Spanning edge betweenness

Andreia Sofia Teixeira
INESC-ID at Lisbon
IST, Tech Univ of Lisbon
steixeira@kdbio.inesc-id.pt

Pedro T Monteiro
INESC-ID at Lisbon
ptgm@kdbio.inesc-id.pt

João A Carriço
IMM, Fac of Medicine,
Univ of Lisbon
jcarrico@fm.ul.pt

Mário Ramirez
IMM, Fac of Medicine,
Univ of Lisbon
ramirez@fm.ul.pt

Alexandre P Francisco
IST, Tech Univ of Lisbon
INESC-ID at Lisbon
aplf@ist.utl.pt

ABSTRACT

We present a new edge betweenness metric for undirected and weighted graphs. This metric is defined as the fraction of minimum spanning trees where a given edge is present and it was motivated by the necessity of evaluating phylogenetic trees. Moreover we provide results and methods concerning the exact computation of this metric based on the well known Kirchhoff's matrix tree theorem.

Categories and Subject Descriptors

G.2.1 [**Discrete Mathematics**]: Combinatorics—*counting problems*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*graph algorithms, network problems*; G.2.3 [**Discrete Mathematics**]: Applications; H.2.8 [**Database management**]: Database Applications—*Data mining*

General Terms

Theory, Measurement, Algorithms

Keywords

Edge Centrality, Minimum Spanning Trees, Network Analysis, Graph Algorithms

1. INTRODUCTION

Centrality measures are important in a large number of graph applications, from search and ranking to social and biological network analysis. Most of this measures are calculated upon the nodes/vertices. With node centrality we can measure the relative importance of nodes within a graph [1] but sometimes our interest is to study the importance of links/edges on a network. A first approach was done by Girvan and Newman [15] where they define edge betweenness, generalizing Freeman's betweenness centrality [12] to edges, as the number of shortest paths between pairs of vertices that run along it, with a direct application on the identification of community structures in networks. There are however other problems where alternative definitions of edge centrality are demanded, as is the case with phylogenetic trees statistical evaluation.

The use of trees for phylogenetic representations started in the middle of the 19th century. One of their most popular uses is Charles Darwin's sole illustration in "The Origin of Species" [4]. The simplicity of the tree representation still makes it the method of choice today to easily convey the diversification and relationships between species. Many different methods have been proposed to reconstruct phylogenies, mostly concerned with recovering evolutionary relationships over long time periods [9]. However, at shorter timescales and with limited diversity, which are conditions encountered in population genetics and microevolutionary studies of a single species, the assumptions made by these methods may not be equally valid [7] and a number of other methods have been used when analyzing this data. Each algorithm or method used to infer and draw a tree, makes a series of implicit or explicit assumptions that conditions the types of trees generated. This variability has important repercussions that frequently go unappreciated by those who use them.

Minimum Spanning Trees (MSTs) are becoming increasingly used for representing relationships between strains in epidemiological and population studies of bacterial pathogens. Although MST computation is a classical mathematical problem and its application to evolutionary studies had already been suggested more than a decade ago [7], it wasn't until recently, with the advent of multilocus sequence typing (MLST) [20], that they gained popularity has an alternative to eBURST [8]. One appeal of MSTs is the simplicity of their assumptions that reflect the concept of minimal evolution. MSTs simply link together the more closely related individuals in the population, generating a single tree representing all individuals. The Steiner trees [23] generated by the more classical methods for phylogenetic inference, place individuals exclusively in branch tips. By allowing individuals to be placed at interior nodes, spanning trees and MSTs in particular, better convey the peculiarities of short-term intraspecific evolution [7].

It was also recently pointed out that the optimal implementation of the BURST rules in goeBURST results in a set of disjoint MSTs [10]. These trees group sequence types (STs) that differ by a maximum threshold number of alleles from at least one other ST of the group. In fact, goeBURST

Eleventh Workshop on Mining and Learning with Graphs. Chicago, Illinois, USA

addresses maximum weight problems that, together with MSTs, are particular cases of graphic matroids [10]. But, as is well known, MSTs are in general not unique for a given network and this fact has been also observed in the context of phylogenetic trees [7, 22]. The fact that a single tree is reported from a multitude of possible and equally optimal solutions and that no statistical metrics exist to evaluate them, justified a recent heuristic approach to address these issues [22]. The authors suggest a method based on a mark-recapture approach to estimate the number of possible trees and a bootstrap procedure to evaluate tree credibility.

