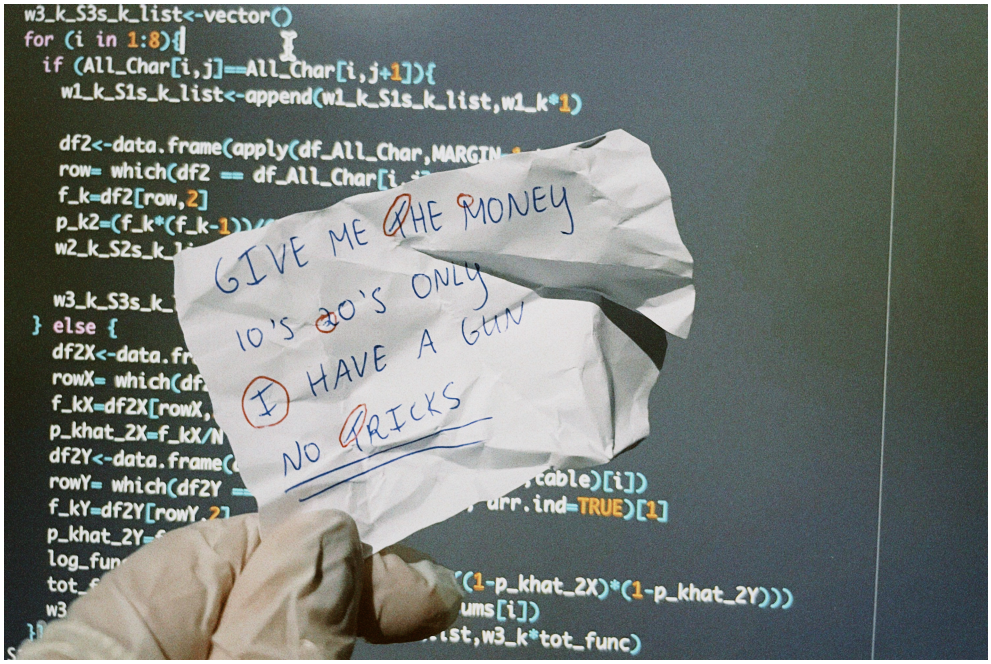


Mathematics as a secret weapon against criminals:

Employing score-based likelihood ratio systems for the comparison of handwriting and studying their quality of performance

Bachelor Thesis

A. J. Wijker
4816870

Delft University of Technology

July 2, 2022

Mathematics as a secret weapon against criminals

Employing score-based likelihood ratio systems for the comparison of handwriting and studying their quality of performance

by

A. J. Wijker

to obtain the degree of Bachelor of Science

at the Delft University of Technology,

to be defended publicly on Friday July 1, 2022 at 14:15.

Student number:	4816870	
Project duration:	April 18, 2022 – July 1, 2022	
Thesis committee:	Dr. J. Söhl,	TU Delft, supervisor
	Drs. A. T. Hensbergen,	TU Delft

TU Delft

Abstract

In this report a new approach to (forensic) handwriting analysis is presented; score-based likelihood ratio (SLR) systems are employed and their quality of performance is studied.

Forensic handwriting analysis is an important part of a forensic investigation which can be used when there are threatening letters, hold-up notes, ransom letters, etc. involved. It can also be used in fraud investigations.
The handwriting analysis, as it is performed now, is in general done by certified forensic document examiners. One of the drawbacks of analyzing in this manner is that there is no expression for the degree of uncertainty of the statement that two writings have the same writer. The analysis is also time consuming and the uniqueness of characteristics is not taken into account.
For the handwriting analysis of this research, SLR systems will be employed. This approach has already been proven to be useful for forensic analyses. These systems do not have the three drawbacks that the traditional approach has. Furthermore, the SLR systems are objective, transparent and their behavior is known. However, they only take a small part of the available information into account and examining their accuracy is complicated. SLRs are best used in combination with the (subjective) opinion of forensic examiners.

The handwriting samples of 800 writers (three documents each) are considered. After the letter combinations "er" are extracted, the characteristics are entered into a user interface for each document which makes the analysis more time efficient. The likelihood ratio (LR) is, given two mutually exclusive hypotheses, the ratio of the probabilities of the evidence. However, because the LR is difficult to compute when the number of variables is large, score-based likelihood ratios (SLRs) are utilized. SLRs use score functions which are a measure for how similar two writings are and they transform multidimensional data to one dimensional data. The SLR expresses the degree of uncertainty that a hypothesis is true. The common source problem is considered due to the unavailability of suspect specific data. So, it is tested if two writings from unknown writers originate from the same unknown writer. For this research four score functions are considered; Overlap (does not take uniqueness of matching and mismatching values into account), Goodall3 (takes uniqueness of matching values (not mismatching ones) into account), Burnaby (takes uniqueness of mismatching values (not matching ones) into account) and Anderberg (takes uniqueness of matching and mismatching values into account). This results in four SLR systems that are evaluated based on three performance characteristics; the leave-one-out method, 95% bootstrap interval and misleading evidence. **Score 2 (Goodall3) performs the best based on the performance characteristics** (then score 1 (Overlap), then score 4 (Anderberg) and score 3 (Burnaby) (tied for third place)). If an SLR system is required that performs the best based on the leave-one-out method (so one that has the greatest discriminating power), score 1 has to be used. If a system is required that performs the best based on the 95% bootstrap confidence interval (so one that has the highest precision) and on misleading evidence (so one that produces the least number of SLRs that support false hypotheses), score 2 has to be used. Thus, **what score function (either score 1 or 2) is chosen for the SLR system depends on the desired qualities of the system**.

Contents

List of variables

LR	The likelihood ratio
SLR	The score-based likelihood ratio
x_{u1}	Piece of handwriting from unknown writer 1 (in common source problem)
x_{u1}	Piece of handwriting from unknown writer 2 (in common source problem)
x_u	Piece of handwriting from an unknown writer (in specific source problem)
x_s	Piece of handwriting from a known specific writer (in specific source problem)
H_1	The hypothesis that the handwriting with unknown source and the handwriting with known source are from the same person (prosecution hypothesis)
H_2	The hypothesis that the handwriting with unknown source and the handwriting with known source are from different people (defence hypothesis)
x	The handwriting with known source
y	The handwriting with unknown source
$P(x, y)$	The joint probability function of x and y
I	The relevant background information
$s(x, y)$	The (similarity) score (that is a measure for how similar known (x) and unknown source (y) are to each other)
$P(s(x, y))$	The probability function of the (similarity) score
$\mathbf{x}$	The set of features of a handwriting with a known source ($[x_1, \ldots, x_n]$)
x_i	The ith feature of the handwriting with known source which can take *exactly one* of the n_i values (in $\{x_i^1, \ldots, x_i^{n_i}\}$)
$\mathbf{y}$	The set of features of a handwriting with an unknown source ($[y_1, \ldots, y_n]$)
y_i	The ith feature of the handwriting with unknown source which can take *exactly one* of the n_i values (in $\{y_i^1, \ldots, y_i^{n_i}\}$)
$s_i(x_i, y_i)$	The (similarity) score of x_i and y_i
w_i	The weight assigned to the ith feature of the handwriting

n	The total number of characteristics
$f_i(x_i)$	The number of times the ith feature takes the value x_i in the data set
N	The number of data points
A_i	The set of all possible values (of size n_i) that the ith feature can take
n_i	The number of possible values that the ith feature can take
$f(x)$	The probability density function of the random variable X
S_1	The similarity scores given H_1
S_2	The similarity scores given H_2
n_1	The number of similarity scores given H_1 (so the size of S_1) (this definition is only used in subsections 6.3.2 and 6.3.4)
n_2	The number of similarity scores given H_2 (so the size of S_2) (this definition is only used in subsections 6.3.2 and 6.3.4)
Ω	The logarithm with base 10 of the prior odds
$P(s\|H_1)$	The probabilities of the same source scores
$P(s\|H_2)$	The probabilities of the different source scores

These variables will mainly be used in the formulas of appendix A.

Chapter 1

Introduction

On March 1 1932, an intruder kidnapped the sleeping newborn Charles Lindbergh, Jr. and two months later the baby's body was found dead. However, a ransom note was left on the window of the bedroom which is shown in figure 1.1.

Figure 1.1: This figure shows the ransom note left in the bedroom of infant Charles Lindbergh, Jr. by the intruder. [15]

Richard Hauptmann was arrested based on other evidence, but over the course of Hauptmann's trial eight handwriting experts testified; they said that the ransom note and samples of Hauptmann's handwriting showed a lot of resemblance. Four years after the kidnapping, in 1936, Hauptmann was convicted of capital murder and was sentenced to death.

So, in the instance of this criminal case (and many more), the (forensic) analysis of handwriting leads to an essential piece of evidence. [15]

Forensic handwriting analysis is an important part of a forensic investigation which can be used when there are threatening letters, hold-up notes, ransom letters, etc. involved. [9] For this analysis two types of writing are compared based on handwriting characteristics in order to determine if they are written

by the same person or not. [6]
Note that this handwriting analysis can be done in the case of other investigations than forensic ones as well. Most often these are fraud investigations; wills, contracts, seals, bank checks, handwritten documents, identification cards, etc. can all be examined in this way. Even when there is suspicion of signature forgery, this analysis can be applied. [9]

There are two ways to perform a handwriting analysis. The handwriting analysis as it is done now, so the more traditional way, is described first. After this, it is explained how the handwriting analysis will be done in this research.
The handwriting analysis, as it is performed right now, is in general done by certified forensic document examiners. [9] First, suspects are asked to write the same text multiple times. Because, although writing is consciously done, repeatedly writing the same words happens almost automatically. In that case the handwriting is individual and unique and therefore can be subjected to a forensic analysis. Key characteristics that the forensic document examiners take into account, when doing this handwriting analysis, are:

- *Letter formations*: How the letters are written in terms of strokes. (How many strokes are used? Is there a continuous stroke or are there multiple strokes that form the letters?)
- *Line quality*: How the writing instrument (for example a pen or pencil) is used. (Features such as pen pressure, speed, number and places of pen lifts, rhythm and writing skill)
- *Alignment*: How the letters are aligned. (Do they all lie on the same baseline?)
- *Arrangement of the writing*: How the letters are arranged. (How much space is between the letters and words? What are the proportions of the letters? How big are the margins?)

If two writings show a lot of similarities based on these characteristics, there is a high probability that they are written by the same person. [6]
For the handwriting analysis that will be done in this report, score-based likelihood ratio (SLR) systems will be employed. This approach has already been proven to be useful for forensic analyses in other research (this was, for example, described by Leegwater et al. [14] and Tang et al. [27]).

One of the benefits of analyzing in the traditional manner, is that every piece of handwriting is compared to the handwriting that has been found (at a crime scene for example) with great attention to detail.
However, this is also drawback since it is very time consuming. Furthermore, every piece of handwriting is compared to one another; only when it is known

to what extent every writing matches the found writing compared to other writings, it is possible to conclude which handwriting matches with the found writing "the most" (so which has the most similarities with the found writing based on the characteristics). This is time consuming as well.
Moreover, this way of performing the analysis does not take the uniqueness of some characteristics into account. For example; say that the found handwriting and that of a suspect both contain "z's" with a stroke in the middle (so it is written as "ƶ"). Then that characteristic leads to a higher likelihood that they have the same writer, if no other person in the world writes the "z" in that way compared to if half of the population writes it in that way. So, the more unique a characteristic is that appears in two writings, the higher the likelihood is that these are written by the same individual. For the same reason, when fingerprints, which are highly unique for every person, match, there is a very high likelihood that they have originated from the same individual.
Lastly, this analysis leads to a categorical conclusion: a decision of identification (two writings have the same writer), a decision of exclusion (two writings have a different writer) or an inconclusive statement. This last conclusion is the case when the writing that has been found does not have enough features for comparison or when there are features that are similar (between the writing that has been found and that of the suspect), but these are insufficient for a decision of identification or exclusion (for example when a lot of people have these features in their writing). A problem in using the categorical conclusions, is that the inconclusive statement only reveals that there is an amount of uncertainty. It does not express the degree of uncertainty. [14]

In this report, systems will be designed that compare handwriting and that do not have these three drawbacks; they are less time consuming, they take the uniqueness of characteristics into account and they give an insight into the degree of uncertainty of the statement that two writings have the same writer. However, the approach of this report also has drawbacks; SLR systems only take a small part of the available information into account and examining the accuracy and performance of SLR systems is complicated. More information on the drawbacks can be found in the Discussion.

Figure 1.2 shows the steps of the procedure of the handwriting comparison system of this report. These steps will be further explained in the following chapters.
Chapter 2 contains a literature review that explains some of the fundamental parts of the analysis of this research. Furthermore, it will describe how an expression can be found for the degree of uncertainty (of the statement that two writings have the same writer) by using the SLR (score-based likelihood ratio). So, it will present a solution to one of the three drawbacks that were stated above. A solution to the second drawback (the analysis is time consuming) is given in chapter 3 by introducing a user interface. This chapter will also explain how the data (that consists out of handwriting samples) is

transformed in such a way that it can be used for the handwriting analysis of this research and it will define the characteristics of the handwriting. So it will focus on the first three boxes of figure 1.2. Chapter 4 solves the third drawback; it describes how the analysis of this research takes the uniqueness of characteristics into account. For this, four different analysis systems are constructed (fourth and fifth box in figure 1.2). The results of these systems are shown in chapter 5. The systems are evaluated in chapter 6 based on the leave-one-out method (cross validation), the 95% bootstrap intervals and the rates of misleading evidence (sixth box in figure 1.2). The system that performs the best, based on these three performance characteristics, will be considered the "best" handwriting analysis system.

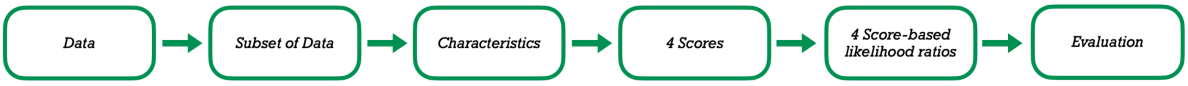


Figure 1.2: This figure shows the steps of the procedure of the handwriting comparison system of this report.

Throughout the report, the variables in the List of variables will be used.
In the appendices, at the end of the report, among other matters, the formulas, detailed calculations and codes (Python and R) can be found.

Chapter 2

Literature Review

A drawback of the current approach to handwriting comparison, is that the inconclusive statement does not express a degree of uncertainty. In this chapter a solution to this problem will be described using LRs and SLRs.
Section 2.1 will explain what LRs and SLRs are, section 2.2 will give the definitions of the common source and specific source problems, section 2.3 will describe the way to calculate the LR and SLR and section 2.4 will demonstrate how to calculate the LR and SLR *specifically* for handwriting.

2.1 What are LRs and SLRs?

The likelihood ratio (LR) is, given two mutually exclusive hypotheses, the ratio of the probabilities of the evidence. For example, these two hypotheses could be those of the decision of identification and of the decision of exclusion. So, in the case of handwriting comparison, if one source is unknown (for example a handwriting that has been found at the crime scene) and one source is known (for example the handwriting of a suspect), then the formula for the LR is given by:

$$LR = \frac{\text{Probability to have this instance of u, k source when they are the same}}{\text{Probability to have this instance of u, k source when they are different}}$$

with u, k: unknown and known.

So, in this way, there is an expression for the degree of uncertainty that a hypothesis is true. Namely the LR. (Note that if $LR > 1$ there is a higher likelihood that u, k source are the same and if $LR < 1$ there is a higher likelihood that u, k source are different.)

The Dutch Forensic Institute (NFI) uses the LR to report the strength of evidence of fingerprints, speaker recognition and weapons and ammunition. [13] Other institutes over the world use LRs to express a degree of uncertainty in the evidence as well. [14]

However, the distributions of the probabilities in the formula (of the LR) above are computationally and statistically infeasible when the number of variables is large. [27] For example, the features of fingerprints are multidimensional,

therefore it is hard to find their distribution. [14] In other words: it is difficult to figure out to what extent two fingerprints are similar without some sort of simplification of these prints. The same holds for the distributions of the characteristics of handwriting (which are multidimensional as well).

A solution to this problem is to use score-based likelihood ratios (SLRs). SLRs use score functions that transform multidimensional data to one dimensional data. This score function is a measure for how similar unknown and known source are to each other. So, only the distributions of the scores when sources are the same and those of the scores when sources are different are required to calculate the SLR. In this way, distributions of one dimensional data (which are less complicated to compute) need to be found instead of those of multi-dimensional data (as for the LR).
For example, the NFI calculates scores for fingerprints with AFIS (automated fingerprint identification system). AFIS indicates to what extent fingerprints are alike by assigning a score after comparing the location and orientation of the ridges of the found fingerprint (unknown source) to all known fingerprints. [14]
In general, the formula for the SLR is given by:

$$SLR = \frac{\text{Probability to have this score when sources are the same}}{\text{Probability to have this score when sources are different}}$$

In 2017 the NFI published a table (in [19]) that contains the verbal expressions connected to LR (and SLR) values. This table is shown in figure 2.1.

Order of magnitude of LR/SLR	*Verbal equivalent*
1-2	As probable
2-10	Slightly more probable
10-100	More probable
100-10,000	Much more probable
10,000-1,000,000	Far more probable
>1,000,000	Exceedingly more probable

Figure 2.1: This figure shows the verbal expressions connected to LR (and SLR) values. [19]

Using this table, LRs (and SLRs) can be reported in a verbal context which is useful in court. For example if the LR of two pieces of handwriting is equal to 8, one could say:

"The information found by comparing the two pieces of handwriting is 8 times

more probable if they come from the same writer than if they come from different writers."

The verbal expression that can be used in court is:

"The information found by comparing the two pieces of handwriting is slightly more probable if they come from the same writer than if they come from different writers."

Now it is possible to express the degree of uncertainty of the statement that two writings (unknown and known source) have the same writer.
However, it is important to note two things;

1. Because the score function transforms multidimensional data to one dimensional data, information is lost. [27] For example, in the case of fingerprints; multiple fingerprints can have the same AFIS score (so they are similar to another fingerprint to the same degree) while not being the same fingerprints.
2. As described by Leegwater et al. [14], because of this information loss and other reasons, LRs and SLRs are best used in combination with the (subjective) opinion of forensic examiners. This way, it is possible to benefit from both the objectivity and transparency of the LR (and SLR) systems and from the knowledge and expertise of forensic examiners (who take more information into account than the LR and SLR systems).

What the common source and specific source problems are will be explained in the next section and how the LR and SLR are generally calculated will be explained in section 2.3. Section 2.4 will describe a way to find the LR and SLR when the sources are handwriting.

2.2 Common source and specific source problems

With the help of the LR and SLR, the hypotheses of two problems can be tested. Namely, the common source problem and the specific source problem as developed and specified in [21] and [22].
In the case of the common source problem, it is tested if two pieces of evidence x_{u1} and x_{u2} (both with an unknown source) originate from the same unknown source. So, it is not investigated who the unknown source is. The hypotheses of the common source problem, in the case that the pieces of evidence are handwriting, are;

H_1: The pieces of handwriting (x_{u1}, x_{u2}) from unknown writers originate from the same unknown writer,

H_2: The pieces of handwriting (x_{u1}, x_{u2}) from unknown writers originate from

two different unknown writers.

In the case of the specific source problem, it is tested if a piece of evidence x_u (with an unknown source) originates from a known specific suspected source x_s. The hypotheses of the specific source problem, in the case that the pieces of evidence are handwriting, are;

H_1: The piece of handwriting (x_u) from an unknown writer and the piece of handwriting from a known specific writer (x_s) originate from the same known specific writer,

H_2: The piece of handwriting (x_u) from an unknown writer does not originate from the known specific writer (x_s), but from an alternative writer.

Which of the two problems should be used in forensic science, is still being researched.
In the case of an ongoing investigation, the common source problem is more appropriate. So for example when one wants to investigate if two pieces of handwriting from different crime scenes originate from the same source. In court the specific source problem is more appropriate, since in that case it is of interest whether the piece of handwriting originates from the suspect.
Furthermore, for the common source problem one background population data set is needed. This data set is required for the specific source problem as well, but this problem also needs a specific source data set. However, obtaining this specific source data set is complicated. For example, in an ongoing investigation the suspect, so the specific writer, can change their handwriting or can be uncooperative. Therefore, most of the time, there is not enough handwriting available to create the specific source data set and thus the LR or SLR cannot be used for the specific source problem. So, the common source problem is more suitable when data is limited. For this reason, only the common source problem is considered in this report. [23]

2.3 Calculation of LRs and SLRs

In section 2.1, it was explained that the LR is an expression for the degree of uncertainty that a hypothesis is true. However, since LRs are computationally and statistically infeasible when the number of variables are large, the SLR is used. This SLR uses a score function to transform multidimensional data to one dimensional data and is therefore less complicated to compute.
This section will elaborate on how the LR and SLR are calculated when the sources are writings, but the formulas can be generalized for all kinds of sources. The notation as described in [14] is used.

As mentioned in section 2.1, the LR is, given two mutually exclusive hypotheses, the ratio of the probabilities of the evidence. In the case where the sources are handwriting, the following hypotheses are used:

H_1: The handwriting with unknown source and the handwriting with known source are from the same person,

H_2: The handwriting with unknown source and the handwriting with known source are from different people.

Here the writing with unknown source is, for example, found at the crime scene and the writing with known source could be of a suspect. Note that these two hypotheses are mutually exclusive; if one is false, the other must be true (and vice versa) (so they cannot both be true or both be false).
Hypotheses H_1 and H_2 are sometimes also called the prosecution hypothesis and the defence hypothesis, respectively, since they are the hypotheses that the prosecutor and defendant want to prove to be true in the courtroom. [8]

2.3.1 Calculation of LR

The formula for the LR given in section 2.1 was:

$$LR = \frac{\text{Probability to have this instance of u, k source when they are the same}}{\text{Probability to have this instance of u, k source when they are different}}$$

with u, k: unknown and known.

This can now be rewritten into:

$$LR(x, y) = \frac{P(x, y \mid H_1, I)}{P(x, y \mid H_2, I)} \tag{2.1}$$

with:
x = The handwriting with known source
y = The handwriting with unknown source
$P(x, y)$ = The joint probability function of x and y
H_1, H_2 = The hypotheses (as defined before)
I = The relevant background information

Relevant background information (I) could be additional information about the case or the sources. In the case where the sources are writings, information that could be added to I is the uniqueness of a characteristic, the number of characteristics, if the suspects are left- or right-handed and additional information about the found writing. Adding this information to I is called conditioning or anchoring. However, as mentioned (for fingerprints) in [14], conditioning with respect to the writing with unknown source is impossible and conditioning with respect to the writing with known source is impractical. So, in this report, no further information relevant to the writing is added to I. (However, in chapter 4 score functions will be proposed that take the

uniqueness of characteristics into account.)

2.3.2 Calculation of SLR

The formula for the SLR given in section 2.1 was:

$$SLR = \frac{\text{Probability to have this score when sources are the same}}{\text{Probability to have this score when sources are different}}$$

Which can be rewritten into:

$$SLR(x, y) = \frac{P(s(x, y) \mid H_1, I)}{P(s(x, y) \mid H_2, I)} \tag{2.2}$$

Where the notation is the same as for the formula of the LR, but with:
$s(x, y)$ = The (similarity) score (that is a measure for how similar known (x) and unknown source (y) are to each other)
$P(s(x, y))$ = The probability function of the (similarity) score

As described by Morrison et al. [17], scores should take both the similarity and the typicality of the evidence into account (anchored approach). Here typicality means that the same and different source scores of the suspect should be used and not those of the general population (so this is the specific source problem as described in section 2.2). In the case of sources that are handwriting, this means that, for the numerator, the characteristics of the suspect's writing are required. For the denominator, the characteristics of the writing of the general population compared to those of the suspect are needed. However, as explained in section 2.2, this data (of the suspect) is unavailable. Therefore the nonanchored approach is applied; scores only take similarity into account (and not typicality). So, for the calculation of the SLR, hypotheses are used that consider the general population instead of a specific suspect (so this is the common source problem as described in section 2.2). This means that the SLR is an expression for the degree of uncertainty that a hypothesis is true for the general population, not a specific suspect in the case. [14]

2.3.3 Calculation of posterior odds

Now, when the SLR is multiplied with the prior odds (the ratio of the probabilities of H_1 and H_2 beforehand) the posterior odds are obtained (the ratio of the probabilities of H_1 and H_2 afterwards, so it can be based on other information in the case and on the score). Therefore:

$$\text{Posterior Odds} = SLR \cdot \text{Prior Odds} \tag{2.3}$$

Which can be rewritten into:

$$\text{Posterior Odds} = SLR \cdot \text{Prior Odds} = \frac{P(s(x,y) \mid H_1, I)}{P(s(x,y) \mid H_2, I)} \cdot \frac{P(H_1 \mid I)}{P(H_2 \mid I)} \quad (2.4)$$

So:

$$\text{Posterior Odds} = \frac{P(s(x,y) \mid H_1, I)}{P(s(x,y) \mid H_2, I)} \cdot \frac{P(H_1 \mid I)/P(s(x,y))}{P(H_2 \mid I)/P(s(x,y))}$$

Now, using Bayes' Theorem, that states that $P(A \mid B) = \frac{P(B|A) \cdot P(A)}{P(B)}$ [10], gives:

$$\text{Posterior Odds} = \frac{P(H_1 \mid s(x,y), I)}{P(H_2 \mid s(x,y), I)} \quad (2.5)$$

Lastly, combining equations (2.4) and (2.5), gives:

$$\frac{P(H_1 \mid s(x,y), I)}{P(H_2 \mid s(x,y), I)} = \frac{P(s(x,y) \mid H_1, I)}{P(s(x,y) \mid H_2, I)} \cdot \frac{P(H_1 \mid I)}{P(H_2 \mid I)} \quad (2.6)$$

So, the posterior odds is the ratio of the probabilities of H_1 and H_2 given a score and relevant background information.

