

PROJET

UE Méthodes de ranking et recommandations

Sujet 5 : Simulation d'un Google Bombing

Maxime Gonthier - Laureline Martin

17 avril 2019

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Manuel utilisateur</b>	<b>2</b>
<b>3</b>	<b>Plan</b>	<b>2</b>
<b>4</b>	<b>I. Tests initiaux et hypothèses</b>	<b>2</b>
4.1	Explication du code . . . . .	2
4.2	Résultats des tests initiaux . . . . .	2
4.2.1	Stanford.txt sans modification . . . . .	2
4.2.2	Résultats . . . . .	3
4.2.3	Analyses . . . . .	3
4.2.4	Hypothèses . . . . .	3
4.3	Conclusion des tests initiaux et hypothèses . . . . .	3
<b>5</b>	<b>II. Tests avec un nombre de graphes générés aléatoirement</b>	<b>3</b>
5.1	Explication du code . . . . .	4
5.2	Résultats des tests . . . . .	5
5.3	Analyse graphes en anneau . . . . .	5
5.4	Analyses Graphes complet . . . . .	7
5.5	Analyses Arbres . . . . .	9
5.6	Conclusion sur l'impact de chaque structure . . . . .	11
<b>6</b>	<b>III. Expériences avec un graphe aléatoire</b>	<b>11</b>
<b>7</b>	<b>Conclusion</b>	<b>11</b>

## 1 Introduction

## 2 Manuel utilisateur

## 3 Plan

Dans un premier temps nous allons effectu  des tests simples en utilisant quatre type de structures diff rentes ainsi que trois cibles de pertinences diff rentes. Nous en d duirons des hypoth ses sur l'efficacit  de chaque structure et l'impact de la pertinence de la cible sur cette m me efficacit . Dans un second temps nous  tudierons l'impact sur la pertinence de graphes g n r s en nombre al atoire pour  tudier qu'elle structure est globalement la plus efficace. Puis nous testerons la meme choses avec cette fois ci un graphe dont les arcs le reliant a lui meme sont g n r es al atoirement. Enfin nous ferons varier le nombre de sommets du graphe ataquant pour en d duire son efficacit  et nous conclurons.

## 4 I. Tests initiaux et hypoth ses

L'algorithme power calculant les pertinences est contenu dans le fichier *ranking.c*. Il n'est pas d taill  dans ce rapport mais le fichier est comment . On consid re ici le graphe du web Stanford.txt. Ce graphe est modifi  par l'ajout de sommets et d'arcs afin d'augmenter la valeur d'un sommet cibl . On consid re que :

- Les sommets repr sentent les pages du web.
- Les arcs repr sentent les liens dirigeant vers d'autres pages.
- Les valeurs des sommets repr sentent les pertinences calcul s par l'algorithme pagerank.
- Le sommet cible repr sente la page dont on souhaite augmenter la pertinence.
- 100 sommets sont ajout s pour chaque test.

### 4.1 Explication du code

### 4.2 R sultats des tests initiaux

#### 4.2.1 Stanford.txt sans modification

281903 pages  
2312497 liens  
132 it rations  
27.627466 secondes

Voici les pertinences de base des trois sommets étudiés :

Pertinence forte : Page 280545 - 9.96199e-05

Pertinence moyenne : Page 281466 - 7.53954e-06

Pertinence faible : Page 281574 - 6.05222e-07

#### 4.2.2 Résultats

Pertinences des sommet cible	Forte	Moyenne	Faible
Sans modification	9,96E-05	7,54E-06	6,05E-07
Difference sans modification	0,00E+00	0,00E+00	0,00E+00
Attaque a 100 sommets seuls	1,82E-04	5,89E-05	5,19E-05
Differences avec 100 sommets seuls	8,25E-05	5,13E-05	5,13E-05
Attaque avec un anneau de 100 sommets	1,71E-04	5,22E-05	4,52E-05
Differences avec un anneau a 100 sommets	7,17E-05	4,46E-05	4,46E-05
Attaque avec un graphe complet de 100 sommets	1,05E-04	1,08E-05	3,84E-06
Differences avec un graphe complet a 100 sommets	5,17E-06	3,23E-06	3,24E-06
Attaque avec un arbre de 100 sommets	1,72E-04	5,25E-05	4,55E-05
Differences avec un arbre a 100 sommets	7,22E-05	4,49E-05	4,49E-05
Attaque avec un arbre a attaquant unique de 100 sommets	1,38E-04	3,17E-05	2,47E-05
Differences avec un arbre a attaquant seul a 100 sommets	3,88E-05	2,41E-05	2,41E-05