Here, we present an improvement to this approach by introducing a new edge centrality metric and showing how to determine exactly the number of possible trees and proportion of this universe that includes each of the possible links/edges through an expansion of the Kirchhoff's matrix tree theorem [16, 17].

2. PROBLEM

Let $G = (V, E)$ be a connected, undirected and weighted graph, with weight function $w : E \to I\!R$, where V is the set of vertices and $E \subset V \times V$ is the set of edges. A Minimum Spanning Tree (MST) $T = (V, E')$ is a subgraph of G that is a tree and contains all the vertices of G, i.e., that spans over all vertices in V, with $|E'| = |V| - 1$, and such that $\sum_{e \in E'} w(e)$ is minimum among all spanning trees. It is clear that we can have more than one MST for a given graph G and we would like to count how many MSTs exist in G. The solution to this problem is provided by the Kirchhoff's matrix tree theorem [16] for unweighted graphs and by Eppstein [6] for weighted graphs, where the Kirchhoff's matrix tree theorem is still used but only after some graph transformations.

However, in this paper we are interested in a slightly different question. Given an edge $e \in E$ we want to know the fraction $\delta_G(e)$ of MSTs where e occurs. The value $\delta_G(e)$ is what we call the *spanning edge betweenness* for e and it is formally defined as

$$\delta_G(e) = \frac{\tau_G(e)}{\tau_G}, \tag{1}$$

where τ_G is the number of different MSTs for G and $\tau_G(e)$ is the number of different MSTs for G where e occurs. Note that $\tau_G(e)$ may be zero whenever an edge e is not present in any MST, causing $\delta_G(e)$ to be zero. In what follows we write $\delta(e)$, $\tau(e)$ and τ whenever G is clear from the context.

Therefore, the problem addressed in this paper is how to compute, as efficiently as possible, the spanning edge betweenness $\tau_G(e)$ for a given $e \in E$, where $G = (V, E)$ is a connected, undirected and weighted graph.

3. METHODS AND RESULTS

We will start by showing how to compute $\tau_G(e)$ and $\delta_G(e)$ when $G = (V, E)$ is a connected, undirected and unweighted graph, with $n = |V|$ vertices and $m = |E|$ edges. Note that in this case the number τ of MSTs in G is equal to the number of spanning trees in G and it can be computed directly from the Kirchhoff's matrix tree theorem [17]. Then we will extend our result to weighted graphs and present some experimental results.

3.1 Unweighted graphs

Let $F \in \{-1, 0, 1\}^{n \times m}$ be the incidence matrix for G such that $F_{i,e} = 1$ and $F_{j,e} = -1$, for $e = (i, j) \in E$. Let us also consider the reduced incidence matrix $F^{(i)}$ obtained from F by deleting row i. Note that we have both $rank(F) = n - 1$ and $rank(F^{(i)}) = n - 1$, since G is connected and rows in F are linearly dependent (the row sum at each column is zero and, hence, any row can be expressed as a linear combination of other rows). Moreover, the determinant for any square submatrix of $F^{(i)}$, for any i, is either 0, -1, or 1. A more interesting observation due to Kirchhoff is that a submatrix $(n-1) \times (n-1)$ of $F^{(i)}$, for any i, is non-singular if and only if its columns correspond to the edges of a spanning tree.

THEOREM 1 (KIRCHHOFF [17]). *The spanning trees of a connected and undirected graph G with n vertices are the non-singular $(n-1) \times (n-1)$ submatrices of the reduced incidence matrix $F^{(i)}$, for any i, and the determinants of the submatrices are all ± 1.*

Hence, by using the Cauchy-Binet theorem on determinants, the number of spanning trees τ is given by the Kirchhoff's well known formula

$$\tau = det(L^{(i)}) \tag{2}$$

$$= \sum_S det(F_S^{(i)}) det({F_S^{(i)}}^\top) \tag{3}$$

$$= \sum_S det(F_S^{(i)})^2, \tag{4}$$

where S ranges over the subsets of E with size $n-1$, $L = FF^\top$ is the Laplacian matrix for G, and $L^{(i)}$ denotes the matrix obtained from L by deleting row and column i.

