Clustering of Leaf-Labelled Trees on Free Leafset*

Jakub Koperwas and Krzysztof Walczak

Institute of Computer Science, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland
J.Koperwas@elka.pw.edu.pl, K.Walczak@ii.pw.edu.pl

Abstract. This paper focuses on the clustering of leaf-labelled trees on free leafset. It extends the previously proposed algorithms, designed for trees on the same leafset. The term z-equality is proposed and all the necessary consensus and distance notions are redefined with respect to z-equality. The clustering algorithms that focus on maximizing the quality measure for two representative trees are described, together with the measure itself. Finally, the promising results of experiments on tandem duplication trees are presented.

1 Introduction

This paper is a part of a larger work on applying data mining techniques to tree data - tree mining. Tree mining techniques have large applications in bioinformatics, image processing, text mining and others. This paper concentrates on clustering techniques for leaf-labelled trees, which have their main applications in the bioinformatics field. Previously in [1], we have presented techniques for clustering leaf-labelled trees, where all the trees where built on the same leafset. In this paper we enhance these methods so that they can be used for trees which do not contain exactly the same leafsets. We call them trees on a free leafset. In the first part of the paper we enhance the basic notions considering a tree representation, distance measure and consensus methods so that they are applicable to trees with free leafset. We introduce z-distance and z-consensus methods. The next section concentrates on the clustering of leaf-labelled trees with a free leafset. We show how to construct the algorithms for strict and majority rule consensus tree as a representative tree. We also discuss the quality measure used for assessing the clustering. Finally, we describe the results of clustering of tandem duplication trees, which are the leaf-labelled trees on a free leafset taken from bioinformatics field.

2 Basic Notions

2.1 Splits

One of the most popular leaf-labelled tree representations is the set of splits, which highlights the leaf-labelled trees interpretation as a space partition.

* The research has been partially supported by grant No 3 T11C 002 29 received from Polish Ministry of Education and Science.

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 736–745, 2007.

Definition 1 (Split). *Split $A|B$ (of a tree T with leafset L), corresponding to an edge e is a pair of leafsets A and B, which originated by splitting tree T into two disconnected trees whilst removing an edge e from a tree T, $A \cup B = L$.*

If $|A|$ or $|B|$ is equal to 1, the split is trivial. Split $A|B$ is a valid split if both sets A and B are non-empty. The splits of tree T_1 from Fig. 1 are: $a|bcde, b|acde, c|abde, d|abce, e|abcd, abe|cd, be|acd$; among them $abe|cd, be|acd$ are non-trivial splits.

Definition 2 (Split Equality). *Two splits $A|B$ and $C|D$ are considered equal iff ($A = C$ and $B = D$) or ($A = D$ and $B = C$).*

The trees with free leafset cannot be compared easily if they are not built on the same leafset. In particular, the conventional distance or consensus techniques cannot be used, because splits, built on a different leafset cannot be equal. On the other hand, there is a need to compare such trees to determine whether they share common information or not. We present therefore, the restricted equality as an efficient and well-interpretable method of comparing two trees on free leafset.

Definition 3 (Restricted Split). *Split s_1 is a restricted version of split s_2 on the leafset z if it is built with removing leafs not in z from s_2: $s_2^z = s_1$.*

Split restriction is a complementary term to the term restricted tree described in [2]. It can be shown that the restricted tree of a tree T is built of restricted splits of a tree T on the same set z.

Definition 4 (Restricted Split Equality(z-equality)). *Splits s_1 and s_2 are restrictedly equal on the leafset z, if their restricted versions on the leafset z are equal: $s_1 =^z s_2 \iff s_1^z = s_2^z$.*

For example: $abc|def$ and $fabc|deg$ are restrictedly equal on the leafset $abcde$, because their corresponding restricted splits: $abc|de$ and $abc|de$ are equal, however they are not equal on a leafset $abcdef$ because their corresponding restricted splits: $abc|def$ and $fabc|de$ are not equal.

Definition 5 (Split Coherence). *Splits s_1 and s_2 are coherent if they are z-equal on the leafset z that is an intersection of their leafsets*
$s_1 \sim s_2 \iff s_1 =^z s_2 \wedge z = L(s_1) \cap L(s_2)$.