2.4 LR and SLR for multinomial features

This section will describe a way to find the LR and SLR *specifically* when the sources are handwriting.
Handwriting, just like a fingerprint, has distinct features. Examples of these features are: alignment of the letters with respect to a baseline, the space between letters, proportions of the letters, etc.
These features can be binary; handwriting either has these feature or it does not (so it can take on two values). For example; (in general) words are either written in cursive or in print.
Features can also be multinomial (sometimes also called categorical); a feature can take on multiple values. For example; the height of the cross of the staff of the letter "t" (it can be in the middle, in the upper part or in the lower part). Note that a multinomial feature can also take on two values and therefore it can be a binomial feature as well. In this section, it will be explained how the LR and SLR are calculated for handwriting with multinomial features.

The notation as described in [27] is used. So, let $\mathbf{x} = [x_1, \ldots, x_n]$ be the set of features of a handwriting. x_i is one feature of the handwriting which can take

exactly one of the n_i values. So, x_i is *exactly one* element in $\{x_i^1, \ldots, x_i^{n_i}\}$. Figure 2.2 shows an example; the features of the writing of "th" that were given by forensic document examiners [18] are shown. The table in figure 2.3 shows the features (x_i) of this example and the values they can take ($\{x_i^1, \ldots, x_i^{n_i}\}$). "NSP" means "no set pattern".
So, in this example, there are six different features which means that $n = 6$. x_1 (height relationship of t to h) can be exactly one of four different values; x_1^1 (t even with h), x_1^2 (t shorter than h), x_1^3 (t taller than h) or x_1^4 (NSP). So, $n_1 = 4$. In the same way; $n_2 = 4$, $n_3 = 3$, $n_4 = 4$, $n_5 = 4$ and $n_6 = 5$. [27]

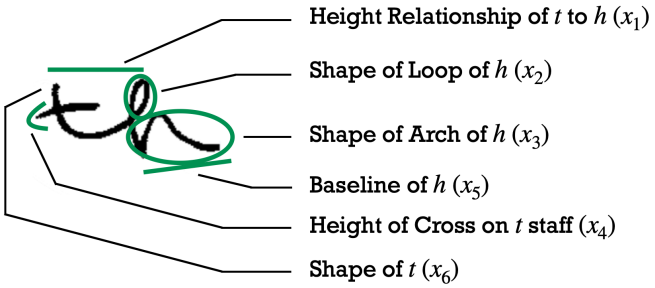


Figure 2.2: This figure shows the features of the writing of "th" that were given by forensic document examiners. [27]

Variable	*Description*	*Values*
x_1	t / h relative heights	x_1^1 : t even with h; x_1^2 : t shorter than h; x_1^3 : t taller than h; x_1^4 : NSP
x_2	Shape of loop of h	x_2^1 : curved left, straight right; x_2^2 : curved right, straight left; x_2^3 : both sides curved; x_2^4 : retraced; x_2^5 : NSP
x_3	Shape of arch of h	x_3^1 : pointed; x_3^2 : rounded; x_3^3 : NSP
x_4	Cross of t	x_4^1 : cross above; x_4^2 : lower half cross of t; x_4^3 : upper half cross of t; x_4^4 : NSP
x_5	Baseline of h	x_5^1 : even; x_5^2 : slanting down; x_5^3 : slanting up; x_5^4 : NSP
x_6	Shape of t	x_6^1 : closed; x_6^2 : looped; x_6^3 : tented; x_6^4 : single stroke; x_6^5 : NSP

Figure 2.3: This figure shows a table with the features (x_i) of the example and the values they can take ($\{x_i^1, \ldots, x_i^{n_i}\}$). "NSP" means "no set pattern". [27]

In the previous section, the formulas for the LR (equation (2.1)) and the SLR (equation (2.2)) were given. They are repeated below.

$$LR(x, y) = \frac{P(x, y \mid H_1, I)}{P(x, y \mid H_2, I)}$$

$$SLR(x, y) = \frac{P(s(x, y) \mid H_1, I)}{P(s(x, y) \mid H_2, I)}$$

Now, let the handwriting with known source (x) and the handwriting with unknown source (y) have multinomial features. That is; let $\mathbf{x} = [x_1, \ldots, x_n]$ be the set of features of a handwriting with a known source and let $\mathbf{y} = [y_1, \ldots, y_n]$ be the set of features of a handwriting with an unknown source. x_i is one feature of the handwriting with known source which can take *exactly one* of the n_i values and y_i is one feature of the handwriting with unknown source which can take *exactly one* of the n_i values. (So, both sources have the same amount of features which can take on the same values.) The features are chosen (by the forensic document examiners) in such a way that it can be assumed that they are independent of one another.
The probabilities in the formula of the LR above are calculated by taking all of the characteristics x_i and y_i (of x and y respectively) into account. The SLR is calculated with a similarity score function $s(x, y)$ that takes the similarity scores of each x_i and y_i into account. This can be done in multiple ways. An example of such a score function is:

$$s(x, y) = \sum_{i=1}^{n} w_i \cdot s_i(x_i, y_i)$$

Where;
x_i = The ith feature of the handwriting with known source
y_i = The ith feature of the handwriting with unknown source
$s_i(x_i, y_i)$ = The (similarity) score of x_i and y_i
w_i = The weight assigned to the ith feature of the handwriting
n = The total number of characteristics [2]

This formula will be further explained in chapter 4. In that chapter the formulas for w_i and $s_i(x_i, y_i)$ will be given as well.

Figure 2.4 shows how the SLR is calculated from handwriting data. Therefore, it serves as a summary of this chapter. So, the background population handwriting data is used to calculate the same source and different source scores with the help of a score function that transforms characteristics to scores. A graph with the parametrizations of the distributions of the same source and

different source scores is created (with the score on the x-axis and the probability on the y-axis). The SLR is obtained when the probabilities of the same source scores in this graph are divided by the probabilities of the different source scores. When, in the common source problem, unknown handwriting data is found at crime scenes, the test score can be calculated with the help of the score function. The SLR of this unknown handwriting data is acquired when the probability of the same source score corresponding to the test score is divided by the probability of the different source score corresponding to the test score.

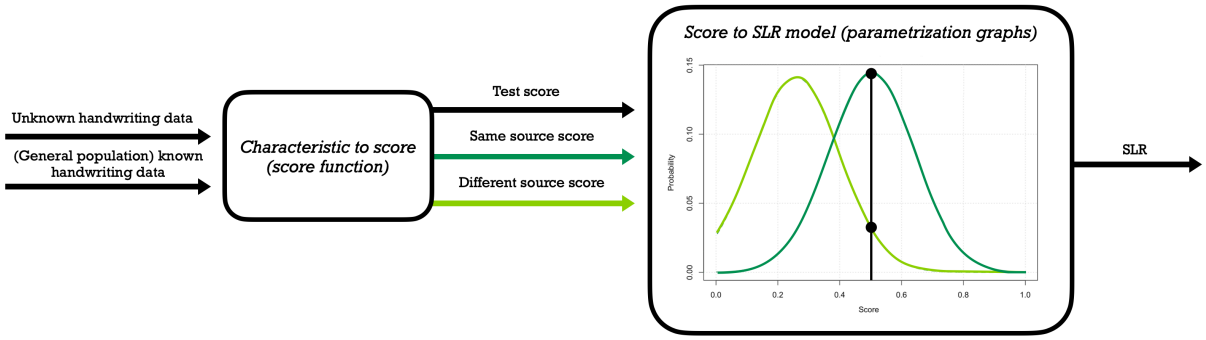


Figure 2.4: This figure shows how the SLR is calculated from handwriting data. Therefore, it serves as a summary of this chapter.

In this report a different letter combination (than "th") will be used to compare handwriting, but it will still follow this method of calculation for four different score functions. [2] contains several score functions that can be employed for sources with multinomial features. Overlap, Goodall3, Burnaby and Anderberg will be applied in this report and in chapter 4 it will be explained how these scores are calculated. The resulting SLR systems will be discussed in chapter 5. Furthermore, the evaluation of the quality of performance of the four SLR systems will be done in the same way as in [14] (the SLR system that performs the best according to this evaluation, will be considered the "best" SLR system). More on this in chapter 6.

In this chapter, one of the three problems of the current approach to handwriting comparison was solved; first there was not an expression for the degree of uncertainty that a hypothesis is true, but now there is. Namely, the SLR. In the next two chapters the other two problems (the analysis is time consuming and does not take uniqueness into account) are solved by introducing a user interface and by applying certain kinds of score functions.

Chapter 3

Methods: Data preparation

In this chapter, the data of handwriting (for which the SLR systems will be constructed) is determined. But, in order to calculate the SLR, the data is required to be of a specific kind. This chapter will also describe how the data is prepared in such a way that in the following chapters the SLR can be calculated without having to transform the data first. Lastly, in the previous chapter the first problem of the current approach to handwriting comparison, was solved; an expression for the degree of uncertainty that a hypothesis is true was given. In this chapter the second problem will be solved; the analysis will become less time consuming.

The data preparation is done by following the steps of the procedure of data preparation as given in [25]. That is;

1. *Handwriting Samples*: The collection of samples of handwriting data is determined (section 3.1).

2. *Letter Combination*: The letter combination, that is extracted from the samples and will be used to calculate the SLR, is determined (section 3.2).

3. *Characteristics*: The characteristics of the letter combination are determined. This is done in the same manner as for "th" in figures 2.2 and 2.3 (section 3.3).

4. *Extraction of Snippets*: Snippets of samples (of handwriting) are extracted that contain the letter combination of interest (section 3.4).

5. *User Interface*: A user interface will be created in order to make the analysis more time efficient (so it will solve the second problem) (section 3.5).

6. *Ground-truthing*: By utilizing the user interface, values are assigned for each of the characteristics such that the data can be used to calculate the SLR. This step completes the data transformation (section 3.6).

3.1 Determination of handwriting samples

Samples of handwriting data are collected from the CEDAR data set. CEDAR is the Center of Excellence for Document Analysis and Recognition at the University at Buffalo [3]. The data set can be found in [4] under "CEDAR-LETTER" and it contains writing samples from over 1500 individuals (in the United States). For this, a piece of text, which contains every letter of the English alphabet at least once, was written by each individual three times (because repeatedly writing the same words happens almost automatically). The piece of text that was used can be found in appendix B. This appendix also contains a handwritten sample of the text provided by a writer.

For this report, the first 800 (of 1500) writers (of the CEDAR data set) were considered. This was done in order to accelerate the research while still taking more than half of the data into account. More on this in the Discussion.
As mentioned before, every writer has three corresponding written documents. All three will be taken into account for every writer.
Figure 3.1 shows which documents are compared to each other in order to obtain the same source and different source scores with the common source problem. The comparison of the documents with unknown sources x_{u1} and x_{u2} (as in this common source problem) is shown as well.
So, for the same source scores, all three documents of every writer have to be compared with one another (see figure 3.1). This results in three same source scores per writer; a score for documents 1 and 2, a score for documents 1 and 3 and a score for documents 2 and 3. So, in total there are $800 \cdot 3 = 2400$ same source scores.
Note that, for example, the score for documents 1 and 2 is the same as the score for documents 2 and 1. Taking these other scores into account as well would result in $2400 \cdot 2 = 800 \cdot 6 = 4800$ same source scores. This would not change the SLR models, since each of the 2400 scores would occur twice in this case and because the parametrization, that is used to obtain the SLR models, is performed on the histograms which use probabilities. However, using a data set of 2400, instead of 4800 same source scores, saves time when doing computations in R. This is why the smaller data set of 2400 same source scores was chosen for this research.
For the different source scores, each of the documents of the first writer are compared with the other $800 \cdot 3 - 3 = 2397$ documents (all the documents of the other writers) (see figure 3.1). So, the three documents of the first writer have $3 \cdot 2397 = 7191$ different source scores. For the second writer, each of the documents are compared with the other $800 \cdot 3 - 3 - 3 = 2394$ documents. This is the case, because the different source scores of the documents of the first and second writer were already computed when comparing the documents of the first writer with those of the other writers. This results in $3 \cdot 2394 = 7182$ different source scores for the three documents of the second writer. Repeating this process for the other writers results in $3 \cdot 2397 + 3 \cdot 2394 + 3 \cdot 2391 + \cdots + 3 \cdot 3 +$

$3{\cdot}0 = 3{\cdot}3{+}\cdots{+}3{\cdot}2391{+}3{\cdot}2394{+}3{\cdot}2397 = \sum_{i=1}^{799} 3{\cdot}(3i) = \sum_{i=1}^{799} 9i = 2,876,400$ different source scores in total.
Note that, for example, the score for document 1 of writer 1 and document 1 of writer 2 is the same as the score for document 1 of writer 2 and document 1 of writer 1. Taking these other scores into account as well would result in $2,876,400 \cdot 2 = 2400 \cdot 2397 = 5,752,800$ different source scores. But, as explained before for the same source scores, this would not change the SLR models. However, using a data set of 2,876,400, instead of 5,752,800 different source scores, saves time when doing computations in R. This is why the smaller data set of 2,876,400 different source scores was chosen for this research.
The documents with unknown sources x_{u1} and x_{u2} (as in the common source problem) only need to be compared with each other (see figure 3.1). This results in one test score (where the test score is defined as in figure 2.4).

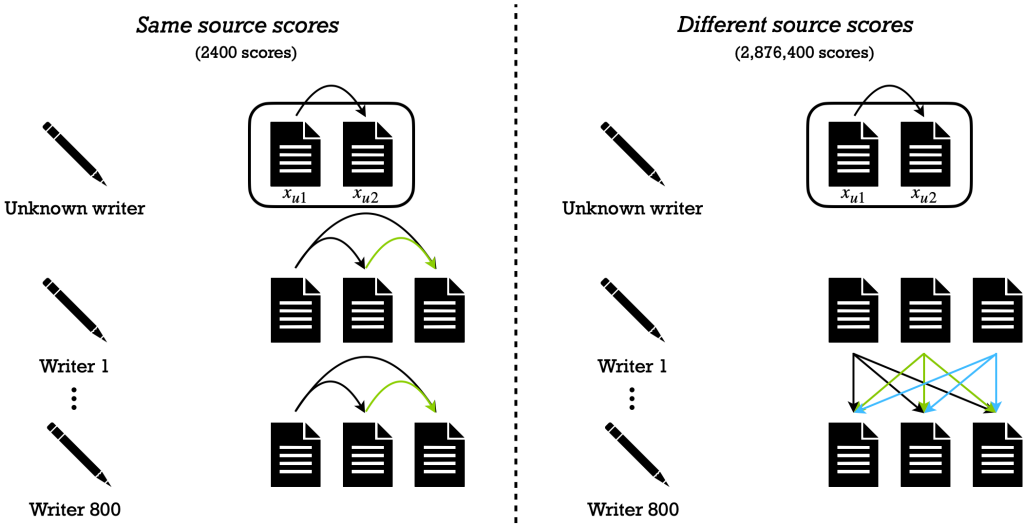


Figure 3.1: This figure shows which documents are compared to each other in order to obtain the same source and different source scores (with the common source problem and with unknown sources x_{u1} and x_{u2}).

Note that the scores of each of the documents are calculated, not the scores of the writers; the degree of similarity between two documents needs to be found, not the degree of similarity between the three documents of one writer and the three documents of another.
More information about these scores can be found in chapters 4 and 5.

3.2 Determination of the letter combination

In section 2.4 the handwriting was compared based on the characteristics of the letter combination "th". In this report, the bigram "er" will be used. Here a bigram is the combination of two letters. [7]
The letter combination "er" was chosen, because, if only the bigrams that do not span across consecutive words are taken into account, it is the second most frequent occurring letter pair in the English language (See figure 3.2) and the third most frequent occurring one in the Dutch language (See figure 3.3). This way, the SLR systems created in this report can be used for documents that are written in English or in Dutch.
In the next section the characteristics of the letter combination "er" will be described.

Bigram	Count	Bigram	Count	Bigram	Count
th	50	at	25	st	20
er	40	en	25	io	18
on	39	es	25	le	18
an	38	of	25	is	17
re	38	or	25	ou	17
he	33	nt	24	ar	16
in	31	ea	22	as	16
ed	30	ti	22	de	16
nd	30	to	22	rt	16
ha	26	it	20	ve	16

Figure 3.2: This figure shows the most frequent occurring letter pairs in the English language per 2000 letters (that do not span across consecutive words). "er" is the second most frequent. [25]

Bigram	Count	Bigram	Count	Bigram	Count	Bigram	Count
n<	384	te	135	el	96	<g	70
en	370	in	131	st	93	ar	70
e<	308	ee	119	s<	87	00	69
de	258	r<	118	nd	84	<b	69
er	232	aa	111	<o	79	ng	69
t<	194	he	106	va	77		
<d	178	et	105	ch	76		
an	172	<h	105	re	75		
<v	143	<e	97	ve	74		
ge	137	ie	97	or	74		

Figure 3.3: This figure shows the most frequent occurring letter pairs in the Dutch language per 10,000 bigrams. "<" represents a space. "er" is the third most frequent (that does not span across consecutive words).[5]

3.3 Determination of the characteristics

In section 2.4, the characteristics of the bigram "th" were given by forensic document examiners. However, there is no data available on what these examiners classify as the characteristics of the letter combination "er". That is why the characteristics of "er" were obtained by looking at those of "th" (as described in section 2.4) and by looking at the CEDAR data base. The characteristics are chosen in such a way that it can be assumed that they are independent of one another (for the reason that was explained in section 2.4). Figure 3.4 shows the features of the writing of "er". The table in figure 3.5 shows the features (x_i) of this bigram and the values they can take ($\{x_i^1, \ldots, x_i^{n_i}\}$). Again "NSP" means "no set pattern".
The feature x_2 (Shape of "r") might be difficult to visualize. For that reason appendix C contains examples out of the CEDAR data set of the different kinds of shapes (so examples of $\{x_2^1, x_2^2 \ldots, x_2^5\}$, because x_2^6 is the case if the feature is "NSP").
The bigram "er" has $4 \cdot 6 \cdot 4 \cdot 4 \cdot 3 \cdot 4 \cdot 4 \cdot 5 = 92,160$ possible combinations of characteristics (for this calculation the number of values that each of the eight characteristics can take are multiplied).

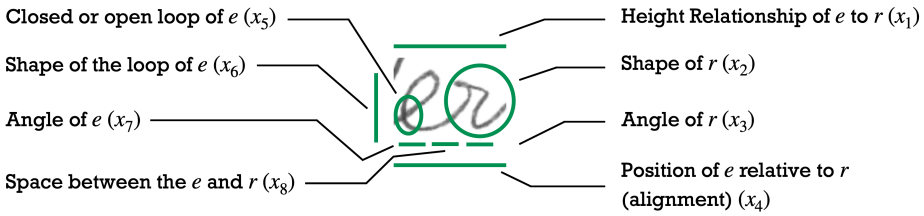


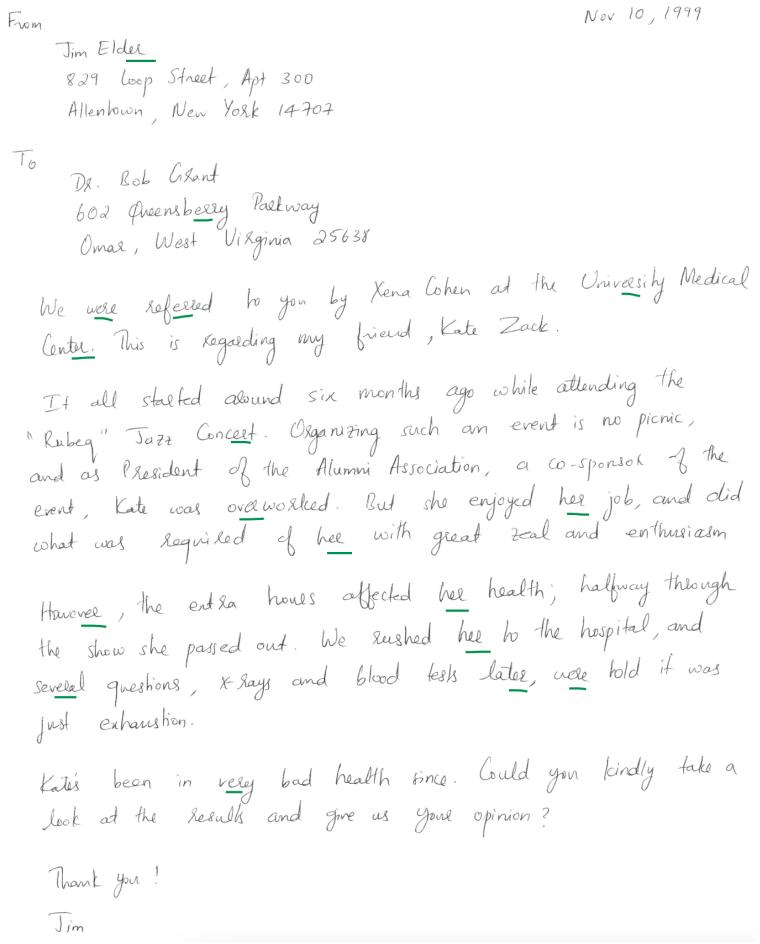
Figure 3.4: This figure shows the features of the writing of "er".

Variable	*Description*	*Values*
x_1	e / r relative heights	x_1^1 : e even with r; x_1^2 : e shorter than r; x_1^3 : e taller than r; x_1^4 : NSP
x_2	Shape of r	x_2^1 : cursive with loop; x_2^2 : cursive without loop; x_2^3 : cursive without horizontal piece; x_2^4 : in print (one stroke); x_2^5 : in print (two strokes); x_2^6 : NSP
x_3	Angle of r	x_3^1 : r leans to the right; x_3^2 : r leans to the left; x_3^3 : r stands upright; x_3^4 : NSP
x_4	Position of e relative to r (alignment)	x_4^1 : e is higher than r; x_4^2 : e is lower than r; x_4^3 : e and r lie on the same baseline; x_4^4 : NSP
x_5	Closed or open loop of e	x_5^1 : open loop; x_5^2 : closed loop; x_5^3 : NSP
x_6	Shape of the loop of e	x_6^1 : curved up; x_6^2 : curved down; x_6^3 : not curved; x_6^4 : NSP
x_7	Angle of e	x_7^1 : e leans to the right; x_7^2 : e leans to the left; x_7^3 : e stands upright; x_7^4 : NSP
x_8	Space between the e and r	x_8^1 : no space between e and r; x_8^2 : small space between e and r; x_8^3 : medium space between e and r; x_8^4 : large space between e and r; x_8^5 : NSP

Figure 3.5: This figure shows a table with the features (x_i) of "er" and the values they can take ($\{x_i^1, \ldots, x_i^{n_i}\}$). "NSP" means "no set pattern".

3.4 Extraction of snippets

Figure 3.6 shows the seventeen locations of "er" (in the written documents) which are underlined in green.



From

Nov 10, 1999

Jim Elder
829 Loop Street, Apt 300
Allentown, New York 14707

To

Dr. Rob Grant
602 Queensberry Parkway
Omar, West Virginia 25638

We were referred to you by Xena Cohen at the University Medical Center. This is regarding my friend, Kate Zack.

It all started around six months ago while attending the "Rubeq" Jazz Concert. Organizing such an event is no picnic, and as President of the Alumni Association, a co-sponsor of the event, Kate was overworked. But she enjoyed her job, and did what was required of her with great zeal and enthusiasm

However, the extra hours affected her health; halfway through the show she passed out. We rushed her to the hospital, and several questions, X-Rays and blood tests later, were told it was just exhaustion.

Kate's been in very bad health since. Could you kindly take a look at the results and give us your opinion?

Thank you!

Jim

Figure 3.6: This figure shows the seventeen locations of "er" (in the written documents) (this is document 1 of writer 1) which are underlined in green.

These snippets were extracted from the document and merged into one image by the use of [12]. This is done such that all of the snippets of "er" can be analyzed at the same time.
The result of this for the seventeen snippets of "er" of document 1 of writer 1, is shown in figure 3.7.

