The Distribution of the Minimum Spanning Tree of the Complete Graph

Author:

Karl-Friedrich Israel

St. Cross College

Supervisor:

Dr Christina Goldschmidt

Lady Margaret Hall

UNIVERSITY OF OXFORD

DEPARTMENT OF STATISTICS

September 2013

A dissertation submitted in partial fulfilment of the requirements for the degree of

Master of Science in Applied Statistics

Abstract

In this study we consider an i.i.d. random version of the minimum spanning tree (MST) of the complete graph on n vertices and investigate distributional and structural properties of the MST limit object, as $n \to \infty$, established in [1]. Relatively little is known about this graph limit. We approximate this object by simulation of trees on up to 4000 vertices. Simulated trees are endowed with the graph distance renormalized by $n^{1/3}$.

Analysing the total length of reduced trees on up to $k = 4$ randomly chosen vertices using non-parametric and parametric density estimation techniques, suggests a unimodal distribution with exponential tails. Moreover, it seems to be log-concave and its overall shape is similar to a $Gamma$-distribution.

$Dirichlet$-models provide a very good approximation for the distribution of segmental lengths in reduced trees on up to $k = 6$ randomly chosen vertices, when measured as proportions of the total length. However, these proportions do not follow the same marginal distribution and are thus not exchangeable, unlike in the limit object of the uniform random tree - the Brownian Continuum Random Tree (BCT). The simulated data supports the existence of two groups of segmental lengths in which elements are exchangeable. The structural shape of reduced trees has no significant influence on the distribution of segmental lengths or the total length. Yet, in contrast to the BCT, genuinely different possible shapes are not equally likely to occur.

Acknowledgements

I am truly and deeply indebted to everyone who afforded me with the opportunity to complete the Master of Science programme in Applied Statistics at the University of Oxford. I would also like to express my heartfelt gratitude to my project supervisor, Dr Christina Goldschmidt, for her stimulating suggestions and encouragement which made this project possible.

Contents

1 Introduction 1

2 Scientific Background 4

2.1 Finding the Minimum Spanning Tree 4

2.2 The Difference between Minimum Spanning Trees and Uniform Random Trees 6

2.3 Illustration of Theoretical Results for Uniform Random Trees 9

3 Summary of Methods to be used 13

3.1 The Simulation of Trees 13

3.2 The Exploration of Trees 14

3.3 Probability Density Estimation 15

4 Detailed Analysis of Simulated Data 19

4.1 The Total Length of Reduced Trees on up to 4 Randomly Chosen Vertices 19

4.2 The Structure and Shape of Reduced Trees on up to 6 Randomly Chosen Vertices 25

5 Conclusion 35

Appendix: *R*-Code i

Simulation of Data i

Analysis of Data vi

List of Figures

1 A weighted complete graph on 4 vertices (left) and the corresponding minimum spanning tree (right) 5

2 Possible shapes for trees on 4 vertices (labelled $\{1, 2, 3, 4\}$): there are 4 combinations with "Star Shape" and 12 combinations for which all vertices lie in a row 7

3 Difference between uniform random trees and minimum spanning trees on 4 vertices: bar shart of the number of selected trees with distinct shape (1 to 4 represent trees with "Star Shape"; 5 to 16 represent trees in which all vertices lie in a row) - 200,000 simulations for each selection procedure 7

4 A unniform random tree (left) and a minimum spanning tree (right) on 3000 vertices 9

5 Histograms of total length (distance) between 2 randomly chosen vertices from uniform random trees on 2000, 3000 and 4000 vertices (rescaled by $n^{-1/2}$) and limiting distribution $Te^{-T^2/2}$ (red) 11

6 Histograms of total length of reduced trees on 3 randomly chosen vertices from uniform random trees on 2000, 3000 and 4000 vertices (rescaled by $n^{-1/2}$) and limiting distribution $\frac{1}{2}T^3e^{-T^2/2}$ (red) 11

7 Histograms of segmental lengths (as proportions of total length) in reduced trees on 3 randomly chosen vertices from uniform random trees on 4000 vertices and limiting distribution $Dirichlet(1, 1, 1)$ - marginal distirbutions, $Beta(1, 2)$, are plotted over each histogram (red) 12

8 Histograms of total length (distance) between 2 randomly chosen vertices from minimum spanning trees on 2000, 3000 and 4000 vertices (rescaled by $n^{-1/3}$) 19

9 Kernel density estimates using a normal distribution with standard deviation 0.17 for total length (distance) between 2 randomly chosen vertices from minimum spanning trees on 2000 (blue), 3000 (green) and 4000 (red) vertices. 20

10 Histograms of total length of reduced trees on 3 randomly chosen vertices from minimum spanning trees on 2000, 3000 and 4000 vertices (rescaled by $n^{-1/3}$) 20

11 Kernel density estimates using a normal distribution with standard deviation 0.21 for the total length of reduced trees on 3 randomly chosen vertices from minimum spanning trees on 2000 (blue), 3000 (green) and 4000 (red) vertices. 21

12 Total length of reduced trees on 4 randomly chosen vertices from minimum spanning trees on 4000 vertices (rescaled by $n^{-1/3}$): histogram (left), kernel density estimate using a normal distribution with standard deviation 0.19 (right), based on 20,000 observations 21

13 Kernel density estimates using a normal distribution with standard deviation 0.20 for the total length of reduced trees on 2 (blue), 3 (green) and 4 (red) randomly chosen vertices from minimum spanning trees on 4000 vertices (left); corresponding cumulative distribution functions (right) . . 22

14 Right tail of the kernel density estimate (black) of the total length of reduced trees on 4 randomly chosen vertices as seen in Figure 13 and fitted exponential curve (yellow); on original scale (left) and log-scale (right) . 23

15 Log-concave density estimate (green) and maximum likelihood estimate of $Gamma$-model (blue) compared to the normal kernel density estimate (black) and the histogram of simulated data for the total length of reduced trees on 4 randomly chosen vertices from minimum spanning trees on 4000 vertices, based on 20,000 observations 23

16 Shape of the reduced tree on 3 randomly chosen vertices (relabelled $\{1, 2, 3\}$) 25

17 Histograms of segmental lengths (as proportions of total length) in reduced trees on 3 randomly chosen vertices from minimum spanning trees on 4000 vertices with normal kernel density estimate (red) and $Beta(1.52, 3.04)$-density function (blue) (marginal distribution of a $Dirichlet(1.52, 1.52, 1.52)$), based on 28,994 observations 26

18 Shape of the reduced tree on 4 randomly chosen vertices (relabelled $\{1, 2, 3, 4\}$) 28

19 Histograms of segmental lengths (as proportions of total length) in reduced trees on 4 randomly chosen vertices from minimum spanning trees on 4000 vertices with normal kernel density estimate (red) and $Beta$-density functions (blue) (marginal distributions of a $Dirichlet(1.57, 1.57, 1.57, 1.57, 1.24)$), based on 18,173 observations 28

20 Shape of the reduced tree on 5 randomly chosen vertices (relabelled $\{1, 2, 3, 4, 5\}$). These two trees are not genuinely different when we think of them as being unrooted. One tree is just a relabelled version of the other. . 30

21 Normal kernel density estimates for segmental lengths (as proportions of total length) in reduced trees on 5 randomly chosen vertices, based on 17,038 observations 30

22 The two genuinely different shapes of reduced trees on 6 randomly chosen vertices (relabelled $\{1, 2, 3, 4, 5, 6\}$) 32

23 Comparison of total lengths between the two genuinely different shapes of reduced trees on 6 randomly chosen vertices. Histogram for shape 1 for 13,264 observations (left); Histogram for shape 2 for 2415 observations (middle); QQ-Plot for both samples (right) - shapes labelled as in Figure 22. 32

24 Normal kernel density estimates for segmental lengths (as proportions of total length) in reduced trees on 6 randomly chosen vertices, based on 13,264 observations for shape 1 (top) and 2,415 observations for shape 2 (bottom) 33

1 Introduction

A graph consists of a set V, whose elements we call vertices, and a set E of unordered pairs of elements of V, which we refer to as edges. Such a graph, in which every pair of distinct vertices is connected by a unique edge, is called a complete graph. The complete graph on n vertices, labelled by $\{1, 2, \cdots, n\}$, is denoted by K_n, and contains $n(n-1)/2$ edges.

A tree is defined to be a connected graph with no cycles. Therefore, a tree on n vertices has exactly $n-1$ edges. We can think of trees on n vertices as being subgraphs of K_n. In this context, we call them spanning trees. From Cayley's formula we know that there are n^{n-2} distinct trees on n vertices, and hence K_n possesses n^{n-2} spanning trees [5].

Consider a complete graph in which we assign each edge a non-negative weight. Finding the minimum spanning tree (MST), that is, the spanning tree with the minimum sum of weights, is one of the founding problems of combinatorial optimisation.

In this project, we are going to study an i.i.d. random version of the MST problem. For this purpose we assign independent random weights from a uniform distribution between 0 and 1 to the $n(n-1)/2$ edges of the complete graph. Notice that for the MST problem, only the ranking of the edge-weights matters, rather than the exact values. It would therefore make no difference if we assigned i.i.d. weights from any other continuous distribution.

Assigning distinct independent random weights to the $n(n-1)/2$ edges of K_n and extracting the resulting minimum spanning tree, is merely a random selection procedure on the set S_n of all trees on n vertices. Although seemingly similar, this procedure is substantially different from uniform random trees (URT), defined by sampling uniformly at random from S_n.

After the selection of a tree, we assign equal length of 1 to all edges. Naturally, distances on the tree between randomly chosen vertices become larger as n, the number of vertices in the tree, increases. In the case of uniform random trees, distances grow by a factor of $n^{1/2}$. Hence, one would rescale distances by a factor of $n^{-1/2}$. The limiting object then, as

n approaches infinity, is called the Brownian Continuum Random Tree. This random limit object is rather well understood. However, relatively little is known about the minimum spanning tree as n goes to infinity.

We know that distances on the minimum spanning tree grow by a factor of $n^{1/3}$. Recently, it has been shown that the minimum spanning tree of K_n, with edge-lengths rescaled by $n^{-1/3}$, converges towards some random limit object, which belongs to the class of $\mathbb{R}$-trees [1]. We refer to it as the MST limit object and like to think of it as a measured metric space. The sense in which the convergence to the MST limit object occurs is beyond the scope of this study. However, it is important to know that both the convergence and the limit object can be understood via sampling. Given k vertices, the k-reduced tree is defined to be the subtree which only contains the paths between these vertices. If we pick k vertices from the measure on the MST limit object and look at the distribution of the k-reduced tree, then doing this for every k determines the distribution of the limit object. Moreover, the sequence of k-reduced trees for MSTs on finite n converges to the limit k-reduced tree, for every $k \geq 2$, as $n \to \infty$.

Also from [1], we know that reduced trees on k randomly chosen vertices of the MST limit object are binary and have all k vertices on the leaves (almost surely). They consist of $2k - 3$ segmental lengths, defined by the edges between vertices on the leaves and internal vertices at which paths split. The arrangement of these segmental lengths up to relabeling, we refer to as the tree shape. The aim of this study is to investigate other distributional and combinatorial properties by simulating minimum spanning trees for a relatively large number of vertices. We are interested in the total length and the segmental lengths of reduced trees on up to 6 randomly chosen vertices as well as the influence of the tree shape on these quantities.

Section 2 provides some scientific background. We illustrate fundamental concepts such as the MST identified by Prim's algorithm, compare uniform random trees with minimum spanning trees and illustrate some distributional properties of the Brownian Continuum Random Tree. Next, the scientific methods that are used for the simulation and analysis of data on minimum spanning trees are outlined. Finally, results for the simulated data are presented in section 4. We finish with some concluding remarks.

Both the simulation of data as well as the analysis of the simulated data are implemented in R. The accompanying code as well as explanatory remarks for simulation of data and subsequent data analysis can be found in the appendix.

2 Scientific Background

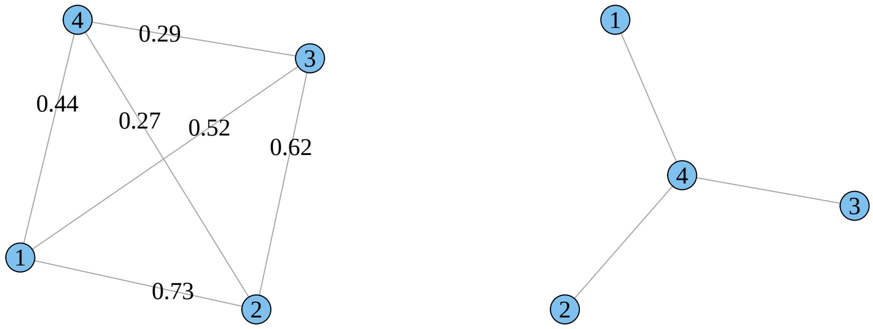
In this section the question under scrutiny is outlined in more detail. Some of the necessary scientific background that is needed to understand the aim and the scope of the study is presented here. We explain Prim's algorithm for finding minimum spanning trees and we emphasise the difference between minimum spanning trees and uniform random trees. Finally, we illustrate some theoretical results for the limiting object of the uniform random tree, as the number of vertices goes to infinity, known as the Brownian Continuum Random Tree. A detailed summary of these can be found in [2]. This problem is similar to the topic that we are concerned with, but, as opposed to the limiting behaviour of minimum spanning trees, it is an area of graph limits that is rather well understood and might therefore give us an idea of where to go and where to look in our analysis of minimum spanning trees.

2.1 Finding the Minimum Spanning Tree

Once independent and uniformly distributed weights between 0 and 1 have been assigned to all edges of the complete graph on n vertices, labelled as $\{1, 2, \cdots, n\}$, several algorithms are at our disposal for indentifying the minimum spanning tree, that is, the tree containing all vertices with the lowest sum of weights. These algorithms include for example Kruskal's algorithm and Prim's algorithm. In this study we make use of Prim's algorithm which provides more computational efficiency and implementation ease. For Prim's algorithm we can think of graphs in the form of matrices, where row i and column j represent the vertices i and j respectively, and the (i, j)-matrix entry represents the weight assigned to the edge between those vertices. The matrix of a complete graph is therefore symmetric and has missing values on the diagonal.

Figure 1 shows a simple example of a complete graph on 4 vertices. All edges have been assigned independent uniform weights between 0 and 1. The equivalent matrix form of the weighted complete graph is given underneath. The corresponding minimum spanning tree is also shown in the figure.