Z-equality/coherence relations as opposed to normal split equality relations do not determine whether two splits carry the same information but whether two splits do not contain contradictory information with respect to given leafset. For example $abc|def$ and $fabc|deg$ are not equal but they are restrictedly equal on the leafset $abcde$, which means that set of leaves $abcde$ in both splits is divided identically. Both the z-equality and the coherence are the equivalence relations.

2.2 Distance Between Leaf-Labelled Trees

One of the most popular distances for leaf labelled trees is a Robinson-Foulds distance. R-F distance between two trees T_1 and T_2 with set of splits S_1 and S_2 respectively is defined as follows:

$$d_{R-F}(T_1, T_2) = |S_1 \cup S_2| - |S_1 \cap S_2|. \tag{1}$$

For the reasons described earlier, classic R-F distance will not work if even one leaf is not present in both of the compared trees. Therefore, we extend the R-F distance with respect to leaf-labelled trees on free leafset.

Definition 6 (z-distance). *The z-distance for a given z is number of splits that are not z-equal on some leafset z.*

$$\begin{gathered} d_z(T_1, T_2) = |S_1 \div^z S_2| = |S_1 \cup^z S_2| - |S_1 \cap^z S_2|, \\ , where \\ S_1 \cup^z S_2 = \{s : (r \in S_1 \vee r \in S_2) \wedge (s = r^z)\}, \\ S_1 \cap^z S_2 = \{s : (r \in S_1 \wedge r \in S_2) \wedge (s = r^z)\}. \end{gathered} \tag{2}$$

Let us consider trees form Fig. 1 as an example. They are built on the following splits:
$T_1 : a|bcde, b|acde, c|abde, d|abce, e|abcd, abe|cd, be|acd.$
$T_2 : a|bcdef,\ b|acdef,\ c|abdef,\ d|abcef,\ e|abcdf,\ f|abcde,\ ab|cdef,\ ef|abcd,\ def|abcd.$
The z-distance, where $z = abcd$, is counted as follows:
The restricted splits are the following:
$T_1 : a|bcd, b|acd, c|abd, d|abc, ab|cd.$ $T_2 : a|bcd, b|acd, c|abd, d|abc, ab|cd.$
Therefore the z-distance on set *abcd* equals 0.
Z-distance on set *abcde* is equal to 4 the same as for set *abcdexy*.
It may seem more natural to count the distance for two trees where z contains common leaves of compared trees i.e. with respect to coherence relation rather than z-equality. However, the distance defined in this way could not meet triangle inequality, therefore it is not a metrics. There are more possible ways to define the distance between leaf labelled trees on free leafset. However the z-distance is both efficient and has a good interpretation. The value of z-distance for two trees indicates the amount of contradictory information in those trees, with respect to a given leafset. For an interpretation in phylogenetic analysis we may imagine that we have two species trees that share common taxa a, b, c, d among others, that are not shared. Counting z-distance on *abcd*, we want to check how much the information about relations of these particular taxa differ in given trees. Z-distance is a natural extension of R-F distance, because for trees with the same leafset it will give the same result.

2.3 Consensus Methods Extensions for Free Leafset

Consensus methods in phylogenetic analysis are used to extract common information from set of trees and represent it as a single tree. The most popular are a

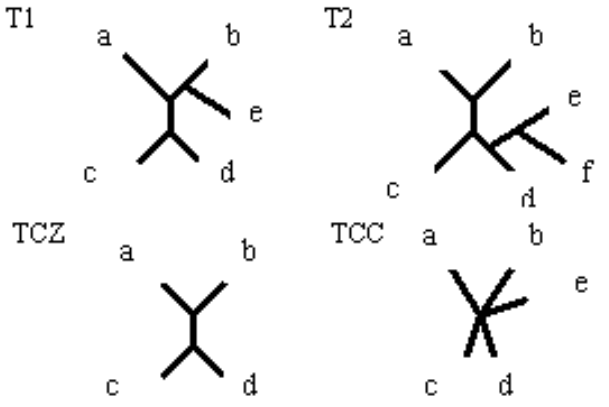

Fig. 1. Two leaf-labelled trees on free leafset and their z-restricted on a leafset *abcd* and common-restricted strict consensus trees

strict consensus tree and a majority rule consensus tree. Strict consensus tree is built of splits that occur in all of the input trees. Majority-rule consensus tree is built of splits that occur in the majority of the input trees. Consensus methods used for trees with free leafset will result in empty consensus tree split-set (always for strict and often for a majority-rule). Therefore we extend these terms with respect to restricted splits.