We extend this result to compute $\tau(e)$, for $e \in E$, as follows.

THEOREM 2. *Given $G = (V, E)$ an undirected and connected graph, let $e = (i, j) \in E$ and $L^{(ij)}$ denote the matrix obtained from L by deleting rows i and j and columns i and j. Then, $det\left(L^{(ij)}\right)$ is the number of spanning trees $\tau(e)$ that contain e.*

PROOF. As discussed above, the total number of spanning trees is given by $det(L^{(i)})$, for any i. Let G' be the graph where we remove the edge (i, j) and L' be the Laplacian for G'. Hence, the total number of spanning trees for G' is given by $det(L'^{(i)})$, for any i, and the number of MSTs that contain (i, j) is simply given by $det(L^{(i)}) - det(L'^{(i)})$. Let us show that $det(L^{(ij)}) = det(L^{(i)}) - det(L'^{(i)})$ or, equivalently, that $det(L^{(i)}) = det(L'^{(i)}) + det(L^{(ij)})$. We have that $L^{(i)} = F^{(i)} {F^{(i)}}^\top$ and $L^{(ij)} = F^{(i,j)} {F^{(i,j)}}^\top$, where $F^{(i,j)}$ is obtained from F by removing rows i and j, and, using Cauchy-Binet's formula, we can show instead that

$$\sum_S det\left(F_S^{(i)}\right)^2 = \sum_{S'} det\left({F'}_{S'}^{(i)}\right)^2 + \sum_{S^*} det\left(F_{S^*}^{(i,j)}\right)^2 \tag{5}$$

where F' is the incidence matrix for G', S ranges over the subsets of E with size $n-1$, S' ranges over the subsets of $E \setminus \{(i, j)\}$ with size $n-1$, and S^* ranges over the subsets of E with size $n-2$. Since S' ranges over the subsets of $E \setminus \{(i, j)\}$, we can replace F' by F in previous equation.

Note also that

$$\sum_{S^*} det\left(F_{S^*}^{(i,j)}\right)^2 = \sum_{S^* \cup \{(i,j)\}} det\left(F_{S^*}^{(i,j)} \times \pm 1\right)^2 \quad (6)$$

$$= \sum_{S^* \cup \{(i,j)\}} det\left(F_{(S^* \cup \{(i,j)\})}^{(i)}\right)^2 \quad (7)$$

because adding edge (i,j) to S^* and considering $F^{(i)}$ instead of $F^{(i,j)}$ just adds a term ± 1 to each matrix determinant. Therefore,

$$\sum_{S} det\left(F_{S}^{(i)}\right)^2 = \sum_{S'} det\left(F_{S'}^{(i)}\right)^2 + \sum_{S^* \cup \{(i,j)\}} det\left(F_{(S^* \cup \{(i,j)\})}^{(i)}\right)^2 \quad (8)$$

which is an equality as the first term on the right side ranges over all subsets of E with size $n-1$ that do not contain (i,j) and the second term ranges over all subsets of E with size $n-1$ that do contain (i,j). □

Hence, using both results, we can easily compute $\delta(e)$ for any $e \in E$. Note also that the same is true for multigraphs, graphs that allow multiple edges between the same pair of vertices, as both results above hold with the following changes in the Laplacian matrix L [19]: if vertex i is adjacent to vertex j in G, then L_{ij} is equal to the number of edges between i and j; when counting the degree of a vertex, all loops are excluded.

3.2 Weighted graphs

Let $G = (V, E)$ be a connected, undirected and weighted graph, with weight function $w : E \to \mathbb{R}$. We can compute a MST for G by using the Kruskal's algorithm [18]:

1. sort E with respect to w in increasing order;
2. create a forest M where for each $u \in V$, $(\{u\}, \{\})$ is a tree of the forest;
3. iterate over E in increasing order and, for each $(u,v) \in E$, if u and v are in different trees, add (u,v) to M combining both trees as single tree;
4. return M.