Figure 1: A weighted complete graph on 4 vertices (left) and the corresponding minimum spanning tree (right)



	1	**2**	**3**	**4**
1	-	0.73	0.52	0.44
2	0.73	-	0.62	0.27
3	0.52	0.62	-	0.29
4	0.44	0.27	0.29	-

Prim's algorithm proceeds as follows. At each stage we have a tree. Start from a single vertex and, at each step, add the edge between the current tree and a new vertex of lowest weight. Translated into the matrix based algorithm, we can just start with any column (vertex), say the column labelled 1. From our initial matrix we pick the entry with the lowest value in column 1. In the example of Figure 1, this entry corresponds to the edge between vertex 1 and 4 weighted 0.44. This edge is part of the minimum spanning tree. Next we cancel out all values in row 1 and 4, since these two vertices are now included in the tree:

$$\begin{pmatrix} - & 0.73 & 0.52 & 0.44 \\ \mathbf{0.73} & - & 0.62 & 0.27 \\ \mathbf{0.52} & 0.62 & - & 0.29 \\ \mathbf{0.44} & 0.27 & 0.29 & - \end{pmatrix} \longrightarrow \begin{pmatrix} - & - & - & - \\ 0.73 & - & 0.62 & 0.27 \\ 0.52 & 0.62 & - & 0.29 \\ 0.44 & - & - & - \end{pmatrix}$$

Iteration step: From the respective columns of all vertices that are already included in the tree we pick the entry with the lowest value. We then cancel out all values of the row in which this entry lies. This step must be repeated until all vertices are included in the tree. This algorithm identifies the unique minimum spanning tree. In our example:

$$\begin{pmatrix} - & - & - & - \\ \mathbf{0.73} & - & 0.62 & \mathbf{0.27} \\ \mathbf{0.52} & 0.62 & - & \mathbf{0.29} \\ 0.44 & - & - & - \end{pmatrix} \longrightarrow \begin{pmatrix} - & - & - & - \\ - & - & - & 0.27 \\ \mathbf{0.52} & \mathbf{0.62} & - & \mathbf{0.29} \\ 0.44 & - & - & - \end{pmatrix}$$

$$\longrightarrow \begin{pmatrix} - & - & - & - \\ - & - & - & 0.27 \\ - & - & - & 0.29 \\ 0.44 & - & - & - \end{pmatrix}$$

The minimum spanning tree in our example contains the edges between the vertex pairs $(1,4)$, $(2,4)$ and $(3,4)$ as illustrated in Figure 1. Notice that in the figure we are not interested in the absolute positions of vertices on the plane or the lengths of the edges. For later computations all edges are assumed to have the same rescaled length of $n^{-1/3}$. The assigned weights are not of interest either as soon as the minimum spanning tree has been identified. They are merely a tool for the selection of the tree. The only important piece of information is the existence or nonexistence of an edge between two vertices in the selected tree.

2.2 The Difference between Minimum Spanning Trees and Uniform Random Trees

Assigning independent uniformly distributed weights to the edges of the complete graph on n vertices and identifying the minimum spanning tree is a selection procedure seeking a particular tree that contains all n vertices. Intuitively, one might think that this selection procedure is equivalent to sampling from S_n, the set of all trees on n vertices, uniformly at random, in other words, that the minimum spanning tree is nothing but a uniform random tree. However, the intuition is misleading in this case. In fact, the results turn out to be quite different.

As before we consider a complete graph on 4 vertices, labelled $\{1, 2, 3, 4\}$. Cayley's formula tells us that there are 16 $(= 4^{4-2})$ distinct trees on 4 vertices and therefore there are 16 spanning trees in a complete graph on 4 vertices. Choosing one of these trees uniformly at random leads to a uniform random tree.

Figure 2: Possible shapes for trees on 4 vertices (labelled $\{1, 2, 3, 4\}$): there are 4 combinations with "Star Shape" and 12 combinations for which all vertices lie in a row

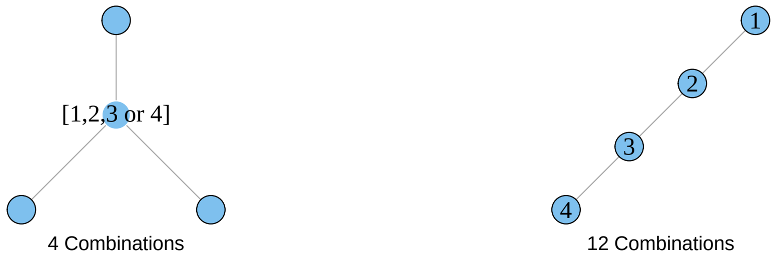


Figure 3: Difference between uniform random trees and minimum spanning trees on 4 vertices: bar shart of the number of selected trees with distinct shape (1 to 4 represent trees with "Star Shape"; 5 to 16 represent trees in which all vertices lie in a row) - 200,000 simulations for each selection procedure

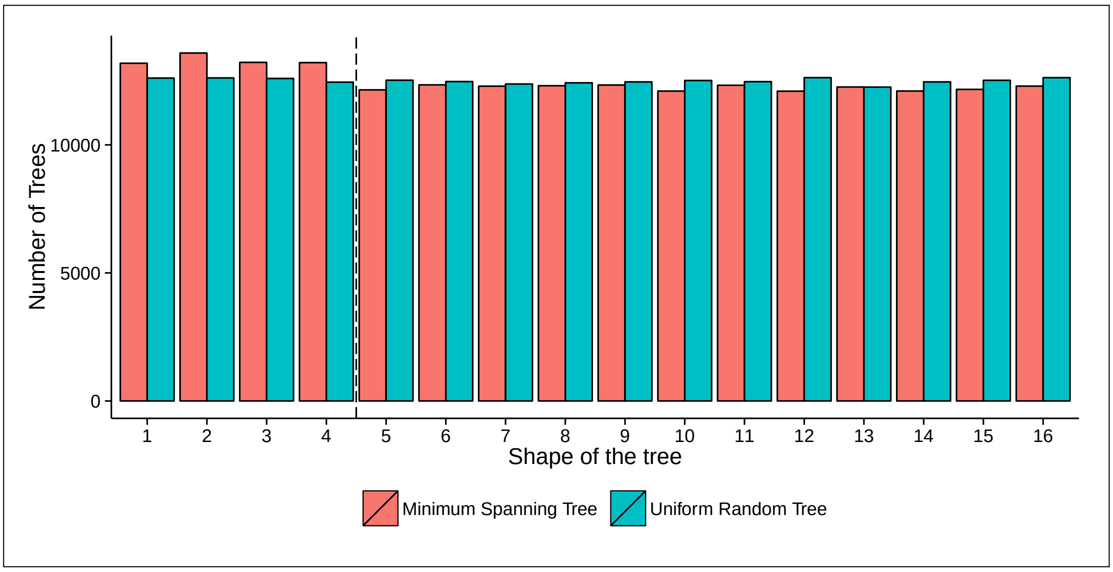


Figure 2 illustrates the two distinct structures that the spanning trees on 4 vertices could adopt; either one vertex is directly connected to all other vertices forming the "star shape", or all vertices lie in a row, that is, each node has at most two edges. For the former, there are obviously 4 different combinations, whereas for the latter, there are 12 distinct combinations. In fact, there are 24 ways of arranging 4 objects in a row. However, in this case, combinations of inverted order are considered to be equivalent ($[1 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 4] \equiv [4 \leftrightarrow 3 \leftrightarrow 2 \leftrightarrow 1]$). Hence, $4!/2 = 12$ distinct combinations are possible.

When selecting a uniform random tree, each of these distinct combinations has equal probability of being chosen. From Figure 3 we can see that this feature is not present in minimum spanning trees. Star shaped trees, in which one vertex is directly connected to all other vertices, have higher probability of being chosen as the Monte Carlo experiment illustrates.

It is in fact easy to demonstrate that this ought to be the case. Remember that the MST is determined by the ranking of the edges, rather than the actual values. There are 6 edges in the complete graph on 4 vertices, $V_n = \{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\}$, where (i,j) denotes the edge that connects the vertices i and j. Hence, there are $6! = 720$ possible rankings, each of which is equally likely to occur in the i.i.d. setting. We show that there are 48 rankings which lead to the star shaped MST in which vertex 1 is directly connected to all other vertices, that is, the tree with the subset of vertices $\{(1,2),(1,3),(1,4)\}$, denoted T_n.

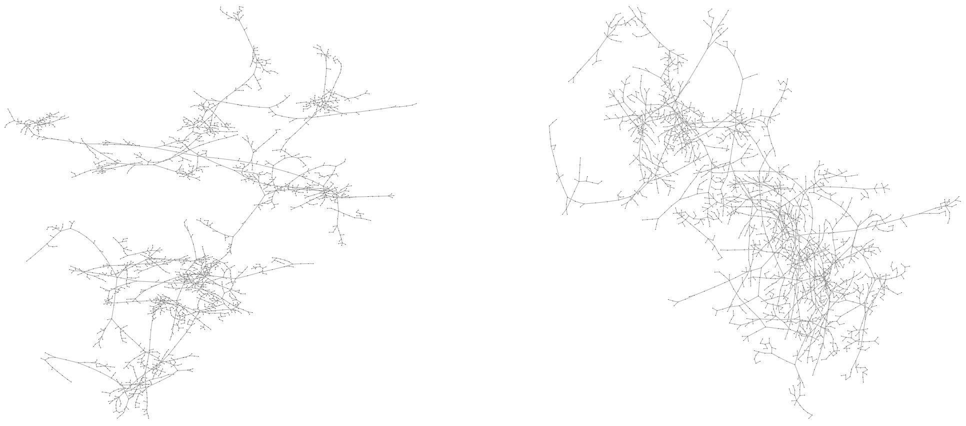
It is easy to see that the first two edges of the ranking (ranked by increasing weight) are always included in the MST. Hence, there have to be 2 out of 3 edges from T_n on the first two ranks. Then, the following two possibilities exist:

A) Obviously, if the remaining third edge is on the third rank, we find the desired MST represented by T_n. For this event there are $3! \times 3! = 36$ possible rankings.

B) It is also possible that the remaining third edge of T_n is on the fourth rank. Then, however, it would be a necessary condition that the unique edge that forms a cycle with the first two edges is on the third rank. For, this edge could not be included in the MST. There are $\binom{3}{2} \times 1 \times 1 \times 2! = 12$ possible rankings for this event. The following ranking would be an example: $(1,2)$, $(1,3)$, $(2,3)$, $(1,4)$, $\cdots$.

There are no other rankings than those provided under **A)** and **B)** that would lead to the desired subset T_n. By symmetry, the probability for any star shaped MST to occur is thus $48/720$. Yet, $48/720 = 1/15 > 1/16$. We have thereby shown that star shaped trees are more likely to occur than trees for which all vertices lie in a row.

Figure 4: A unniform random tree (left) and a minimum spanning tree (right) on 3000 vertices



This proof as well as its empirical illustration correspond to the fact that the distances on minimum spanning trees, as $n \to \infty$, grow by a factor of $n^{1/3}$, whereas the distances on uniform random trees grow faster, by a factor of $n^{1/2}$. Roughly speaking, uniform random trees tend to be more stretched than minimum spanning trees. This is also illustrated in Figure 4.

2.3 Illustration of Theoretical Results for Uniform Random Trees

Consider a uniform random tree on n vertices, labelled $\{1, 2, \cdots, n\}$. As n approaches infinity and distances of the tree are rescaled by $n^{-1/2}$, the tree converges to the so-called Brownian Continuum Random Tree. It satisfies the following conditions [2]:

1. the reduced tree on k randomly chosen vertices is a binary tree with all chosen vertices on the leaves (almost surely). The tree shape of the k-reduced tree is uniformly distributed on the set of binary trees with k labelled leaves,

2. the $2k - 3$ segmental lengths of the reduced tree follow a joint probability distribu-

tion of the form:

$$f(\ell_1, \cdots, \ell_{2k-3}) = \left(\sum_{i=i}^{2k-3} \ell_i \right) \exp \left(-\frac{1}{2} \left(\sum_{i=i}^{2k-3} \ell_i \right)^2 \right),$$

3. points 1. and 2. are independent.

Consider for example Figure 16 on page 25 or Figure 18 on page 28 as an illustration of the $2k - 3$ segmental lengths of a k-reduced tree for $k = 3$ and 4 respectively.

In the following we try to illustrate these theoretical results by simulation of uniform random trees on 2000, 3000 and 4000 vertices. We pick $k = 2$ and 3 of those vertices uniformly at random and compute the $2k - 3$ (1 for $k = 2$ and 3 for $k = 3$) resulting distances of the reduced tree. Instead of considering only the segmental lengths, we will also focus on the total length of the reduced trees, that is, the sum of all segmental lengths. The density function against which we compare the simulated total lengths can be derived from the above expression (joint density of segmental lengths) as follows.