Definition 7 (z-restricted Strict Consensus Tree). *For a profile of trees $T_1, \ldots, T_n$ z-restricted strict consensus tree is built of valid splits s such that s is restrictedly equal on z to at least one split in each tree, in other words, split s is a restricted version of at least one split in each tree on leafset z.*

$$T_{zc}(T_1, \ldots, T_m) : S_{zc} = \left(\bigcap_{i=1}^{m} \right)^{z} S_i. \tag{3}$$

Definition 8 (Common-restricted Strict Consensus Tree). *Common-restricted consensus tree is a z-restricted consensus tree where z is an intersection of all corresponding leafsets $L_1, \ldots, L_n$.*

In order to construct z-restricted or common-restricted tree, we restrict all splits to a leafset z, and count classic consensus tree. For trees from the Fig. 1, the z-restricted strict consensus tree on a leafset *abcd* contains $a|bcd, b|acd, c|abd, d|abc$, $ab|cd$ (see Fig. 1 - TCZ) and the common-restricted strict consensus tree consists of: $a|bcde, b|acde, c|abd, d|abceande|abcd$ (see Fig. 1 - TCC)

Property 1. For any given set of trees $T_1, \ldots, T_m$ and a set z.

$$T_{zc}(T_1, \ldots, T_m) = T_{zc}(T_1, T_{zc}(T_2, \ldots, T_m)), \tag{4}$$

where T_{zc} is z-restricted strict consensus tree on leafset z.

Proof.

$$T_{zc}(T_1,\ldots,T_m) : S_{zc} = (\bigcap_{i=1}^{m})^z S_i = \bigcap_{i=1}^{m} S_i^z,$$
$$T_{zc}(T_1, T_{zc}(T_2,\ldots,T_m)) : S_{zc} = (\bigcap_{i=2}^{m} S_i^z) \cap S_1^z = \bigcap_{i=1}^{m} S_i^z. \quad (5)$$

Definition 9 (z-restricted Majority-rule Consensus Tree). *For a profile of trees $T_1,\ldots,T_n$, z-restricted majority-rule consensus tree is built of valid splits s such that s is restrictedly equal on z to some split, from the majority of trees.*

Definition 10 (Common-restricted Majority-rule Consensus Tree). *Common-restricted majority-rule consensus tree is a z-restricted consensus tree, where z is an intersection of all corresponding leafsets $L_1 \ldots L_n$ of the whole profile.*

In the Fig. 2 and Fig. 3 there are examples on z-restricted on abcdef and common-restricted majority rule consensus trees.

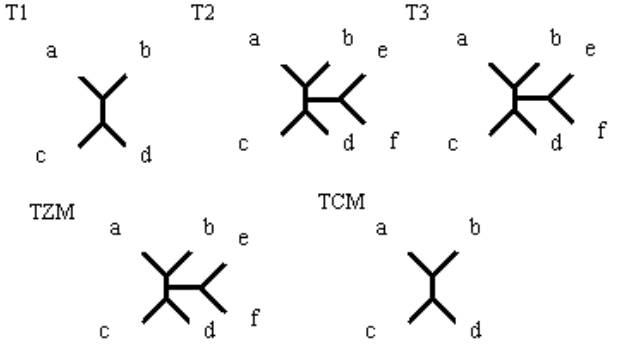

Fig. 2. Profile of trees together with their z-restricted on abcdef and common-restricted majority- rule consensus trees

The examples from Fig. 2 and Fig. 3 show that choosing a set z is not obvious. If an intersection of leaves is used, sometimes the tree may lose some interesting information, like in example from Fig. 2, however taking a larger leafset may bring totally uninformative tree like in example from Fig. 3. Finding most informative z-restricted majority-rule consensus tree is another interesting task for future considerations.

