Advanced Studies in Biology, Vol. 4, 2012, no. 2, 77 – 84

Properties of the Triset Metric for Phylogenetic Trees

Pawel Noga

Gdansk University of Technology
Faculty of Electronics, Telecommunication and Informatics
ul. Gabriela Narutowicza 11/12, 80-233 Gdansk, Poland
pawel.noga@eti.pg.gda.pl

Abstract

The following paper presents a new polynomial time metric for unrooted phylogenetic trees (based on weighted bipartite graphs and the method of determining a minimum perfect matching) and its properties. Also many its properties are presented.

Keywords: phylogenetic trees, metric, triset

1 Introduction

Demand for fast (polynomial) metrics for phylogenetic trees is enormous. Since we can not identify an perfect tree representing the evolution of given organisms, we use consensus trees, which are the result of comparison and averaging a large set of trees generated by algorithms that create phylogenetic trees [2]. Depending on the used metric [3] as a result we can get a variety of consensus trees. This diversity is one of the ways used by biologists to obtain a more accurate tree of evolution. Therefore, it is important to create new metrics of various properties.

Recently new method of creation metrics was presented by Damian Bogdanowicz [1]. His perfect-matching method allow us to create many metrics with different properties. Presented in this paper triset metric is one of them and allow us to compare unrooted phylogenetic trees.

2 Definitions

Phylogenetic tree T built on a non-empty set of organisms Φ of size n will be a graph G, where:

a) leaves (vertices of degree *1*) each has one assigned one organism from the set Φ. Will be marked with lowercase letters ($a, b, c, d... \in \Phi$);
b) root (vertex of degree *2*), occurs only in the rooted tree;
c) nodes (vertices of degree *3*, and root), mean symbolic distinction of two different species. In this paper they will be numbers.

Going from a rooted tree by removing the root (and connecting the edges outgoing from it into one) clearly sets the unrooted tree. The unambiguous transition in the opposite direction (by inserting the root on one edge) is not possible, because the tree can create root on any of the *n-1* edges.

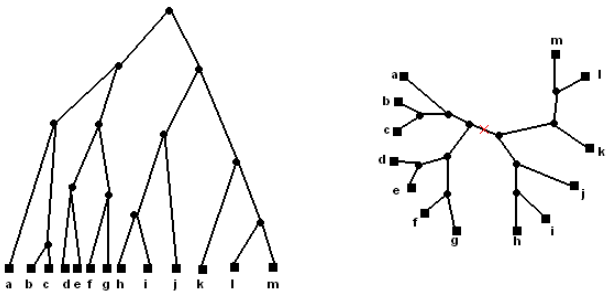

Fig.1. On the left tree rooted tree created from unrooted one (right) by adding the root of the selected edge.

The practical application of the unrooted tree is that, it can be used for the representation of the evolution of organisms that are very similar (eg, species of dogs, and orchids). In such cases, the information when diversity occurred is much less important than the information about similarities of species.

3 Triset Form

For a unrooted tree T we set following description of all nodes::
$T_x = \{A|B|C\}$ – this figure we will called **triset form** (triset)
T_x – a node x in the tree T
A, B, C – disjoint subsets, each containing a organisms list of one of the three consistent trees that will be created by removing a node T_x
Is also accepted that, for fixed sets A, B and C all permutations of a triset form are equivalent and indistinguishable:

$$\{A|B|C\} \equiv \{B|C|A\} \equiv \{B|A|C\} \equiv \{C|A|B\} \equiv \{C|B|A\} \equiv \{A|C|B\}$$

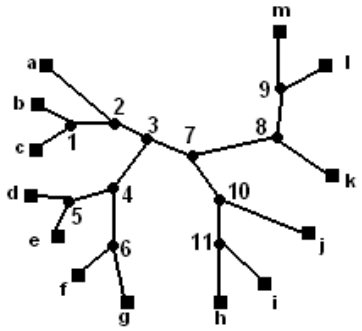

Fig.2. Unrooted phylogenetic tree. Individuals (organisms) marked with letters, nodes with numbers.

The node definition (vertex degree 3) shows that every set in triset contains at least one element. In any given tree, for each node triset is different, and always includes all organisms.

Sample trisets for the tree of Figure 2.