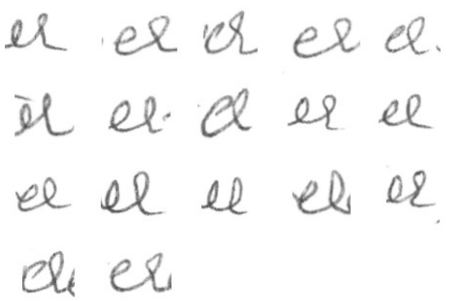


Figure 3.7: This figure shows the seventeen snippets of “er” of document 1 of writer 1 merged into one image.

The next section will discuss the construction of a user interface that will make the analysis more time efficient and that uses images like figure 3.7.

3.5 Creation of a user interface

In order to analyze the writing of every document in a time efficient manner, a user interface was created. This was done by using the images that were described in the previous section. Figure 3.8 shows what this looks like.
The Python code, that was used in order to create this user interface, can be found in appendices D.1 (the creation of the option menus) and D.2 (displaying the bigrams).

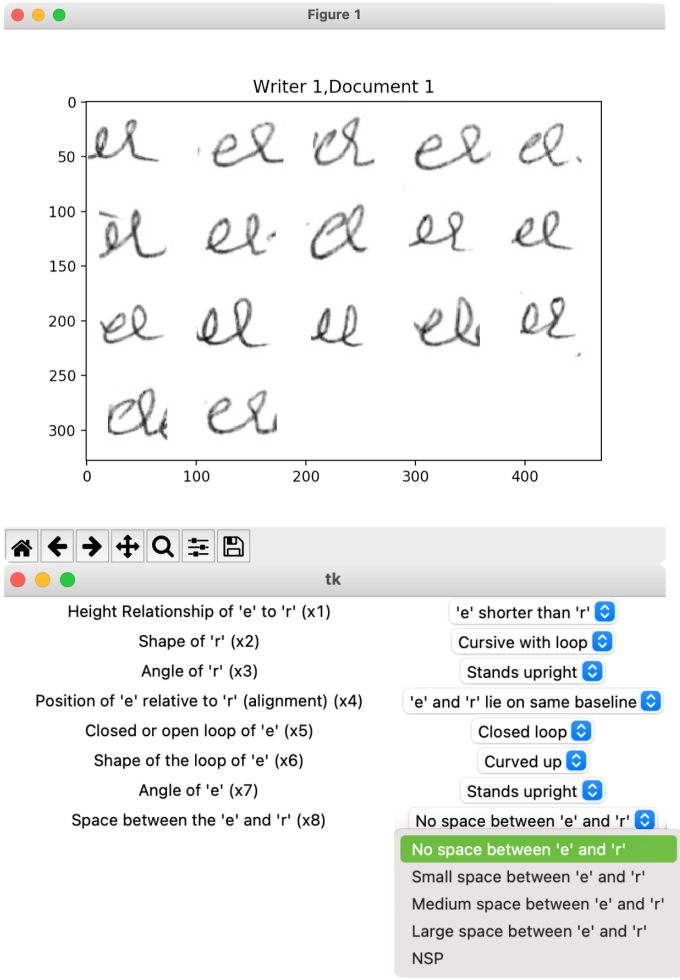


Figure 3.8: This figure shows what the user interface (that is used to analyze the writing of every document) looks like.

Every characteristic still needs to be entered into the computer by hand. However, this is an improvement in terms of time efficiency from the current approach to handwriting comparison; now every document needs to be analyzed once instead of once to compare with the unknown source and then once to compare with every other document to obtain the degree of similarity. Besides, with the user interface, only the characteristics need to be entered into the computer by hand; the computer compares the documents. This means that, now, two of the three problems are solved; this analysis has an expression for the degree of uncertainty that a hypothesis is true and the analysis is less time consuming.

The next section will explain, by utilizing the user interface, what values are assigned for each of the characteristics such that the data can be used to calculate the SLR.

3.6 Ground-truthing the letter combination

The user interface of the previous section is utilized; for every document of every writer the characteristics are entered into the computer by using the option menus. This information is then transformed into a binary vector (a vector that only has entries "1" or "0"). For example; if the feature x_1 (height relationship of "e" to "r") (with options {"e" even with "r", "e" shorter than "r", "e" taller than "r", NSP}) takes the value "e" even with "r", then the x_1 binary vector becomes $[1, 0, 0, 0]$. This vector is then entered into an Excel file. The result for the first two writers is shown in figure 3.9. The Python code used for this transformation can be found in appendix D.1.

Characteristic	*W 1, D1*	*W 1, D2*	*W1, D3*	*W2, D1*	*W2, D2*	*W2,D3*
x_1	[0, 1, 0, 0]	[0, 1, 0, 0]	[0, 1, 0, 0]	[1, 0, 0, 0]	[1, 0, 0, 0]	[1, 0, 0, 0]
x_2	[1, 0, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0]	[0, 0, 0, 1, 0, 0]	[0, 0, 0, 1, 0, 0]	[0, 0, 0, 1, 0, 0]
x_3	[0, 0, 1, 0]	[0, 0, 1, 0]	[0, 0, 1, 0]	[0, 0, 1, 0]	[0, 0, 1, 0]	[0, 0, 1, 0]
x_4	[0, 0, 1, 0]	[0, 0, 1, 0]	[0, 0, 1, 0]	[0, 1, 0, 0]	[0, 1, 0, 0]	[0, 0, 1, 0]
x_5	[0, 1, 0]	[0, 1, 0]	[0, 1, 0]	[1, 0, 0]	[0, 1, 0]	[0, 1, 0]
x_6	[1, 0, 0, 0]	[1, 0, 0, 0]	[1, 0, 0, 0]	[1, 0, 0, 0]	[1, 0, 0, 0]	[1, 0, 0, 0]
x_7	[0, 0, 1, 0]	[0, 0, 1, 0]	[0, 0, 1, 0]	[0, 0, 1, 0]	[0, 0, 1, 0]	[0, 0, 1, 0]
x_8	[1, 0, 0, 0, 0]	[1, 0, 0, 0, 0]	[1, 0, 0, 0, 0]	[0, 1, 0, 0, 0]	[0, 1, 0, 0, 0]	[0, 1, 0, 0, 0]

Figure 3.9: This figure shows the table of the vectors of the eight characteristics for the first two writers. The number after "W" is the number of the writer and the number after "D" is the document number.

The transformation from characteristics into binary vectors is necessary in order to calculate the scores that are described in the next chapter. This is the case because, for the calculation of these scores, it is required to know if $x_i = y_i$ or $x_i \neq y_i$ for every i (where x_i is a feature of the handwriting with known source and y_i is that of the handwriting with unknown source). It is less complicated to know when this equality or inequality holds if binary vectors are used (instead of just the characteristics); every entry of the vector x_i and y_i need to be the same to have an equality, otherwise $x_i \neq y_i$. So, for example; $[1, 0, 0, 0] = [1, 0, 0, 0]$ and $[1, 0, 0, 0] \neq [0, 1, 0, 0]$.

This step completes the data transformation. The next chapter will explain how the SLR systems are constructed.

Chapter 4

Methods: Construction of SLR systems

This chapter will describe the four different scores that are used to construct four SLR systems. Three of these systems take the uniqueness of the characteristics of the handwriting into account. This means that the third problem of the current approach to handwriting comparison is solved in this chapter.

As explained in chapter 2, by [2];

$$s(x, y) = \sum_{i=1}^{n} w_i \cdot s_i(x_i, y_i) \tag{4.1}$$

Where;
x_i = The ith feature of the handwriting with known source
y_i = The ith feature of the handwriting with unknown source
$s_i(x_i, y_i)$ = The (similarity) score of x_i and y_i
w_i = The weight assigned to the ith feature of the handwriting
n = The total number of characteristics (in this report equal to 8)

The following four sections (sections 4.1 to 4.4) will discuss one score each for which the formulas of $s_i(x_i, y_i)$ and w_i in equation (4.1) will be given.

4.1 SLR construction with score 1: Overlap

$$s_i(x_i, y_i) = \begin{cases} 1 & \text{if } x_i = y_i \\ 0 & \text{otherwise} \end{cases}$$

And $w_i = \frac{1}{n} = \frac{1}{8}$.

So, the "overlap" score counts the number of characteristics that match between known source x and unknown source y and divides it by the total number of characteristics n (so 8). $s_i(x_i, y_i)$ (and therefore also $s(x, y)$) has range $[0, 1]$ where $s(x, y) \in \{0, \frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{4}{8}, \frac{5}{8}, \frac{6}{8}, \frac{7}{8}, 1\}$. [2]

The "overlap" score is the least complicated score of the four and the next three scores are all extensions of this one. It is expected that those three will outperform the first score, because they take the uniqueness of the characteristics into account and the "overlap" score does not. More on this in chapter 6.

4.2 SLR construction with score 2: Goodall3

The "Goodall 3" score assigns higher scores if the matching values are unique. The formulas are;

$$s_i(x_i, y_i) = \begin{cases} 1 - p_i^2(x_i) & \text{if } x_i = y_i \\ 0 & \text{otherwise} \end{cases}$$

And $w_i = \frac{1}{n} = \frac{1}{8}$.

Here;

$$p_i^2(x_i) = \frac{f_i(x_i) \cdot (f_i(x_i) - 1)}{N \cdot (N-1)}$$

With;
$f_i(x_i)$ = The number of times the ith feature takes the value x_i in the data set
N = The number of data points (in this report equal to $800 \cdot 3 = 2400$)

$s_i(x_i, y_i)$ takes the minimum value when, for every data point, the ith feature takes the value x_i. So, when x_i is the only value for the ith feature. Then $f_i(x_i) = N$, so $p_i^2(x_i) = \frac{f_i(x_i) \cdot (f_i(x_i)-1)}{N \cdot (N-1)} = \frac{N \cdot (N-1)}{N \cdot (N-1)} = 1$ which means that $s_i(x_i, y_i) = 1 - 1 = 0$.
$s_i(x_i, y_i)$ takes the maximum value when x and y are the only two data points for which the ith feature takes the value x_i. In that case $f_i(x_i) = 2$, so $p_i^2(x_i) = \frac{2 \cdot (2-1)}{N \cdot (N-1)} = \frac{2}{N \cdot (N-1)}$. This means that $s_i(x_i, y_i) = 1 - \frac{2}{N \cdot (N-1)}$. Note that if there is only one data point x for which the ith feature takes the value x_i, there is no y for which $x_i = y_i$. Therefore $s_i(x_i, y_i) = 0$.
So, it can be concluded that $s_i(x_i, y_i)$ (and therefore also $s(x, y)$ (for which $s_i(x_i, y_i)$ is summed over i and divided by 8)) has range $[0, 1 - \frac{2}{N \cdot (N-1)}] \approx [0, 1]$ (for $N = 2400$ in this report).[2]

4.3 SLR construction with score 3: Burnaby

The "Burnaby" score assigns lower scores if the values that do not match are unique. If the values that do not match are common, so not unique, a higher score is assigned. Matching values receive a score of 1 regardless of their uniqueness.

$$s_i(x_i, y_i) = \begin{cases} 1 & \text{if } x_i = y_i \\ \frac{\sum_{q \in A_i} 2\log(1-\hat{p}_i(q))}{\log\frac{\hat{p}_i(x_i)\hat{p}_i(y_i)}{(1-\hat{p}_i(x_i))(1-\hat{p}_i(y_i))} + \sum_{q \in A_i} 2\log(1-\hat{p}_i(q))} & \text{otherwise} \end{cases}$$

And $w_i = \frac{1}{n} = \frac{1}{8}$.

Here;

$$\hat{p}_i(x_i) = \frac{f_i(x_i)}{N}$$

With;
A_i = The set of possible values (of size n_i) that the ith feature can take
$f_i(x_i)$ and N are the same as in section 4.2.

$s_i(x_i, y_i)$ takes the minimum value when, the ith feature has N possible values it can take (that all occur exactly once). Then, for all data points x_i, $f_i(x_i) = 1$, so $\hat{p}_i(x_i) = \frac{1}{N}$ which means that $\log\frac{\hat{p}_i(x_i)\hat{p}_i(y_i)}{(1-\hat{p}_i(x_i))(1-\hat{p}_i(y_i))} = \log\frac{\frac{1}{N}\cdot\frac{1}{N}}{(1-\frac{1}{N})(1-\frac{1}{N})} = \log\frac{\frac{1}{N}\cdot\frac{1}{N}}{(\frac{N-1}{N})(\frac{N-1}{N})} = \log\frac{\left(\frac{1}{N^2}\right)}{\left(\frac{(N-1)^2}{N^2}\right)} = \log((N-1)^{-2}) = -2\log(N-1)$. So, $s_i(x_i, y_i) = \frac{\sum_{q \in A_i} 2\log(1-\hat{p}_i(q))}{-2\log(N-1) + \sum_{q \in A_i} 2\log(1-\hat{p}_i(q))} = \frac{2\sum_{q \in A_i} \log(1-\frac{1}{N})}{-2\log(N-1) + 2\sum_{q \in A_i} \log(1-\frac{1}{N})} = \frac{\sum_{q \in A_i} \log(1-\frac{1}{N})}{-\log(N-1) + \sum_{q \in A_i} \log(1-\frac{1}{N})} = \frac{N\log(1-\frac{1}{N})}{N\log(1-\frac{1}{N}) - \log(N-1)}$ (because the ith feature has N possible values it can take, so A_i has size N).
If $x_i \neq y_i$, $s_i(x_i, y_i)$ takes the maximum value when, for every data point, the ith feature only takes the value x_i or y_i with equal probability. So, when x_i and y_i are the only values for the ith feature (that occur with equal probability). In that case $f_i(x_i) = f_i(y_i) = \frac{N}{2}$, so $\hat{p}_i(x_i) = \hat{p}_i(y_i) = \frac{\left(\frac{N}{2}\right)}{N} = \frac{1}{2}$. This means that $\log\frac{\hat{p}_i(x_i)\hat{p}_i(y_i)}{(1-\hat{p}_i(x_i))(1-\hat{p}_i(y_i))} = \log\frac{\frac{1}{2}\cdot\frac{1}{2}}{(1-\frac{1}{2})(1-\frac{1}{2})} = \log\frac{\left(\frac{1}{4}\right)}{\left(\frac{1}{4}\right)} = \log(1) = 0$. Therefore, $s_i(x_i, y_i) = \frac{\sum_{q \in A_i} 2\log(1-\hat{p}_i(q))}{0 + \sum_{q \in A_i} 2\log(1-\hat{p}_i(q))} = 1$. Note that if $x_i = y_i$, $s_i(x_i, y_i) = 1$ as well.

So, it can be concluded that $s_i(x_i, y_i)$ (and therefore also $s(x, y)$ (for which $s_i(x_i, y_i)$ is summed over i and divided by 8)) has range $[\frac{N\log(1-\frac{1}{N})}{N\log(1-\frac{1}{N})-\log(N-1)}, 1] \approx [0.1139, 1]$ (for $N = 2400$ in this report).[2]

4.4 SLR construction with score 4: Anderberg

In [1], Anderberg argues that matching values that are unique indicate a lot of similarity and should therefore receive a higher weight. In the same way, values that do not match and are unique indicate that they are distinct and should receive a lower weight. So, the "Anderberg" score assigns higher scores if the matching values are unique and lower scores if the values that do not match are unique.
This score is calculated with a function for $s(x, y)$ (it cannot be written in the form of functions for $s_i(x_i, y_i)$ and w_i in equation (4.1)).

$$s(x,y) = \frac{\sum\limits_{i\in\{1\leq i\leq n : x_i = y_i\}} \left(\frac{1}{\hat{p}_i(x_i)}\right)^2 \frac{2}{n_i(n_i+1)}}{\sum\limits_{i\in\{1\leq i\leq n : x_i = y_i\}} \left(\frac{1}{\hat{p}_i(x_i)}\right)^2 \frac{2}{n_i(n_i+1)} + \sum\limits_{i\in\{1\leq i\leq n : x_i \neq y_i\}} \left(\frac{1}{2\hat{p}_i(x_i)\hat{p}_i(y_i)}\right)^2 \frac{2}{n_i(n_i+1)}}$$

Here;
n_i = The number of possible values that the ith feature can take
$\hat{p}_i(x_i)$ and $\hat{p}_i(y_i)$ are the same as in section 4.3.

$s(x, y)$ takes the minimum value when known source x and unknown source y have no matching values for any of the $n = 8$ features. Then $\{1 \leq i \leq n : x_i = y_i\}$ is empty, so $\sum\limits_{i\in\{1\leq i\leq n : x_i = y_i\}} \left(\frac{1}{\hat{p}_i(x_i)}\right)^2 \frac{2}{n_i(n_i+1)} = 0$. This means that $s(x,y) = \frac{0}{0+\sum\limits_{i\in\{1\leq i\leq n : x_i \neq y_i\}} \left(\frac{1}{2\hat{p}_i(x_i)\hat{p}_i(y_i)}\right)^2 \frac{2}{n_i(n_i+1)}} = 0.$
$s(x, y)$ takes the maximum value when known source x and unknown source y have matching values for all of the $n = 8$ features. In that case $\{1 \leq i \leq n : x_i \neq y_i\}$ is empty, so $\sum\limits_{i\in\{1\leq i\leq n : x_i \neq y_i\}} \left(\frac{1}{2\hat{p}_i(x_i)\hat{p}_i(y_i)}\right)^2 \frac{2}{n_i(n_i+1)} = 0$. This means that $s(x,y) = \frac{\sum\limits_{i\in\{1\leq i\leq n : x_i = y_i\}} \left(\frac{1}{\hat{p}_i(x_i)}\right)^2 \frac{2}{n_i(n_i+1)}}{\sum\limits_{i\in\{1\leq i\leq n : x_i = y_i\}} \left(\frac{1}{\hat{p}_i(x_i)}\right)^2 \frac{2}{n_i(n_i+1)}+0} = 1.$
So, it can be concluded that $s(x, y)$ has range $[0, 1]$.[2]

It is expected that the "Anderberg" score will outperform the other scores, because it takes the uniqueness of characteristics into account for values that match *and* for values that do not match. More on this in chapter 6.

Note that now all of the three problems of the current approach to handwriting comparison are solved; the method of this report is more time efficient (by utilizing a user interface), it has an expression for the degree of uncertainty that a hypothesis is true (by using the SLR) and the uniqueness of characteristics is taken into account (by using score 2 (Goodall3), 3 (Burnaby) and 4 (Anderberg)).

Substituting the formulas for $s_i(x_i, y_i)$ and w_i into equation (4.1) (for scores 1, 2 and 3) gives the $s(x, y)$. Doing this for all the documents of all the writers gives, for each of the four score functions, 2400 same source scores and 2,876,400 different source scores (as calculated in section 3.1). By using "R", the probability distributions of these scores can be found and thus the four SLRs can be calculated with equation (2.2). This will be done in the next chapter.

Chapter 5

Results of the SLR systems

Figure 5.1 shows the steps of the procedure of the handwriting comparison system of this report (which was also shown at the beginning of the report). After the documents of 800 writers of the CEDAR data set are obtained, the letter combinations "er" are extracted with which the subset of the data is created. Next, the characteristics of the letter combinations are entered into the user interface for each document. After this, the four scores (Overlap, Goodall3, Burnaby and Anderberg) can be calculated for the same source and different source documents and with these scores the four SLR systems can be obtained. In this chapter the same source scores, different source scores and SLRs will be calculated and discussed for each of the four scores (as described in the previous chapter). Each section (sections 5.1 to 5.4) considers one score. The SLR systems of all of the scores will be compared in section 5.5 and the evaluation of the SLR systems will be done in the next chapter.
The "R" code, that was used to calculate the scores and to create the graphs of this chapter, can be found in appendix E.

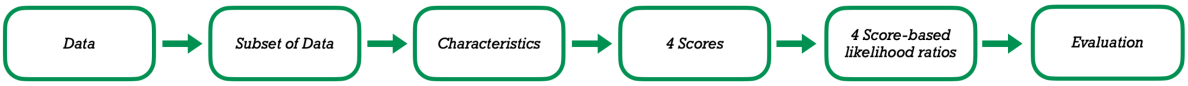


Figure 5.1: This figure shows the steps of the procedure of the handwriting comparison system of this report.

5.1 SLR results with score 1: Overlap

5.1.1 Results same source scores (score 1)

After calculating the 2400 same source scores, a histogram of the same source scores (of score 1) is created. Figure 5.2 shows that histogram.

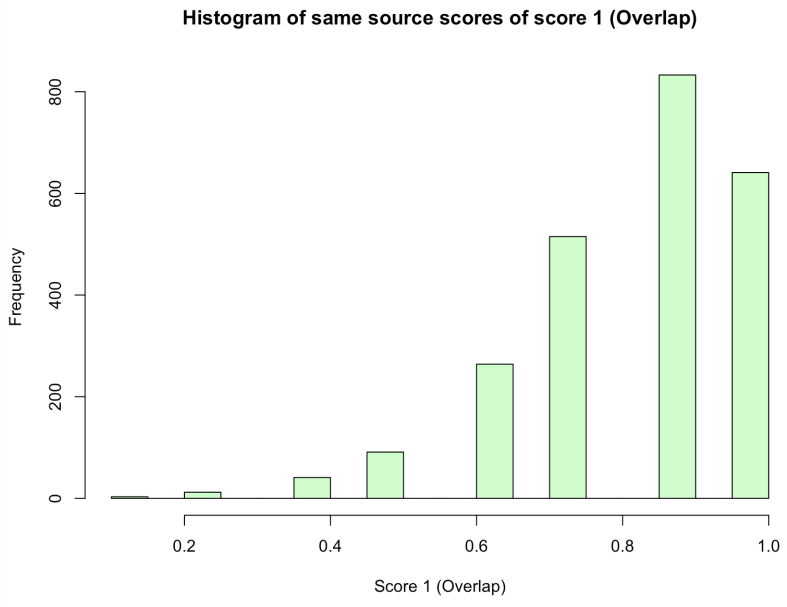


Figure 5.2: This figure shows the histogram of the same source scores of score 1 (Overlap).

Probabilities can be placed on the y-axis instead of frequencies. Figure 5.3 shows the histogram of the *probabilities* of the same source scores of score 1 (Overlap) and parametrization with the Weibull distribution. The Weibull distribution was chosen with the help of "R" and the parameters of the distribution (scale = 6.8677993 and shape = 0.8874142) were found using maximum likelihood estimation. The Q-Q plot of the theoretical quantiles (Weibull distribution) and the empirical quantiles (distribution of the same source scores) is shown in figure 5.4. The quantiles of both distributions are (approximately) on the same line.
Appendix F.1 contains more information on the decision of choosing the Weibull distribution for parametrization.

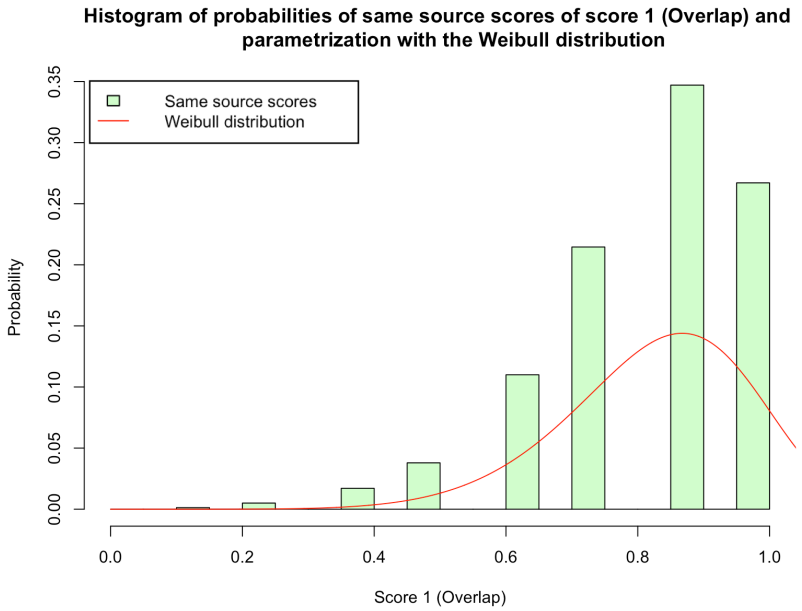


Figure 5.3: This figure shows the histogram of the probabilities of the same source scores of score 1 (Overlap) (in green) and parametrization with the Weibull distribution (in red).