Let $n = 2k-3$ and think about doing the change of variables $x_1 = \ell_1, x_2 = \ell_2, \ldots, x_{n-1} = \ell_{n-1}, T = \sum_{i=1}^{n} \ell_i$. This transformation has Jacobian 1 and so we conclude that the joint density of the new variables is $Te^{-T^2/2}$. But now we need to integrate out the variables $x_1, x_2, \ldots, x_{n-1}$ under the constraint that $x_1 + x_2 + \ldots + x_{n-1} \leq T$. It is clear that the answer has to be proportional to T^{n-1}. Hence, the marginal density in T is proportional to:

$$T^n e^{-T^2/2} = T^{2k-3} e^{-T^2/2}.$$

One can then check that the correct normalising constant to make this a density that integrates to 1 must be:

$$1/(2^{k-2}(k-2)!).$$

This is, in fact, the density of the square root of a Gamma random variable, $\Gamma(k-1, 1/2)$. Hence, in the cases of $k = 2$ and $k = 3$, we compare our simulations of reduced trees with total length T against the following two density functions respectively:

$$Te^{-T^2/2} \text{ and } \frac{1}{2}T^3e^{-T^2/2}.$$

Figure 5: Histograms of total length (distance) between 2 randomly chosen vertices from uniform random trees on 2000, 3000 and 4000 vertices (rescaled by $n^{-1/2}$) and limiting distribution $Te^{-T^2/2}$ (red)

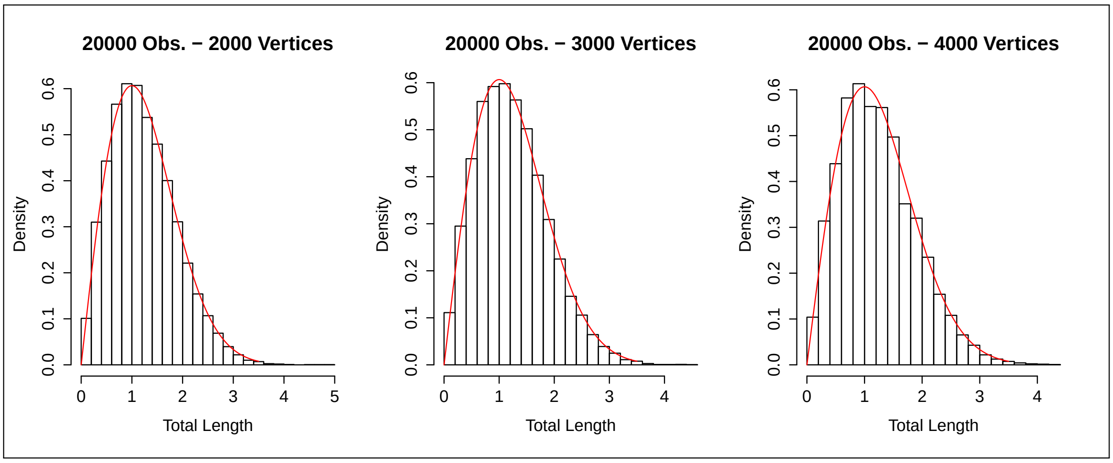


Figure 6: Histograms of total length of reduced trees on 3 randomly chosen vertices from uniform random trees on 2000, 3000 and 4000 vertices (rescaled by $n^{-1/2}$) and limiting distribution $\frac{1}{2}T^3e^{-T^2/2}$ (red)

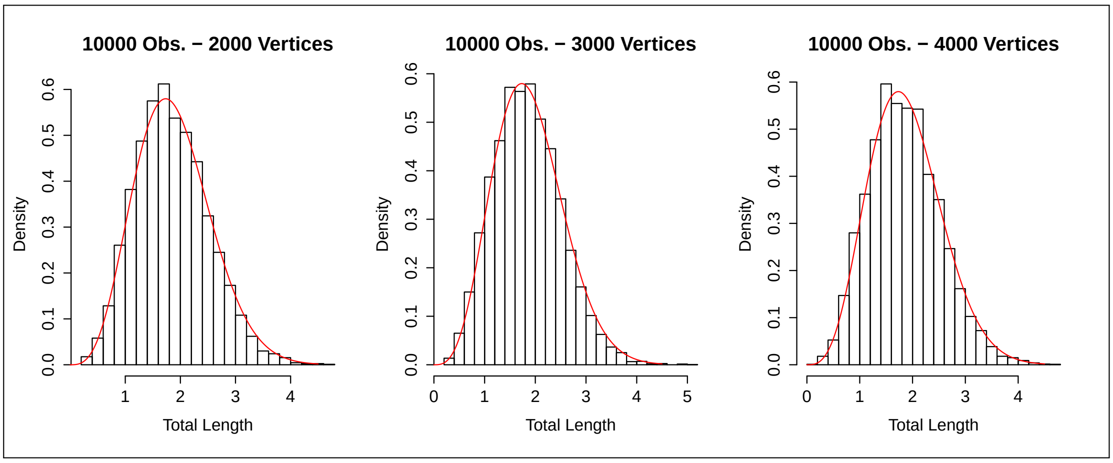


Figure 5 contains histograms of the rescaled (by $n^{-1/2}$) lengths between two randomly chosen vertices from uniform random trees on 2000, 3000 and 4000 vertices. We can see that the empirical results based on 20,000 simulations concur with the overlaid theoretical density.

Figure 6 shows histograms of the total length of the reduced trees on 3 randomly chosen vertices. For 10,000 simulations the empirical results are again in close agreement with the theoretical density.

Figure 7: Histograms of segmental lengths (as proportions of total length) in reduced trees on 3 randomly chosen vertices from uniform random trees on 4000 vertices and limiting distribution $Dirichlet(1, 1, 1)$ **- marginal distirbutions,** $Beta(1, 2)$**, are plotted over each histogram (red)**

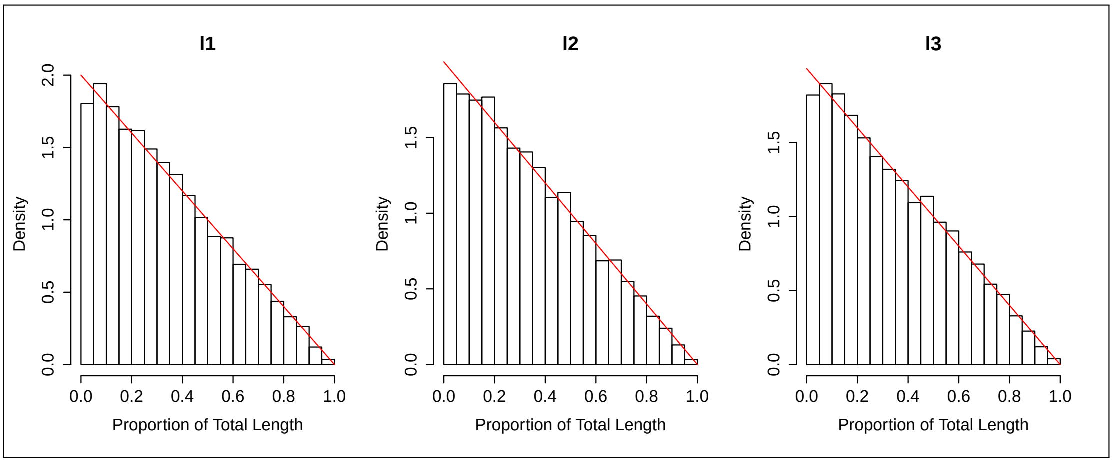


Figure 7 shows histograms of proportions of the total length for the 3 segmental lengths of reduced trees on 3 randomly chosen vertices. These proportions follow a $Dirichlet(1, 1, 1)$, that is, the proportions (l_1, l_2, l_3) are uniformly distributed on the 2-dimensional simplex. Their marginal distributions are therefore $Beta(1, 2)$ for each proportion. Use of 10,000 simulations of uniform random trees on 4000 vertices seems sufficient to observe the asymptotic distribution.

The analysis implemented above serves as a prototype of what follows in section 4 on the properties of the MST limit object.

3 Summary of Methods to be used

This section contains a brief discussion of the methods and tools of analysis to be used in this study. Firstly, we describe how minimum spanning trees are simulated. Secondly, the exploration of simulated trees is outlined. Lastly, two non-parametric density estimation techniques, namely kernel smoothing and log-concave density estimation, as well as two parametric approaches are introduced.

3.1 The Simulation of Trees

In order to simulate a minimum spanning tree on n vertices, labelled $\{1, 2, \cdots, n\}$, we start with the complete graph on n vertices, that is, an undirected graph in which every pair of distinct vertices is connected by a unique edge. We assign independent uniformly distributed weights between 0 and 1 to each of the $n(n-1)/2$ edges in the complete graph and arrange them in matrix form as described in section 2.1. Then, Prim's algorithm can be applied to the complete graph in matrix form in order to identify the minimum spanning tree. This is computationally straightforward.

In R the independent uniform (pseudo) random numbers between 0 and 1 are by default generated using the Marsenne twister MT-19937 [16], a pseudo random number generator developed by Makoto Matsumoto and Takuji Nishimura in 1997 [9]. It is based on the Marsenne prime number $2^{19937} - 1$.

Once these pseudo random numbers are assigned to the edges of the complete graph and the minimum spanning tree is identified using Prim's algorithm, the only information needed for subsequent analysis is which vertices are directly connected by the $n-1$ edges of the MST. This information can be stored in the form of $n - 1$ pairs of labels of the respective vertices that are directly connected. This can be done by storing information as graph objects, which is a special class of objects in the *igraph*-package in R. Note that the generated weights are no longer of interest.

A large set of data of different tree sizes (2000, 3000 and 4000 vertices) was simulated. For the gathering of each observation, we simulated a new minimum spanning tree. Up to

30,000 repetitions are computationally very expensive and can take more than 48 hours even when multiple core processors are used. For this study a computer with quad-core Intel(R) Core(TM) i7-3770S CPU @ 3.10GHz processor and 8GB RAM has been used. Selected parts of the R-code that generated the data can be found in the appendix. The entire code and data sets are available on request.

3.2 The Exploration of Trees

Once a minimum spanning tree is identified, we are interested in the distances between randomly chosen vertices on the tree. We can think of the tree as a measured metric space, $\mathbb{M}^n$, in which distances are rescaled by $n^{-1/3}$ and mass $1/n$ is assigned to each vertex [1, p. 5].

We are interested in the combinatorial structure and the distribution of distances in reduced trees on k randomly chosen vertices from minimum spanning trees. We start with $k = 2$, that is, the simple question: how is the distance between two randomly chosen vertices on the minimum spanning tree distributed? By definition, on a tree there is only one simple path between any two vertices. Imagine that on a specific tree there are 4 vertices on the unique path between two randomly chosen vertices. That means the length between those two vertices is 5, since one had to traverse 5 edges to reach the other vertex. In the rescaled metric of $\mathbb{M}^n$ this distance would then be $5 \times n^{-1/3}$.

As in the case of uniform random trees, the reduced tree on k vertices of a minimum spanning tree is binary in the limit and has the k vertices on the leaves (almost surely) [1]. As k gets larger, naturally, the total length of the reduced tree grows. However, given that the tree is binary in the limit, for $k \leq 5$ there is only one genuinely distinct structural shape for the reduced (unrooted) tree. For $k = 6$ there exist two genuinely distinct structural shapes as illustrated in the following section in Figure 22 on page 32. In fact, as we work with trees on up to 4000 vertices as an approximation to the limiting object, we will sometimes find reduced trees on randomly chosen vertices that are not binary or in which some of the chosen vertices are not on the leaves. When considering the proportions of segmental lengths of the total length of the reduced tree, we will ignore these cases. Theoretically speaking, we therefore examine the distribution of these proportions conditioned on the event that the reduced tree is indeed binary and has the chosen vertices on the leaves, that is, an event which has probability 1 in the limit.

3.3 Probability Density Estimation

In order to investigate the distribution of the total length and the proportions of the segmental lengths of the reduced trees on up to $k = 6$ vertices, we make use of different non-parametric and parametric approaches.

Kernel Density Estimation

Without imposing the constraints of prescribing a parametric model, kernel density estimation techniques provide a way of finding structure in data. For a simulated data set, $X_1, X_2 \cdots, X_m$, we fit a smooth density function according to the following formula [15, p. 11]:

$$\hat{f}_\sigma(x) = m^{-1} \sum_{i=1}^{m} K_\sigma(x - X_i) = (m\sigma)^{-1} \sum_{i=1}^{m} K((x - X_i)/\sigma),$$

where the *kernel* K is a function that integrates to 1, $\int K(x)dx = 1$. We choose the kernel to be the density of a standard normal distribution. We thus call it the normal kernel. K_σ is the rescaled kernel, defined as $K_\sigma(x) = \sigma^{-1}K(x/\sigma)$, where σ is called the *bandwidth*.

The choice of the bandwidth is essential. On the one hand, one doesn't want to smear out real peaks, and on the other hand, we would like to get rid of insignificant noise. In mathematical terms, there is a trade-off between bias and variance of $\hat{f}_\sigma(x)$. The larger σ is chosen to be, the larger the bias of the estimate and the smaller its variance.

In our analysis, σ is chosen to minimize an estimation of the *mean integrated square error*:

$$MISE = \mathbb{E}\int [\hat{f}_\sigma(x) - f(x)]^2 dx,$$

which after some computation leads to the optimal bandwidth [14, pp. 128-129]:

$$\sigma_{AMISE} = \left(\frac{\int K^2}{m\int (f'')^2 \{\int x^2 K\}^2}\right)^{1/5}.$$

This corresponds to the *direct plug-in estimator* as described in Sheather and Jones (1991) and Wand and Jones (1995) [14, p. 129] and it involves the integral of an unknown function, $\int (f'')^2$. This requires knowledge of the very function we want to estimate. Using a reference bandwidth, we estimate $\int (f'')^2$. This bandwidth for estimating $\int (f'')^2$ is taken as a function of σ (proportional to $\sigma^{5/7}$) [13–15]. This concludes the discussion on how the bandwidths of normal kernel density estimates are selected in this study.

Log-Concave Density Estimation

For kernel density estimations, no restrictive assumptions are made about the shape of the underlying distribution. We will see, however, that based on the kernel estimates some reasonable assumptions can be made in order to narrow the class of distributions we are dealing with. One of these assumptions is log-concavity. A density function on $\mathbb{R}$ is log-concave if and only if:

$$f(x) = \exp(\phi(x)),$$

for some concave function $\phi : \mathbb{R} \to [-\infty, \infty)$. Given a sample of i.i.d. random variables, $X_1, X_2, \cdots, X_m$, we maximize the normalized log-liklihood function, which is a functional in this setting [12, p. 2]:

$$l(\phi) = m^{-1}\sum_{i=1}^{m} \log(f(X_i)) = m^{-1}\sum_{i=1}^{m} \phi(X_i),$$

over all concave functions $\phi : \mathbb{R} \to [-\infty, \infty)$, such that $\int \exp(\phi(x))dx = 1$. The resulting maximum likelihood estimates $\hat{\phi}$ and $\hat{f} = \exp(\hat{\phi})$ are unique and consistent [3,7].

The maximum likelihood estimator over the whole class of unimodal density functions does not exist [4, 12]. However, it exists for log-concave density functions which are a subclass of unimodal density functions. This is one advantage of applying log-concave density estimation. It can be a useful restriction on the class of possible density functions. The implementation of log-concave density estimation in this study is based on the R-package *logcondens* developed by Rufibach and Dümbgen [11]. An example for a log-concave density function would be the density of a $Gamma$-distribution. The $Gamma$-model is one of the parametric approaches used in this study.

