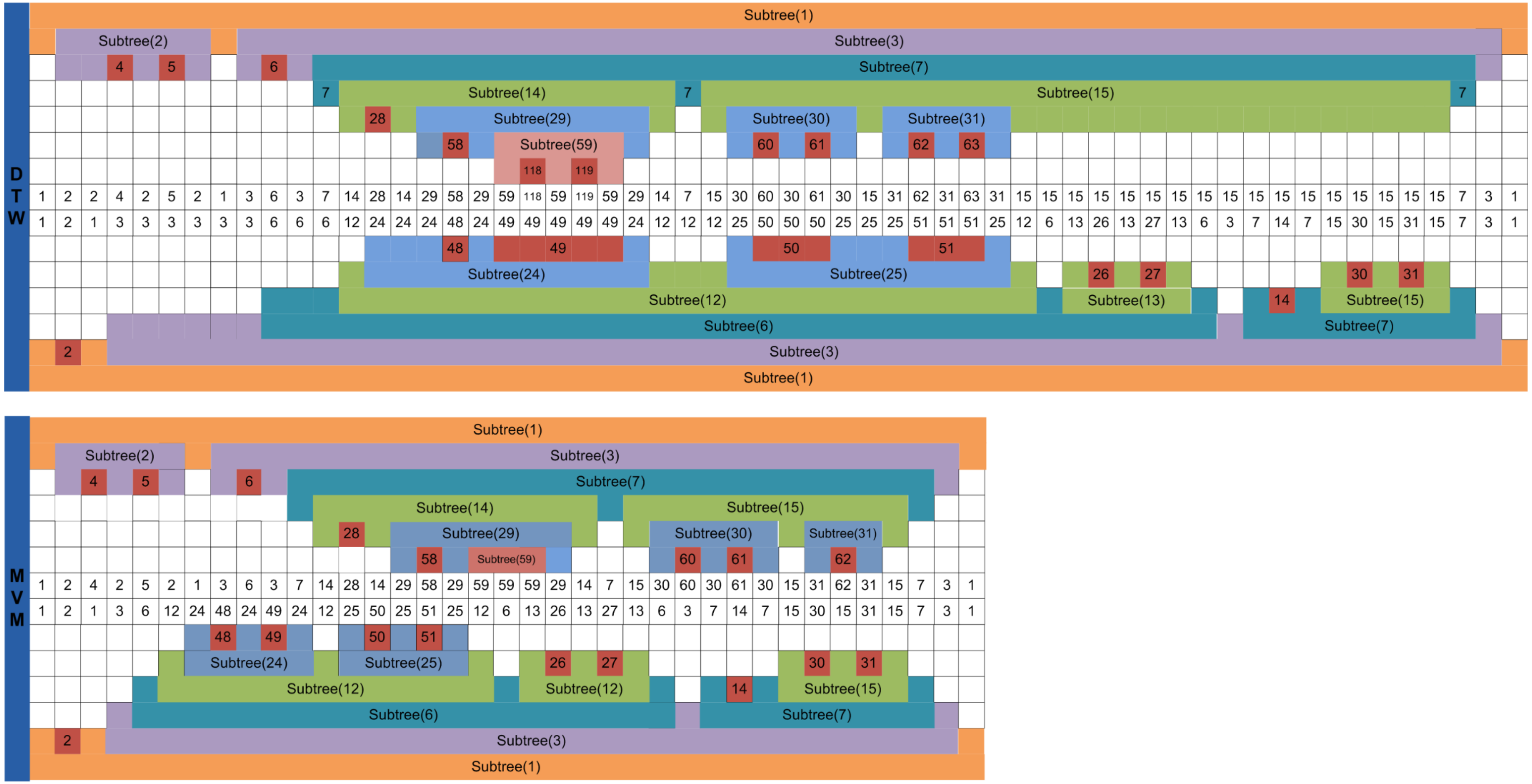
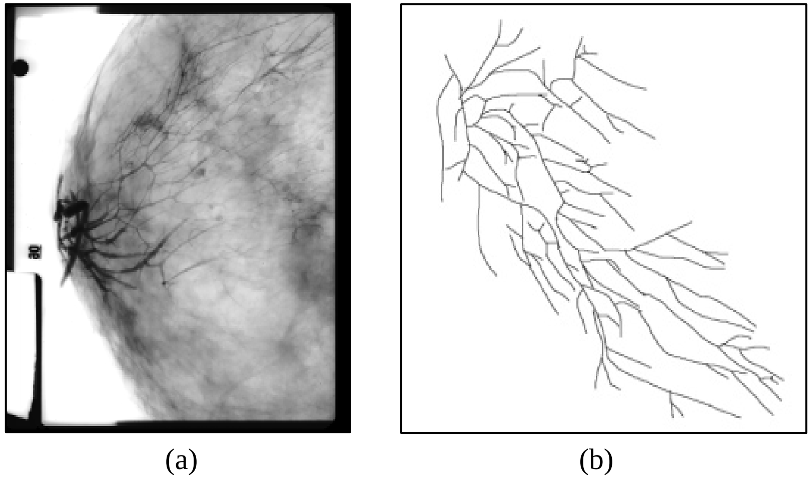


Fig. 6. (a) An original medical image of galactogram, (b) the magnified ductal tree segmented using manual tracing.



(a)

(b)