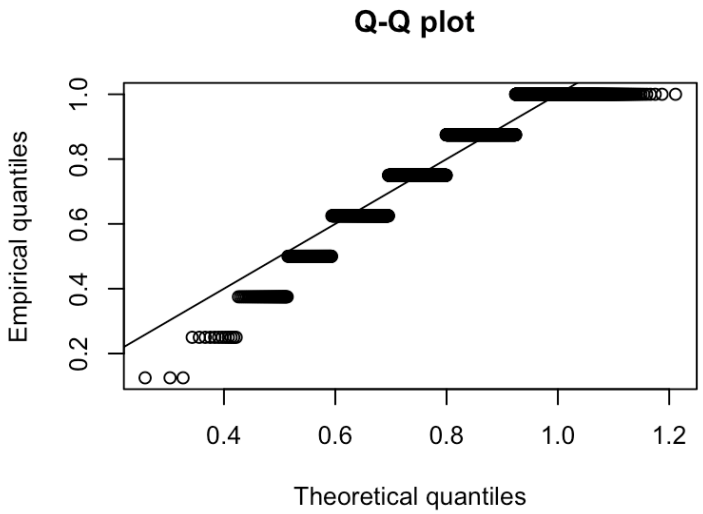


Figure 5.4: This figure shows the Q-Q plot of the theoretical quantiles (Weibull distribution) and the empirical quantiles (distribution of the same source scores).

5.1.2 Results different source scores (score 1)

After calculating the 2,876,400 different source scores, a histogram of the probabilities of the different source scores of score 1 (Overlap) and parametrization with the normal distribution was created. Figure 5.5 shows that histogram. The normal distribution was chosen with the help of "R" and the parameters of the distribution (mean = 0.5254443 and variance = 0.2164346) were found using maximum likelihood estimation. The Q-Q plot of the theoretical quantiles (normal distribution) and the empirical quantiles (distribution of the different source scores) is shown in figure 5.6. The quantiles of both distributions are (approximately) on the same line.
Appendix F.2 contains more information on the decision of choosing the normal distribution for parametrization.

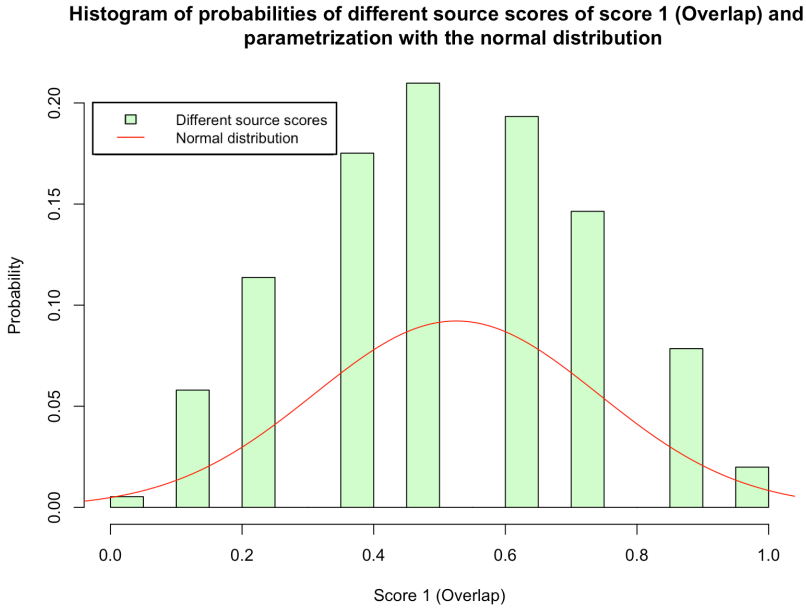


Figure 5.5: This figure shows the histogram of the probabilities of the different source scores of score 1 (Overlap) (in green) and parametrization with the normal distribution (in red).

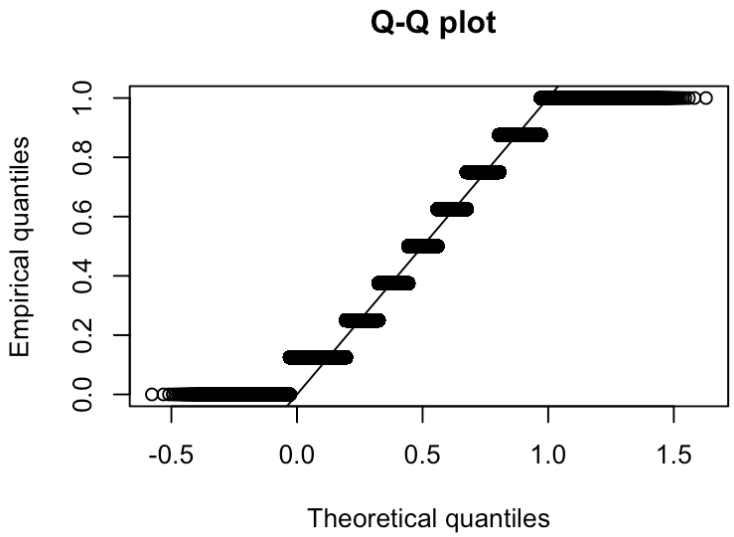


Figure 5.6: This figure shows the Q-Q plot of the theoretical quantiles (normal distribution) and the empirical quantiles (distribution of the different source scores).

Note that for the "overlap" score it is more complicated to find a parametrization with a distribution, since there are only eight possible scores (see section 4.1).

5.1.3 Results SLR (score 1)

Figure 5.7 shows the parametrization of the same source scores and different source scores of score 1. Dividing the parametrization of the same source scores by that of the different source ones, gives the SLR (by equation (2.2)). Figure 5.8 shows the score-based likelihood ratio (SLR) as a function of score 1. However, from a score of 0 to about 0.5, the SLR is close to zero. Therefore, it was decided to change the y-axis to a logarithmic scale (with a base of 10). Figure 5.9 shows the $\log_{10}$ score-based likelihood ratio ($\log_{10}$(SLR)) as a function of score 1.

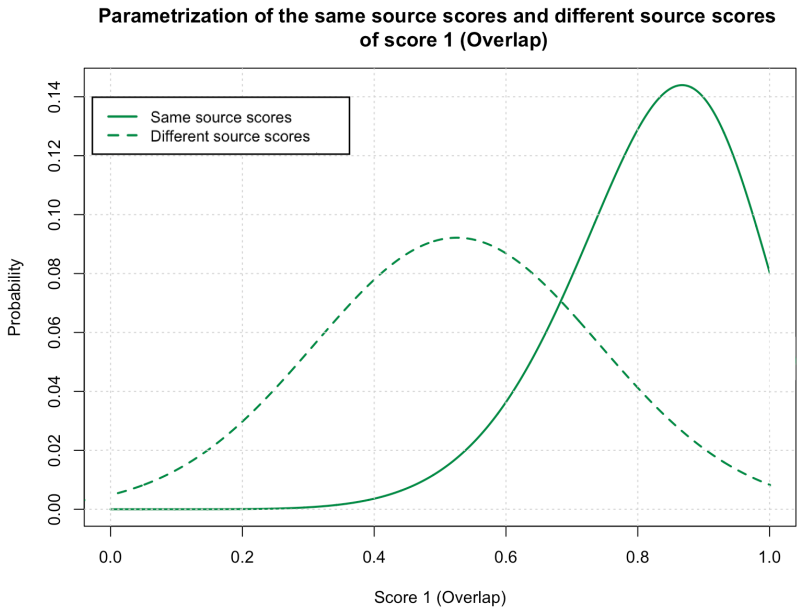


Figure 5.7: This figure shows the parametrization of the same source scores (solid green line) and different source scores (dashed green line) of score 1 (Overlap).

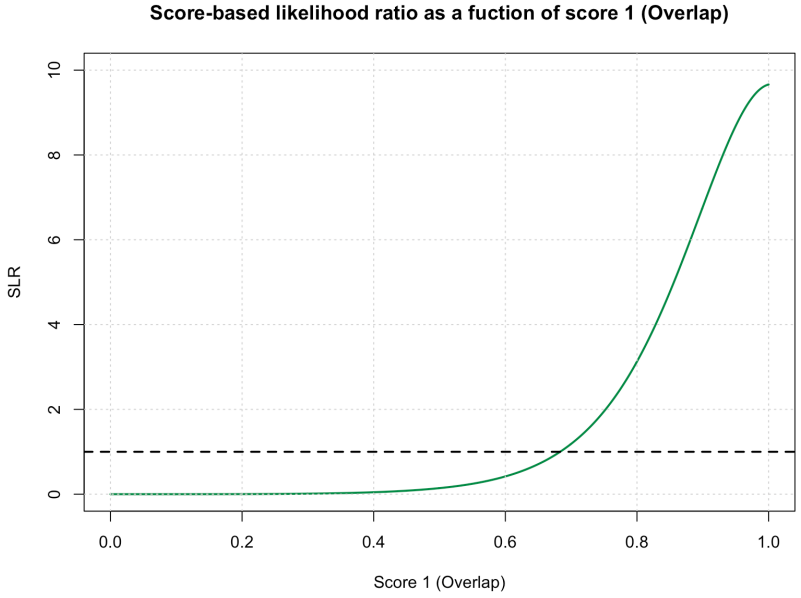


Figure 5.8: This figure shows the score-based likelihood ratio (SLR) as a function of score 1 (Overlap) (in green). The dashed black line represents an SLR value of 1.

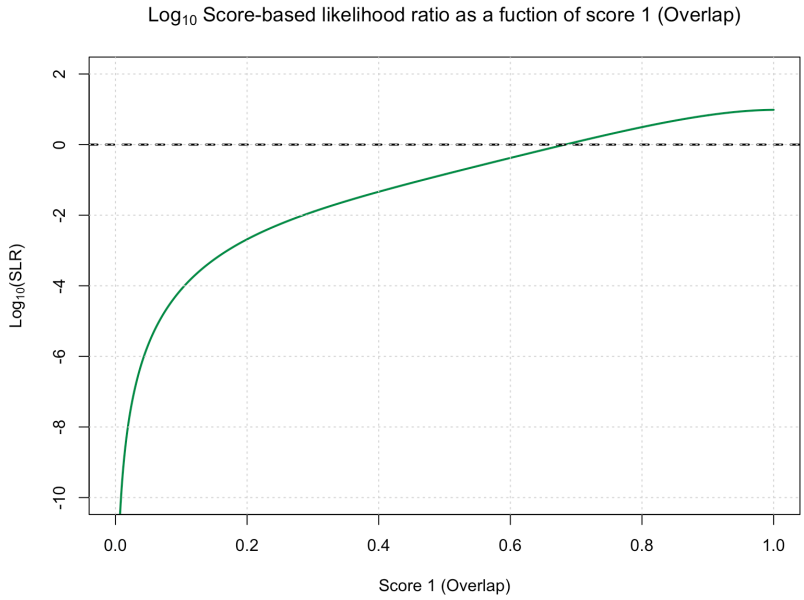


Figure 5.9: This figure shows the $\log_{10}$ score-based likelihood ratio ($\log_{10}$(SLR)) as a function of score 1 (Overlap) (in green). The dashed black line represents an SLR value of 1 (so a $\log_{10}$(SLR) value of 0).

5.2 SLR results with score 2: Goodall3

5.2.1 Results same source scores (score 2)

After calculating the same source scores, a histogram of the probabilities of the same source scores of score 2 (Goodall3) and parametrization with the normal distribution was created. Figure 5.10 shows that histogram. Again, the normal distribution was chosen with the help of "R" and the parameters of the distribution (mean = 0.5033672 and variance = 0.1380562) were found using maximum likelihood estimation. The Q-Q plot of the theoretical quantiles (normal distribution) and the empirical quantiles (distribution of the same source scores) is shown in figure 5.11. The quantiles of both distributions are (approximately) on the same line.
Appendix F.3 contains more information on the decision of choosing the normal distribution for parametrization.

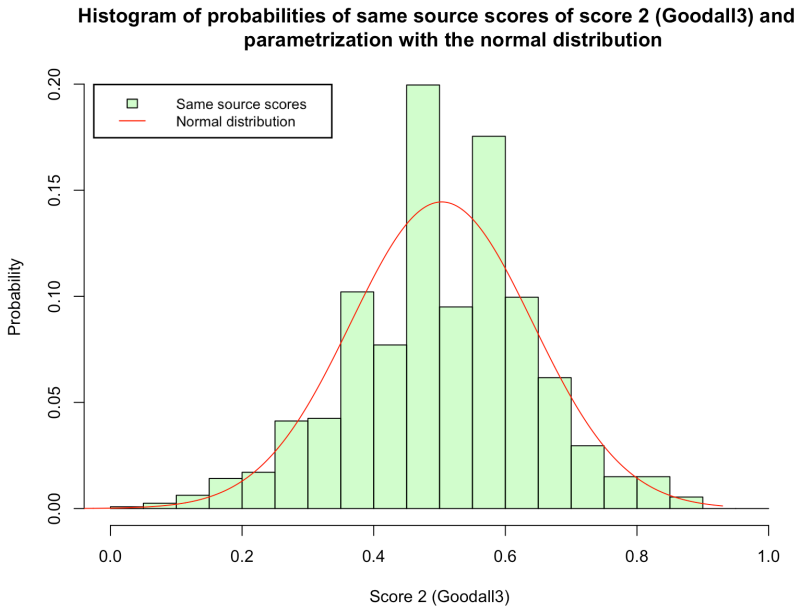


Figure 5.10: This figure shows the histogram of the probabilities of the same source scores of score 2 (Goodall3) (in green) and parametrization with the normal distribution (in red).

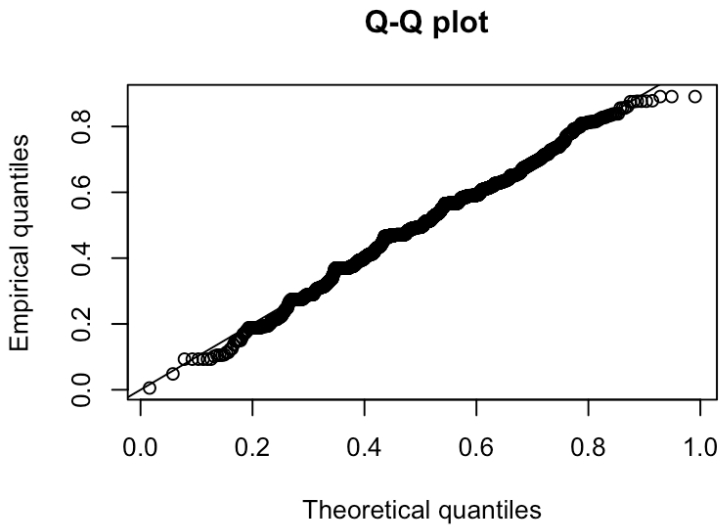


Figure 5.11: This figure shows the Q-Q plot of the theoretical quantiles (normal distribution) and the empirical quantiles (distribution of the same source scores).

Note that the score with the highest probability (of the parametrization of

the same source scores) is higher for score 1 than for score 2. This is the case, because a lot of the documents have the same characteristics. Therefore, most of the documents do not have characteristics that are unique, so the "Goodall3" score is lower for most of the documents (see section 4.2) than the "overlap" score.

5.2.2 Results different source scores (score 2)

After calculating the different source scores, a histogram of the probabilities of the different source scores of score 2 (Goodall3) and parametrization with the normal distribution was created. Figure 5.12 shows that histogram. The normal distribution was chosen with the help of "R" and the parameters of the distribution (mean = 0.2583087 and variance = 0.1412754) were found using maximum likelihood estimation. The Q-Q plot of the theoretical quantiles (normal distribution) and the empirical quantiles (distribution of the different source scores) is shown in figure 5.13. The quantiles of both distributions are (approximately) on the same line.
Appendix F.4 contains more information on the decision of choosing the normal distribution for parametrization.

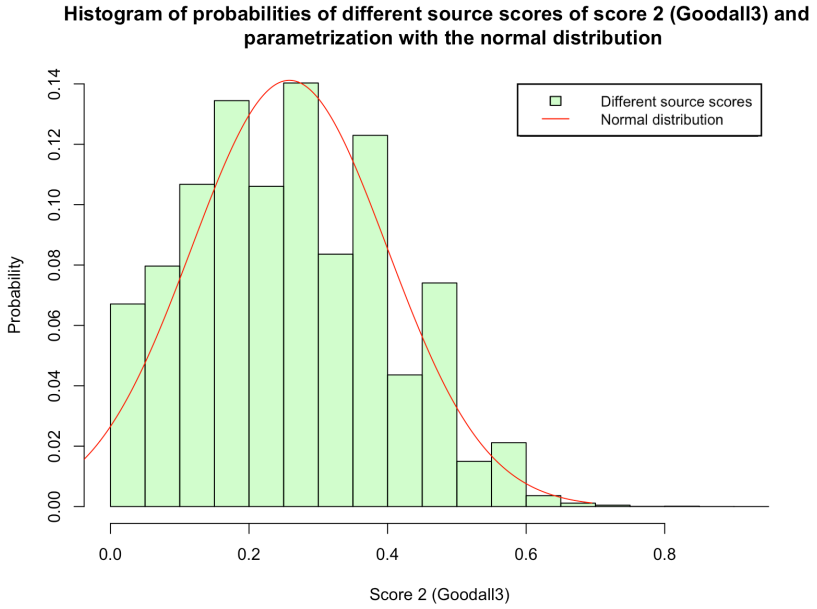


Figure 5.12: This figure shows the histogram of the probabilities of the different source scores of score 2 (Goodall3) (in green) and parametrization with the normal distribution (in red).

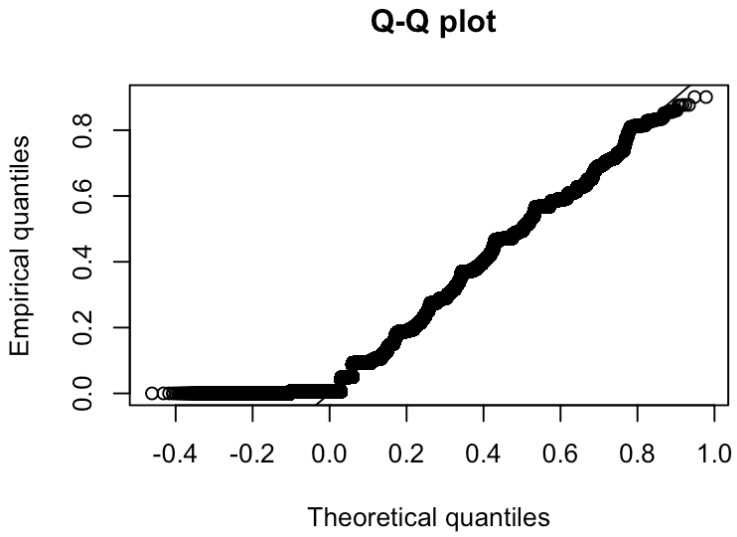


Figure 5.13: This figure shows the Q-Q plot of the theoretical quantiles (normal distribution) and the empirical quantiles (distribution of the different source scores).

For the same reason as before (that was explained in the previous subsection), the score with the highest probability (of the parametrization of the different source scores) is higher for score 1 than for score 2.

5.2.3 Results SLR (score 2)

Figure 5.14 shows the parametrization of the same source scores and different source scores of score 2. Again, dividing the parametrization of the same source scores by that of the different source ones, gives the score-based likelihood ratio (SLR) which is shown in figure 5.15 as a function of score 2. However, from a score of 0 to about 0.18, the SLR is close to zero. Furthermore, the SLR seems to increase exponentially. Therefore, it was decided to change the y-axis to a logarithmic scale (with a base of 10). Figure 5.16 shows the $\log_{10}$ score-based likelihood ratio ($\log_{10}$(SLR)) as a function of score 2.

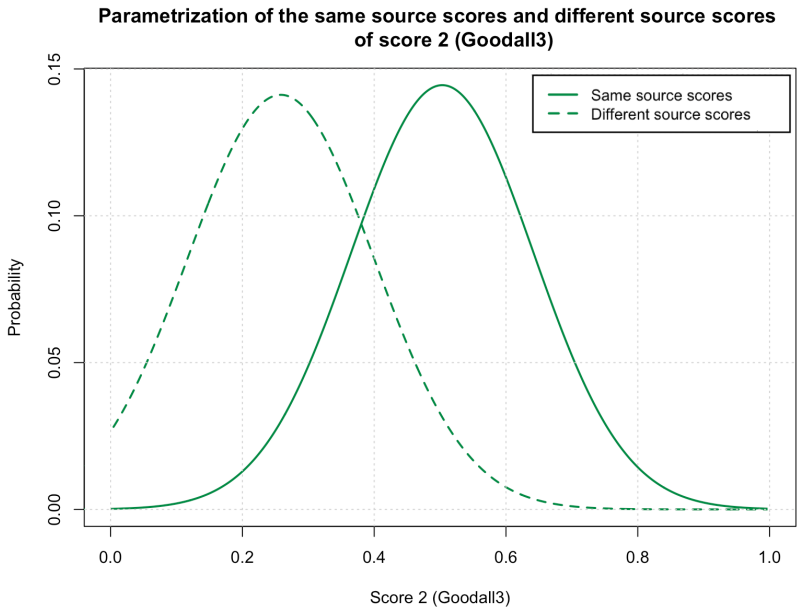


Figure 5.14: This figure shows the parametrization of the same source scores (solid green line) and different source scores (dashed green line) of score 2 (Goodall3).

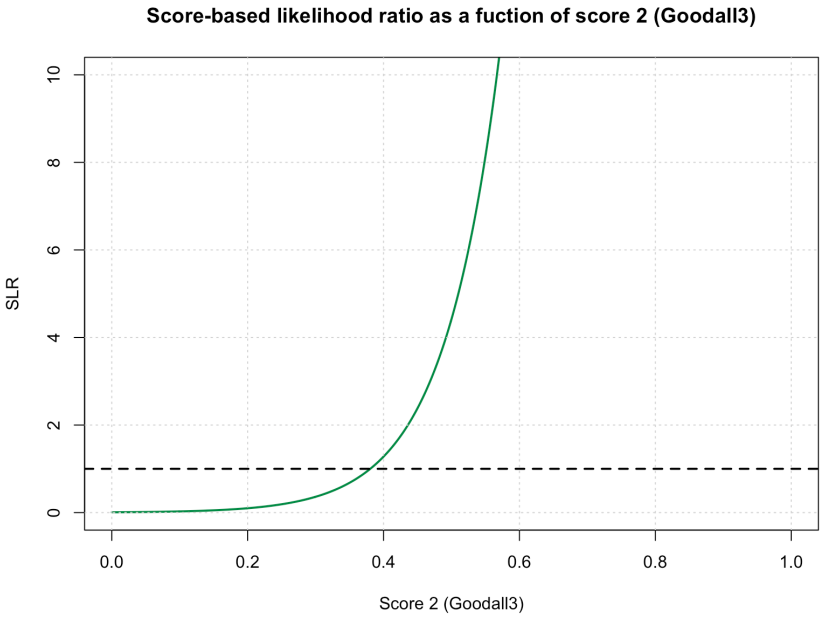


Figure 5.15: This figure shows the score-based likelihood ratio (SLR) as a function of score 2 (Goodall3) (in green). The dashed black line represents an SLR value of 1.

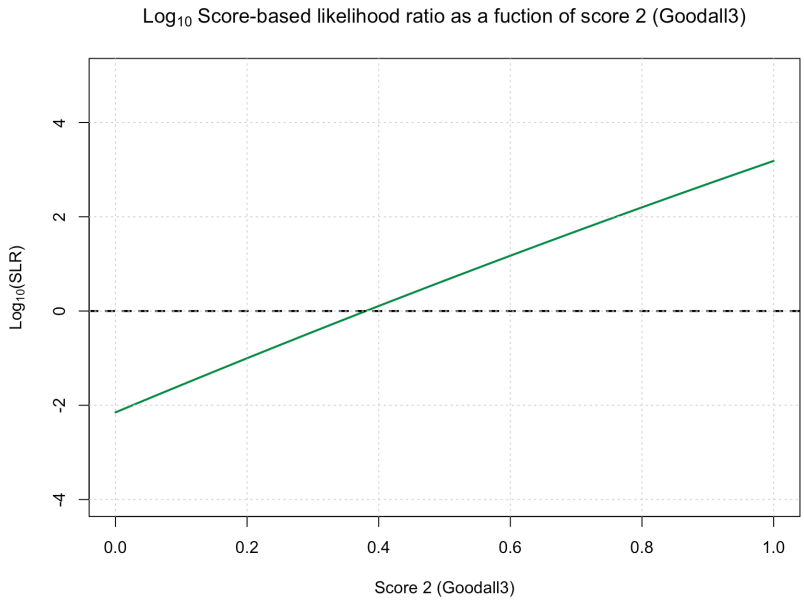


Figure 5.16: This figure shows the $\log_{10}$ score-based likelihood ratio ($\log_{10}$(SLR)) as a function of score 2 (Goodall3) (in green). The dashed black line represents an SLR value of 1 (so a $\log_{10}$(SLR) value of 0).