$T_1 = \{b \mid c \mid a, d, e, f, g, h, u, j, k, l, m\}$

$T_4 = \{a, b, c, h, i, j, k, l, m \mid d, e \mid f, g\}$

$T_7 = \{a, b, c, d, e, f, g \mid h, i, j \mid k, l, m\}$

Lemma 1

For two adjacent nodes, x, y their trisets look as follows:

$T_x = \{A|B|C\}$ $T_y = \{A \cup B|C'|C''\}$, where $C = C' \cup C''$

Lemma 2

If two nodes x and y are adjacent to a z, and $T_x=\{A|B|C\}$ $T_y=\{D|E|F\}$ such that:

a) $C \cap F \neq \varnothing$,

b) A, B, D and E are disjoint, then:

$T_z = \{A \cup B|D \cup E|C \cap F\}$

Proof:

From the first lemma, we have z that is a neighbor of x, so should be in the form $T_z = \{A \cup B|C'|C''\}$, while z also a neighbor of y, so should be in the form $T_z = \{D \cup E|F'|F''\}$. Disjoint of A, B, D, E shows that the triset should be in the form $T_z = \{A \cup B|D \cup E|G\}$.

Because $A \cup B \cup C = D \cup E \cup F$, and A, B, D, E are disjoint, it follows that $G = C \cap F$. So triset has a form $T_z = \{A \cup B|D \cup E|C \cap F\}$.

Based on these two lemmas we can quickly (linear time) compute trisets for all nodes in the tree.

4 Operation >>>

Let take $T_j = \{A, B, C\}$ and $T_k = \{A', B', C'\}$. Define the operation of $\boldsymbol{T_j >>> T_k}$ as the smallest number of transformations (actions) needed to get from triset T_j to triset T_k. By action we understand:
- the swap of any two elements between any two sets of A, B or C,
- the removal of one element for any set A, B, C and adding it to other set.

Properties of operation >>>

$|T_i| = |A| + |B| + |C|$, where $T_i = \{A|B|C\}$
1. $T_i >>> T_i = 0$
2. $\forall\ j, k \quad T_j >>> T_k = T_k >>> T_j$
3. $\forall\ i, j, k \quad (T_i >>> T_j) + (T_j >>> T_k) \geq (T_i >>> T_k)$

This operation can be done in polynomial time (depending on the number of organisms): create a complete bipartite graph, where the vertexes on the one side are organisms of one of triset, on the other side – are organisms of the other triset. Edge weight is the number of operations needed to convert sets from one into another. The solution is a weight of the minimum perfect matching of the graph.

5 Algorithm of comparison of two trees

Input*:* two unrooted phylogenetic trees constructed on the same set of organisms.
1. for each node in each tree determinate it triset form,
2. create complete bipartite graph, where the vertexes of the one side are nodes of one tree, at the other side vertexes of the second tree [3]. Compute weights of all edges as follows:

2.1. for each edge xy, where $T_X = \{A,B,C\}$ i $T_Y = \{A',B',C'\}$
2.2. weight(xy) = $T_X >>> T_Y$
3. compute the minimal weight edge cover for bipartite graph,
4. give sum of edge weights for this cover (metric value).

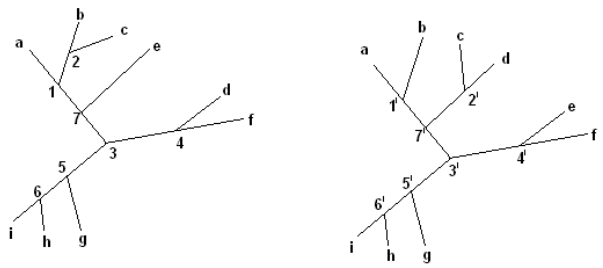

Fig.3. Examples of two trees, for which we want to calculate the metric

Example:
We have two similar phylogenetic trees. First we determinate triset form for each node:

1 {a|b,c|d,e,f,g,h,i}
2 {b|c|a,d,e,f,g,h,i}
3 {a,b,c,e|d,f|g,h,i}
4 {d|f|a,b,c,e,g,h,i}
5 {i,h|g|a,b,c,d,e,f}
6 {i|h|a,b,c,d,e,f,g}
7 {a,b,c|e|d,f,g,h,i}

1' {a|b|c,d,e,f,g,h,i}
2' {d|c|a,b,e,f,g,h,i}
3' {a,b,c,d|e,f|g,h,i}
4' {e|f|a,b,c,d,g,h,i}
5' {i,h|g|a,b,c,d,e,f}
6' {i|h|a,b,c,d,e,f,g}
7' {a,b|c,d|e,f,g,h,i}