Note that we may get different MSTs by changing the order obtained in step 1, where we can exchange positions of edges with the same weight. In particular, since it is well known that the sorted list of edge weights is the same for any MST, changing the order allow us to obtain all different MSTs.

We can take a step further. Let $e \in E$ and let M' be the forest obtained by the Kruskal's algorithm after processing all edges $e' \in E$ such that $w(e') < w(e)$. Let also G' be a graph where each tree in M' is a vertex and where we add all edges in E with weight $w(e)$. Note that G' may be a multigraph and, since all edges have the same weight, we may look at it as an unweighted multigraph. Moreover, if we consider the connected component C of G' that contains edge e, using results from the previous section, we can compute the number τ_C of spanning trees for that component and also the number $\tau_C(e)$ of spanning trees for that component where e occurs. The key observations are that we can use this approach to compute the number of spanning trees in G and that $\delta_G(e) = \delta_C(e)$.

It is clear that an edge $e \in E$ can only permute with another edge $e' \in E$ to form a different MST iff $w(e) = w(e')$ and, if a MST M contains e, adding e' to M leads to a cycle. Moreover, that cycle can only contain edges with weight equal or lower than $w(e)$, otherwise M would not be a MST. If we add all edges with weight $w(e)$ to M and contract all edges with weight lower than $w(e)$, we obtain the graph G' and the product of the number of trees in each connected component of G' is the number of ways we can select edges with weight $w(e)$ for each MST of G. By doing this for each different weight in G and by multiplying all values, we obtain the number of MSTs τ for G.

Since a given edge e only has influence on the number of trees for the component of G' where it occurs and the number of trees for all other components and weights remain the same, it follows that $\delta_G(e) = \delta_C(e)$.

Hence, given a connected, undirected and weighted graph $G = (V, E)$, with weight function $w : E \to \mathbb{R}$, we can compute the number of MSTs and the spanning edge betweenness for each edge as follows:

1. sort E with respect to w in increasing order;
2. let $H = (V, \emptyset)$ and $\tau_G = 1$;
3. iterate over E in increasing order and, while edges have the same weights, add them to H;
4. for each connected component C in H, compute τ_C using Theorem 1, update $\tau_G = \tau_G \times \tau_C$, and, for each edge $e \in C$, compute $\tau_C(e)$ using Theorem 2 and $\delta_C(e)$ using Equation 1;
5. contract all edges in H such that each connected component becomes a single vertex;
6. if H has more than one vertex, repeat from step 3, otherwise return τ_G.

3.3 Implementation and evaluation

We have implemented our approach in Java as a module for PHYLOViZ [11] that, given a dataset, allows the user to compute the number of MSTs for a given phylogenetic graph and also spanning edge betweenness values for each edge. Our implementation uses the Colt library[1] for linear algebra operations, including in particular the computation of matrix determinants. Since we are dealing with relatively large sparse graphs, we use the class `SparseDoubleMatrix2D` in Colt. We also use a disjoint-set data structure to track connected components similarly to what is common in Kruskal's algorithm implementations [3].

The time complexity of the proposed approach is dominated by the time required to compute the determinants, since the Kruskal's runs in $O(m \log n)$ time, for a graph with n vertices and m edges. Computing the determinant for a $n \times n$ matrix can be done in $O(n^{3/2})$ time [14]. Hence, for sparse graphs with $m = O(n)$, this method runs in $O(n^{2.5})$ time since we have to compute a determinant for each edge. In practice, it runs faster since connected components are usually much smaller than the original graph.

We provide in Table 1 details for five different datasets and the running times to compute spanning edge betweenness for all edges in phylogenetic trees computed by PHYLOViZ.

[1] `http://acs.lbl.gov/software/colt/`

Table 1: Details and running time for some datasets where we compute the spanning edge betweenness for evaluating phylogenetic trees.