5.3 SLR results with score 3: Burnaby

5.3.1 Results same source scores (score 3)

After calculating the same source scores, a histogram of the probabilities of the same source scores of score 3 (Burnaby) and parametrization with the Weibull distribution was created. Figure 5.17 shows that histogram. Since no scores smaller than 0.6 occur, the histogram was magnified. The result can be seen in figure 5.18.
Again, the Weibull distribution was chosen with the help of "R" and the parameters of the distribution (scale = 24.8003921 and shape = 0.9682415) were found using maximum likelihood estimation. The Q-Q plot of the theoretical quantiles (Weibull distribution) and the empirical quantiles (distribution of the same source scores) is shown in figure 5.19. The quantiles of both distributions are (approximately) on the same line.
Appendix F.5 contains more information on the decision of choosing the Weibull distribution for parametrization.

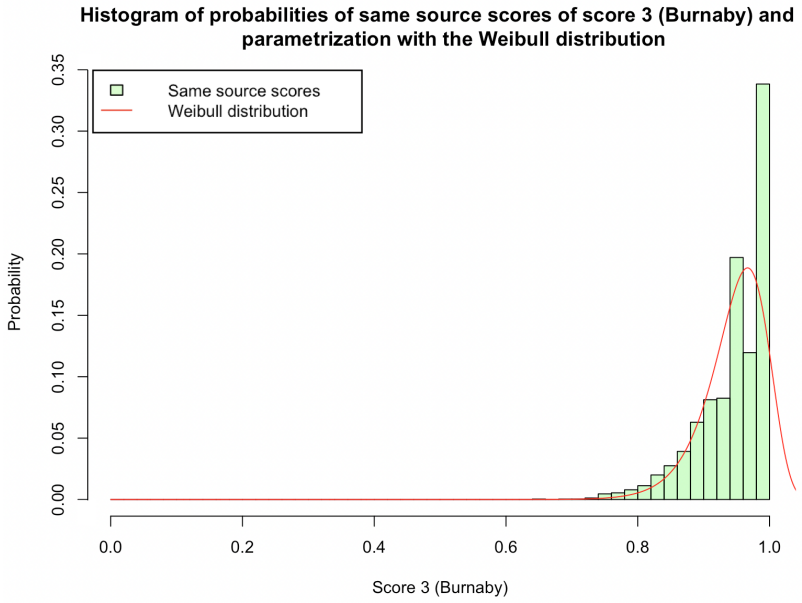


Figure 5.17: This figure shows the histogram of the probabilities of the same source scores of score 3 (Burnaby) (in green) and parametrization with the Weibull distribution (in red).

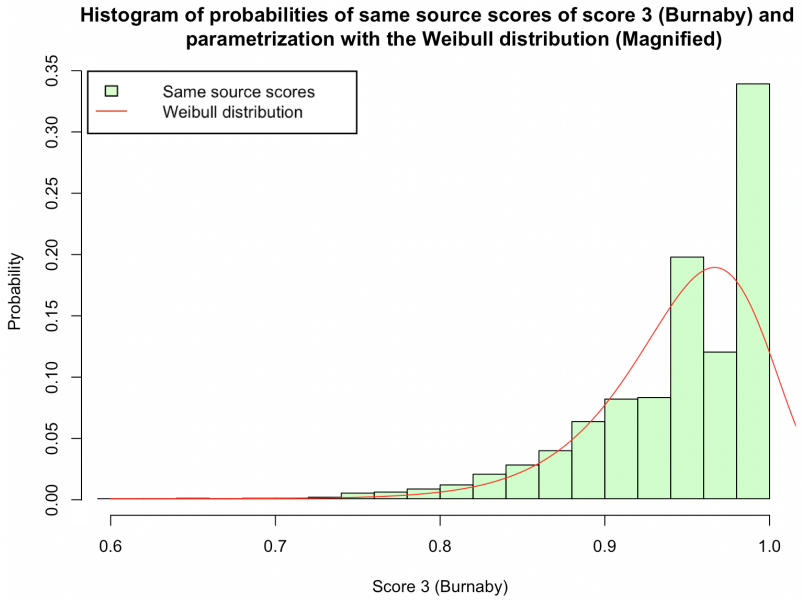


Figure 5.18: This figure shows the **magnified** histogram of the probabilities of the same source scores of score 3 (Burnaby) (in green) and parametrization with the Weibull distribution (in red).

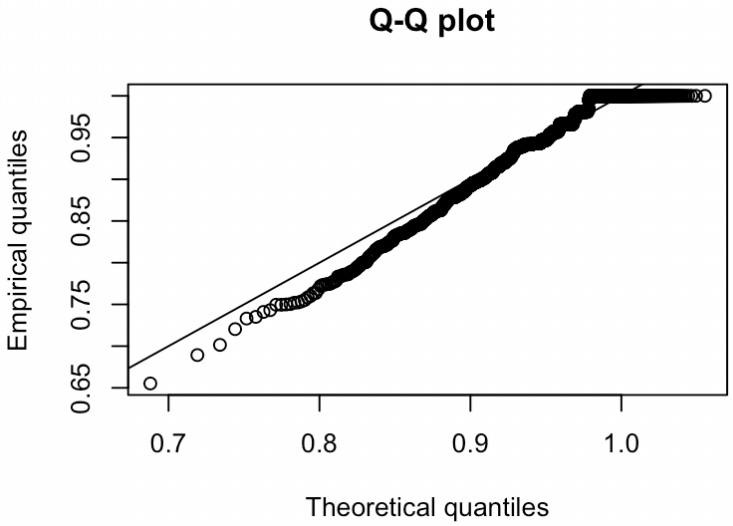


Figure 5.19: This figure shows the Q-Q plot of the theoretical quantiles (Weibull distribution) and the empirical quantiles (distribution of the same source scores).

Note that the score with the highest probability (of the parametrization of the same source scores) is higher for score 3 than for scores 1 and 2. This is the case, because, with the "Burnaby" score, matching characteristics receive a score of 1 and characteristics that do not match get a higher score if the characteristics are common (so not unique) (see section 4.3). A lot of the documents have the same characteristics. Therefore, most of the documents do not have characteristics that are unique, so the "Burnaby" score is higher for most of the documents than the "overlap" and "Goodall3" scores.

5.3.2 Results different source scores (score 3)

After calculating the different source scores, a histogram (figure 5.20) of the probabilities of the different source scores of score 3 (Burnaby) and parametrization with the normal distribution was created. Since no scores smaller than 0.5 occur, the histogram was magnified. The result can be seen in figure 5.21. The normal distribution was chosen with the help of "R" and the parameters of the distribution (mean = 0.85762356 and variance = 0.07077304) were found using maximum likelihood estimation. The Q-Q plot of the theoretical quantiles (normal distribution) and the empirical quantiles (distribution of different source scores) is shown in figure 5.22. The quantiles of both distributions are (approximately) on the same line.
Appendix F.6 contains more information on the decision of choosing the normal distribution for parametrization.

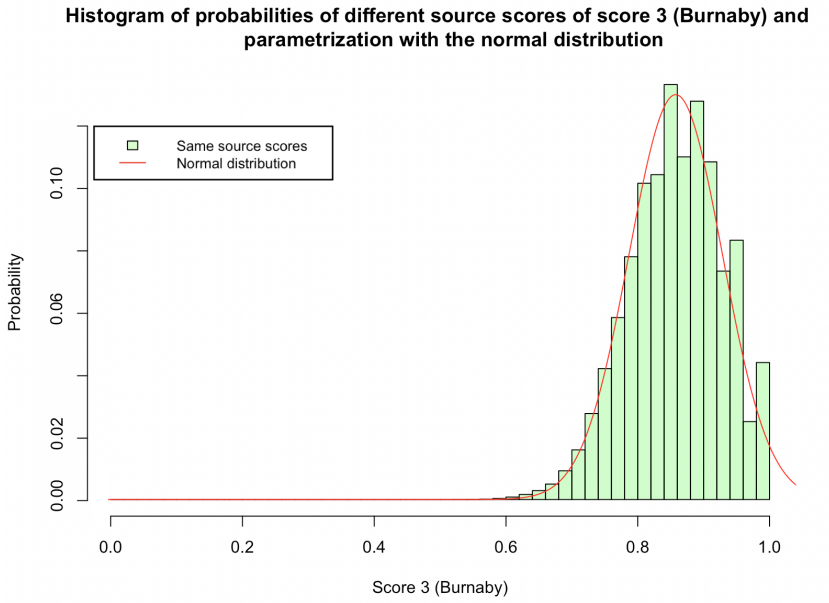


Figure 5.20: This figure shows the histogram of the probabilities of the different source scores of score 3 (Burnaby) (in green) and parametrization with the normal distribution (in red).

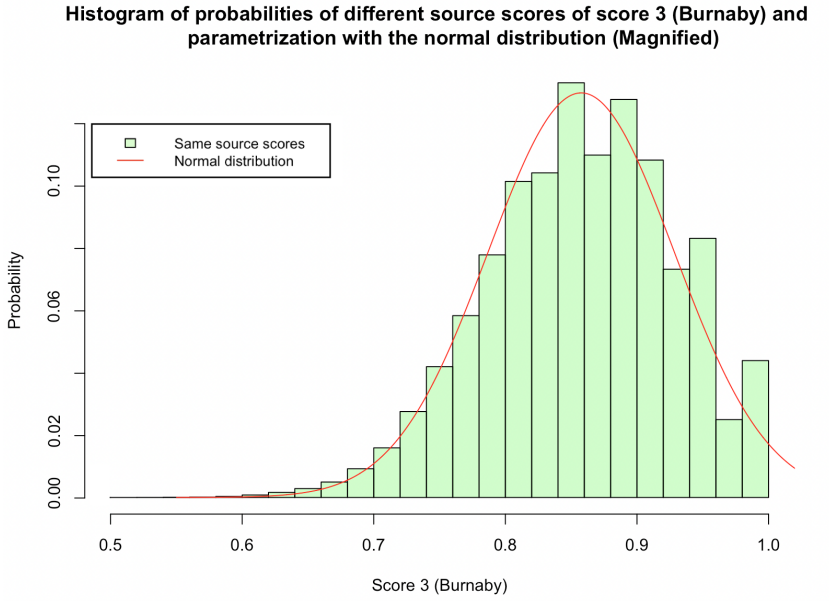


Figure 5.21: This figure shows the **magnified** histogram of the probabilities of the different source scores of score 3 (Burnaby) (in green) and parametrization with the normal distribution (in red).

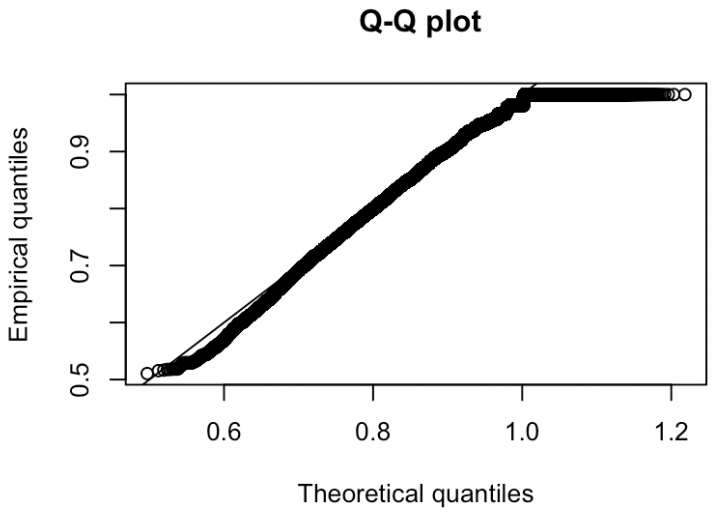


Figure 5.22: This figure shows the Q-Q plot of the theoretical quantiles (normal distribution) and the empirical quantiles (distribution of the different source scores).

5.3.3 Results SLR (score 3)

Figure 5.23 shows the parametrization of the same source scores and different source scores of score 3. Again, dividing the parametrization of the same source scores by that of the different source ones, gives the score-based likelihood ratio (SLR) which is shown in figure 5.24 as a function of score 3. However, from a score of 0 to about 0.05 and of 0.55 to 0.8, the SLR is close to zero. Furthermore, the SLR seems to increase exponentially. Therefore, it was decided to change the y-axis to a logarithmic scale (with a base of 10). Figure 5.25 shows the $\log_{10}$ score-based likelihood ratio ($\log_{10}$(SLR)) as a function of score 3.

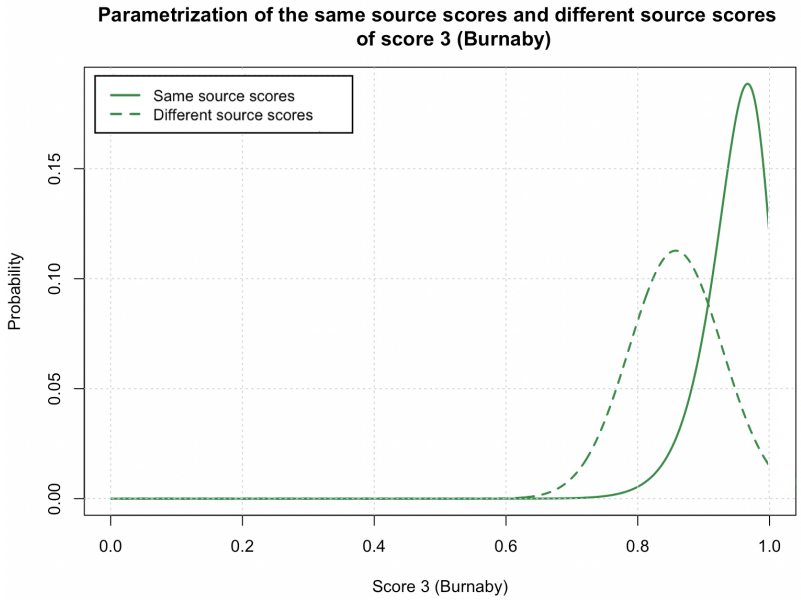


Figure 5.23: This figure shows the parametrization of the same source scores (solid green line) and different source scores (dashed green line) of score 3 (Burnaby).

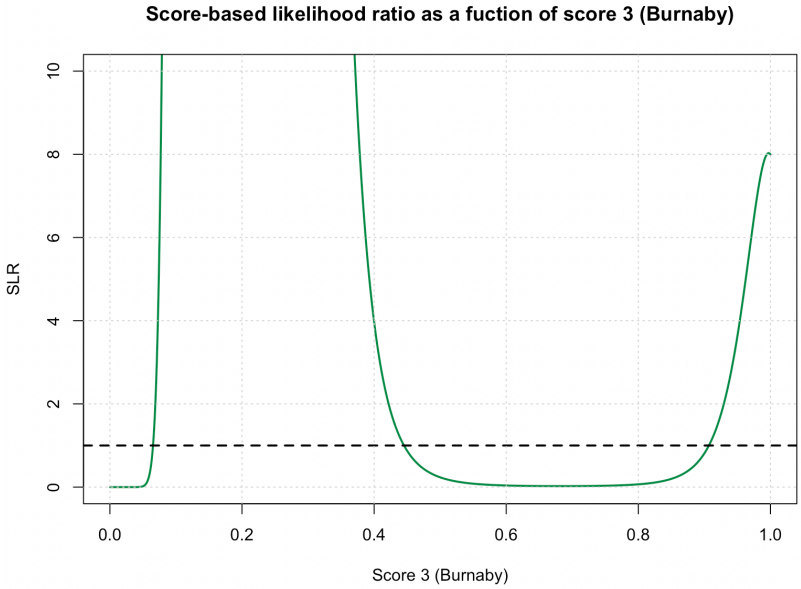


Figure 5.24: This figure shows the score-based likelihood ratio (SLR) as a function of score 3 (Burnaby) (in green). The dashed black line represents an SLR value of 1.

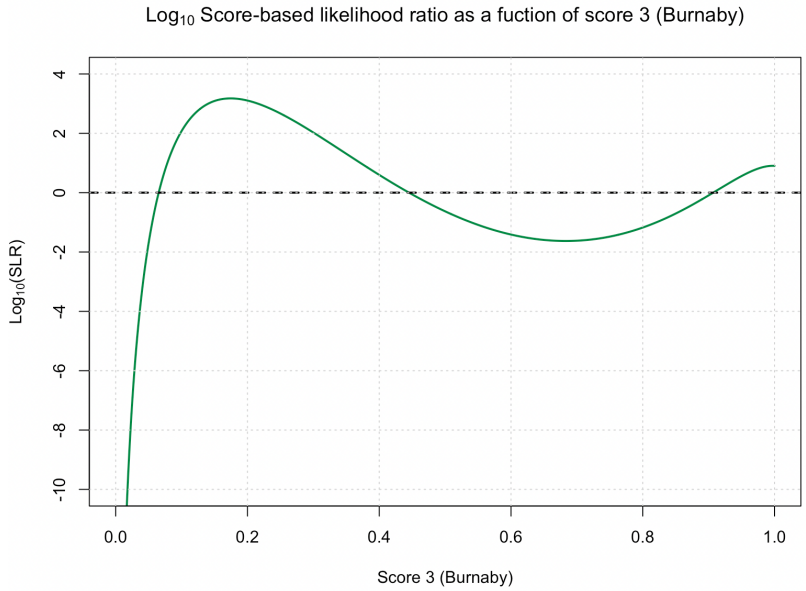


Figure 5.25: This figure shows the $\log_{10}$ score-based likelihood ratio ($\log_{10}$(SLR)) as a function of score 3 (Burnaby) (in green). The dashed black line represents an SLR value of 1 (so a $\log_{10}$(SLR) value of 0).

In figures 5.24 and 5.25, it can be seen that the SLR is greater than 1 (so the $\log_{10}$(SLR) is greater than 0), when the score is greater than 0.9 or when the score is between the 0.05 and 0.45. This means that, for those values, the parametrization of the same source scores is greater than that of the different source scores. For all other values the parametrization of the same source scores is smaller than that of the different source scores. This switch, in which parametrization is greater, can be seen (for scores between 0 and 0.5) in figure 5.26; it shows the magnified parametrization of figure 5.23.

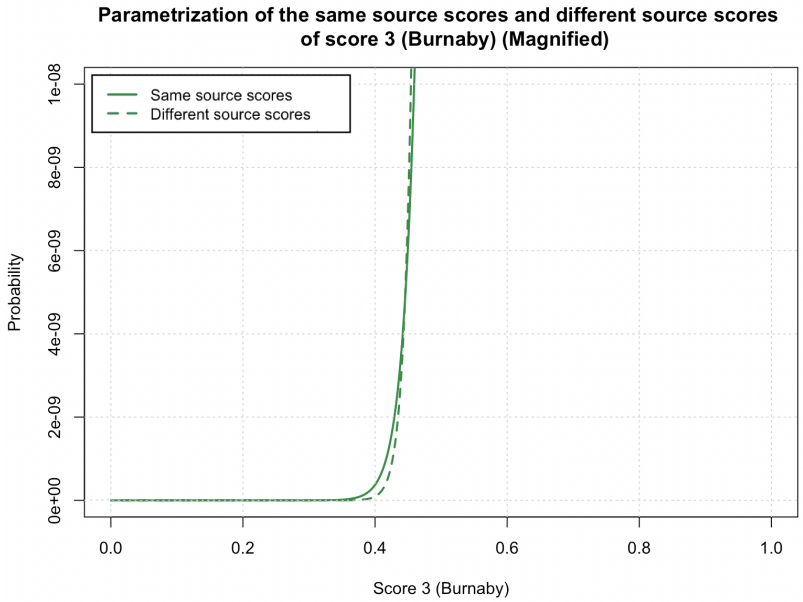


Figure 5.26: This figure shows the **magnified** parametrization of the same source scores (solid green line) and different source scores (dashed green line) of score 3 (Burnaby).

Figure 5.24 shows that a score of 0.2 has a higher SLR than a score of 0.6. So, a score of 0.2 indicates that the sources are more similar than a score of 0.6. This is impossible, because the score itself is already a measure for the similarity. Therefore, the SLR function (and thus also the $\log_{10}$(SLR) function) should be an increasing function of the score. Most likely this is not the case for score 3, because the Weibull (of same source scores) and normal distribution (of different source scores) behave too differently in the tails. Therefore, the decision was made to fit the second best distribution to the same source scores; the gamma distribution (see appendix F.5). Figure 5.27 shows the histogram of the probabilities of the same source scores of score 3 (Burnaby) and parametrization with the gamma distribution. The parameters of the distribution (shape = 294.0646 and scale = 310.9852) were found using maximum likelihood estimation. Furthermore, since no scores smaller than 0.6 occur, the histogram was magnified. The result can be seen in figure 5.28.

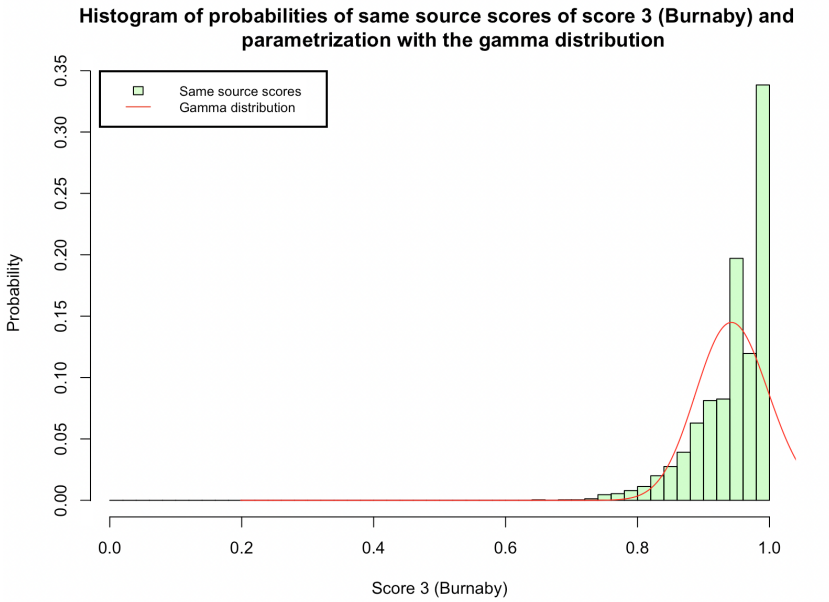


Figure 5.27: This figure shows the histogram of the probabilities of the same source scores of score 3 (Burnaby) (in green) and parametrization with the gamma distribution (in red).

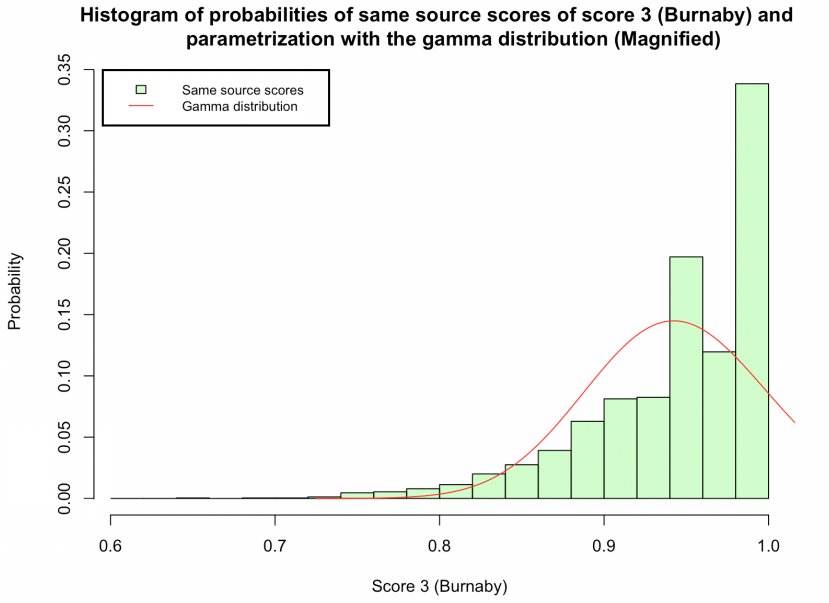


Figure 5.28: This figure shows the **magnified** histogram of the probabilities of the same source scores of score 3 (Burnaby) (in green) and parametrization with the gamma distribution (in red).

Using the gamma distribution for the parametrization of the same source scores, figure 5.29 shows the parametrization of the same source scores and different source scores of score 3. Again, dividing the parametrization of the same source scores by that of the different source ones, gives the score-based likelihood ratio (SLR) which is shown in figure 5.30 as a function of score 3. However, from a score of 0 to about 0.8, the SLR is close to zero. Therefore, it was decided to change the y-axis to a logarithmic scale (with a base of 10). Figure 5.31 shows the $\log_{10}$ score-based likelihood ratio ($\log_{10}$(SLR)) as a function of score 3. In that figure, it can be seen that the $\log_{10}$(SLR) is relatively small for scores between 0 and 0.8. Most likely this is because none of the documents had a score lower than 0.6. Therefore, it is more complicated to find a distribution that fits these scores. This results in very small SLRs.