#### 4.2.3 Analyses

#### 4.2.4 Hypothèses

#### 4.3 Conclusion des tests initiaux et hypothèses

### 5 II. Tests avec un nombre de graphes générés aléatoirement

Nous allons désormais insérer des graphes générés aléatoirement. Ce qui est aléatoire n'est pas la structure des graphes mais le nombre de garphes générés. L'objectif de cette démarche est de déterminer quel structure est la plus efficace globalement. C'est a dire quelle structure influe le plus sur la pertinence quelle que soit la situation. De plus les resultats nous aiderons aussi a determiner l'impact qu'a une certaine structure sur la cible de manière plus générale que dans les cas prédéfinis de la partie précédente. Le nombre de sommet des graphes ajoutées est fixé a 100. La cible est également fixé ainsi que la structure des graphes ajoutées. On va par exemple insérez 5 graphes complet de nombre de sommet respectivement : 25, 10, 5, 6 et 4. Tous reliées au sommet cible. Dans un premier temps nous allons expliquer le code derrière cette démarche puis nous analyserons les résultats.

## 5.1 Explication du code

Tous est dans le fichier *ajoutsommetsattanquants.c*. Les fonctions utilisées sont *ajoutanneaualeatoire*, *ajoutcompletaleatoire* et *ajoutarbrealeatoire*. Ces fonctions reprennent en partie le code des trois fonctions presque éponyme décrite précédemment. Regardons ce qui a changé.

```
while (nbsommetrestant > 0) {  
    nbajout = rand()%(100-nouveausommets);  
    if (nbajout <= 3) { nbajout = 3; }  
    nbsommetrestant -= nbajout;
```

*nbajout* représente le nombre de sommet que l'on va ajouter dans la première structure créer. Il choisit donc un nombre aléatoire entre 0 et 100 car le nombre de sommet total est limité à 100. Si le nombre choisit est inférieur à 3 on le fixe à 3 car créer des anneaux ou des graphes complets de tailles inférieures à 3 revient juste à créer des sommets seuls. *nbajout* est enlevé au nombre de sommet restant à ajouter. Puis on lance la construction de la structure comme vu précédemment avec *nbajout* représentant le nombre de sommets. À la fin de cette itération *nbajout* reprends un nombre aléatoire qui cette fois prend une valeur entre 0 et 100 moins le nombre de sommets ajouté précédemment représenté par *nouveausommets*.

## 5.2 Résultats des tests

Type de graphe	Sans modification	Anneau	Graphe complet	Arbre
Pertinences	9,96E-05	0,000172787	0,000110716	0,00017433
	9,96E-05	0,00017207	0,000111977	0,00017345
	9,96E-05	0,000172787	0,000120062	0,00017417
	9,96E-05	0,000171353	0,000110397	0,00017326
	9,96E-05	0,000171353	0,000111791	0,00017372
	9,96E-05	0,000171353	0,000110417	0,00017387
	9,96E-05	0,000172787	0,000111729	0,00017378
	9,96E-05	0,00017207	0,000112341	0,00017456
	9,96E-05	0,00017207	0,00011073	0,00017265
	9,96E-05	0,000172787	0,000117964	0,00017477
	9,96E-05	0,00017207	0,000117294	0,00017273
	9,96E-05	0,00017207	0,000119603	0,00017287
	9,96E-05	0,000171353	0,000124625	0,0001745
	9,96E-05	0,00017207	0,000114558	0,00017338
	9,96E-05	0,00017207	0,000114431	0,00017474
	9,96E-05	0,00017207	0,000113345	0,00017332
	9,96E-05	0,000171353	0,000112464	0,00017427
	9,96E-05	0,00017207	0,000113455	0,00017272
	9,96E-05	0,000172787	0,000124734	0,00017454
	9,96E-05	0,00017207	0,00011858	0,00017256
	9,96E-05	0,000172787	0,000110571	0,00017301
	9,96E-05	0,000171353	0,000114567	0,00017436
	9,96E-05	0,000171353	0,000114828	0,00017329
	9,96E-05	0,000172787	0,000115583	0,00017326
	9,96E-05	0,00017207	0,000117809	0,00017281
	9,96E-05	0,000172787	0,000120914	0,0001739
	9,96E-05	0,000172787	0,00012048	0,00017202
	9,96E-05	0,000172787	0,00011457	0,00017362
	9,96E-05	0,000172787	0,000114598	0,0001741
	9,96E-05	0,000171353	0,000114455	0,00017333