Dataset	|V|	|E|	Edges without ties	Number MSTs	Time (sec.)
Bacillus licheniformis	16	107	1	78177	0
Staphylococcus epidermis	470	16995	95	1×10^{194}	84
Enterococcus faecium	797	47717	141	∞	1180
Burkholdaria pseudomallei	976	52499	210	∞	1252
Candida albicans	1694	57855	312	∞	1264

These phylogenetic trees are just MSTs computed for single locus variant graphs for each species, where each vertex denotes a strain and where there is an edge between two vertices whenever strains differ in one single locus [10]. These experiments were conducted using an Intel i7 processor at 2.3GHz, with 6GB of RAM. The column *Edges without ties* represents the number of edges that are always present in every MST, i.e., no other edges could replace them to build a new MST. Note also that, for the last three datasets, the number of MSTs is too large to be represented as a double, but we can still compute the spanning edge betweenness as it depends only on the number of MSTs for a given reduced connected component as described before.

As an illustration for the methods proposed in this paper, we provide in Figures 1-3 an example with the second largest connected component in the SLV graph (with edges between strains with a single locus variation) for the *Burkholdaria pseudomallei* dataset, as computed by PHYLOViZ. Note that, in this case, we have an unweighted graph as all edges are in the same level. We have the spanning tree selected by PHYLOViZ in Figure 2 and the spanning edge betweenness values, as well as the Laplacian matrix, in Figure 3. The usefulness of the proposed metric becomes clear when we want to evaluate the proposed spanning tree, since we can easily verify that there are edges that only occur in about 50% of all possible spanning trees and, thus, while analyzing phylogenetic trees, we can decide about our confidence in the proposed tree as a phylogenetic hypothesis. This is of particular relevance if we take into consideration the fact that locus variations can be due to recombination and mutation events, where the evolutionary origin of a given strain may not be unique. Since a phylogenetic tree is just a possible explanation for such origin, this kind of analysis allows us to evaluate how many alternatives the underlying phylogenetic model can provide for a given phylogenetic link/edge.

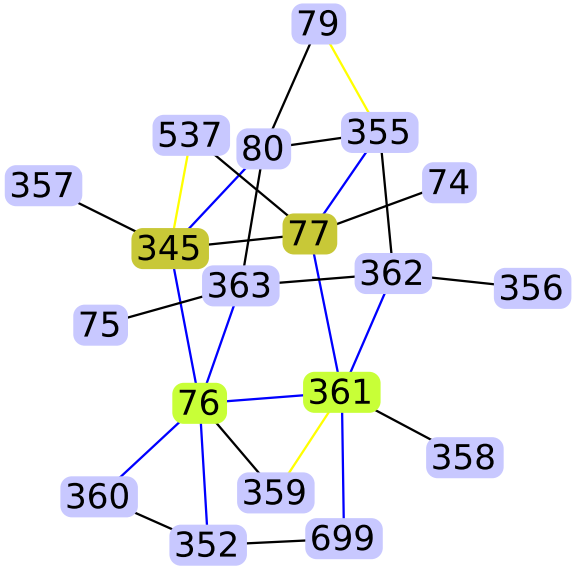

Figure 1: Graph for the second largest component in the SLV graph for *Burkholdaria pseudomallei* dataset as computed by PHYLOViZ.

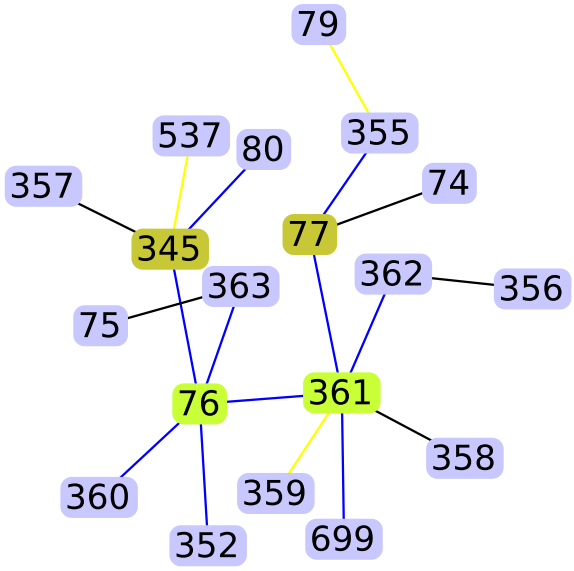

Figure 2: A spanning tree for the graph in Figure 1.