Parametric Models

Two parametric approaches are implemented in this study. Firstly, for the distribution of the total length of the reduced tree on 4 vertices, a $Gamma$-distribution, $f(x, k, \theta) = \frac{\theta^k}{\Gamma(k)} x^{k-1} e^{-\theta x}$ with shape parameter k and scale parameter θ, is fitted. The log-likelihood for this model is given by:

$$l(x_1, x_2, \cdots, x_m | k, \theta) = (k-1) \sum_{i=1}^{m} x_i - \theta \sum_{i=1}^{m} x_i - m \log(\Gamma(k)) + mk \log(\theta).$$

The maximum likelihood estimate for θ is easily found to be $\hat{\theta}_{MLE} = k/(m^{-1} \sum_{i=1}^{m} x_i)$. However, there is no closed form solution for $\hat{k}_{MLE}$. An iterative quasi-Newton method is implemented in order to solve the equations. The method was published simultaneously in 1970 by Broyden, Fletcher, Goldfarb and Shanno and is specified by “BFGS” in *R* [10,17].

Secondly, a $Dirichlet$-distribution is fitted to the proportions of the total length of each segmental length in reduced trees on $k = 3, 4, 5$ and 6 vertices. A reduced tree on 3 vertices (if it is binary and has the three vertices on the leaves) has 3 segments as illustrated in Figure 16 on page 25. The proportions of the total length of these segments sum up to 1. Hence, the $Dirichlet$-distribution for 3 (4, 5 or 6 respectively) variables as a generalization of the $Beta$-distribution is a reasonable choice. Its density is given by:

$$f(p_1, p_2, p_3|\alpha_1, \alpha_2, \alpha_3) = \frac{1}{B(\alpha)} \prod_{i=1}^{3} p_i^{\alpha_i - 1},$$

where B is the normalizing $Beta$-function, defined as $B(\alpha) = \frac{\prod_{i=1}^{3} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{3} \alpha_i)}$ and the exponents α_i for $i \in \{1, 2, 3\}$ are the shape parameters of the distribution. The likelihood for this model for m observations is given by:

$$L(P|\alpha_1, \alpha_2, \alpha_3) = \prod_{k=1}^{m} \left(\frac{1}{B(\alpha)} \prod_{i=1}^{3} p_{i,k}^{\alpha_i - 1} \right).$$

The log-likelihood can be written as:

$$l(P|\alpha_1, \alpha_2, \alpha_3) = \sum_{k=1}^{m} \left(\sum_{i=1}^{3} (\alpha_i - 1) \log(p_{i,k}) \right) - m \log(B(\alpha)).$$

The first order conditions on α_i, $i \in \{1, 2, 3\}$ are given by:

$$\frac{\partial}{\partial \alpha_i} l(P|\alpha_1, \alpha_2, \alpha_3) = \sum_{k=1}^{m} \log(p_{i,k}) - m \frac{\partial}{\partial \alpha_i} \log(B(\alpha)) \overset{!}{=} 0.$$

In order to find a numerical solution for the maximum likelihood estimates of these parameters the “BFGS” method is used as in the $Gamma$-model described above [8, p. 10].

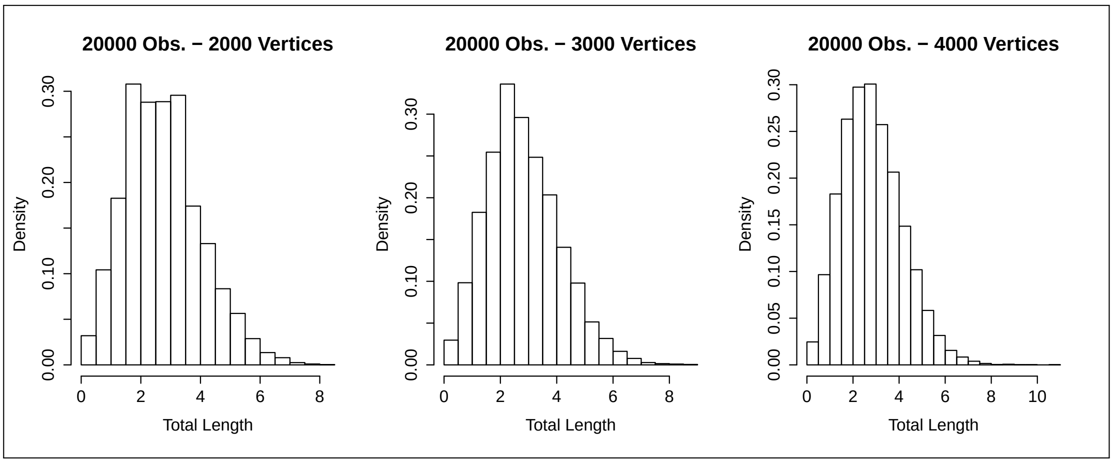
4 Detailed Analysis of Simulated Data

In this section the simulated data is analysed in detail. We start with the distance between 2 randomly chosen vertices. Thereafter, the total lengths of reduced trees on 3 and 4 randomly chosen vertices are examined. Finally, we focus on the combinatorial structure of reduced trees on up to 6 randomly chosen vertices. We apply non-parametric approaches for density estimation such as kernel smoothing and log-concave density estimation, as well as parametric models such as the $Gamma$-model for the total length and a $Dirichlet$-regression for the proportions of segmental lengths.

4.1 The Total Length of Reduced Trees on up to 4 Randomly Chosen Vertices

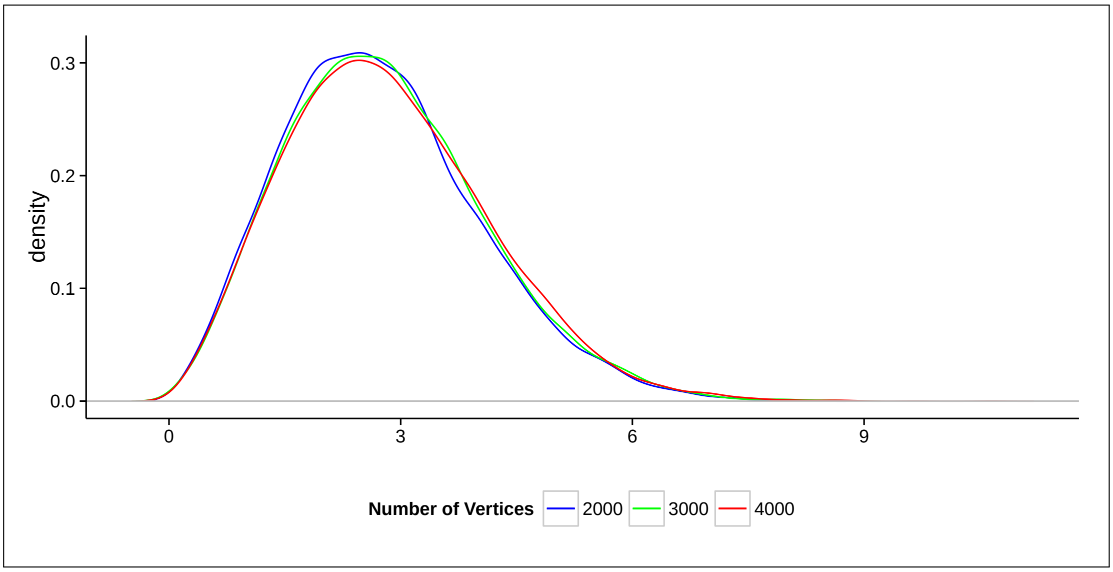
Once a minimum spanning tree is simulated we choose 2 vertices uniformly at random and calculate the distance of the unique path from one vertex to the other on the tree. As we wish to approximate the limit object of the rescaled MST, it is important to simulate trees on as many vertices as possible. However, the need for accuracy is limited by computational capacity. Figure 8 shows three histograms for the rescaled distance between 2 randomly chosen vertices for minimum spanning trees on 2000, 3000 and 4000 vertices respectively.

Figure 8: Histograms of total length (distance) between 2 randomly chosen vertices from minimum spanning trees on 2000, 3000 and 4000 vertices (rescaled by $n^{-1/3}$)



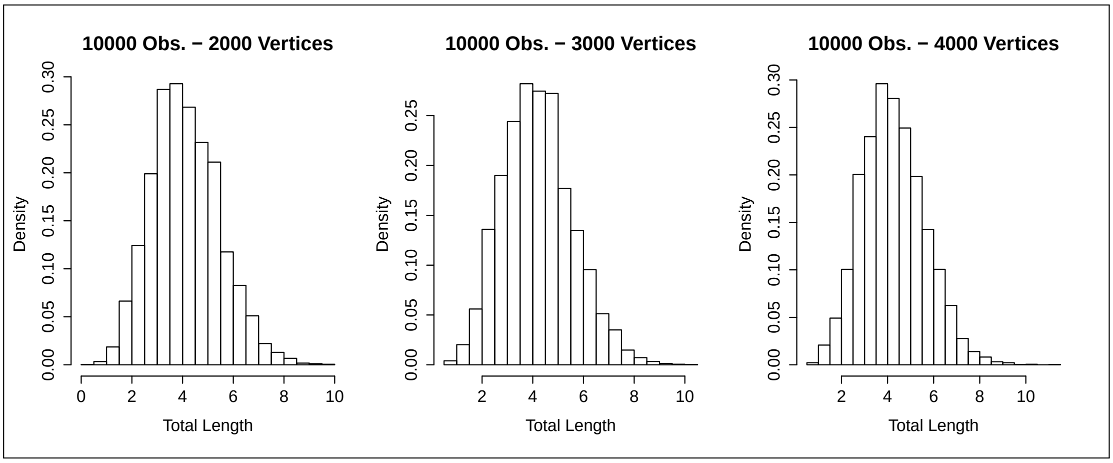
We can see that as the number of vertices in the tree grows, the distribution becomes more regular. The first important result that we can infer from this figure is that the distance between two randomly chosen vertices seems to follow a unimodal distribution. There are no unusual peaks or bumps observed in the histograms.

Figure 9: Kernel density estimates using a normal distribution with standard deviation 0.17 for total length (distance) between 2 randomly chosen vertices from minimum spanning trees on 2000 (blue), 3000 (green) and 4000 (red) vertices.



The kernel density estimates plotted in Figure 9 also illustrate the seemingly unimodal form of the distribution. For the different tree sizes the distributions do not vary substantially. However, we can see that for fewer vertices the kernel estimates become less regular.

Figure 10: Histograms of total length of reduced trees on 3 randomly chosen vertices from minimum spanning trees on 2000, 3000 and 4000 vertices (rescaled by $n^{-1/3}$)



The histograms and the normal kernel density estimates for the total length of reduced trees on 3 vertices are presented in Figure 10 and Figure 11. We observe very similar distributional properties. Again, the simulated data suggests that the total length follows a unimodal distribution. In the following, only data for minimum spanning trees on 4000 vertices is considered, as we want to avoid noise because of being too far from convergence to the limit object.

Figure 11: Kernel density estimates using a normal distribution with standard deviation 0.21 for the total length of reduced trees on 3 randomly chosen vertices from minimum spanning trees on 2000 (blue), 3000 (green) and 4000 (red) vertices.

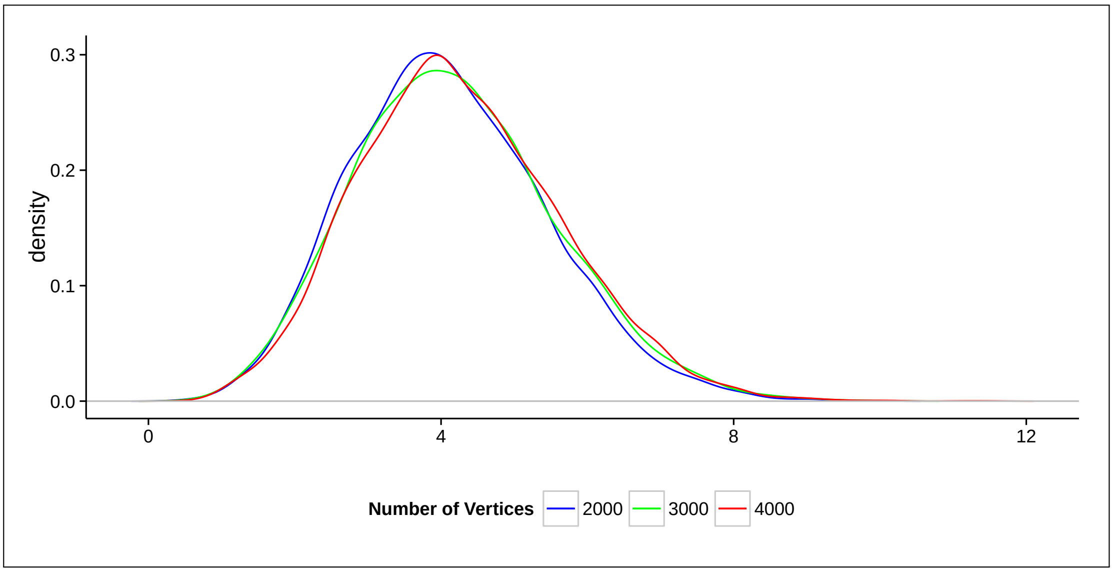


Figure 12: Total length of reduced trees on 4 randomly chosen vertices from minimum spanning trees on 4000 vertices (rescaled by $n^{-1/3}$): histogram (left), kernel density estimate using a normal distribution with standard deviation 0.19 (right), based on 20,000 observations