	1	2	3	4	5	6	7
1'	1	1	5	2	3	2	2
2'	2	1	4	1	3	2	3
3'	5	5	1	4	3	3	2
4'	3	2	4	2	3	2	3
5'	3	3	3	3	0	2	4
6'	3	2	4	2	2	0	4
7'	2	2	3	3	4	4	2

Fig.4. Bipartite tree with weights which are the result of the operation >>>
Bipartite tree created during step 2 of an algorithm is shown in Figure 4. As we can see perfect matching for this graph is 1-1, 2-2, 3-3, 4-4, 5-5, 6-6 and its weight is equal 7.

Proof that it is a metric.
Denote by $M(T,T')$ the value returned by the above algorithm for given trees T, T':

(i) $\forall T \quad M(T,T) = 0$

As the two trees are identical, the corresponding nodes are connected in a bipartite tree edges with a weight of 0 (no change in the triset form for corresponding nodes), these edges are also the smallest perfect matching of our graph, whose value is equal to 0.

(ii) $\forall T, T' \quad M(T,T') = M(T',T)$

The algorithm does not favor any of the trees, the sequence of their computing does not change anything in the shape of nor the values of the bipartite tree and thus does not change the minimum edge cover, neither the value of the metric.

(iii) $\forall T, T' \quad M(T,T') \geq 0$

The definition of >>> shows that the weight on the edges of bipartite graphs have values ≥ 0. The value of the minimum edge cover will also be ≥ 0.

(iv) $\forall T, T', T'' \quad M(T,T'') \leq M(T,T') + M(T',T'')$

Lets take trees T and T' now we can take pairs of nodes from perfect matching (directly as a result of algorithm). Denote those pairs as (T_i, T_i'). Take trees T' and

T” and pairs of nodes from perfect matching. Denote those pairs as (T_i', T_i”). $i = 1, 2 \ldots n$ (where n is number of nodes in tree)
$M(T,T') =$ sum of all $T_i >>> T_i'$
$M(T',T'') =$ sum of all $T_i' >>> T_i''$
$M(T,T'') =$ sum of all $T_i >>> T_i''$
From properties of >>> operation we have
$\forall\ i, j, k \quad (T_i >>> T_j) + (T_j >>> T_k) \geq (T_i >>> T_k)$
So $M(T,T'') \leq M(T,T') + M(T',T'')$.

6. Triset Metric Properties

6.1. Minimum non zero value of triset metric is 2

If we swap two leafs that have common node – new tree will be exactly the same, so we can not get 1 as a minimum value (only one node is different in trees).
If we choose two leafs that between them are two nodes and we swap them (Figure 5.) only two nodes will get different triset form:
$T_1 = \{a|b|c,O\}$ $T_1 = \{a|c|b,O\}$
$T_2 = \{a,b|c|O\}$ $T_2 = \{a,c|b|O\}$
$T_1 >>> T_1' = 1$, $T_2 >>> T_2' = 1$, and we get 2 as metric value for such trees.

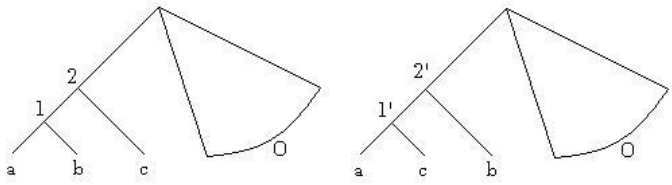

Fig.5. Examples of two trees that give minimum non zero value of triset metric. By O we mean subtree of any non zero size.

6.2. Mimimum value of triset metric for any topological change is 3

Swapping leafs is not only change we can made in trees. We also can choose to reconstruct topological shape of the tree. Figure 6. shows example of the smallest change we can make. As we can see only two nodes will get different triset form:
$T_1 = \{a|b|c,d,O\}$ $T_1' = \{a|b|c,d,O\}$
$T_2 = \{a,b|c|d,O\}$ $T_2' = \{a,b|c,d|O\}$
$T_3 = \{a,b,c|d|O\}$ $T_3' = \{a,b,O|c|d\}$

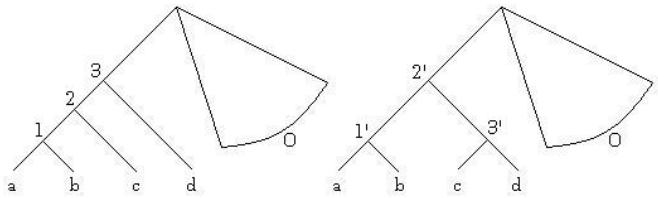

Fig.6. Examples of two different (in a topological way) trees that give minimum non zero value of triset metric. By O we mean subtree of any non zero size.