4. RELATED WORK

The problem of counting MSTs has been a challenge for the last decades, namely in what concerns the development of efficient approaches for counting MSTs in weighted graphs. As stated before, for unweighted graphs, best solutions rely on the Kirchhoff's matrix tree theorem [16, 17]. In what concerns counting exactly the number of trees where a given edge occurs, to our knowledge, the method described here is the first one to be proposed.

For weighted graphs, the problem becomes more complex and we have seen different approaches. In 1987, Gavril [13] addressed the problem of counting the number of MSTs by constructing a treelike recursive structure, the root of which is the subgraph G' formed by removing all non-maximum-weight edges from G, and each subtree of which is constructed recursively from the components of $G \backslash G'$. The minimum spanning trees of G can then be counted by multiplying together the numbers of spanning trees at each node of this structure. This method runs in $O(nM(n))$ time, where

```
19 x 19 matrix
 1 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
-1  5  0  0  0 -1  0  0  0 -1 -1 -1  0  0  0  0  0  0  0
 0  0  1 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0 -1  4 -1  0  0  0  0  0  0  0  0 -1  0  0 -1  0  0
 0  0  0 -1  6 -1 -1 -1 -1 -1  0  0  0  0  0  0  0  0  0
 0 -1  0  0 -1  5  0  0  0  0  0 -1  0 -1 -1  0  0  0  0
 0  0  0  0 -1  0  3  0 -1  0  0  0  0  0  0 -1  0  0  0
 0  0  0  0 -1  0  0  2  0 -1  0  0  0  0  0  0  0  0  0
 0  0  0  0 -1  0 -1  0  2  0  0  0  0  0  0  0  0  0  0
 0 -1  0  0 -1  0  0 -1  0  6  0  0  0  0  0  0 -1 -1  0 -1
 0 -1  0  0  0  0  0  0  0  0  4  0 -1 -1  0  0 -1  0  0
 0 -1  0  0  0 -1  0  0  0  0  0  2  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0 -1  0  2 -1  0  0  0  0  0
 0  0  0 -1  0 -1  0  0  0  0 -1  0 -1  4  0  0  0  0  0
 0  0  0  0  0 -1  0  0  0  0  0  0  0  0  1  0  0  0  0
 0  0  0  0  0  0 -1  0  0 -1  0  0  0  0  0  2  0  0  0
 0  0  0 -1  0  0  0  0  0 -1 -1  0  0  0  0  0  4 -1  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1  1  0
 0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  1

Det: 61215.99999999997
CC 1 has 6.1216E4 MSTs
Spanning edge betweenness:
 74 -  77, level: 1, freq: 100.00%  (1E0)
 75 - 363, level: 1, freq: 100.00%  (1E0)
 76 - 345, level: 1, freq:  53.73%  (5.373105E-1)
 76 - 352, level: 1, freq:  52.38%  (5.237846E-1)
 76 - 359, level: 1, freq:  59.51%  (5.951385E-1)
 76 - 360, level: 1, freq:  63.09%  (6.309462E-1)
 76 - 361, level: 1, freq:  38.06%  (3.805541E-1)
 76 - 363, level: 1, freq:  55.69%  (5.569132E-1)
 77 - 345, level: 1, freq:  43.60%  (4.359645E-1)
 77 - 355, level: 1, freq:  54.67%  (5.467198E-1)
 77 - 361, level: 1, freq:  53.73%  (5.373105E-1)
 77 - 537, level: 1, freq:  60.90%  (6.089911E-1)
 79 -  80, level: 1, freq:  61.08%  (6.108207E-1)
 79 - 355, level: 1, freq:  61.08%  (6.108207E-1)
 80 - 345, level: 1, freq:  54.67%  (5.467198E-1)
 80 - 355, level: 1, freq:  44.33%  (4.432828E-1)
 80 - 363, level: 1, freq:  56.48%  (5.647543E-1)
345 - 357, level: 1, freq: 100.00%  (1E0)
345 - 537, level: 1, freq:  60.90%  (6.089911E-1)
352 - 360, level: 1, freq:  63.09%  (6.309462E-1)
352 - 699, level: 1, freq:  67.85%  (6.785154E-1)
355 - 362, level: 1, freq:  56.48%  (5.647543E-1)
356 - 362, level: 1, freq: 100.00%  (1E0)
358 - 361, level: 1, freq: 100.00%  (1E0)
359 - 361, level: 1, freq:  59.51%  (5.951385E-1)
361 - 362, level: 1, freq:  55.69%  (5.569132E-1)
361 - 699, level: 1, freq:  67.85%  (6.785154E-1)
362 - 363, level: 1, freq:  55.62%  (5.561945E-1)
```