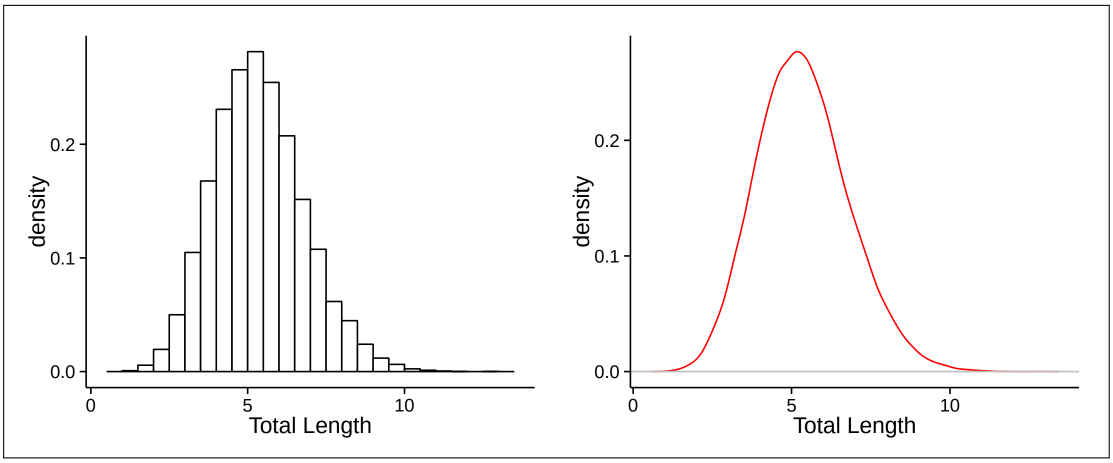
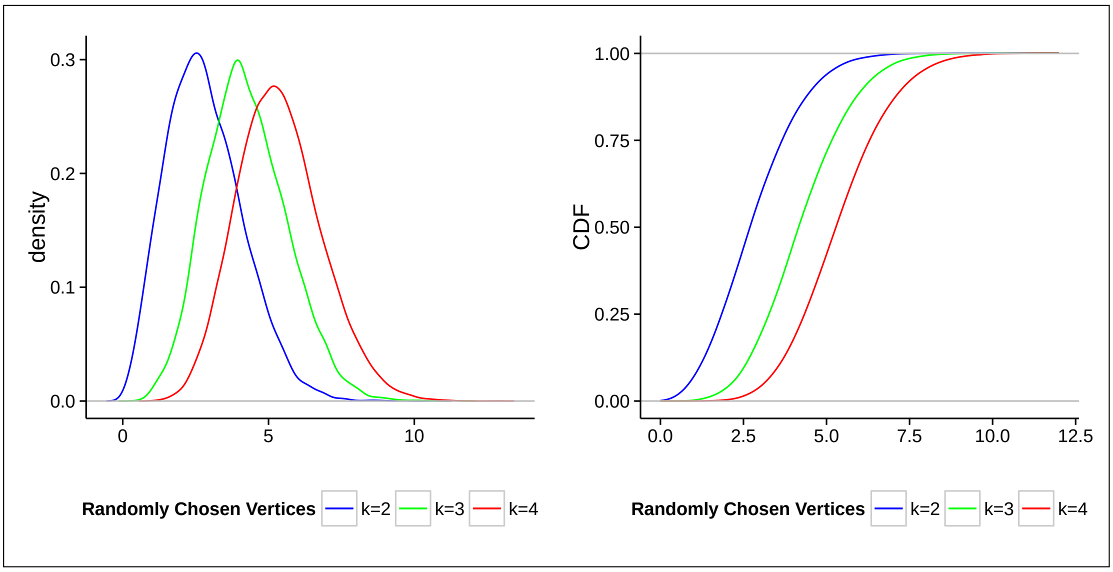


Figure 12 contains a histogram and a normal kernel density estimate for the reduced tree on 4 randomly chosen vertices. For all three previous cases, the total length of subtrees on 2, 3 and 4 vertices, we observe fairly similar unimodal distributions that are shifted towards higher values as the number of chosen vertices increases. This natural tendency is illustrated in Figure 13 which also contains normal kernel estimates of the cumulative distribution function.

Figure 13: Kernel density estimates using a normal distribution with standard deviation 0.20 for the total length of reduced trees on 2 (blue), 3 (green) and 4 (red) randomly chosen vertices from minimum spanning trees on 4000 vertices (left); corresponding cumulative distribution functions (right)



A second important property that we can infer from the kernel density estimates is that the right tails seem to be exponential. In fact, the following exponential curve model:

$$\hat{f}(x_i) = \beta_1 \exp(-\beta_2 x_i) + \epsilon_i$$

provides a very good fit to the right tail of the kernel density estimate for the total length of the reduced tree on 4 vertices as plotted in Figure 13. Heuristically, we define the right tail as the area where the rescaled total length is higher than 8.5. The non-linear least squares estimates for the parameters of the exponential curve on this area are $\beta_1 = 4487.00$ and $\beta_2 = 1.39$. They have standard errors of 233.800 and 0.006 respectively and both p-values are less than 0.001. The residual standard error is 0.0002 on 99 degrees of freedom. For the estimation of the parameters, 100 evenly spaced data points, in steps of 0.025 along the x-axis, have been used.

Figure 14: Right tail of the kernel density estimate (black) of the total length of reduced trees on 4 randomly chosen vertices as seen in Figure 13 and fitted exponential curve (yellow); on original scale (left) and log-scale (right)

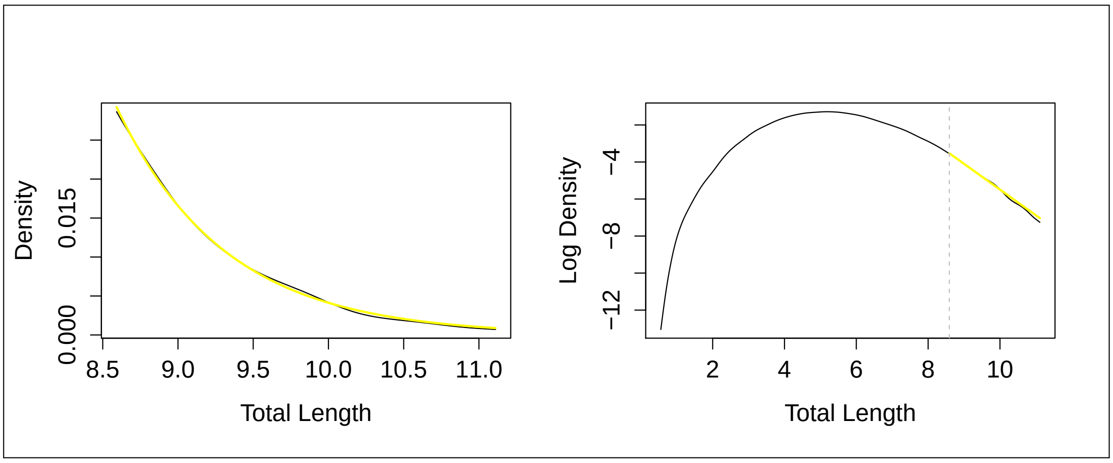


Figure 15: Log-concave density estimate (green) and maximum likelihood estimate of $Gamma$-model (blue) compared to the normal kernel density estimate (black) and the histogram of simulated data for the total length of reduced trees on 4 randomly chosen vertices from minimum spanning trees on 4000 vertices, based on 20,000 observations

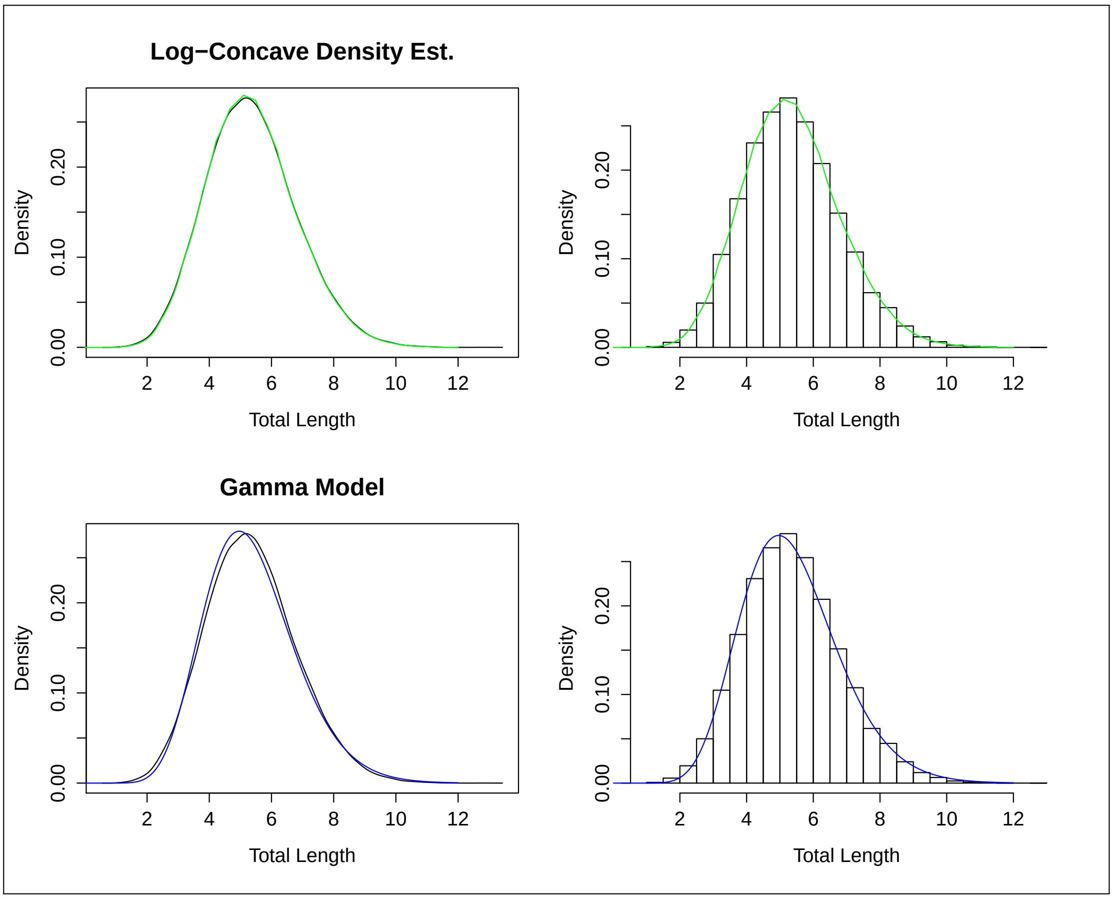


Figure 14 illustrates the goodness of fit for the exponential curve model. On a log-scale we can see that the right tail of the density is almost linear. We also notice from Figure 14 that the log-density appears to be concave which gives us access to more specific tools of density estimation, known as log-concave density estimation.

As described in section 3.3 the functional (normalized log-likelihood) has been maximized over the set of all concave functions $\phi : \mathbb{R} \to [-\infty, \infty)$, such that $\int \exp(\phi)dx = 1$. The result is shown in the first row of Figure 15. We can see that the kernel density estimate and the log-concave density estimate are almost identical. This observation indicates that the assumption of log-concavity is indeed supported by the simulated data.

We can push the analysis one step further by implementing a parametric model. Since we are concerned with a positive random variable, given its log-concavity and overall shape, the $Gamma$-distribution seems to be an intuitive choice. A maximum likelihood approach leads to estimates of the shape parameter of $\hat{k} = 13.235$ and the scale parameter of $\hat{\theta} = 2.467$, in the parametrisation of section 3.3. Their standard deviations are given by 0.107 and 0.020 respectively. Since the $Gamma$-density meets the regularity conditions, for instance formulated in [6, p. 118], we may use normal approximation to compute the following 95% confidence intervals for the estimates:

$$CI_k^{95\%} = (13.026, 13.444) \text{ and } CI_\theta^{95\%} = (2.427, 2.506).$$

From the two lowermost plots in Figure 15 we can see that the estimated $Gamma$-density, $\Gamma(13.235, 2.467)$, does not match the kernel density estimate as accurately as the log-concave density estimate. We can see that the kernel density estimate is slightly more skewed to the right in the highest density area. However, the simulated data suggests that the $Gamma$-distribution can serve as a useful approximation to the distribution of rescaled total lengths of reduced trees on 4 randomly chosen vertices.

4.2 The Structure and Shape of Reduced Trees on up to 6 Randomly Chosen Vertices

We will now focus on the combinatorial structure and shape of the reduced trees on up to 6 randomly chosen vertices from minimum spanning trees on 4000 vertices. As mentioned before, the reduced trees satisfy two properties in the limit [1]:

1. they are binary trees,
2. they have the chosen vertices on the leaves (almost surely).

Therefore, especially for low numbers of chosen vertices, there is only a relatively small set of possible shapes for reduced trees. However, as we are only working with an approximation of the MST limit object, we will always find reduced trees of other shapes. These will be neglected in the following analysis. This means that we are in fact investigating distributions conditional on the event that the reduced trees satisfy the conditions 1 and 2 - an event that has probability 1 in the limit.

Reduced Trees on 3 Randomly Chosen Vertices

Figure 16: Shape of the reduced tree on 3 randomly chosen vertices (relabelled $\{1, 2, 3\}$**)**

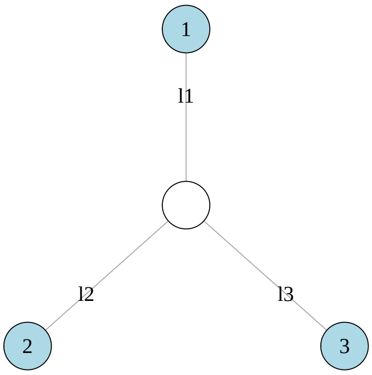
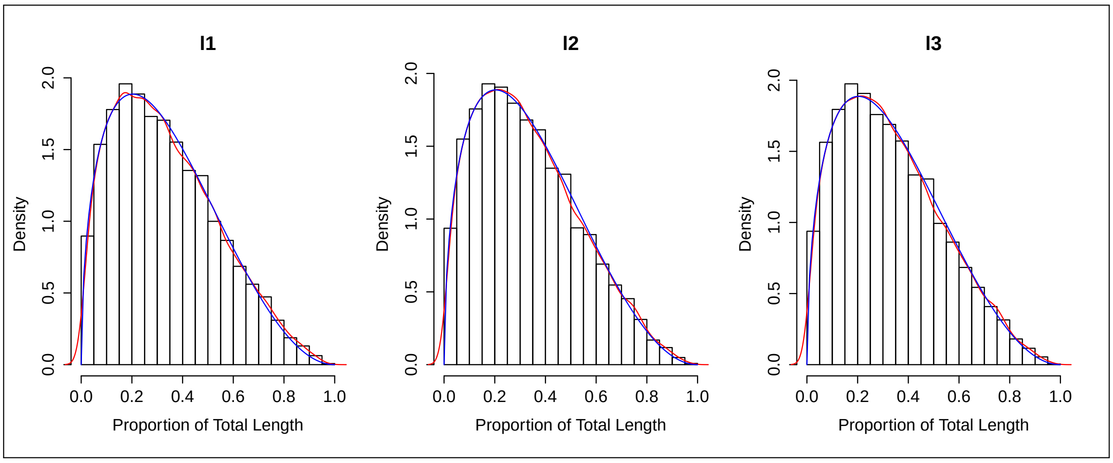


Figure 16 shows the only reduced tree for 3 chosen vertices that satisfies the conditions 1 and 2. As we can see the reduced tree on 3 vertices has 3 segmental lengths that sum up to the total length. We denote l_i the length next to vertex i. A reduced tree on k vertices would have $2k-3$ segmental lengths as illustrated later. We will now investigate the proportions of these segments, that is, the percentage of the total length for each individual segment of the tree.