Using our algorithm we get perfect matching $T_1 >>> T_{1'} = 0$, $T_2 >>> T_{2'} = 1$, $T_3 >>> T_{3'} = 2$, and we get 3 as metric value for such trees.

6.3. Two most different trees

We can construct two most different tress in terms of treeset metric using this method:
As a first tree take "worm" graph (each node has one or two leafs outgoing from it), as a second tree take balanced tree (maximum numbers of nodes has 3 other nodes as neighbors) and make specific permutation of leafs (example in Fig.7.)

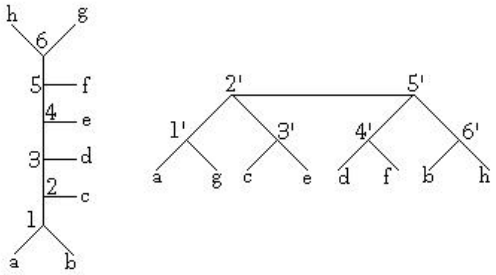

Fig.7. Examples of two different trees that give maximum value of triset metric for any given trees with 8 leafs.

For each node we get triset form:

T_1 = {a|b|c,d,e,f,g,h} $T_{1'}$ = {a|g|c,e,d,f,b,h}
T_2 = {a,b|c|d,e,f,g,h} $T_{2'}$ = {a,g|c,e|d,f,b,h}
T_3 = {a,b,c|d|e,f,g,h} $T_{3'}$ = {c|e|a,g,d,f,b,h}
T_4 = {a,b,c,d|e|f,g,h} $T_{4'}$ = {d|f|a,g,c,e,b,h}
T_5 = {a,b,c,d,e|f|g,h} $T_{5'}$ = {d,f|b,h|a,g,c,e}
T_6 = {a,b,c,d,e,f|g|h} $T_{6'}$ = {b|h|a,g,c,e,d,f}

Now we create bipartite graph using those nodes, we count value of >>> operation between each node in different tree, and we find minimum weight perfect matching.
Our solution is $T_1 >>> T_{1'} = 1$, $T_2 >>> T_{2'} = 2$, $T_3 >>> T_{3'} = 3$, $T_4 >>> T_{4'} = 3$, $T_5 >>> T_{5'} = 2$, $T_6 >>> T_{6'} = 1$, so metric value is 12.

6.4. For every tree *T* (with 4 or more leafs) the is a tree *T'* that *M*(*T*,*T'*) = *2*

This fact is very important for describing metric space properties. T' is constructed using method shown in 6.1.

6.5. For every two trees *T, T'* (constructed over 4 or more the same leafs) we can show path of trees $T^1 = T, T^2, T^3 \ldots T^{k-1}, T^k = T'$, where $M(T^i,T^{i+1}) = 2$ or 3 for $i = 1,2 \ldots k-1$.

This fact can be proven by showing that combination of operations: swapping two leafs form 6.1. and changing topological shape from 6.2. is enough for making transition from *T* to *T'*.

6 Summary

There are lot of possibilities of modification this metric. In the above considerations we have assumed that a single action increases the value >>> by one. However, we can differentiate the importance those operations in any way. This gives us the ability to compare for example: weighted phylogenetic trees.

One of the most important properties of this metric is fact that local changes do not affect metric value too much. Also using bipartite graph, and perfect matching method allow us to compute metric value in polynomial time.

References

[1] D. Bogdanowicz, Comparing Phylogenetic Trees Using a Minimum Weight Perfect Matching, Information Technology (ICIT) 2008

[2] D. Bryant, Building Trees, Hunting for Trees, and Comparing Trees – Theory and Methods in Phylogenetic Analysis. Ph.D. Thesis. Department of Mathematics. University of Canterbury, 1997.

[3] D.F. Robinson, L. R. Foulds, Comparison of phylogenetic trees. Mathematical Biosciences, 53(1-2):131–147, February 1981.

Received: November, 2011