Property 2. For any given set of trees z-restricted majority-rule consensus tree is a middle tree with respect to z-distance i.e. it minimizes the sum of z-distances between itself and all the trees. (Theorem 1 is a proof of this property)

Lemma 1. *For any set of trees $T_1,\ldots,T_m$ on the same leafset if T_M is a majority-rule consensus tree then*

$$T_M : \min \sum_{i=1}^{m} d(T_i, T_M) \quad (6)$$

(this was proved by [3]).

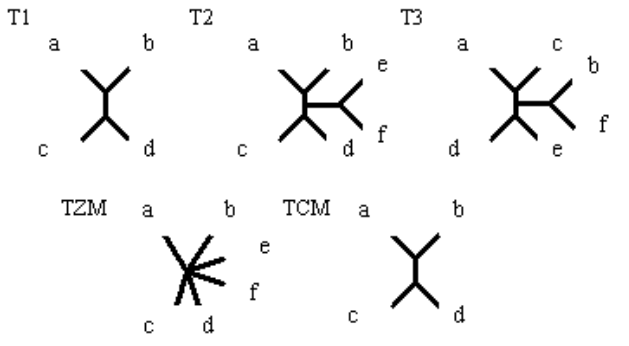

Fig. 3. Profile of trees together with their z-restricted on abcdef and common-restricted majority- rule consensus trees

Theorem 1.

$$T_{Mz} : \min \sum_{i=1}^{m} d^z(T_i, T_{Mz}). \tag{7}$$

Proof.

$$\begin{aligned} T_M : \min \textstyle\sum_{i=1}^{m} d(T_i, T_M) &\Rightarrow T^z_{Mz} : \min \textstyle\sum_{i=1}^{m} d(T^z_i, T^z_{Mz}) \\ &\Rightarrow T^z_{Mz} : \min \textstyle\sum_{i=1}^{m} d^z(T_i, T_{Mz}), \\ &\text{because } d^z(T_i, T_{Mz}) = d(T^z_i, T^z_{Mz}) \\ &\Rightarrow T_{Mz} : \min \textstyle\sum_{i=1}^{m} d^z(T_i, T_{Mz}), \\ &\text{because } T_{Mz} = T^z_{Mz}. \end{aligned} \tag{8}$$

Consensus methods presented above are suitable for representing common information in leaf-labelled trees on free leafset.

3 Clustering of Leaf-Labelled Trees on Free Leaf-Set

The aim of our clustering techniques is to divide trees into k groups in such a way the clustering is possibly the best towards our quality measure.

3.1 Quality Measure

The quality measure is based on the informativity of the representative trees of each cluster. The representative may be any predefined tree that shares common knowledge of all the trees, it can possibly be strict consensus tree, majority-rule consensus tree or other. The representative tree shall only contain the knowledge present in input trees but nothing more. We can state that $S_R \subseteq \bigcup_{i \in C} S^z_i$, which is again the free-leafset extension of what was proposed in [1]. Here we focus on z-restricted (also common-restricted as a special case) strict consensus tree and majority-rule consensus tree, because these trees can be efficiently counted with simple algorithms. The quality is counted as follows:

1. select the k representative trees, one for each cluster
2. count how much information is lost when replacing the whole dataset of trees with k representative trees - this is information loss
3. count how much information is lost when replacing the whole dataset with a single representative tree - this is one-cluster information loss
4. count information gain as follows:

$$IG = \frac{\Delta I_{C_0} - \Delta I}{\Delta I_{C_0}}, \tag{9}$$

which shows how much our clustering is better from no clustering.

The informativity of a tree is simply the amount of non-trivial splits contained by tree [4], therefore information loss for a cluster is counted with formula:

$$\Delta I_{C_x} = \sum_{i=1}^{l} |S_i \div^z S_R|. \tag{10}$$

For further information on informativity and information gain please refer to [1].