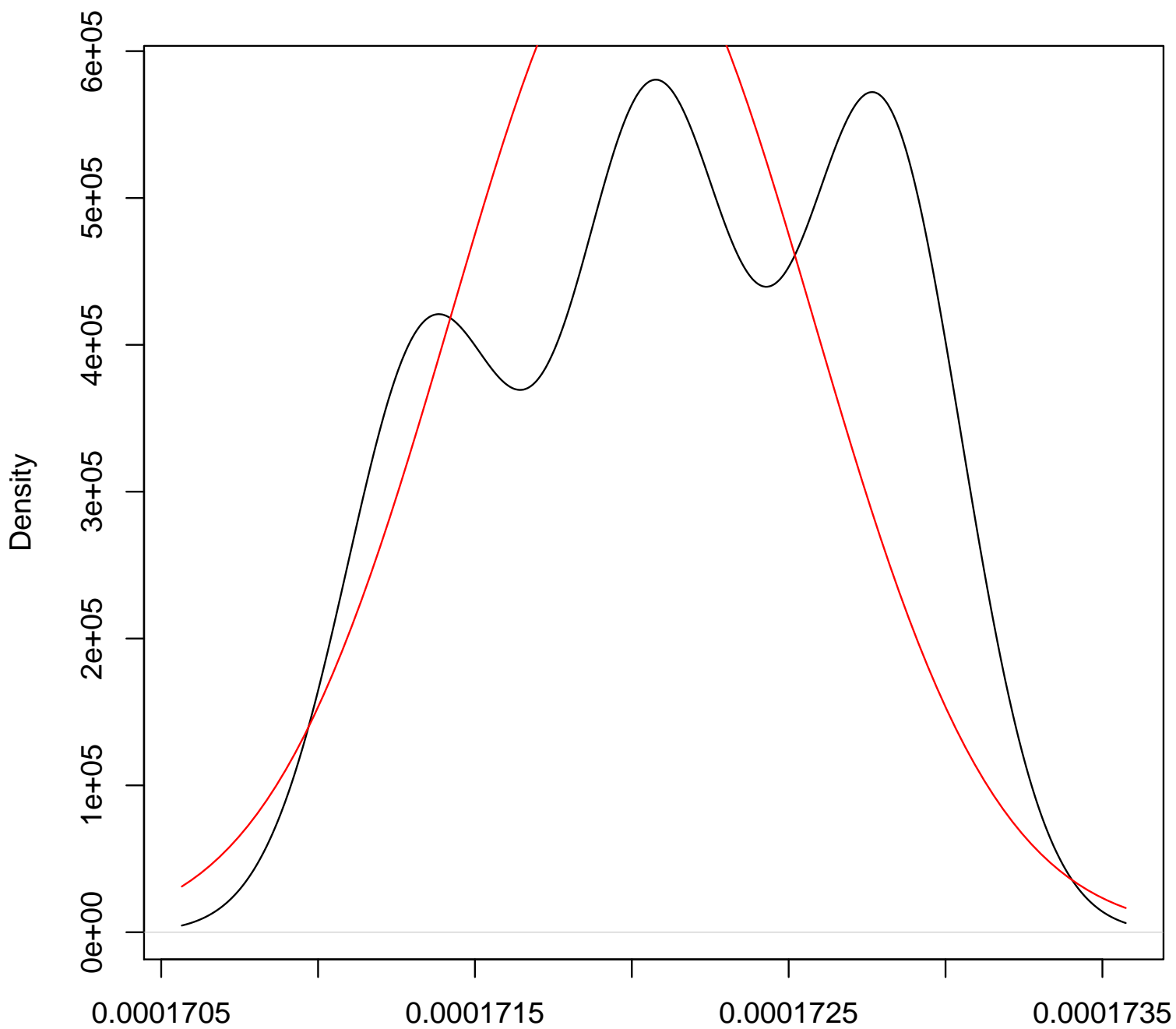
### ANALYSE

Nb tirages	30	30	30	30
Espérance	9,96E-05	1,72E-04	1,15E-04	1,74E-04
Ecart type	0	5,75759E-07	4,06743E-06	7,347E-07
Intervalle de confiance : 95% de la moyenne				
t-normale	1,645	1,645	1,645	1,645
Erreur	0	1,7292E-07	1,22159E-06	2,2065E-07