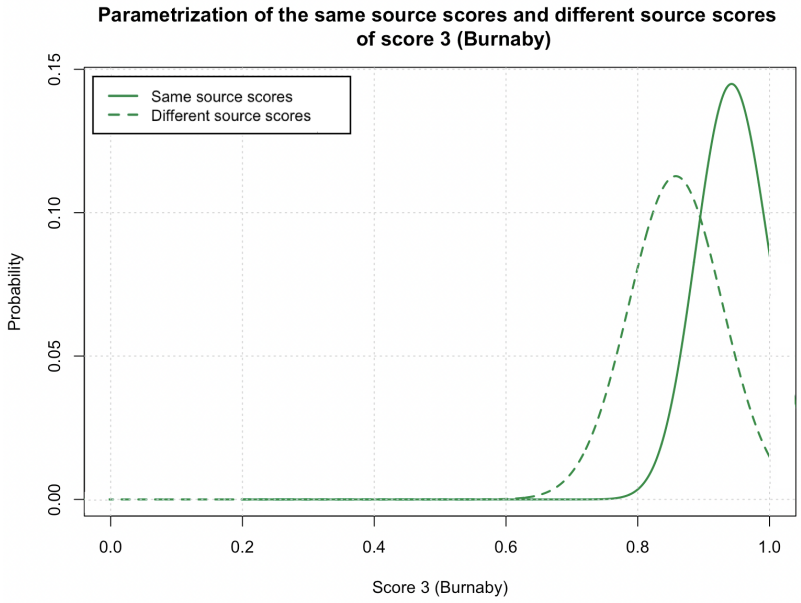


Figure 5.29: This figure shows the parametrization of the same source scores (solid green line) and different source scores (dashed green line) of score 3 (Burnaby).

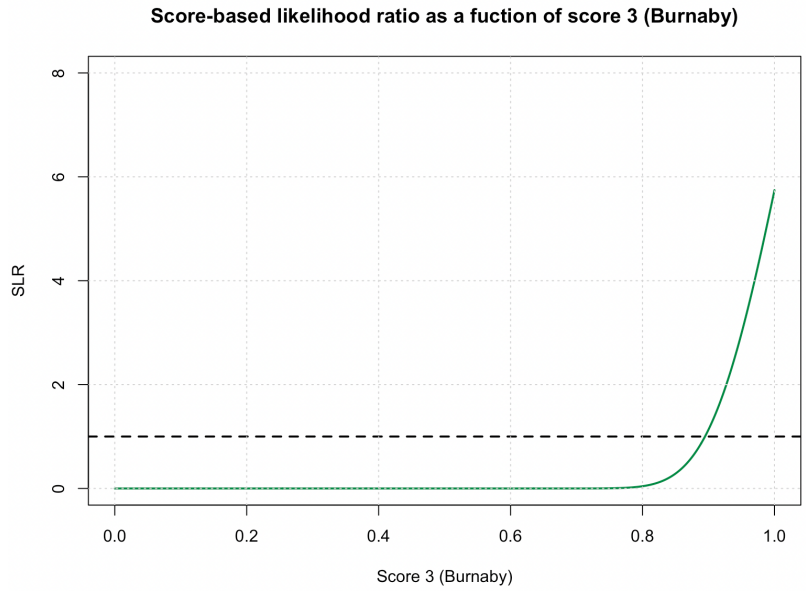


Figure 5.30: This figure shows the score-based likelihood ratio (SLR) as a function of score 3 (Burnaby) (in green). The dashed black line represents an SLR value of 1.

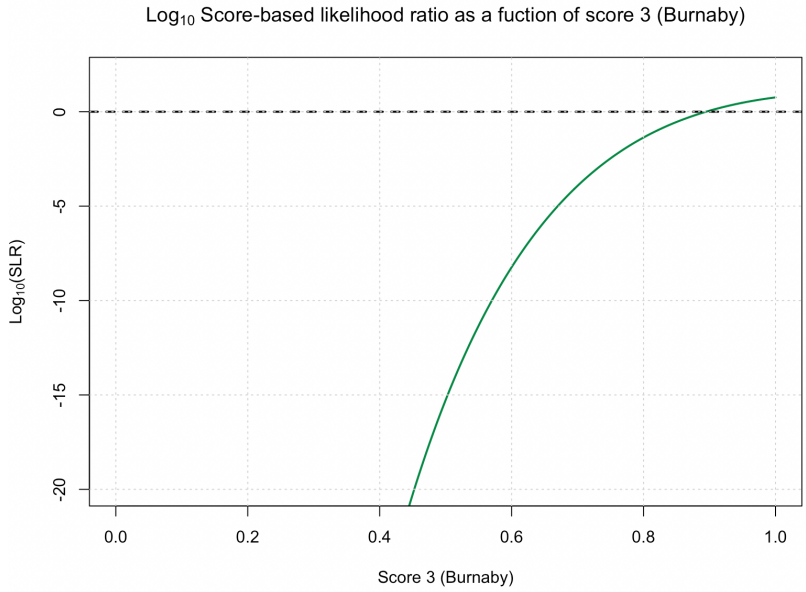


Figure 5.31: This figure shows the $\log_{10}$ score-based likelihood ratio ($\log_{10}$(SLR)) as a function of score 3 (Burnaby) (in green). The dashed black line represents an SLR value of 1 (so a $\log_{10}$(SLR) value of 0).

5.4 SLR results with score 4: Anderberg

5.4.1 Results same source scores (score 4)

After calculating the same source scores, a histogram of the probabilities of the same source scores of score 4 (Anderberg) and parametrization with the beta distribution was created. Figure 5.32 shows that histogram. Again, the beta distribution was chosen with the help of "R" and the parameters of the distribution (shape alpha = 1.7348483 and shape beta = 0.3720472) were found using maximum likelihood estimation. The Q-Q plot of the theoretical quantiles (beta distribution) and the empirical quantiles (distribution of the same source scores) is shown in figure 5.33. The quantiles of both distributions are (approximately) on the same line.
Appendix F.7 contains more information on the decision of choosing the beta distribution for parametrization.

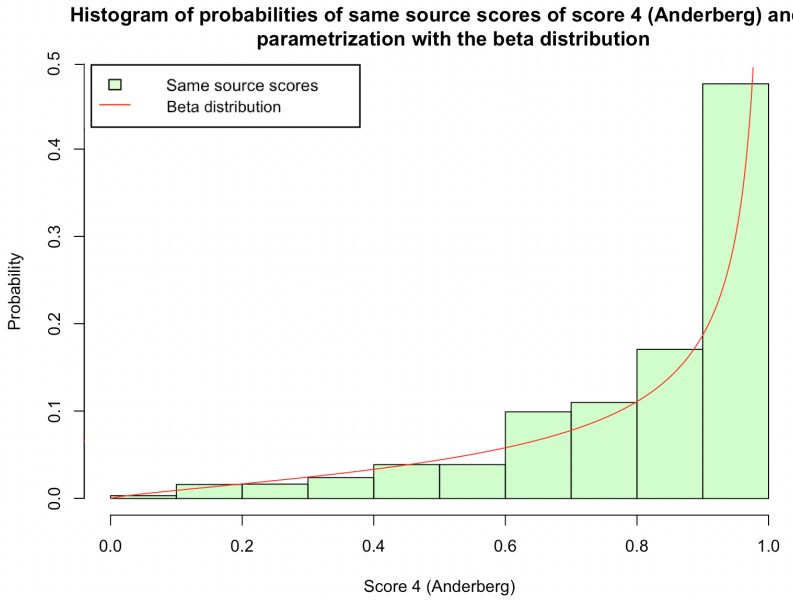


Figure 5.32: This figure shows the histogram of the probabilities of the same source scores of score 4 (Anderberg) (in green) and parametrization with the beta distribution (in red).

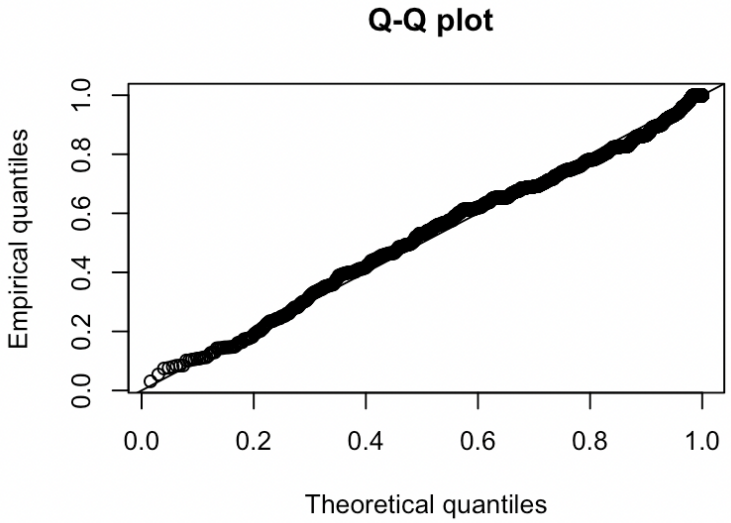


Figure 5.33: This figure shows the Q-Q plot of the theoretical quantiles (beta distribution) and the empirical quantiles (distribution of the same source scores).

Note that a score of 1 occurs the most for score 4 (with a probability of 0.5) in comparison to scores 1, 2 and 3. This is the case, because, if two documents are from the same source, most of the time they only have characteristics that match. In section 4.4 it was explained that, in that case, the "Anderberg" score is equal to 1.

5.4.2 Results different source scores (score 4)

After calculating the different source scores, a histogram of the probabilities of the different source scores of score 4 (Anderberg) and parametrization with the beta distribution was created. Figure 5.34 shows that histogram. Again, the beta distribution was chosen with the help of "R" and the parameters of the distribution (shape alpha = 0.9371025 and shape beta = 1.1337795) were found using maximum likelihood estimation. The Q-Q plot of the theoretical quantiles (beta distribution) and the empirical quantiles (distribution of the different source scores) is shown in figure 5.35. The quantiles of both distributions are (approximately) on the same line.
Appendix F.8 contains more information on the decision of choosing the beta distribution for parametrization.

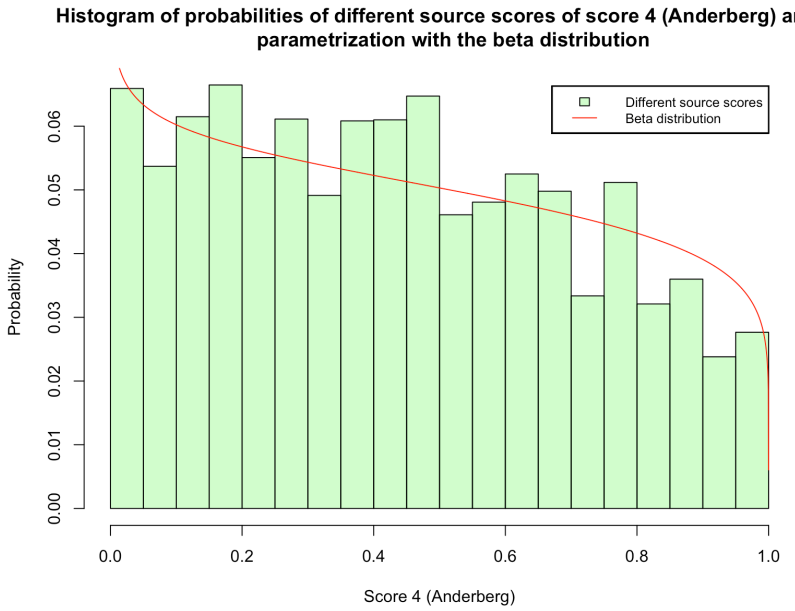


Figure 5.34: This figure shows the histogram of the probabilities of the different source scores of score 4 (Anderberg) (in green) and parametrization with the beta distribution (in red).

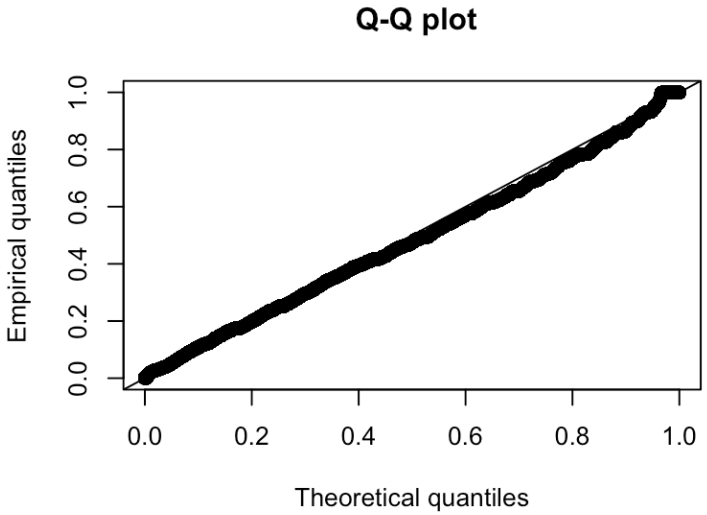


Figure 5.35: This figure shows the Q-Q plot of the theoretical quantiles (beta distribution) and the empirical quantiles (distribution of the different source scores).

5.4.3 Results SLR (score 4)

Figure 5.36 shows the parametrization of the same source scores and different source scores of score 4. Again, dividing the parametrization of the same source scores by that of the different source ones, gives the score-based likelihood ratio (SLR) which is shown in figure 5.37 as a function of score 4. However, from a score of 0 to about 0.15, the SLR is close to zero. Furthermore, the SLR seems to increase exponentially. Therefore, it was decided to change the y-axis to a logarithmic scale (with a base of 10). Figure 5.38 shows the $\log_{10}$ score-based likelihood ratio ($\log_{10}$(SLR)) as a function of score 4.

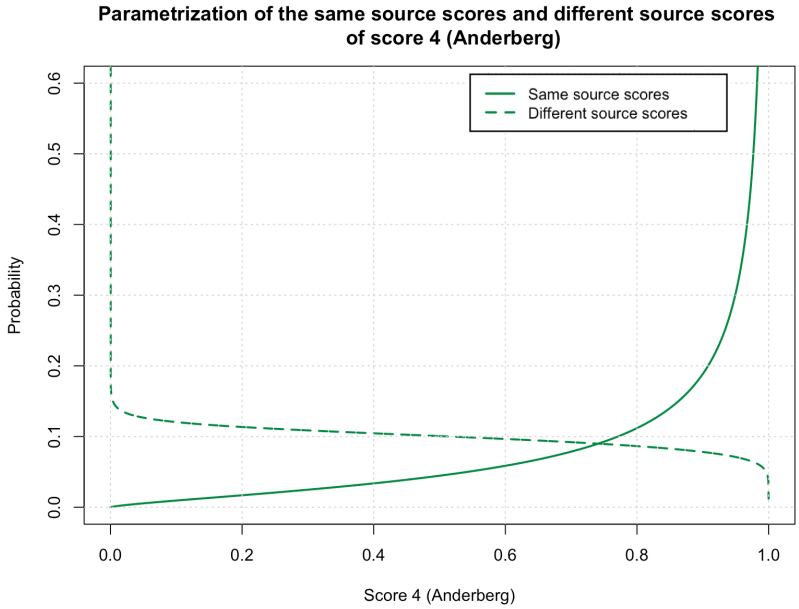


Figure 5.36: This figure shows the parametrization of the same source scores (solid green line) and different source scores (dashed green line) of score 4 (Anderberg).

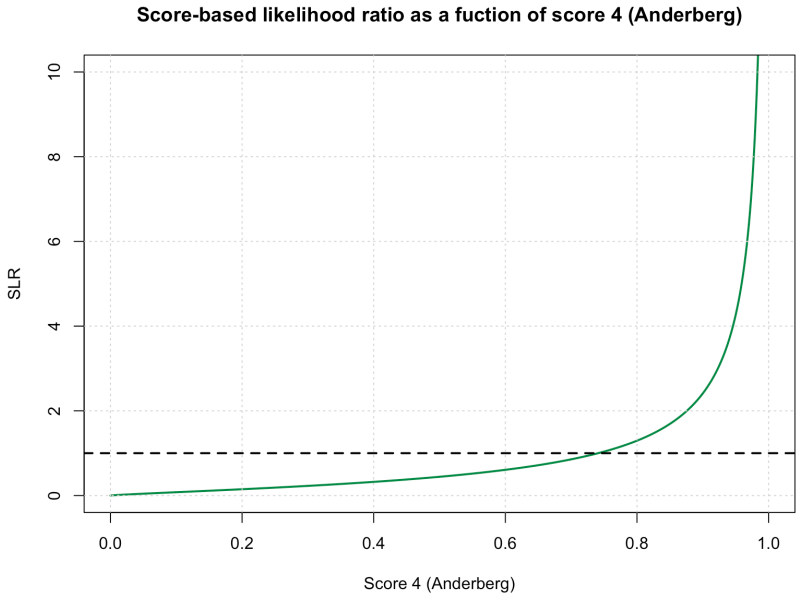


Figure 5.37: This figure shows the score-based likelihood ratio (SLR) as a function of score 4 (Anderberg) (in green). The dashed black line represents an SLR value of 1.

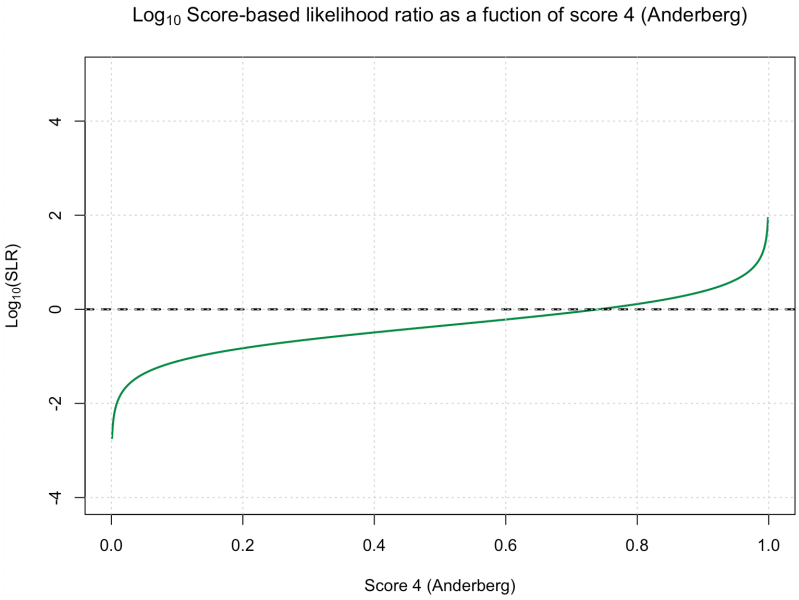


Figure 5.38: This figure shows the $\log_{10}$ score-based likelihood ratio ($\log_{10}$(SLR)) as a function of score 4 (Anderberg) (in green). The dashed black line represents an SLR value of 1 (so a $\log_{10}$(SLR) value of 0).

5.5 Results SLRs (all scores)

Figures 5.39 and 5.40 show the score-based likelihood ratios (SLRs) and the $\log_{10}$ score-based likelihood ratios ($\log_{10}$(SLR)) (respectively) as functions of all scores.
For example, if two documents have a "Burnaby" score of 0.8, the SLR (so $\log_{10}$(SLR)) is lower than if two documents have an "Anderberg", "Overlap" or "Goodall3" score (from smallest to greatest corresponding SLR) of 0.8.

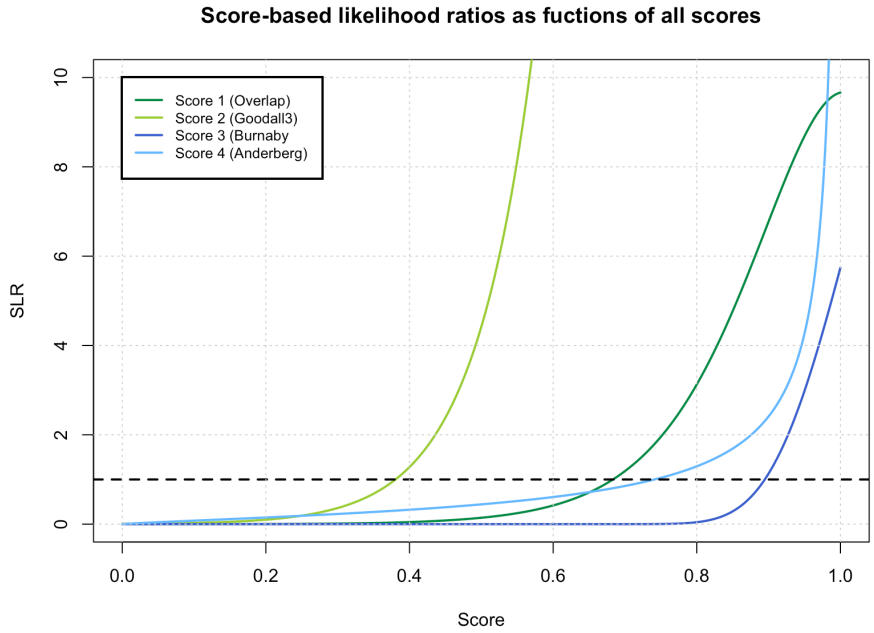


Figure 5.39: This figure shows the score-based likelihood ratios (SLRs) as functions of all scores. The dashed black line represents an SLR value of 1.

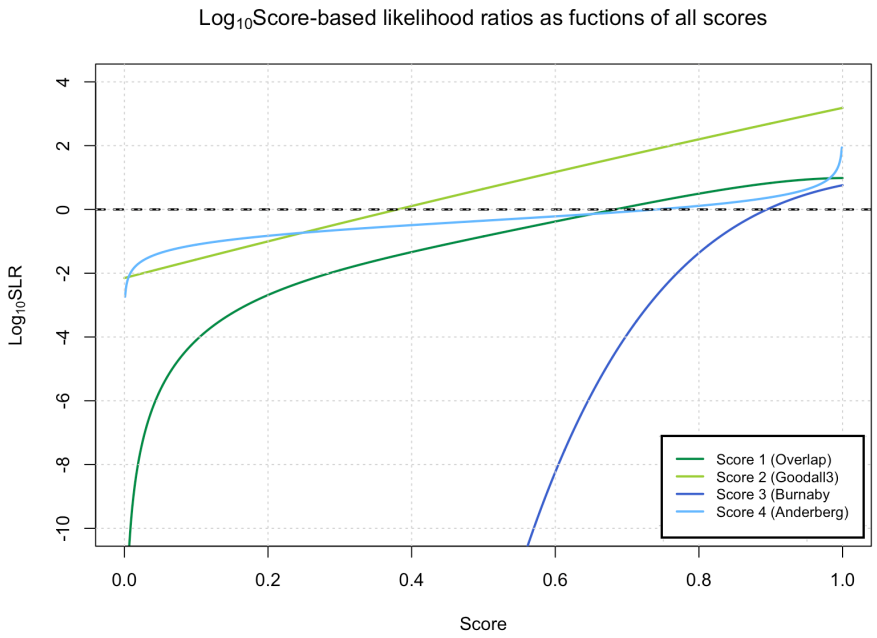


Figure 5.40: This figure shows the $\log_{10}$ score-based likelihood ratios ($\log_{10}$(SLR)) as functions of all scores. The dashed black line represents an SLR value of 1 (so a $\log_{10}$(SLR) value of 0).

As mentioned in chapter 2, if S$LR > 1$, there is a higher likelihood that the two documents have the same writer and, if $SLR < 1$, there is a higher likelihood that the two documents have a different writer. This concept will be used in the next chapter and that is why, in the SLR graphs of this chapter, a line was plotted that represents an SLR value of 1 (so a $\log_{10}$(SLR) value of 0). Figures 5.39 and 5.40 also show that score 3 (Burnaby) requires the highest similarity score (compared to the other SLR systems) in order to have $SLR > 1$ (so $\log_{10}(SLR) > 0$). After this score 4 (Anderberg) requires the highest similarity score, then score 1 (Overlap) and score 2 (Goodall3) requires the lowest similarity score.

In the next chapter the four SLR systems, that were obtained in this chapter, will be evaluated based on their quality of performance. The SLR system that performs the best according to this evaluation, will considered be the "best" SLR system.

Chapter 6

Evaluation of the quality of performance of SLR systems

Since the LR (which is approximated by the SLR) is unknown, the SLR systems cannot be compared to the LR system. Therefore, examining the accuracy and performance of SLR systems is complicated. [14]. However, their behavior can be investigated; in this chapter, the SLR systems of the previous chapter are evaluated in the same way as in [14]. This means that a leave-one-out method (cross-validation) is performed (section 6.1) and the 95% SLR bootstrap confidence intervals are calculated (section 6.2). Lastly, the misleading evidence is quantified (section 6.3). The SLR system that performs the best, based on these three performance characteristics, will be considered the "best" SLR system. For this, every SLR system is ranked (from 1 to 4) for each of the performance characteristics, then the sum of these rankings over all of the performance characteristics is taken. This sum can range from $1 \cdot 3 = 3$ to $4 \cdot 3 = 12$. The SLR system with the lowest sum of rankings, is considered the "best" SLR system.