3.2 Clustering of Leaf-Labelled Trees with Free Leafset with a z-Restricted Strict Consensus Tree as a Representative Tree

The aim of this clustering is to divide trees into k groups in such a way that information gain towards the z-restricted strict consensus tree is maximal. For this purpose we choose an agglomerative clustering algorithm, but we replace common merging strategies min, max and complete linkage with our own: minimum information loss linkage (agg-inf). We choose such two clusters to merge that merging minimizes the information loss of the clustering after the merging.

$$\arg\min_{C_x, C_y} \Delta I' - \Delta I. \tag{11}$$

This way it automatically maximizes the information gain in each step. Fortunately, while selecting the clusters to merge, we do not need to count complete information loss for all possible mergings. It is enough to count the components of the two candidate clusters (x, y) and one resulting cluster (z). So the merging condition:

$$\arg\min_{C_x, C_y} \Delta I_z - (\Delta I_x + \Delta I_y). \tag{12}$$

Due to this property and Property 1 we can construct the algorithm that is very efficient. In such an algorithm, the clusters in each step are represented only with their consensus trees and the amount of trees assigned. Moreover, the information loss in each step is not completely counted, because the minimum loss linkage in each step can be determined on the basis of consensus trees informativity in the previous step. It can be shown, (which we omit due to lack

of space), that for a agg-inf clustering for a given z used for z-distance and z-restricted strict consensus tree

$$\Delta I' - \Delta I = l_x * (|S_{C_x}| - |S_{C_x} \cap^z S_{C_y}|) + l_y * (|S_{C_y}| - |S_{C_x} \cap^z S_{C_y}|), \quad (13)$$

where l_x and l_y are the amount of trees in clusters candidate for merge and $|S_c|$ is the amount of splits in corresponding consensus trees.

3.3 Clustering of Leaf-Labelled Trees with Free Leafset with z-Restricted Majority Rule Consensus Tree as a Representative Tree

The aim of this clustering is to divide the trees into k groups in such a way that the information gain towards the z-restricted majority-rule consensus tree is maximal. For this purpose we choose k-mean clustering algorithm. Because of Property 2, which states that majority-rule consensus tree is a middle tree, we can use it as a centroid in k-mean algorithm whose objective function will be automatically identical to ours because its objective function is as follows:

$$\min_{C,\{T_{M_k}\}_{k=1}^K} \sum_{k=1}^{K} \sum_{C(i)=k} d(T_i, T_{M_k}). \quad (14)$$

3.4 Z Parameter Selection

The main problem of this approach is the selection of set z. We may think of an application, for example from phylogenetic analysis, where particular taxa let's say a, b, c, d are of a special interest. In this case, the quite obvious thing is to choose a set z as *abcd*. On the other hand, we may also think of such a clustering where no particular taxa is preferred. For a phylogenetic or duplication trees, where all the clustered trees share most but not all leaves, we may choose z as an intersection of leaves. However, when the input data contains a weakly connected set of leaves such an approach will not bring any reasonable results. There is a need to provide a distance measure that does not require arbitrary z selection, for example based on coherence relation. Construction of a middle tree for such distance is required as well. We intend to do it in future studies.

4 Results

Below we describe the results of clustering tandem duplication trees, which are the leaf-labelled trees on free leafset taken from a bioinformatics field. Tandem duplication is a DNA sequence built of the adjacent copies of a pattern. The adjacent copies are not exactly the same as they diverged over time, due to point mutations. Tandem duplications are thought to be a result of events based on the duplication of one or more already existent copies. Tandem duplication process can be illustrated as a leaf-labelled tree where the labels on leaves correspond to

Table 1. Quality of clustering with various algorithms

k	Agg-inf	Agg-min	Agg-max	Agg-compl	K-mean
10	0.83	0.49	0.73	0.73	0.85
9	0.76	0.41	0.73	0.73	0.65
8	0.68	0.32	0.62	0.62	0.68
7	0.61	0.25	0.54	0.54	0.60
6	0.51	0.19	0.50	0.50	0.53
5	0.45	0.11	0.29	0.37	0.49

Table 2. Sample clustering results

k	Agg-inf	Agg-min	Agg-max	K-mean
5-12	48(1.0):81(2.0): 80(2.0):78(1.0): 93(1.0): 77(1.0): 98(1.0):69(2.0)	423(0.0): 39(2.0): 46(2.0): 8(2.0): 34(2.0): 42(2.0): 21(2.0): 11(2.0)	69(1.0): 189(1.0): 134(1.0): 11(2.0): 42(2.0): 77(1.0): 46(2.0): 56(2.0)	202(0.0): 15(2.0): 61(1.0): 23(2.0): 6(6.0): 17(3.0): 11(6.0): 16(2.0)
9-12	40(1.0):170(0.0): 41(1.0):32(1.0): 15(2.0):7(4.0): 14(2.0): 32(1.0)	343(0.0):1(6.0): 1(6.0):1(6.0): 1(6.0):1(6.0): 2(4.0): 1(6.0)	277(0.0):17(1.0): 15(2.0):12(2.0): 8(1.0):7(1.0): 5(2.0): 10(1.0)	158(0.0):54(1.0): 37(1.0):18(2.0): 15(2.0):4(6.0): 21(3.0): 44(1.0)