Evidently, these proportions sum up to 1. By symmetry, in the case of 3 randomly chosen vertices, one would also assume that these proportions all follow the same distribution. Hence, an intuitive model for these dependent distributions would be a $Dirichlet$-distribution with equal shape parameters.

Figure 17: Histograms of segmental lengths (as proportions of total length) in reduced trees on 3 randomly chosen vertices from minimum spanning trees on 4000 vertices with normal kernel density estimate (red) and $Beta(1.52,$ $3.04)$-density function (blue) (marginal distribution of a $Dirichlet(1.52,$ $1.52,$ $1.52)$), based on 28,994 observations



There are several details of Figure 17 that are worth mentioning. Firstly, it contains the histograms of the proportions for 28,994 simulated observations. In fact, 30,000 observations have been simulated, but 1,006 reduced trees did not satisfy the two before mentioned conditions and were removed from the data set. These form only a small proportion of 3.35$\%$ of the whole data set. Secondly, the kernel density estimates are overlaid in red colour. Thirdly, the density function of a $Beta(1.52, 3.04)$-distribution is plotted on each histogram in blue.

We can see that the distributions of the proportions indeed seem to be equal and that the parametric density estimate is close to the kernel density estimate. The coefficients for the $Beta$-distribution have been chosen based on the result of a maximum likelihood estimation for a $Dirichlet$-distribution. The marginal distributions of $Dirichlet$-distributed proportions are $Beta$-distributed. In fact, the maximum likelihood estimates for the three shape parameters are $\alpha_1 = 1.530$, $\alpha_2 = 1.520$ and $\alpha_3 = 1.514$ for the three lengths respectively. Hence, the choice of plotting the density functions of a $Beta(1.52, 3.04)$-distribution is made. Strictly speaking, the marginals are $Beta(\alpha_i, \sum_j \alpha_j - \alpha_i))$. The 95% confidence intervals for the estimates based on normal approximations have an intersection of $(1.512, 1.533)$ that contains 1.52.

The precise 95% confidence intervals are given by:

$$CI^{95\%}_{\alpha_1} = (1.512, 1.549),\ \ CI^{95\%}_{\alpha_2} = (1.502, 1.539)\ \text{ and }\ CI^{95\%}_{\alpha_3} = (1.496, 1.533).$$

The p-value of the Kolmogorov-Smirnov test for any two of the three proportions is well above 5%, indicating that we cannot reject the null hypothesis of identical distributions. In the case of 3 randomly chosen vertices from the minimum spanning tree, we can therefore suggest that the segmental lengths are exchangeable, just like in uniform random trees. However, the proportions of these lengths are not uniformly distributed on the 2-dimensional simplex. This is a difference compared to uniform random trees.

Reduced Trees on 4 Randomly Chosen Vertices

As for 3 vertices, there is only one possible reduced tree shape for 4 randomly chosen vertices that is binary and has the chosen vertices on the leaves. This is shown in Figure 18. Again, by symmetry, we would expect to find equal distributions for the lengths of the 4 external edges, l_1, l_2, l_3 and l_4. However, there is no obvious reason why the length of the internal edge, l_5, should have the same distribution. Surprisingly, it is the case for uniform random trees that all these lengths follow the same distribution and are exchangeable, but as Figure 19 reveals, this might not be the case for minimum spanning trees.

20,000 observations have been simulated for the reduced tree on 4 randomly chosen vertices, 1,827 of which had other shapes than the one shown in Figure 18. These form 9.14% of the simulated observations, which is substantially more than in the previous case. As

Figure 18: Shape of the reduced tree on 4 randomly chosen vertices (relabelled $\{1, 2, 3, 4\}$)

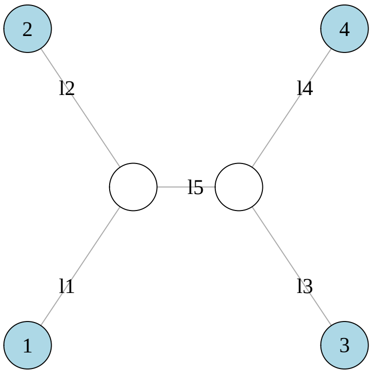
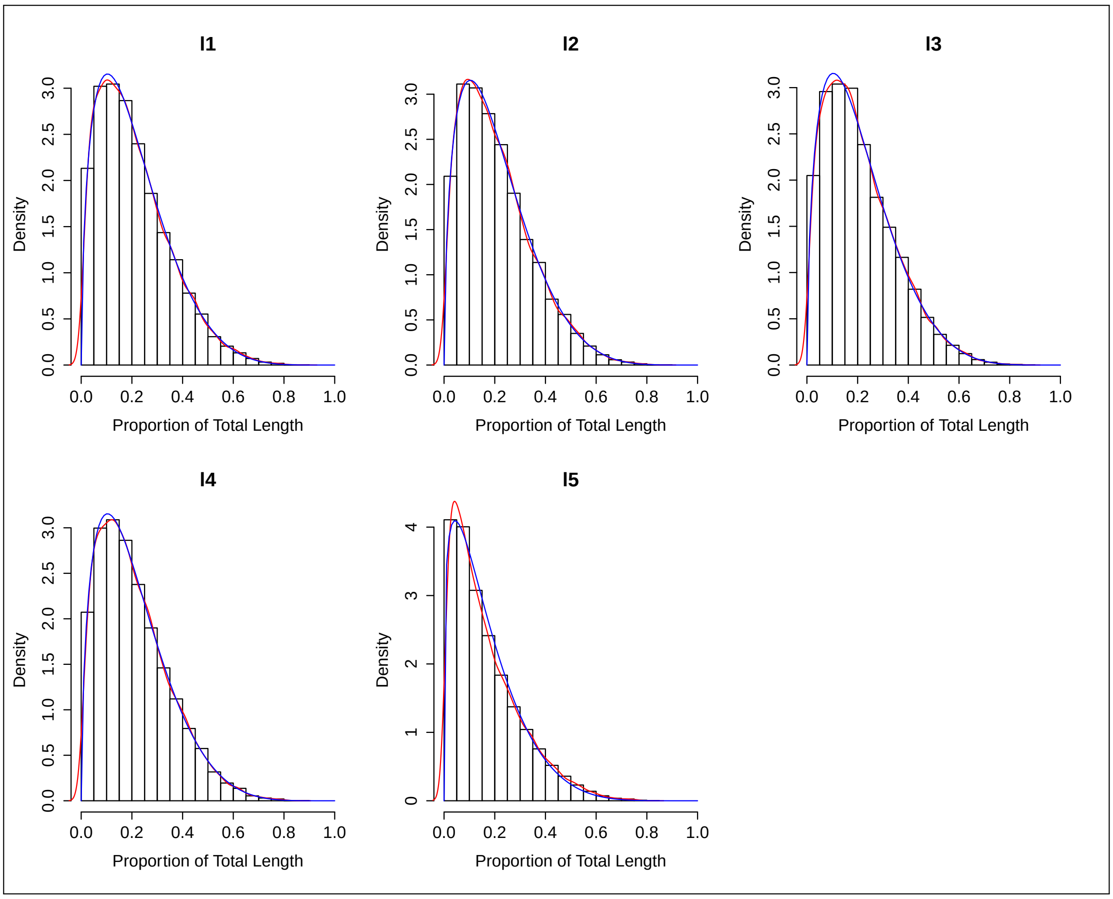


Figure 19: Histograms of segmental lengths (as proportions of total length) in reduced trees on 4 randomly chosen vertices from minimum spanning trees on 4000 vertices with normal kernel density estimate (red) and $Beta$-density functions (blue) (marginal distributions of a $Dirichlet(1.57$, 1.57, 1.57, 1.57, $1.24))$, based on 18,173 observations



before, these observations have been removed from the data set. Figure 19 shows histograms, kernel density estimates and the marginal distributions of a $Dirichlet$-model estimate. In this case the plotted $Beta$-density functions have coefficients $(1.57, 5.92)$ for l_1, l_2, l_3 and l_4 and $(1.24, 6.28)$ for l_5. The estimated coefficients from the $Dirichlet$-model are $\alpha_1 = 1.569, \alpha_2 = 1.564, \alpha_3 = 1.580, \alpha_4 = 1.574$ and $\alpha_5 = 1.240$ with estimated standard deviations of 0.0065 for the first four and 0.0066 for the fifth.

The 95$\%$ confidence intervals for the first four estimates have intersection of $(1.560, 1.584)$ which contains 1.57, the parameter used in the graphical representation. The precise 95$\%$ intervals are given by:

$$CI_{\alpha_1}^{95\%} = (1.549, 1.589) \ \ , \ \ CI_{\alpha_2}^{95\%} = (1.544, 1.584), \ \ CI_{\alpha_3}^{95\%} = (1.560, 1.600),$$

$$CI_{\alpha_4}^{95\%} = (1.554, 1.594) \ \text{ and } \ CI_{\alpha_5}^{95\%} = (1.225, 1.257).$$

Using Kolmogorov-Smirnov tests for any two of the first four proportions leads to high p-values, well above any reasonable significance level. Here, we cannot reject the null hypothesis of equal distributions. Yet, if we apply the test for the fifth and any of the first four proportions, the p-value is always less than 0.001. In this case we strongly reject the null hypothesis and conclude that these proportions follow different distributions. Hence, l_5 and l_i for $i \in \{1, 2, 3, 4\}$ are not exchangeable. However, as we expected, the first four proportions seem to be exchangeable. This is another difference compared to uniform random trees, for which all lengths are exchangeable for any number of chosen vertices.

Reduced Trees on 5 Randomly Chosen Vertices

For the reduced tree on 5 randomly chosen vertices, we have again only one genuinely distinct structural shape that satisfies our two conditions. The two trees shown in Figure 20 are in fact the same up to relabeling. The representation of the trees as being rooted is misleading. In fact, we are considering unrooted trees here, so the two trees are identical. If one designated vertex 3 of the tree on the right to be the root, it would coincide with the tree on the left, except for the labelling. In the previous cases, for 3 and 4 vertices, it does not actually matter whether we think of the trees as being rooted or not, there is only one shape anyway. Yet, for 5 vertices this distinction is essential.

Figure 20: Shape of the reduced tree on 5 randomly chosen vertices (relabelled $\{1, 2, 3, 4, 5\}$). These two trees are not genuinely different when we think of them as being unrooted. One tree is just a relabelled version of the other.

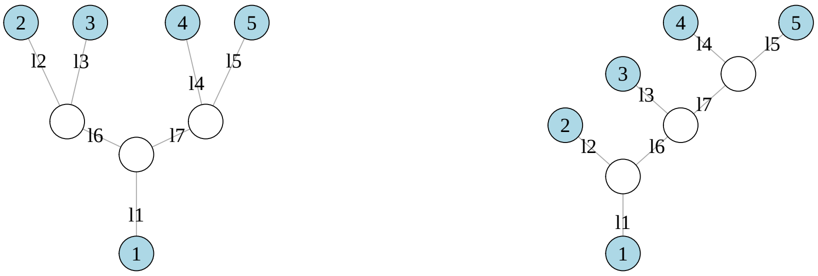
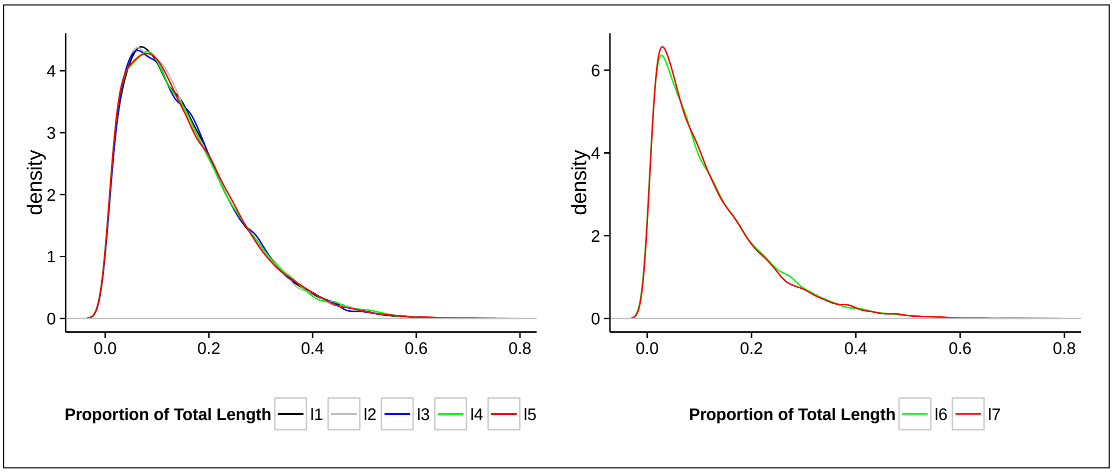


Figure 21: Normal kernel density estimates for segmental lengths (as proportions of total length) in reduced trees on 5 randomly chosen vertices, based on 17,038 observations



For 5 randomly chosen vertices, we have 2 internal edges, l_6 and l_7, and 5 external edges, l_1, l_2, l_3, l_4 and l_5, that connect to the leaves. Again, 20,000 observations have been simulated, 2,962 of which have a different structure than the tree shown in Figure 20, that is, they are either not binary, or some of the chosen vertices are not on the leaves, or both. This constitutes a removal of 14.84% of the data set.

Figure 21 shows the normal kernel density estimates for the proportions of the total length. We notice an obvious difference between external edges and internal edges. The latter tend to be shorter and their distribution possesses a more prominent peak. Within those two groups the distributions appear to be equal. In fact, a $Dirichlet$-regression leads to parameter estimates in the interval $(1.611, 1.619)$ for the first five proportions (external edges), and parameter estimates of 1.248 and 1.256 for l_6 and l_7 respectively (internal edges), which would result in estimated marginal distributions close to $Beta(1.61, 8.94)$ and $Beta(1.25, 9.30)$. The standard deviations of the estimated parameters of the $Dirichlet$-distribution lie in the interval $(0.0062, 0.0065)$, leading to confidence intervals of similar size as in the previous cases.

Applying the Kolmogorov-Smirnov test to the different proportions of external edges, we find again that the null hypothesis of equal distributions cannot be rejected on any common confidence level. The lowest p-value observed here is 0.364 (between l_2 and l_3), and the highest being 0.990 (between l_4 and l_5). At a 5% confidence level we cannot reject the null hypothesis of equal distributions for the proportions of the internal edges (l_6 and l_7), however at a 10% level we could (p-value 0.066). For any combination of an internal and an external edge the test returns p-values that are less than 0.001. Again, we strongly reject the null hypothesis and conclude that these lengths are not exchangeable. Yet, external edges seem to be exchangeable among each other. For internal edges, this claim cannot not be formulated with great confidence, but it might well be the case. In fact, given the symmetry of the reduced tree, it should be.