6.1 Using the leave-one-out method (cross-validation)

In order to examine the distributions of the SLR systems (for the same source and different source comparisons), the leave-one-out method (or cross validation) is used. This means that for each of the 2400 documents, the parametrization (as described in the previous chapter) was performed using all data except the data connected with that specific document, that is, using all data except the same source and different source scores of that document. This results in 2400 same source SLRs and 2,876,400 different source SLRs (see section 3.1). Note that, since each parametrization is done without taking the scores of one specific document into account, the calculation of each of the SLR systems was performed based on a dataset independent of the scores of this document. [14]

Figures 6.1 and 6.2 show the boxplots of the $\log_{10}$(SLR) of same source and different source comparisons (respectively) for the four different score systems using the leave-one-out method (cross-validation). The five horizontal lines in each of the boxplots represent the maximum, third quartile, median, first

quartile and minimum (from top to bottom).

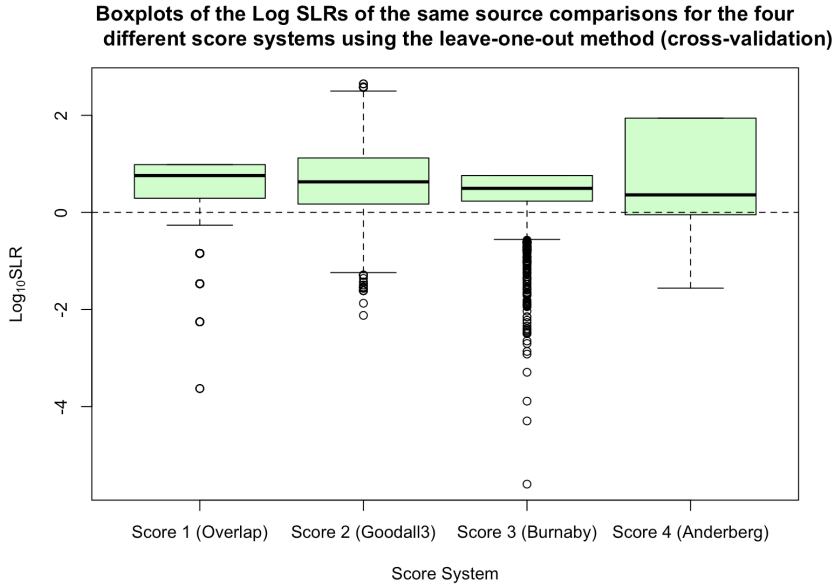


Figure 6.1: This figure shows the boxplots of the $\log_{10}$(SLR) of **same** source comparisons for the scores using the leave-one-out method (cross-validation). The horizontal dashed black line represents a $\log_{10}$(SLR) value of 0.

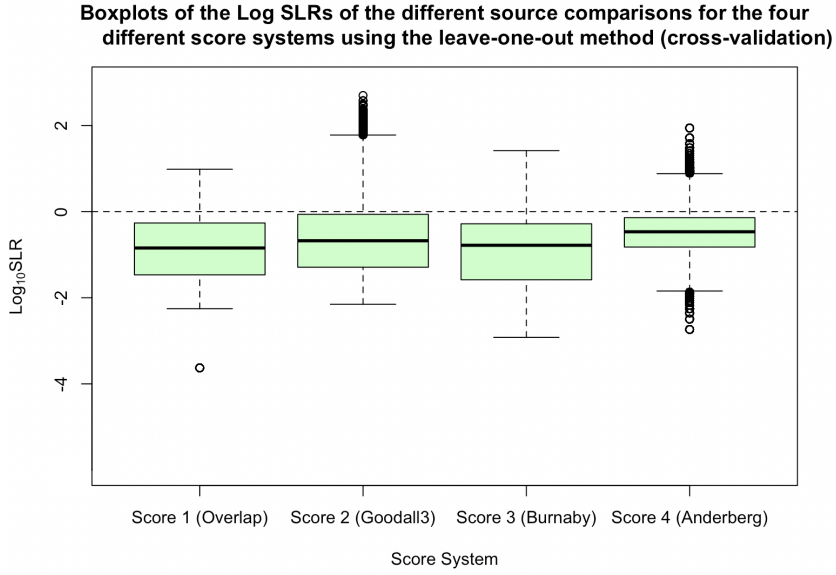


Figure 6.2: This figure shows the boxplots of the $\log_{10}$(SLR) of **different** source comparisons for the scores using the leave-one-out method (cross-validation). The horizontal dashed black line represents a $\log_{10}$(SLR) value of 0.

First, the boxplots in the two figures are analyzed. After this, the systems are ranked based on the distance between the boxplots of the same source and different source comparisons; the larger the distance, the higher the discriminating power. The discriminating power can be defined as the ability of the SLR system to differentiate between the two hypothesis H_1 and H_2. So, the higher the discriminating power, the stronger the evidence that the system forms (see figure 2.1 that displays the verbal expressions connected to the SLR), thus the better the SLR system performs. In summary, the larger the distance between the boxplots of the same source and different source comparisons, the better the SLR system performs.

By looking at figure 6.1, it appears that score 1 has the highest median $\log_{10}$(SLR) (then score 2, then score 3 and score 4 has the lowest median). All medians lie above a $\log_{10}$(SLR) value of 0 which means that it is more likely that the sources of the documents are the same. This makes sense for the same source scores. The entire boxplot (without the whiskers) lies above 0 for scores 1, 2 and 3. Score 2 is the only score to have whiskers on both sides of the boxplot and score 4 has the longest whisker (then score 2, then score 3 and score 1 has the shortest whisker). The box lengths (interquartile ranges) show how the data is dispersed; a longer box means more dispersed data. So score 4 has the most dispersed data (then score 2, then score 1 and score 3 has the least dispersed data). Score 1 appears to be negatively skewed (left skewed), score 2 appears to be symmetrical and scores 3 and 4 appear to be positively skewed (right skewed).[16] Note that score 4 has no outliers (so it has no unusual observations that are far removed from the other values of the data). [20] Score 1 has the least number of outliers (after score 4), then score 2 and score 3 has the most outliers. Score 2 is the only score that has outliers on both sides of the boxplot. Furthermore, the number of outliers below the boxplot is a measure for the misleading evidence of the same source scores. Here misleading evidence occurs if $SLR < 1$ (so $\log_{10}(SLR) < 0$) given H_1. Thus, if the SLR indicates that there is a higher likelihood that two sources are different given that the sources are the same. More on this in section 6.3. Score 4 has no outliers, so it has no outliers below the boxplot. Score 1 has the least number of outliers below the boxplot (after score 4), then score 2 and score 3 has the most outliers below the boxplot. [16]

Looking at figure 6.2, it appears that score 4 has the highest median $\log_{10}$(SLR) (then score 2, then score 3 and score 1 has the lowest median). All medians lie below a $\log_{10}$(SLR) value of 0 which means that it is more likely that the sources of the documents are different. This makes sense for the different source scores. The entire boxplot (without the whiskers) lies below 0 for all of the four scores as well. All of the scores have whiskers on both sides of the boxplot and score 3 has the longest whiskers (then score 2, then score 1 and score 4 has the shortest whiskers). Note that the whiskers are longer for the different source comparisons (in figure 6.2) than for the same source compar-

isons (in figure 6.1). This makes sense, because the same source documents, compared to the different source ones, are more alike, therefore have similar scores, and thus have less scattered data. Score 3 has the most dispersed data (judging by the box lengths as explained before) (then score 2, then score 1 and score 4 has the least dispersed data). Score 3 appears to be negatively skewed (left skewed) and scores 1, 2 and 4 appear to be fairly symmetrical. [16] Note that score 3 has no outliers (so it has no unusual observations that are far removed from the other values of the data). [20] Score 1 has the least number of outliers (after score 3) (only one outlier), then score 2 and score 4 has the most outliers. Score 4 is the only score that has outliers on both sides of the boxplot. Furthermore, the number of outliers above the boxplot is a measure for the misleading evidence of the different source scores. Here misleading evidence occurs if $SLR > 1$ (so $\log_{10}(SLR) > 0$) given H_2. Thus, if the SLR indicates that there is a higher likelihood that two sources are the same given that the sources are different. More on this in section 6.3. Score 3 has no outliers, so it has no outliers above the boxplot. Score 1 also has no outliers above the boxplot. Score 4 has the least number of outliers above the boxplot (after scores 1 and 3) and score 2 has the most. [16]

Based on the leave-one-out method (cross-validation) and judging by the distance between the boxplots of the same source and different source comparisons, score 1 (Overlap) performs the best since it has the greatest discriminating power (because the distance between the boxplots is the greatest). After that, from best to worst; score 3 (Burnaby), score 2 (Goodall3) and score 4 (Anderberg).

The R code, that was used to carry out the leave-one-out method (cross-validation) in this section, can be found in appendix G.1.

6.2 Calculating the 95% SLR bootstrap confidence intervals

The process of acquiring the SLR systems has a sampling uncertainty, because the CEDAR database consists of random samples of handwriting. This uncertainty can be quantified by using the bootstrap technique.
For this, a sample of size 2400 of the 2400 same source scores is drawn with replacement (so the same score can be drawn multiple times). This means that a new sample of scores given H_1 is acquired. After this, for each of the 2400 bootstrapped same source scores, a sample of size 2397 of the 2397 different source scores (belonging to that bootstrapped same source score) is drawn (with replacement). This way, a new sample of scores given H_2 is acquired. Both of these new samples combined gives a data set of size $2400 \cdot 2397 = 5,752,800$ which is twice the size of the original data set (namely 2,876,400). After this, the new likelihood ratio function is determined in the same way as in chapter 5.

Note that the size of the new data set is irrelevant since each of the 2,876,400 scores occurs twice in this set and because the parametrization is performed on histograms which use probabilities. This was explained in section 3.1.
This sampling process is repeated 50 times which results in 50 SLRs for each score.
From the SLRs, the bootstrap confidence interval is derived which quantifies the sampling uncertainty; the larger the interval, the lower the precision of the SLR system. [14] In this report, the 95% bootstrap confidence intervals are calculated. These intervals imply that, if 100 different samples (of scores) are taken and for each sample the 95% bootstrap confidence interval is calculated, then (approximately) 95 of the 100 confidence intervals will contain the true mean value (of the scores) (μ). [26]
Note that the more the sampling process is repeated, the more accurate the resulting bootstrap interval. However, in order to accelerate the calculation, it was repeated 50 times. More on this in the Discussion.

Figures 6.3, 6.4, 6.5 and 6.6 show the medians (50% points) of the SLR bootstrap results with the boundaries of the 95% SLR bootstrap intervals of the four SLR systems.

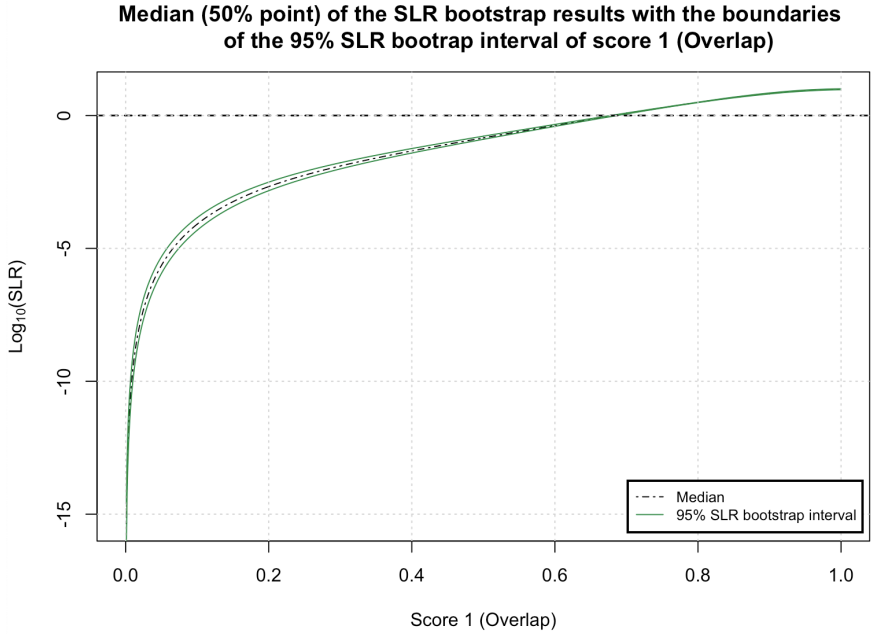


Figure 6.3: This figure shows the median (50% point) of the SLR bootstrap results with the boundaries of the 95% SLR bootstrap interval of score 1 (Overlap).

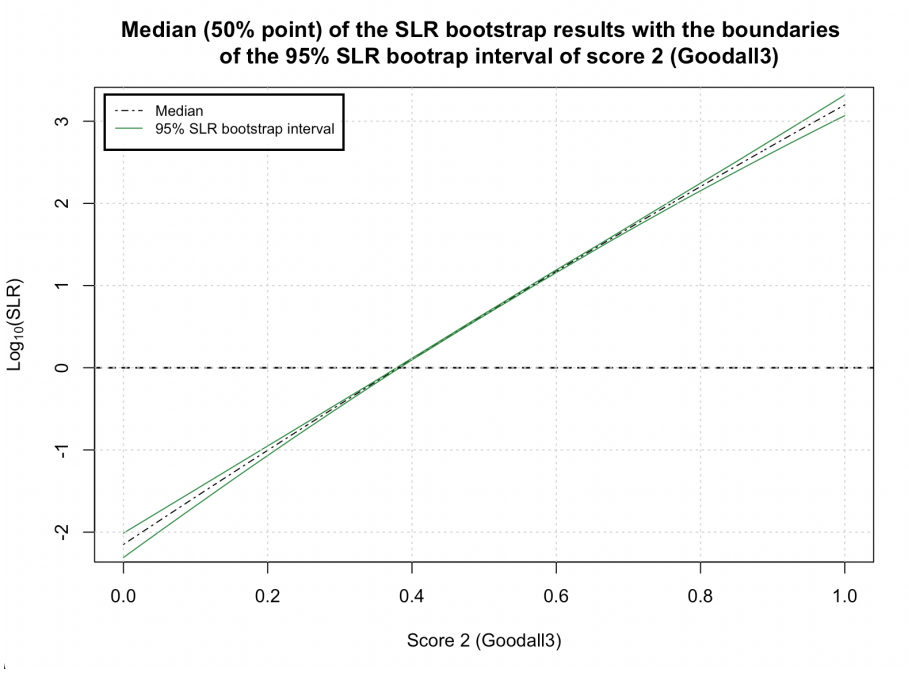


Figure 6.4: This figure shows the median (50% point) of the SLR bootstrap results with the boundaries of the 95% SLR bootstrap interval of score 2 (Goodall3).

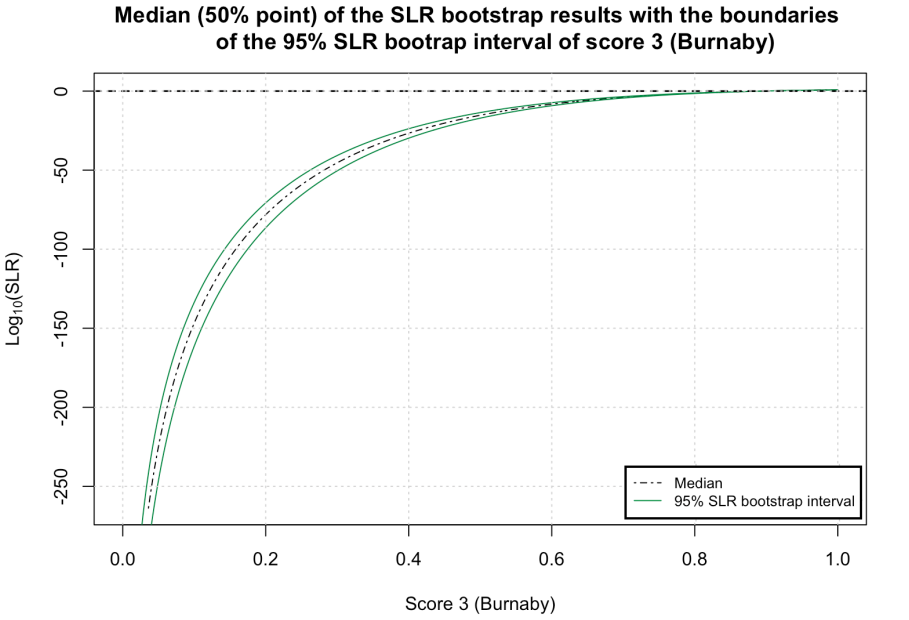


Figure 6.5: This figure shows the median (50% point) of the SLR bootstrap results with the boundaries of the 95% SLR bootstrap interval of score 3 (Burnaby).

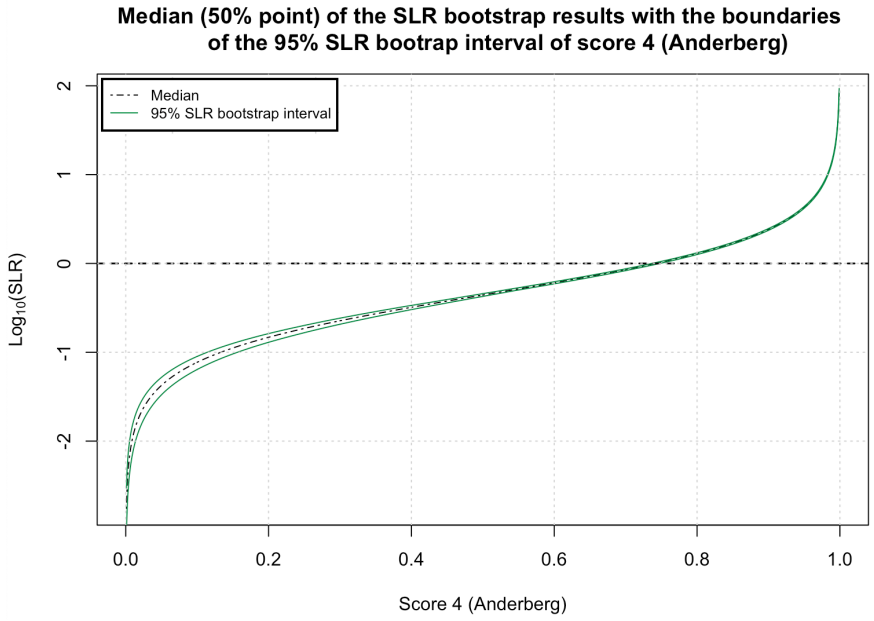


Figure 6.6: This figure shows the median (50% point) of the SLR bootstrap results with the boundaries of the 95% SLR bootstrap interval of score 4 (Anderberg).

When evaluating figures 6.3, 6.4, 6.5 and 6.6, it is evident that ranking the SLR systems based on the widths of the bootstrap intervals is complicated. Therefore the average widths and maximum widths of these intervals are calculated and ranked. Figure 6.7 shows the table of the average and maximum widths of the bootstrap intervals for the four different SLR systems. The sums of the average and maximum widths of the intervals are shown as well. Note that, for score 3 (Burnaby) the average and maximum width (and therefore also their sum) are infinity. This can also be seen in figure 6.5, since the width of the bootstrap interval tends to infinity when the score tends to 0.
From figure 6.7, it can be seen that score 4 has the lowest average width of the 95% bootstrap interval (so this score performs the best in this case). After this score 2 has the lowest average width, then score 1 and score 3 has the highest average width of the 95% bootstrap interval. Furthermore, it can be seen that score 2 has the lowest maximum width of the 95% bootstrap interval (so this score performs the best in this case). After this score 4 has the lowest maximum width, then score 1 and score 3 has the highest maximum width of the 95% bootstrap interval. This ranking is shown in the table of figure 6.8. The cells outlined in red have the highest ranking.
The summed widths (of the average and maximum widths) are ranked as well (which is also shown in the table of figure 6.8). This is done because, for example, scores 2 and 4 have the same sum of rankings (of 3), but score 2 has a lower summed width than score 4. So, using these summed widths, it can

be said that score 2 has the smallest bootstrap interval and thus is the most precise SLR system (so this score performs the best in this case). After this score 4 has the smallest one, then score 1 and score 3 has the largest bootstrap interval and thus is the least precise SLR system. This ranking is later used for the comparison of the SLR systems.

Score function	*Average width of 95% bootstrap interval*	*Maximum width of 95% bootstrap interval*	*Summed widths of 95% bootstrap interval*
1 (Overlap)	0.192	1.492	1.684
2 (Goodall3)	0.098	0.302	0.400
3 (Burnaby)	Infinity	Infinity	Infinity
4 (Anderberg)	0.063	0.500	0.563

Figure 6.7: This figure shows the table of the average widths and maximum widths of the bootstrap intervals for the four different SLR systems (the sums of the widths are shown as well).

Score function	*Ranking of average width of interval*	*Ranking of maximum width of interval*	*Sum of the rankings*	*Ranking of the summed widths of the interval*
1 (Overlap)	3	3	6	3
2 (Goodall3)	2	1	3	1
3 (Burnaby)	4	4	8	4
4 (Anderberg)	1	2	3	2

Figure 6.8: This figure shows the table of the rankings of the average widths and maximum widths of the bootstrap intervals (with the sum and ranking of summed widths). Cells outlined in red have the highest ranking.

The R code, that was used to calculate the 95% bootstrap confidence intervals in this section, can be found in appendix G.2.

6.3 Quantifying the misleading evidence

In order to quantify the misleading evidence, the percentages of misleading evidence are calculated in subsection 6.3.1. Furthermore, the indications of strength of evidence (in subsection 6.3.2) and the expected values (in subsection 6.3.3) are calculated as well. Lastly, in subsection 6.3.4, the ECE plots are obtained. Each one of these four performance characteristics are ranked, then the sum of these rankings over all of the performance characteristics is taken. The SLR system with the lowest sum of rankings, is considered to be the "best" SLR system based on misleading evidence only.

6.3.1 Calculating the percentages of misleading evidence

As mentioned in chapter 2, if $SLR > 1$ there is a higher likelihood that two sources are the same and if $SLR < 1$ there is a higher likelihood that two sources are different. However, every properly working SLR system will also create SLRs that support the false hypothesis. This means that there is an $SLR > 1$ given H_2 (the hypothesis that states that the sources are different) or that there is an $SLR < 1$ given H_1 (the hypothesis that states that the sources are the same). These SLRs, that support the false hypotheses, are called misleading evidence. Note that, if for an SLR system the SLR is always equal to 1, it is not called misleading evidence. Furthermore, other thresholds for misleading evidence can be applied as well. For example, a threshold that depends on the strength of (misleading) evidence can be used (more on this in section 6.3.2). [24] However, since in general a threshold of one is applied, that threshold will be used in this research as well. Lastly, as mentioned in chapter 2, SLRs are best used in combination with the (subjective) opinion of forensic examiners. So, the rate of misleading evidence is only a performance measure of the SLR system. [14]

The table in figure 6.9 shows the rates of misleading evidence for the four different SLR systems (same and different source and the average). For the same source scores, score 3 has the lowest rate of misleading evidence (so this score performs the best in this case). After this score 1 has the lowest rate, then score 2 and score 4 has the highest rate of misleading evidence. For the different source scores, score 4 has the lowest rate of misleading evidence (so this score performs the best in this case). After this score 2 has the lowest rate, then score 1 and score 3 has the highest rate of misleading evidence. This ranking is shown in the table of figure 6.10. The cells outlined in red have the highest ranking.
It appears that, if a score function has a low rate of misleading evidence for the same source scores, it has a high rate of misleading evidence for the different source scores (and vice versa). This can also be seen in the column of the sum of the rankings in the table of figure 6.10; all score functions have a sum of rankings of 5. Therefore, the average percentages of misleading evidence are ranked as well (which is also shown in the table of figure 6.10). So, score 2 has the lowest average rate of misleading evidence (so this score performs the best in this case). After this score 1 has the lowest average rate, then score 4 and score 3 has the highest average rate of misleading evidence. This ranking is later used for the comparison of the SLR systems.

Score function	Percentage of misleading evidence (SLR<1) given H_1 (same source)	Percentage of misleading evidence (SLR>1) given H_2 (different source)	Average percentage of misleading evidence
1 (Overlap)	17.125	24.477	20.801
2 (Goodall3)	19.292	19.199	19.246
3 (Burnaby)	15.500	32.718	24.109
4 (Anderberg)	27.167	17.438	22.303

Figure 6.9: This figure shows the table of the rates of misleading evidence for the four different SLR systems (same and different source and the average).

Score function	Ranking of percentages of misleading evidence (same source)	Ranking of percentages of misleading evidence (different source)	Sum of the rankings	Ranking of the average percentages of misleading evidence
1 (Overlap)	2	3	5	2
2 (Goodall3)	3	2	5	1
3 (Burnaby)	1	4	5	4
4 (Anderberg)	4	1	5	3

Figure 6.10: This figure shows the table of the rankings of percentages of misleading evidence for same and different source (with the sum and ranking of average percentages). Cells outlined in red have the highest ranking.