the position of a given copy in a sequence. There are techniques that are able to reconstruct such a tree, basing on a sequence, especially the differences between the copies [5]. In general cases such trees are unrooted due to problems with estimating time on the basis of those differences. Here we have performed experiments on tandem duplication trees which were reconstructed with DTScore algorithm [5]. The sequences were retrieved from Tandem Repeats Database [6]. We have examined trees that contained from 5 up to 12 copies due to efficiency barriers considering trees reconstruction. The selection of z was natural as an intersection of leafsets of examined trees. So when examining for example trees consisting of 9-12 copies at a time, a leafset containing $1, 2, 3, 4, 5, 6, 7, 8, 9$ is chosen. As a sample of results we present the agg-inf algorithm as opposed to standard min, max and complete linkage clustering for strict consensus and k-mean algorithm for majority-rule consensus. The input trees were pre-processed by removing duplicating trees for more reliable results. In the Table 1 we show the results of clustering the trees with 5-12 copies, for a different number of clusters (k). Because of the large possible number of different trees the clustering results is only reliable for at least 5 groups. In the Table 2 the sample clustering results for 8 groups are presented, where trees with 5-12 and 9-12 leaves were tested. The results are presented in format: 203(0.0):179(1.0): 69(2.0): 80(2.0): 93(1.0) which indicates how many trees were assigned to the following groups: 203,179,69,93 and what was the informativity (numer of non-trivial splits) of a representative tree -(value in brackets). In all cases, the agg-inf strategy was better then others, even up to 75%. For experiments with smaller range of copies, the informativity of representative trees was significantly higher.

5 Discussion

In this paper we have presented the methodology aimed at the clustering leaf-labelled trees on a free leafset. Although we perform experiments for tandem duplication data, our approach is described in general terms. In the future there will be a need to construct a better distance measure that does not require arbitrary z selection and allows more accurate clustering. A middle tree for such a distance is also required. There is also a need to test the proposed methods for trees from other disciplines.

References

1. Koperwas, J., Walczak, K.: Clustering of leaf labeled-trees. In: ICANNGA 2007. Part I, LNCS, vol. 4431, pp. 702–710. Springer, Heidelberg (2007)
2. Ganeshkumar, G., Warnow, T.: Finding a maximum compatible tree for a bounded number of trees with bounded degree is solvable in polynomial time. In: Gascuel, O., Moret, B.M.E. (eds.) Algorithms in Bioinformatics. LNCS, vol. 2149, pp. 156–163. Springer, Heidelberg (2001)
3. Barthelemy, J.P., McMorris, F.R.: The median procedure for n-trees. J. Classif. 3, 329–334 (1986)
4. Bryant, D.: Building trees, hunting for trees, and comparing trees. Theory And Method. In: Phylogenetic Analysis. Ph.D Thesis University of Canterbury (1997)
5. Elemento, O. et al.: Reconstructing the duplication history of tandemly repeated genes. Molecular Biology and Evolution 19, 278–288 (2002)
6. Tandem Repeats Database, `http://tandem.bu.edu`
7. Stockham, C., Wang, L.S., Warnow, T.: Statistically based postprocessing of phylogenetic analysis by clustering. Bionformatics 18, 285–293 (2002)
8. Amenta, N., Klingner, J.: Case study: Visualizing sets of evolutionary trees. 8th IEEE Symposium on Information Visualization, pp. 71–74 (2002)
9. Akutsu, T., Halldrsson, M.: On the approximation of largest common point sets and largest common subtrees. Unpublished manuscript (1997)
10. Gascuel, O. et al.: The combinatorics of tandem duplication trees. Systematic Biology 52(1), 110–118 (2003)
11. Bille, P.: Tree edit distance, alignment distance and inclusion. Technical report TR-2003-23 in IT University Technical Report Series (2003)