Figure 3: Spanning edge betweenness and Laplacian matrix for the graph in Figure 1.

$M(n)$ is the time required to multiply two $n \times n$ matrices. Later, in 1997, Broder and Mayr [2] improved this bound by proposing a method based on a generating function that can be expressed as a simple determinant, where the weights of the edges appear as exponents of polynomials. This method proceeds by factoring the determinant and it works for non-negative integral edge weights. It runs in $O(M(n))$ time.

Eppstein [6] took a different approach and created the concept of equivalent graph. Specifically, we construct from any given edge-weighted graph G an equivalent graph EG without weights, with a *sliding transformation*, such that the minimum spanning trees of G have a one-to-one correspondence with the spanning trees of EG. Having translated the weighted graph to an equivalent unweighted graph, we can compute the number of MSTs by just applying the Kirchhoff's matrix tree theorem to the new graph.

Note that most of these approaches aim at generating and sampling MSTs, an harder problem than just counting the number of MSTs. Hence, although we use some of their ideas in our approach, since we are just counting MSTs, we have a less complex approach and we are able to achieve a better performance. Moreover, our approach may be applied to the general case of graphical matroids. Note that the problem of finding an MST is a particular case of graphic matroids [21] and, thus, finding a solution for a given graph consists of solving an instance of graphic matroids [21, 25, 24], which can be optimally solved with a greedy approach [5]. One of those greedy approaches is precisely the Kruskal's algorithm [18]. In the general case of graphic matroids, edges may not be weighted, which is usually the case. We just need to define a total order for edges based on some criteria and this is precisely what we have in general phylogenetic studies based on trees [10]. Contrary to other methods that depend on edges being weighted, our approach just depends on sorting edges in increasing order and, thus, we just require a total order to be defined.

5. CONCLUSIONS AND FURTHER WORK

We present a new edge centrality metric, the spanning edge betweenness, defined as the fraction of MSTs containing a given edge. We also provide the required results and methods to compute exactly this metric for both unweighted and weighted graphs. Since our method relies just on the existence of a total order on edges, it can be used in the general case of graphical matroids. Although real weighted graphs may have just a single MST, in many problems, such as phylogenetic studies, edges have levels assigned or are categorized accordingly to a set of decision rules which impose a total order on edges. These are the kind of problems where the proposed metric and approach becomes useful to evaluate MST edges and full MST quality, based either on the number of possible MSTs or on spanning edge betweenness statistics.

Since we rely on the Kirchhoff's matrix tree theorem, thus needing to compute several determinants for slightly different matrices, we plan to investigate how to accelerate these computations by reusing previous computations and by using more efficient methods for sparse positive semi-definite matrices decomposition, such as those based on Cholesky decomposition.

The comparison between this metric and other well known centrality metrics should also be investigated in the context of complex network analysis, as it provides a rather different approach for evaluating edge relevance or significance.

Finally, we note that this metric is being used with success to evaluate phylogenetic trees, providing confidence levels for each selected edge in the proposed tree, as described in our example.

Acknowledgments.

This work was partly supported by national funds through FCT – Fundação para a Ciência e Tecnologia, under projects PTDC/EIA-CCO/118533/2010 and PEst-OE/EEI/LA0021 /2013.