### ANALYSE

## 5.3 Analyse graphes en anneau

**density.default(x = Anneau\_aleatoire[, 1])**

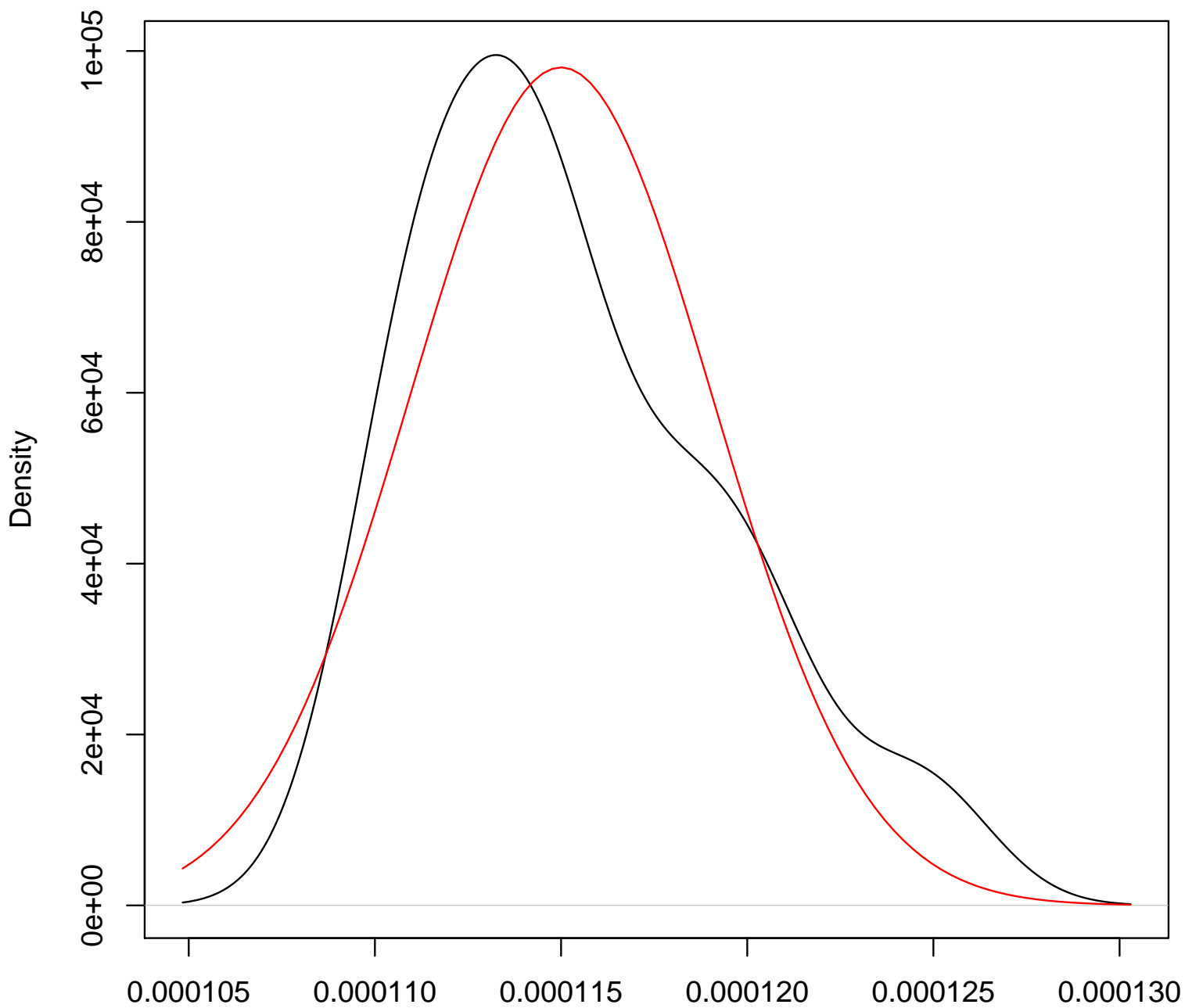


N = 30 Bandwidth = 2.625e-07

## 5.4 Analyses Graphes complet



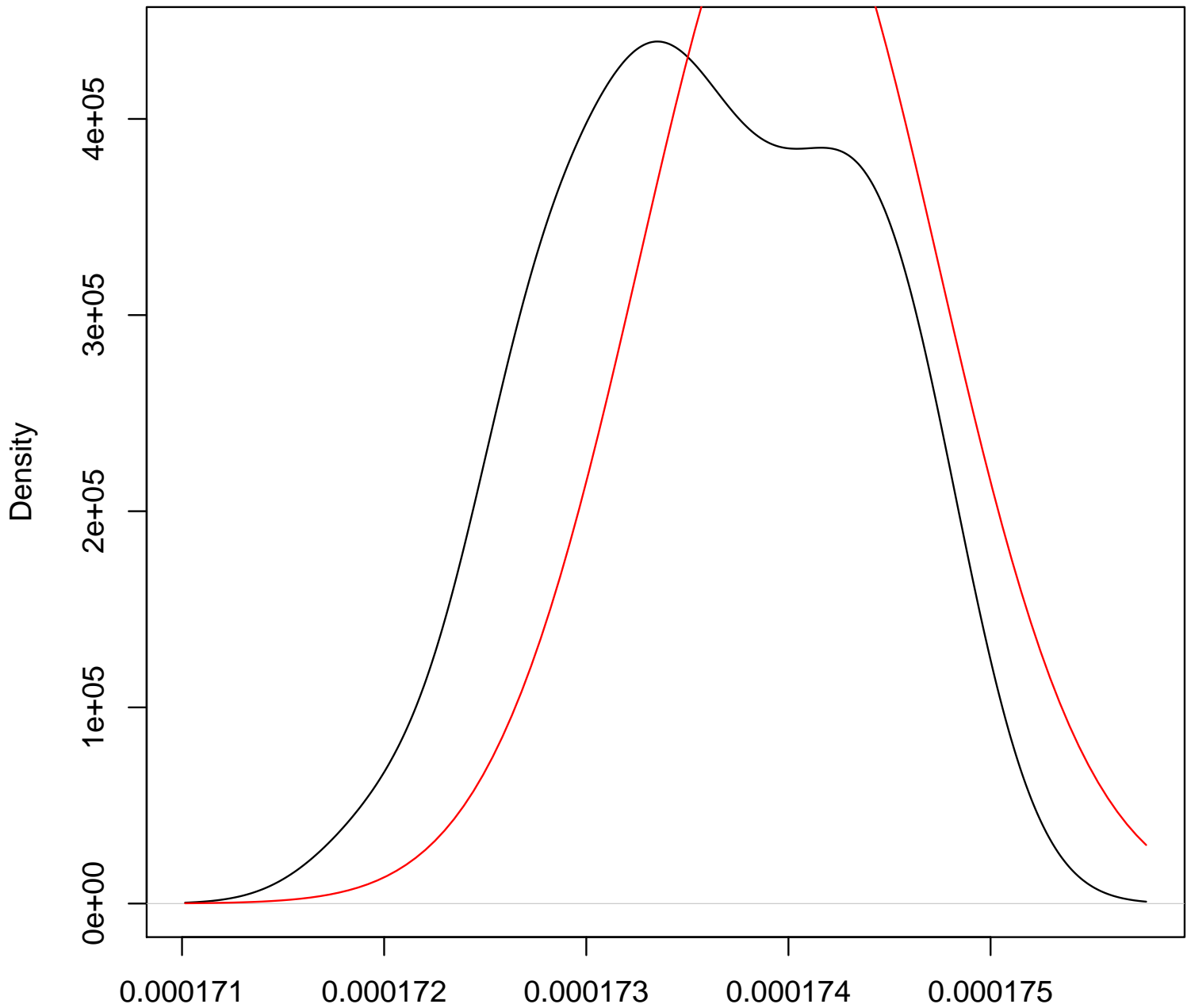
**density.default(x = Anneau\_aleatoire[, 1])**



N = 30 Bandwidth = 1.854e-06

## 5.5 Analyses Arbres

**density.default(x = Anneau\_aleatoire[, 1])**



N = 30 Bandwidth =  $3.349e-07$

5.6	Conclusion sur l'impact de chaque structure
6	III. Expériences avec un graphe aléatoire
7	Conclusion