Reduced Trees on 6 Randomly Chosen Vertices

Finally, we analyse the reduced tree on 6 randomly chosen vertices, which is a special case, since for the first time there exist two genuinely different structural shapes that are binary and have the chosen vertices on the leaves. Figure 22 illustrates these two shapes. It is of interest how often each shape appears and whether these shapes have an impact on the distributions of the total length of the tree as well as its segmental lengths.

We simulated 20,000 reduced trees from minimum spanning trees on 4000 vertices, 4321 (21.61%) of which had a different shape than the two shown in Figure 22 and were re-

Figure 22: The two genuinely different shapes of reduced trees on 6 randomly chosen vertices (relabelled $\{1, 2, 3, 4, 5, 6\}$)

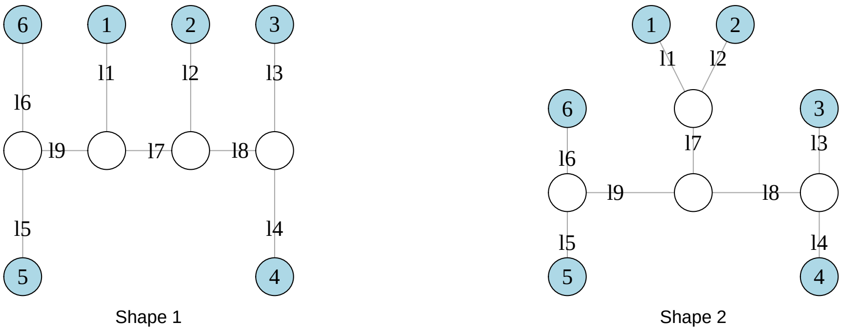
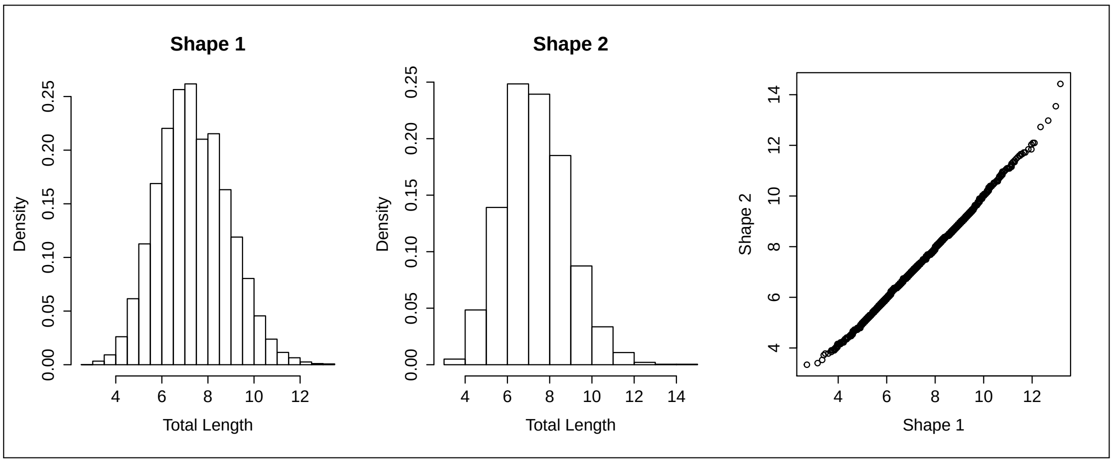


Figure 23: Comparison of total lengths between the two genuinely different shapes of reduced trees on 6 randomly chosen vertices. Histogram for shape 1 for 13,264 observations (left); Histogram for shape 2 for 2415 observations (middle); QQ-Plot for both samples (right) - shapes labelled as in Figure 22.



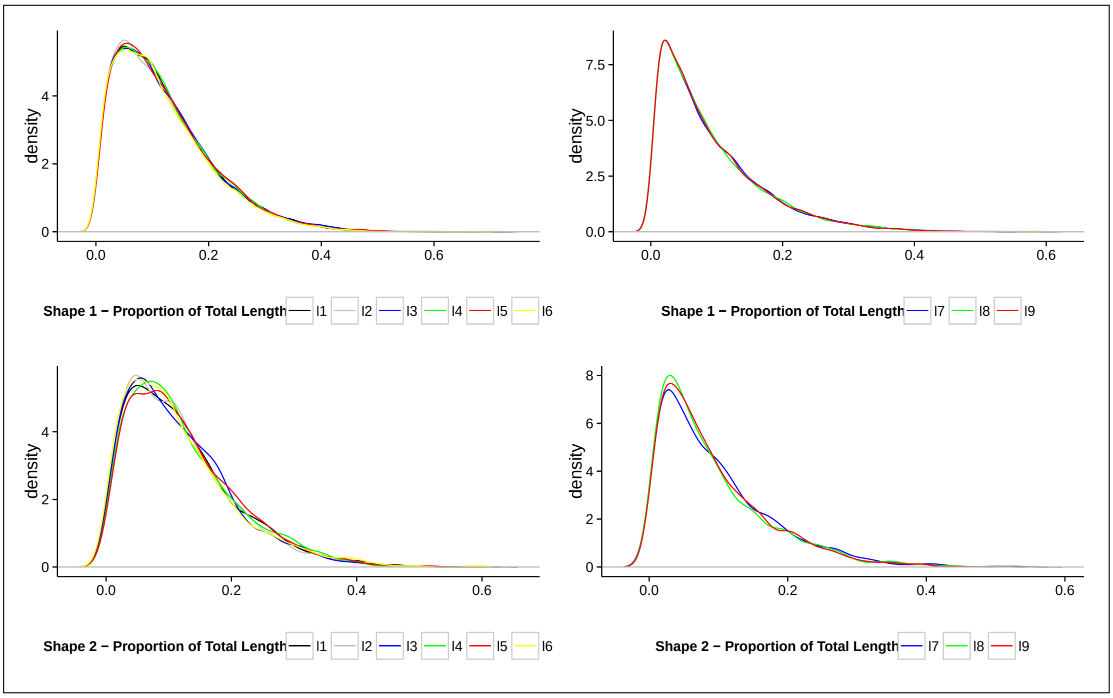
moved. We notice a clear tendency: as the number of randomly chosen vertices increases relative to the number of vertices in the minimum spanning tree, more observations are generated that do not satisfy the two known properties of the limiting object. Interestingly, 13,264 observations took shape 1 (84.60%), but only 2,415 observations (15.40%) took shape 2 (as seen in Figure 22). This means that shape 1 appeared 5.5 times more often than shape 2.

Figure 23 shows histograms of the total length for the two different shapes as well as a QQ-plot for both samples. There is no obvious difference between the two distributions.

Only at the tails the QQ-plot reveals some inconsistencies, but this might be due to the fact that the sample for shape 2 is so much smaller and therefore the tails are not sufficiently covered. The Kolmogorov-Smirnov test leads to a p-value of 0.561, indicating that the null hypothesis of equal distributions cannot be rejected at a reasonable significance level. Hence, the shape of the reduced tree seems to have no impact on the distribution of the total length.

What about the edges of the reduced tree? We have 6 external edges and 3 internal edges. Figure 24 shows kernel density estimates of the proportions of the total length for each of the edges for both possible shapes. As before, there is an obvious difference between the distributions for external and internal edges. We observe the same overall pattern as before. Internal edges tend to be shorter and their distribution possesses a more prominent peak compared to the distribution for external edges. Between the two shapes there are again no obvious differences.

Figure 24: Normal kernel density estimates for segmental lengths (as proportions of total length) in reduced trees on 6 randomly chosen vertices, based on 13,264 observations for shape 1 (top) and 2,415 observations for shape 2 (bottom)



For each of the two distinct shapes, we cannot reject the null hypothesis of the Kolmogorov-Smirnov test for any two of the external edges. For shape 1 this is also the case for the internal edges, which is noteworthy, because, as seen in Figure 22, there is no obvious reason why l_7 should have the same distribution as l_8 and l_9. The edges l_8 and l_9 take symmetrical positions within the reduced tree. We would therefore expect their lengths to be exchangeable. However, l_7 plays a somewhat special role. It is merely the connection of two identical pieces.

For shape 2, the statistical test results are less clear for the internal edges, although we would expect them to be exchangeable, because of the symmetrical structure of the tree. Here, three identical pieces are set together at the middle vertex. For proportions of l_8 and l_7, the p-value is 0.003 which would lead us to reject the null hypothesis. For the other two combinations of internal edges, (l_8, l_9) and (l_7, l_9), the p-values are 0.489 and 0.220 respectively. In these cases we cannot reject the null hypothesis on a common confidence level.

Summing up, we notice that the structural shape of the reduced tree tends to have no influence on the distribution of the total length. Moreover, the shape has no obvious impact on the proportions of the segmental lengths. For both shapes the proportions of external and internal edges follow distinct distributions. Within these two groups of edges, the distributions seem to be rather similar. The estimated coefficients of a $Dirichlet$-regression for shape 1 are $\alpha_1 = 1.647$, $\alpha_2 = 1.637$, $\alpha_3 = 1.658$, $\alpha_4 = 1.653$, $\alpha_5 = 1.656$, $\alpha_6 = 1.657$, $\alpha_7 = 1.249$, $\alpha_8 = 1.244$, $\alpha_9 = 1.248$, for $l_1, l_2 \cdots, l_9$ respectively. For shape 2, the estimates are $\alpha_1 = 1.614$, $\alpha_2 = 1.607$, $\alpha_3 = 1.608$, $\alpha_4 = 1.650$, $\alpha_5 = 1.678$, $\alpha_6 = 1.593$, $\alpha_7 = 1.316$, $\alpha_8 = 1.264$, $\alpha_9 = 1.288$. As in the previous cases, the simulated data suggests that the lengths of external edges, as well as internal edges, are exchangeable within each group. They are certainly not exchangeable between those two groups.

5 Conclusion

Notwithstanding all the similarities, we have seen that the distributional properties of the minimum spanning tree, as n, the number of vertices, increases, are substantially different from the well understood properties of the Brownian Continuum Random Tree - the limiting object of the uniform random tree. We have simulated minimum spanning trees on up to 4000 vertices and investigated the total length and proportions of segmental lengths in reduced trees on up to 6 randomly chosen vertices.

The distribution of the rescaled total length (by $n^{-1/3}$) appears to be unimodal with exponential tails. Moreover, it seems to be log-concave and is close to a $Gamma$-distribution for the reduced tree on 4 randomly chosen vertices ($\Gamma(13.235, 2.467)$ in the parametrisation of section 3.3).

The proportions of total length of the $2k-3$ segments in a reduced tree on k randomly chosen vertices provide more insights into the distributional properties of the minimum spanning tree. For $k = 3$, the proportions of the 3 segments seem to follow the same distribution and are exchangeable, which is also the case for uniform random trees. However, they are not uniformly distributed on the 2-dimensional simplex. Their joint distribution is close to a $Dirichlet(1.52, 1.52, 1.52)$-distribution. For $k \geq 4$, we have observed quite different distributions for the proportions of internal and external segments - those edges that do not connect leaf vertices and those that do. The proportions of internal segments tend to be smaller and their distribution possesses a more prominent peak. For $k = 4$, a $Dirichlet$-model led to shape parameter estimates $(1.57, 1.56, 1.58, 1.57, 1.24)$. For $k = 5$, we obtained the estimates $(1.62, 1.62, 1.62, 1.61, 1.61, 1.26, 1.25)$ for a $Dirichlet$-distribution of 7 proportions. In both cases the lower coefficients, (1.24) and $(1.26, 1.25)$ respectively, correspond to the internal edges in the reduced trees. The higher coefficients correspond to external edges. Lengths of internal and external edges are certainly not exchangeable. This property constitutes a distinctive difference to the uniform random tree. Yet, lengths of edges within each of these two groups might, with some reservation, be exchangeable, even in cases where the symmetrical form of the tree does not provide intuition.

We know that the reduced trees satisfy two properties in the limit. They are binary and have the chosen vertices on the leaves. For $k = 6$, we have shown that there are two genuinely distinct structural shapes for reduced trees that satisfy these conditions. We have seen that one of these shapes appeared almost 5.5 times as often as the other (84.60% vs. 15.40%), but the distributional properties of the total length as well as the proportions did not vary significantly with respect to the shape of the reduced tree.

As k, the number of randomly chosen vertices, increases relative to n, the number of vertices in the minimum spanning tree, the percentage of simulated reduced trees that did not satisfy both of the two limiting conditions increases substantially. For $k = 3, 4, 5$ and 6 ($n = 4000$), the percentage increased from 3.35%, 9.14%, 14.84% to 21.61%. This might be interpreted as a sign of getting further away from the actual limiting distribution. Hence, it is possible that there is more noise in the density estimations for larger numbers of randomly chosen vertices. In order to provide stronger results larger minimum spanning trees need to be simulated. Yet, hardware, time and programming constraints set $n = 4000$ as the limit for this study.

References

[1] L. Addario-Berry, N. Broutin, C. Goldschmidt, G. Miermont (2013), *The scaling limit of the minimum spanning tree of the complete graph*, `http://arxiv.org/abs/1301.1664`

[2] D. Aldous (1991), *The Continuum Random Tree II: An Overview*, in *Stochastic Analysis*; edited by M. T. Barlow, N. H. Bingham, London Math. Soc. Lecture Notes in Math., Cambridge University Press: 23-70

[3] F. Balabdaoui, K. Rufibach, J. A. Wellner (2009), *Limit Distribution Theory for Maximum Likelihood Estimation of a Log-Concave Density*, The Annals of Statistics, 37(3), 1299-1331

[4] L. Birgé (1997), *Estimation of Unimodal Densities Without Smoothness Assumptions*, The Annals of Statistics, 25(3), 970-981

[5] A. Cayley (1889), *A theorem on trees*, Quarterly Journal of Mathematics 23: 376-378.

[6] A. C. Davison (2003), *Statistical Models*, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press

[7] L. Dümbgen, K. Rufibach (2009), *Maximum Likelihood Estimation of a Log-Concave Density and its Distribution Function*, Bernoulli, 15, 40-68. ISSN 1350-7265

[8] M. J. Maier (2012),*Dirichlet Regression in R*, CRAN, `http://cran.r-project.org/web/packages/DirichletReg/DirichletReg.pdf`

[9] M. Matsumoto, T. Nishimura (1998), *Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator*, ACM Transactions on Modeling and Computer Simulation 8 (1): 3-30.