6. REFERENCES

[1] S. P. Borgatti and M. G. Everett. A graph-theoretic perspective on centrality. *Social Networks*, 28(4):466–484, October 2006.

[2] A. Z. Broder and E. W. Mayr. Counting minimum weight spanning trees. *Journal of Algorithms*, 24(1):171–176, 1997.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction To Algorithms*. MIT Press, 2001.

[4] C. Darwin. *On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life*. John Murray, Albemarle Street, London, 1861.

[5] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1(1):127–136, 1971.

[6] D. Eppstein. Representing all minimum spanning trees with applications to counting and generation. Technical Report 95-50, Department of Information and Computer Science, University of California, Irvine, CA 92717, December 1995.

[7] L. Excoffier and P. Smouse. Using allele frequencies and geographic subdivision to reconstruct gene trees within a species: molecular variance parsimony. *Genetics*, 136:343–359, 1994.

[8] E. J. Feil, B. C. Li, D. M. Aanensen, W. P. Hanage, and B. G. Spratt. eburst: Inferring patterns of evolutionary descent among clusters of related bacterial genotypes from multilocus sequence typing data. *J Bacteriol*, 186:1518–1530, 2004.

[9] J. Felsenstein. *Inferring phylogenies*. Sinauer Associates, Sunderland, Massachusetts, 2004.

[10] A. Francisco, M. Bugalho, M. Ramirez, and J. Carriço. Global optimal eburst analysis of multilocus typing data using a graphic matroid approach. *BMC Bioinformatics*, 10:152, 2009.

[11] A. Francisco, C. Vaz, P. Monteiro, J. Melo-Cristino, M. Ramirez, and J. Carriço. PHYLOViZ: phylogenetic inference and data visualization for sequence based typing methods. *BMC Bioinformatics*, 13(1):87, 2012.

[12] L. C. Freeman. A set of measures of centrality based upon betweeness. *Sociometry*, 40(1):35–41, 1977.

[13] F. Gavril. Generating the maximum spanning trees of a weighted graph. *J. Algorithms*, 8(4):592–597, 1987.

[14] A. George and E. Ng. On the complexity of sparse *qr* and *lu* factorization of finite-element matrices. *SIAM Journal on Scientific and Statistical Computing*, 9(5):849–861, 1988.

[15] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *PNAS*, 99(12):7821–7826, April 2002.

[16] J. Harris, J. L. Hirst, and M. Mossinghoff. *Combinatorics and Graph Theory*. Springer, 2008.

[17] G. Kirchhoff. Ueber die auflösung der gleichungen, auf welche man bei der untersuchung der linearen vertheilung galvanischer ströme geführt wird. *Annalen der Physik*, 148(12):497–508, 1847.

[18] J. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Amer. Math. Soc.*, 7:48–50, 1956.

[19] M. Lewin. A generalization of the matrix-tree theorem. *Mathematische Zeitschrift*, 181:55–70, 1982.

[20] M. Maiden, J. A. Bygraves, E. Feil, G. Morelli, J. E. Russell, R. Urwin, Q. Zhang, J. Zhou, K. Zurth, D. A. Caugant, I. M. Feavers, M. A., and B. G. Spratt. Multilocus sequence typing: a portable approach to the identification of clones within populations of pathogenic microorganisms. *Proc Natl Acad Sci USA*, 95:3140–3145, 1998.

[21] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization*. Dover, 1998.

[22] S. J. Salipante and B. G. Hall. Inadequacies of minimum spanning trees in molecular epidemiology. *J Clin Microbiol*, 49:3568–3575, 2011.

[23] P. H. A. Sneath and R. R. Sokal. *Numerical taxonomy; the principles and practice of numerical classification*. W. H. Freeman, San Francisco, 1973.

[24] W. T. Tutte. Lectures on matroids. *J. Res. Nat. Bur. Standards Sect. B*, 69:1–47, 1965.

[25] H. Whitney. On the abstract properties of linear dependence. *American Journal of Mathematics*, 57(3):509–533, 1935.