[10] J. Nocedal, S. J. Wright (1999), *Numerical Optimization*, Springer.

[11] K. Rufibach, L. Dümbgen (2010), *logcondens: Estimate a Log-Concave Probability Density from i.i.d. Observations*, R package version 2.0.1,

http://www.biostat.uzh.ch/aboutus/people/rufibach.html,http://www.staff.unibe.ch/duembgen.

[12] K. Rufibach, L. Dümbgen (2013), *logcondens: Computations Related to Univariate Log-Concave Density Estimation*, R package version 2.0.1, http://cran.r-project.org/web/packages/logcondens/vignettes/logcondens.pdf

[13] S. J. Sheather, M. C. Jones (1991), *A reliable data-based bandwidth selection method for kernel density estimation*, Journal of the Royal Statistical Society B 53: 683-690.

[14] W. N. Venables, B. D. Ripley (2002), *Modern Applied Statistics with S*, New York: Springer

[15] M. P. Wand, M. C. Jones (1995), *Kernel Smoothing*, Monographs on Statistics and Applied Probability 60, Chapman and Hall

[16] CRAN Task View: Probability Distributions, *Random Number Generators*, Retrieved 2012-05-29, http://cran.r-project.org/web/views/Distributions.html

[17] R-Documentation, *Optim: General-purpose Optimization*, http://stat.ethz.ch/R-manual/R-devel/library/stats/html/optim.html

Appendix: Selected Parts of the *R*-Code

Here, selected parts of the R-Code that has been used to simulate the data and implement the above data analysis are presented.

Simulation of Data

The following block of code shows the functions used for the simulation of uniform random trees, weighted complete graphs and minimum spanning trees. For the latter we made use of the function $minimum.spanning.tree()$ provided in the $igraph$-package.

```
# required R-packages:
library(MCMCpack); library(snow); library(snowfall); library(igraph)

# a function to simulate a uniform random tree using the Aldous-Broder algorithm:
ABalg <- function(n){
  state <- sample(1:n,2)
  v <- c(state); v2 <- c(state)
  x <- c(state[1]); y <- c(state[2])
  while (length(v) <= n-1){
    state <- sample(1:n,1)
    v2 <- c(v2,state)
    if (any(v==state)==FALSE){
      v <- c(v,state); x <- c(x,v2[(length(v2)-1)]); y <- c(y,state)
    }
  }
  E <- cbind(x,y); return(E)
}

# a function to simulate a weighted complete graph:
CGraph.three <- function(n){
  edges <- graph.full(n); w <- runif(sum(1:n-1))
  g <- list("edgelist"=edges,"weights"=w)
  return(g)
}

# for minimum spanning trees we use the minimum.spanning.tree() function
# from the R-package igraph.
```

The following block of code shows how data for the distances between 6 randomly chosen vertices have been simulated for minimum spanning trees. The procedure for $k < 6$ as well as the adaptation of it for uniform random trees are analogue and not shown here. They can of course be provided on request.

```
# a function for simulating a minimum spanning tree on n vertices and the distances
# between 6 randomly chosen vertices:
mst.dist.6 <- function(n){
  M <- CGraph.three(n)
  v <- sample(c(1:n),6)
  mst <- minimum.spanning.tree(M$edgelist,weights=M$weights)
  d.1 <- get.shortest.paths(mst,v[1],to=v[2])
  d.2 <- get.shortest.paths(mst,v[3],to=v[4])
  count <- 1
  if (length(intersect(d.1[[1]],d.2[[1]]))>=1){
    d.1 <- get.shortest.paths(mst,v[1],to=v[3])
    d.2 <- get.shortest.paths(mst,v[2],to=v[4])
    count <- count +1
    if (length(intersect(d.1[[1]],d.2[[1]]))>=1){
      d.1 <- get.shortest.paths(mst,v[1],to=v[4])
      d.2 <- get.shortest.paths(mst,v[2],to=v[3])
      count <- count +1
      if (length(intersect(d.1[[1]],d.2[[1]]))>=1){
        count <- count+1
      }
    }
  }
  if (count==4){
    heart <- intersect(d.1[[1]],d.2[[1]])
    d.1 <- get.shortest.paths(mst,heart,to=v[1])[[1]]
    d.2 <- get.shortest.paths(mst,heart,to=v[2])[[1]]
    d.3 <- get.shortest.paths(mst,heart,to=v[3])[[1]]
    d.4 <- get.shortest.paths(mst,heart,to=v[4])[[1]]
    d.7 <- heart
  }
  if (count<=3){
    p <- list()
    for (i in 1:length(d.1[[1]])){
      pp <- get.shortest.paths(mst,from=d.1[[1]][i],to=d.2[[1]])
      p <- c(p,pp)
    }
    l <- sapply(p,length)
    index <- which(l==min(l))
    d.7 <- p[[index]]
    in.1 <-d.7[1]
    in.2 <-d.7[length(d.7)]
    if (count==3){
      d.1 <- get.shortest.paths(mst,in.1,to=v[1])[[1]]
      d.4 <- get.shortest.paths(mst,in.1,to=v[4])[[1]]
      d.2 <- get.shortest.paths(mst,in.2,to=v[2])[[1]]
      d.3 <- get.shortest.paths(mst,in.2,to=v[3])[[1]]
    }
    if (count==2){
      d.1 <- get.shortest.paths(mst,in.1,to=v[1])[[1]]
      d.3 <- get.shortest.paths(mst,in.1,to=v[3])[[1]]
      d.2 <- get.shortest.paths(mst,in.2,to=v[2])[[1]]
      d.4 <- get.shortest.paths(mst,in.2,to=v[4])[[1]]
    }
    if (count==1){
      d.1 <- get.shortest.paths(mst,in.1,to=v[1])[[1]]
      d.2 <- get.shortest.paths(mst,in.1,to=v[2])[[1]]
      d.3 <- get.shortest.paths(mst,in.2,to=v[3])[[1]]
      d.4 <- get.shortest.paths(mst,in.2,to=v[4])[[1]]
    }
  }

  ## at this stage we have the subtree for the first 4 chosen vertices

  stree <- unique(c(d.1,d.2,d.3,d.4,d.7))
  dis <- get.shortest.paths(mst,v[5],to=stree)
  l <- sapply(dis,length)
  index <- which(l==min(l))
  d.5 <- dis[[index]]
  if (d.5[length(d.5)] %in% d.1){
    count2 <- 1; ddd <- d.1; j <- which(ddd==d.5[length(d.5)])
    d.1 <- ddd[j:length(ddd)]; d.8 <- ddd[1:j]
  }
  if (d.5[length(d.5)] %in% d.2){
    count2 <- 2; ddd <- d.2; j <- which(ddd==d.5[length(d.5)])
    d.2 <- ddd[j:length(ddd)]; d.8 <- ddd[1:j]
  }
  if (d.5[length(d.5)] %in% d.3){
    count2 <- 3; ddd <- d.3; j <- which(ddd==d.5[length(d.5)])
    d.3 <- ddd[j:length(ddd)]; d.8 <- ddd[1:j]
  }
  if (d.5[length(d.5)] %in% d.4){
    count2 <- 4; ddd <- d.4; j <- which(ddd==d.5[length(d.5)])
    d.4 <- ddd[j:length(ddd)]; d.8 <- ddd[1:j]
  }
  if (d.5[length(d.5)] %in% d.7){
    count2 <- 7; ddd <- d.7; j <- which(ddd==d.5[length(d.5)])
    d.7 <- ddd[j:length(ddd)]; d.8 <- ddd[1:j]
  }
  d.5 <- rev(d.5)
```

```
## at this stage we have the subtree for the first 5 chosen vertices

stree <- unique(c(d.1,d.2,d.3,d.4,d.5,d.7,d.8))
dis <- get.shortest.paths(mst,v[6],to=stree)
l <- sapply(dis,length)
index <- which(l==min(l))
d.6 <- dis[[index]]
if (d.6[length(d.6)] %in% d.1){
  count3 <- 1; ddd <- d.1; j <- which(ddd==d.6[length(d.6)])
  d.1 <- ddd[j:length(ddd)]; d.9 <- ddd[1:j]
}
if (d.6[length(d.6)] %in% d.2){
  count3 <- 2; ddd <- d.2; j <- which(ddd==d.6[length(d.6)])
  d.2 <- ddd[j:length(ddd)]; d.9 <- ddd[1:j]
}
if (d.6[length(d.6)] %in% d.3){
  count3 <- 3; ddd <- d.3; j <- which(ddd==d.6[length(d.6)])
  d.3 <- ddd[j:length(ddd)]; d.9 <- ddd[1:j]
}
if (d.6[length(d.6)] %in% d.4){
  count3 <- 4; ddd <- d.4; j <- which(ddd==d.6[length(d.6)])
  d.4 <- ddd[j:length(ddd)]; d.9 <- ddd[1:j]
}
if (d.6[length(d.6)] %in% d.5){
  count3 <- 5; ddd <- d.5; j <- which(ddd==d.6[length(d.6)])
  d.5 <- ddd[j:length(ddd)]; d.9 <- ddd[1:j]
}
if (d.6[length(d.6)] %in% d.7){
  count3 <- 7; ddd <- d.7; j <- which(ddd==d.6[length(d.6)])
  d.7 <- ddd[j:length(ddd)]; d.9 <- ddd[1:j]
}
if (d.6[length(d.6)] %in% d.8){
  count3 <- 8; ddd <- d.8; j <- which(ddd==d.6[length(d.6)])
  d.8 <- ddd[j:length(ddd)]; d.9 <- ddd[1:j]
}
d.6 <- rev(d.6)

## at this stage we have all lengths of the subtree on 6 vertices
## last step: find out which shape the tree has (there are two genuinely different shapes)

shape <- NA
starts <- c(d.1[1],d.2[1],d.3[1],d.4[1],d.5[1],d.6[1])
if (max(table(starts))==2 & any(vw %in% starts)==FALSE & count<=3){
  if (length(unique(starts))==4){shape <- ”shape1”}
  if (length(unique(starts))==3){shape <- ”shape2”} # shapes as in figure 22
}

L <- c(length(d.1)-1,length(d.2)-1,length(d.3)-1,
       length(d.4)-1,length(d.5)-1,length(d.6)-1,
         length(d.7)-1,length(d.8)-1,length(d.9)-1,shape)
  return(L)
} # END of function

# simulating 20,000 observations of minimum spanning trees on 4000 vertices
# for the distances between 6 randomly chosen vertices:
sim.mst.4000.20000.dist.6 <- sfLapply(rep(4000,20000), mst.dist.6)
sfStop()
sim.10 <- sapply(sim.mst.4000.20000.dist.6,round)
DATA <- data.frame(t(sim.10))
write.table(DATA, ”C:/Users/Israel/Downloads/mst.4000.dis.6.txt”, sep=”\t”)
```

Analysis of Data

Kernel density estimations, *Gamma*-models, *Dirichlet*-regressions and log-concave density estimations have been implemented using the following commands. The analysis is based on the *R*-packages *logcondens*, developed by Kaspar Rufibach and Lutz Dümbgen, and *DirichletReg*, developed by Marco J. Maier.

```
library(logcondens); library(DirichletReg)

# Kernel smoothing as used in this study:

KER <- density(X, bw=”SJ”)
# where X is a vector that contains simulated rescaled lengths
# and bw=”SJ” specifies the bandwidth selection procedure
# as described in section 3.3 (recommended in Venables and Ripley 2002)

# Implementing a Dirichlet-regression in R for 3 proportions:

names(DATA) <- c(”l1”,”l2”,”l3”)
DATA.DR <- DR_data(DATA)
model <- DirichReg(DATA.DR~1)
# where DATA is a data frame that contains the 3 segmental lengths of the
# 3-reduced tree. For k>3, the case is analogue.

# Log-concave density estimation:

res <- logConDens(X,smoothed=TRUE,print=FALSE)
# X is a vector that contains simulated rescaled lengths
summary(res)
est <- evaluateLogConDens(seq(0,12,0.01),res)
# ”est” is a data frame that contains evaluations of the estimated log-concave density
# as well as a smoothed version of it and the corresponding CDF estimates

# Fit a Gamma-model:

model <- glm(X ~ 1, family = Gamma)
# X is a vector that contains simulated rescaled lengths
gamma.shape(model) # to get an improved estimate of the shape parameter
# alternative:
fitdistr(X, ”gamma”)
# notice that the two alternatives use different parametrisations!
```

The following block of code shows how Figure 22 on page 32 has been produced. The complete code that has been written for graphical representations is not shown here, but is available on request. It makes substantial use of the dafault plotting options in *R* as well as the *R*-package *ggplot2*.

```
par(mfrow=c(1,2))
m <- t(matrix(c(2,1,2,2,1,2,3,1,3,2,4,1,1,1,1,0,4,2,4,0),ncol=10))
plot.igraph(graph.edgelist(t(matrix(c(1,2,1,7,7,3,7,8,1,4,4,5,4,6,6,9,6,10),ncol=9)),directed=FALSE),
           vertex.label=c("","1","6","","2","","","5","3","4"),vertex.size=30,
           vertex.label.color="black",
           edge.label=c("l1","l9","l6","l5","l7","l2","l8","l3","l4"),
           edge.label.color="black",
           vertex.frame.color="black",
           vertex.color=c("white","lightblue","lightblue","white",
                        "lightblue","white","white","lightblue",
                        "lightblue","lightblue"),
           layout = m)
mtext("Shape 1",side=1, cex=0.8)
m <- t(matrix(c(2.5,1,2.5,2,1,2,2,3,3,3,4,1,1,1,1,0,4,2,4,0),ncol=10))
plot.igraph(graph.edgelist(t(matrix(c(1,2,1,7,7,3,7,8,1,6,2,5,4,2,6,9,6,10),ncol=9)),directed=FALSE),
           vertex.label=c("","","6","1","2","","","5","3","4"),vertex.size=30,
           vertex.label.color="black",
           edge.label=c("l7","l9","l6","l5","l8","l2","l1","l3","l4"),
           edge.label.color="black",
           vertex.frame.color="black",
           vertex.color=c("white","white","lightblue","lightblue",
                        "lightblue","white","white","lightblue","lightblue","lightblue"),
           layout = m)
mtext("Shape 2",side=1, cex=0.8)

# the matrix m and the command "layout = m" specify the location
# of vertices on the plane.
# with mtext() we can add captions under the trees.
```