6.3.2 Calculating the indications of the strength of misleading evidence

The tables in figures 6.11, 6.12, 6.13 and 6.14 show the percentages of SLRs in ten intervals (for the same source and different source scores) for the four SLR systems. This is an indication of the strength of misleading evidence toward false hypotheses; if the misleading SLRs are not far from an SLR of 1, the strength of misleading evidence is lower (which is beneficial). [14] In order to calculate the percentages of SLRs in the intervals for the same source and different source scores, $n_1 = 2400$ and $n_2 = 2,876,400$ observations were considered respectively (see chapter 3). Here n_1 and n_2 are the numbers of similarity scores given H_1 and H_2 respectively. Note that these are different definitions for n_i than given previously. Also note that, if the percentages of misleading evidence of figures 6.11, 6.12, 6.13 and 6.14 are summed (so the percentages in columns "$< 1/10,000$" to "$[1/10; 1)$" for same source scores and the percentages in columns "$[1; 10)$" to "$\geq 10,000$" for different source scores), the percentages of figure 6.9 are obtained.
In order to discover which SLR system has misleading SLRs that are the closest to an SLR of 1, a transformation of the data in the four tables is required. This is done as follows. The further an interval (which contains a percentage of misleading SLRs that is greater than 0) is from 1, the higher the factor with which the percentage in that interval is multiplied to obtain the trans-

formed percentage. For example, in the case of same source scores, misleading evidence toward false hypotheses occurs if $SLR < 1$. So, if a percentage of misleading SLRs (that is greater than 0) lies in $[\frac{1}{10}; 1)$, the transformed percentage is equal to 1 multiplied by the percentage in the interval. If a percentage of misleading SLRs (that is greater than 0) lies in $(-\infty; \frac{1}{10{,}000})$, the transformed percentage is equal to 5 multiplied by the percentage in the interval, since it is 5 intervals "away" from 1. For example, if 0.1 percent of the SLRs lie in the interval $(-\infty; \frac{1}{10{,}000})$, then the transformed percentage is equal to $5 \cdot 0.1 = 0.5$. If percentages of misleading SLRs (that are greater than 0) lie in multiple intervals (for example in $[\frac{1}{10}; 1)$ and $(-\infty; \frac{1}{10{,}000})$ for same source), the transformed percentages are summed. Note that, in the case of different source scores, misleading evidence toward false hypotheses occurs if $SLR > 1$. So, if a percentage of misleading SLRs (that is greater than 0) lies in $[1; 10)$, the transformed percentage is equal to 1 multiplied by the percentage in the interval. If it lies in $[10,000; \infty)$, the transformed percentage is equal to 5 multiplied by the percentage in the interval.
The table in figure 6.15 shows the transformed percentages of the misleading SLRs in the intervals for same and different source (with the average transformed percentages). These transformed percentages are used to rank the strength of misleading evidence.

Interval	*Percentage of SLRs in interval given H_1 (same source) (n_1=2400)*	*Percentage of SLRs in interval given H_2 (different source) (n_2=2.876.400)*
<1/10,000	0	0.526
[1/10,000;1/1,000)	0.125	5.798
[1/1,000;1/100)	0.500	11.369
[1/100;1/10)	1.708	17.517
[1/10;1)	14.792	40.312
[1;10)	82.875	24.477
[10;100)	0	0
[100;1,000)	0	0
[1,000;10,000)	0	0
≥10,000	0	0

Figure 6.11: This figure shows the percentages of SLRs in ten intervals (for the same source and different source scores) for **score 1** (Overlap). This is an indication of the strength of misleading evidence toward false hypotheses.

Interval	Percentage of SLRs in interval given H_1 (same source) (n_1=2400)	Percentage of SLRs in interval given H_2 (different source) (n_2=2.876.400)
<1/10,000	0	0
[1/10,000;1/1,000)	0	0
[1/1,000;1/100)	0.042	5.230
[1/100;1/10)	2.333	33.628
[1/10;1)	16.917	41.943
[1;10)	44.208	16.988
[10;100)	33.292	2.196
[100;1,000)	3.208	0.016
[1,000;10,000)	0	0
≥10,000	0	0

Figure 6.12: This figure shows the percentages of SLRs in ten intervals (for the same source and different source scores) for **score 2** (Goodall3). This is an indication of the strength of misleading evidence toward false hypotheses.

Interval	Percentage of SLRs in interval given H_1 (same source) (n_1=2400)	Percentage of SLRs in interval given H_2 (different source) (n_2=2.876.400)
<1/10,000	0.083	1.597
[1/10,000;1/1,000)	0.083	2.672
[1/1,000;1/100)	0.708	7.058
[1/100;1/10)	2.333	18.763
[1/10;1)	12.292	37.192
[1;10)	84.500	32.718
[10;100)	0	0
[100;1,000)	0	0
[1,000;10,000)	0	0
≥10,000	0	0

Figure 6.13: This figure shows the percentages of SLRs in ten intervals (for the same source and different source scores) for **score 3** (Burnaby). This is an indication of the strength of misleading evidence toward false hypotheses.

Interval	Percentage of SLRs in interval given H_1 (same source) (n_1=2400)	Percentage of SLRs in interval given H_2 (different source) (n_2=2.876.400)
<1/10,000	0	0.530
[1/10,000;1/1,000)	0	0
[1/1,000;1/100)	0	0.109
[1/100;1/10)	0.875	15.474
[1/10;1)	26.292	66.500
[1;10)	41.792	15.340
[10;100)	4.333	0.109
[100;1,000)	0	0
[1,000;10,000)	0	0
≥10,000	26.708	1.989

Figure 6.14: This figure shows the percentages of SLRs in ten intervals (for the same source and different source scores) for **score 4** (Anderberg). This is an indication of the strength of misleading evidence toward false hypotheses.

Score function	Transformed percentage of SLRs in interval given H_1 (same source) (n_1=2400)	Transformed percentage of SLRs in interval given H_2 (different source) (n_2=2.876.400)	Average transformed percentage of SLRs in intervals
1 (Overlap)	20.208	24.477	22.343
2 (Goodall3)	21.710	21.428	21.569
3 (Burnaby)	19.830	32.718	26.274
4 (Anderberg)	28.042	25.503	26.773

Figure 6.15: This figure shows the table of the transformed percentages of the misleading SLRs in the intervals for same and different source (with the average transformed percentages).

By the table of figure 6.15, for the same source scores, the misleading SLRs of score 3 are the closest to an SLR of 1 (so this score performs the best in this case). After this the misleading SLRs of score 1 are the closest, then score 2 and the misleading SLRs of score 4 are the furthest from an SLR of 1. For the different source scores, the misleading SLRs of score 2 are the closest to an SLR of 1 (so this score performs the best in this case). After this the misleading SLRs of score 1 are the closest, then score 4 and the misleading SLRs of score 3 are the furthest from an SLR of 1. This ranking is shown in the table of figure 6.16. The cells outlined in red have the highest ranking.
The average strengths of misleading evidence are ranked as well (which is also shown in the table of figure 6.16). This is done for the same reason as described in subsection 6.3.1. That is, for example, scores 1 and 2 have the same sum of rankings (of 4), but score 2 has a lower average strength of misleading evidence than score 1. So, using this average, the misleading SLRs of score 2 are the closest to an SLR of 1 on average (so this score performs the best in this case). After this the misleading SLRs of score 1 are the closest on average, then score

3 and the misleading SLRs of score 4 are the furthest from an SLR of 1 on average. This ranking is later used for the comparison of the SLR systems.

Score function	Ranking of strength of evidence (same source)	Ranking of strength of evidence (different source)	Sum of the rankings	Ranking of the average strength of evidence
1 (Overlap)	2	2	4	2
2 (Goodall3)	3	1	4	1
3 (Burnaby)	1	4	5	3
4 (Anderberg)	4	3	7	4

Figure 6.16: This figure shows the table of the rankings of strength of misleading evidence for same and different source (with the sum and ranking of average strength). Cells outlined in red have the highest ranking.

The strength of misleading evidence can be visualized in the form of a Tippett plot. Figure 6.17 shows the Tippett plot of the four different SLR systems for the same (solid lines) and different source scores (dashed lines). So it shows the proportions of cases in which the SLRs (given H_1 or H_2) exceed certain values. [14] In the case of same source scores, misleading evidence toward false hypotheses occurs if $SLR < 1$ and, in the case of different source scores, it occurs if $SLR > 1$. This is also why the Tippett plots of the same source scores lie more to the right (for all score functions) than those of the different source scores.
The greater the distance between the Tippett plots for the same source and different source scores, the greater the discriminating power (so the better the system is calibrated and the less misleading SLRs occur). However, when evaluating figure 6.17, it is evident that ranking the SLR systems based on the distance between the Tippett plots is complicated.
Note that there is a cutoff on the x axis of figure 6.17 at an SLR of 10^{-5} and at an SLR of 10^5 in order to maintain visibility. So, there is a proportion of cases in which the SLRs (given H_1 or H_2), for score 4 (Anderberg), exceed an SLR value of 10^5.
The shape of the Tippett plot of the SLR system of score 1 (Overlap) can be explained by the fact that the system only has eight possible scores. This was explained in chapter 4.

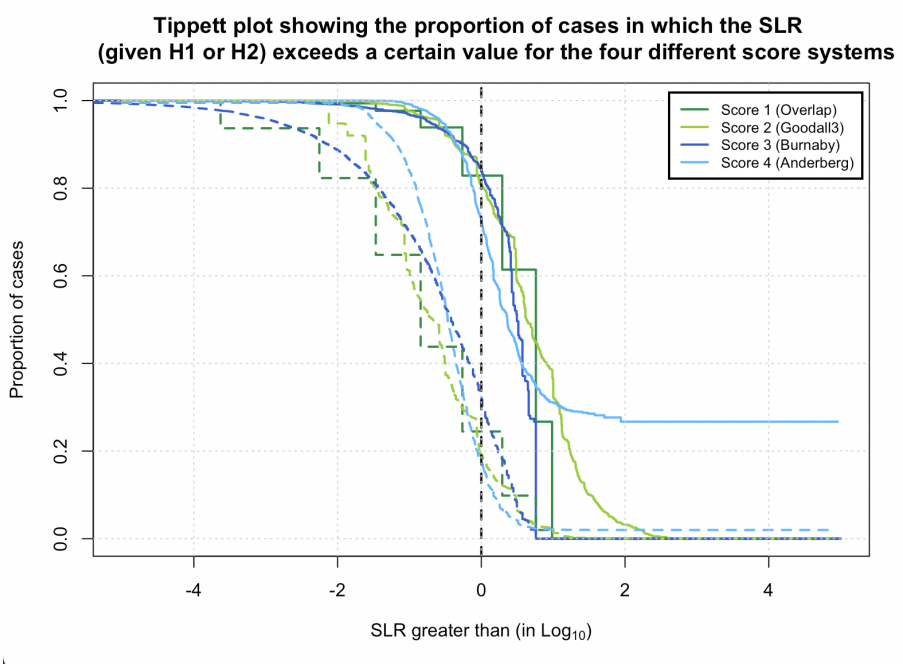


Figure 6.17: This figure shows the Tippett plot for the four different SLR systems and for the same (solid lines) and different source scores (dashed lines).

6.3.3 Calculating the expected values

Every properly working SLR system (for which the possible range of values given each hypothesis is the same) has an expected value of the SLR of 1 for different source comparisons. Furthermore, the expected value of the inverse SLR for same source comparisons is 1 as well. [28] The derivations of these two expected values is shown in equations (6.1) and (6.2) (for which equation (2.2) and Bayes' Theorem are used (see chapter 2)).

$$E(SLR|H_2) = E(\frac{f(s|H_1)}{f(s|H_2)}|H_2) = \int_s \frac{f(s|H_1)}{f(s|H_2)} \cdot f(s|H_2)ds = \int_s f(s|H_1)ds = 1 \tag{6.1}$$

$$E(\frac{1}{SLR}|H_1) = E(\frac{f(s|H_2)}{f(s|H_1)}|H_1) = \int_s \frac{f(s|H_2)}{f(s|H_1)} \cdot f(s|H_1)ds = \int_s f(s|H_2)ds = 1 \tag{6.2}$$

Where;
$f(x)$ = The probability density function of the random variable X [14]

The expected values of the SLR given H_2 and of the $\frac{1}{SLR}$ given H_1 are shown in the table of figure 6.18 for the four different SLR systems (including the average expected values). Note that $E(SLR|H_2)$ is closer to 1 than $E(\frac{1}{SLR}|H_1)$ for all scores except score 4. Also note that the graph of the SLR system with score function 4 goes to infinity, therefore the value of $E(SLR|H_2)$ is infinity. This means that the average expected value for score 4 is infinity as well.
It is desired that the expected values are as close to 1 as possible. So, for the same source scores, score 4 has the expected value that is the closest to 1 (so this score performs the best in this case). After this score 2 has the closest expected value, then score 1 and score 3 has the expected value that is the furthest from 1. For the different source scores, score 2 has the expected value that is the closest to 1 (so this score performs the best in this case). After this score 3 has the closest expected value, then score 1 and score 4 has the expected value that is the furthest from 1. This ranking is shown in the table of figure 6.19. The cells outlined in red have the highest ranking.
The average expected values are ranked as well (which is also shown in the table of figure 6.19). This is done for the same reason as described in subsection 6.3.1. That is, for example, scores 1 and 3 have the same sum of rankings (of 6), but score 1 has a significantly lower average expected value (that is closer to 1) than score 3. So, using this average, score 2 has the average expected value that is the closest to 1 (so this score performs the best in this case). After this score 1 has the closest average expected value, then score 3 and score 4 has the average expected value that is the furthest from 1. This ranking is later used for the comparison of the SLR systems.

Score function	*E(1/SLR) given H_1 (same source)*	*E(SLR) given H_2 (different source)*	*Average expected value*
1 (Overlap)	7.375	1.072	4.224
2 (Goodall3)	1.155	1.015	1.085
3 (Burnaby)	180.906	0.970	90.938
4 (Anderberg)	0.957	Infinity	Infinity

Figure 6.18: This figure shows the table of the expected values of the SLR given H_2 and of the $\frac{1}{SLR}$ given H_1 for the four different SLR systems. It also shows the average expected values.

Score function	Ranking of expected value (same source)	Ranking of expected value (different source)	Sum of the rankings	Ranking of the average expected value
1 (Overlap)	3	3	6	2
2 (Goodall3)	2	1	3	1
3 (Burnaby)	4	2	6	3
4 (Anderberg)	1	4	5	4

Figure 6.19: This figure shows the table of the rankings of the expected values for same and different source (with the sum and ranking of average expected values). Cells outlined in red have the highest ranking.

The R code, that was used to quantify the misleading evidence in this section, can be found in appendix G.3.

6.3.4 Obtaining the ECE plots

Using the ECE (Empirical Cross-Entropy) is an alternative way of weighing misleading evidence. It gives penalties for each posterior probability which depend on the true hypothesis and the strength of the evidence; the stronger the evidence of the true hypothesis, the lower the penalty. So, the weaker the evidence of the true hypothesis (or the stronger the misleading evidence), the higher the penalty.
In order to obtain the posterior probabilities, the prior probabilities are required (this was explained in chapter 2). But these are unknown, therefore the ECE is plotted with respect to the log of the prior odds. Here the prior odds are $\frac{P(H_1)}{P(H_2)}$ with H_1 and H_2 the hypotheses as defined in chapter 2. The ECE is calculated with the following formula;

$$ECE = -\frac{P(H_1)}{n_1}\sum_{s\in S_1}\log_2 P(H_1|s) - \frac{P(H_2)}{n_2}\sum_{s\in S_2}\log_2 P(H_2|s)$$

Where;
S_1 = The similarity scores given H_1
S_2 = The similarity scores given H_2
n_1 = The number of similarity scores given H_1 (so the size of S_1)
n_2 = The number of similarity scores given H_2 (so the size of S_2)

This formula can be rewritten into a function that only depends on the log of the prior odds. Namely;

$$ECE = -\frac{10^{\Omega}}{n_1(1+10^{\Omega})} \sum_{s \in S_1} \log_2 \left(\frac{P(s|H_1) \cdot 10^{\Omega}}{P(s|H_1) \cdot 10^{\Omega} + P(s|H_2)} \right) - \frac{1}{n_2(1+10^{\Omega})} \sum_{s \in S_2} \log_2 \left(\frac{P(s|H_2)}{P(s|H_1) \cdot 10^{\Omega} + P(s|H_2)} \right)$$

Where;
$\Omega = \log_{10}\left(\frac{P(H_1)}{P(H_2)}\right)$ = The logarithm with base 10 of the prior odds
$P(s|H_1), P(s|H_2)$ = The probabilities of the same source and different source scores respectively (these were found in chapter 5)

See appendix H.1 for this derivation.

In the case of a perfectly discriminating system, the ECE is equal to 0 (for all prior odds). Thus, the closer the ECE of a system is to 0 (so the flatter the ECE curve of a system is), the more informative an SLR system is (which is beneficial).
Note that the ECE is used in order to examine the performance of an SLR system. It cannot be used in order to calibrate the SLRs of the SLR system. [14]

Figure 6.20 shows the ECE plots for the four different SLR systems. It also shows the ECE plot in the case of a constant SLR of 1 in a system. This is called a noninformative SLR system. The formula for the ECE of a noninformative SLR system is;

$$ECE = -\frac{10^{\Omega}}{1+10^{\Omega}} \cdot \log_2 \left(\frac{10^{\Omega}}{10^{\Omega}+1} \right) - \frac{1}{1+10^{\Omega}} \cdot \log_2 \left(\frac{1}{10^{\Omega}+1} \right)$$

See appendix H.2 for this derivation.

In figure 6.20 it can be seen that score 2 has the flattest ECE plot (therefore it is the most informative, so this score performs the best in this case). After this score 1 has the flattest plot, then score 4 and score 3 has the highest ECE plot (therefore it is the least informative).

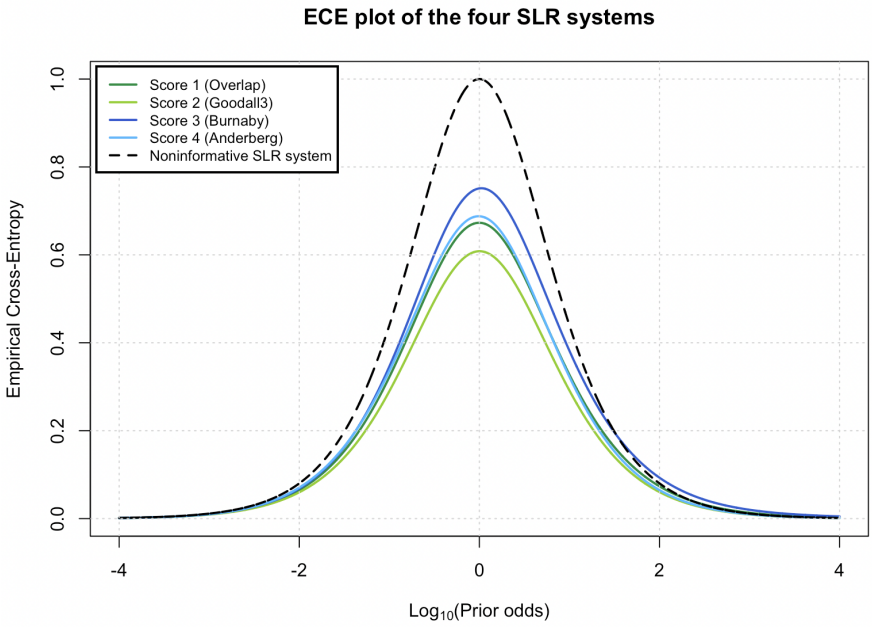


Figure 6.20: This figure shows the ECE plots for the four different SLR systems and the ECE plot for a noninformative SLR system (so when the SLR is always equal to 1).

Note that, for score 4 (Anderberg), $P(s|H_1)$ and $P(s|H_2)$ can take values of infinity and 0. This leads to calculations with logarithms of zero which are undefined. Therefore, for the creation of the ECE plot of score 4, infinity is approximated by 10^{10} and 0 is approximated by $\frac{1}{10^{10}}$.

The R code, that was used to obtain the ECE plots in this subsection, can be found in appendix G.4.

6.3.5 Misleading evidence summary table

The table in figure 6.21 shows the rankings that were obtained in this section. Since it is necessary to know which SLR system performs the best based on all the performance characteristics of misleading evidence, the comparison rankings are summed. After that, these sums are transformed in such a way that they add up to 10 (just as the rankings 1, 2, 3 and 4 do) (for example, the SLR system with the lowest sum of rankings gets transformed ranking 1). This is required for the table in the next section, so when all the SLR systems are compared based on the rankings of all the performance characteristics. Thus, the sums of the rankings and the transformed sums are shown in figure 6.21 as well. The cells outlined in red have the highest ranking. Therefore, based on the misleading evidence, score 2 (Goodall3) performs the best. After

that, from best to worst; score 1 (Overlap), score 4 (Anderberg) and score 3 (Burnaby) (these last two are tied for third place). Note that score 2 performs the best based on all of performance characteristics for misleading evidence.

Score function	Ranking of percentages of misleading evidence	Ranking of indication of strength of evidence	Ranking of expected value	Ranking of ECE plot	Sum of the rankings	Transformed sum of the rankings (misleading evidence)
1 (Overlap)	2	2	2	2	8	2
2 (Goodall3)	1	1	1	1	4	1
3 (Burnaby)	4	3	3	4	14	3.5
4 (Anderberg)	3	4	4	3	14	3.5

Figure 6.21: This figure shows the table of the rankings that were obtained in this section (with their sum and transformed sum). The cells outlined in red have the highest ranking.

6.4 Evaluation summary table

The table in figure 6.22 shows all of the rankings that were obtained in this chapter (with their sum and transformed sum). The cells outlined in red have the highest ranking. Note that every one of the three performance characteristics contributes with the same weight to the sum of rankings (and therefore the transformed sum). This is the case because, for every performance characteristic, the sum of the rankings were transformed in such a way that they add up to 10 (just as the rankings 1, 2, 3 and 4 do). This was explained in subsection 6.3.5.
The table of figure 6.22 will be used in order to draw the conclusion in the next chapter.

Score function	Ranking of leave-one-out method (cross-validation)	Ranking of 95% SLR bootstrap confidence interval	Ranking of misleading evidence	Sum of the rankings	Transformed sum of the rankings
1 (Overlap)	1	3	2	6	2
2 (Goodall3)	3	1	1	5	1
3 (Burnaby)	2	4	3.5	9.5	3.5
4 (Anderberg)	4	2	3.5	9.5	3.5

Figure 6.22: This figure shows the table of all of the rankings that were obtained in this chapter (with their sum and transformed sum). The cells outlined in red have the highest ranking.

Chapter 7

Conclusion

The objective of this research was to employ score-based likelihood ratio systems for the comparison of handwriting and study their performance.

The handwriting samples of 800 writers (three documents each) were considered. After the letter combinations "er" were extracted, the characteristics were entered into a user interface for each document which makes the analysis more time efficient. The SLR expresses the degree of uncertainty that a hypothesis is true. Furthermore, the common source problem was considered due to the unavailability of suspect specific data. So, it was tested if two writings from unknown writers originate from the same unknown writer. For this research four score functions were considered; Overlap (does not take uniqueness of matching or mismatching values into account), Goodall3 (takes uniqueness of matching values (so not mismatching ones) into account), Burnaby (takes uniqueness of mismatching values (so not matching ones) into account) and Anderberg (takes uniqueness of matching and mismatching values into account). This resulted in four SLR systems that were evaluated based on three performance characteristics; the leave-one-out method, 95% bootstrap interval and misleading evidence.

The table in figure 7.1 shows all of the rankings that were obtained in the previous chapter (with their sum and transformed sum). The cells outlined in red have the highest ranking. Therefore, **overall (based on all of the performance characteristics), score 2 (Goodall3) performs the best**. After that, from best to worst; score 1 (Overlap), score 4 (Anderberg) and score 3 (Burnaby) (these last two are tied for third place).
This result indicates that the SLR system, that performs the worst based on the performance characteristics, is obtained when only the uniqueness of mismatching characteristics (so not the matching ones) is taken into account (score 3) or when the uniqueness of matching and mismatching characteristics is taken into account (score 4). Not taking the uniqueness of matching and mismatching characteristics into account (score 1), improves the performance of the SLR system. Furthermore, it is best to use score functions that only take the uniqueness of matching characteristics (so not the mismatching ones) into account (score 2).

Score function	Ranking of leave-one-out method (cross-validation)	Ranking of 95% SLR bootstrap confidence interval	Ranking of misleading evidence	Sum of the rankings	Transformed sum of the rankings
1 (Overlap)	1	3	2	6	2
2 (Goodall3)	3	1	1	5	1
3 (Burnaby)	2	4	3.5	9.5	3.5
4 (Anderberg)	4	2	3.5	9.5	3.5

Figure 7.1: This figure shows the table of all of the rankings that were obtained in the previous chapter (with their sum and transformed sum). The cells outlined in red have the highest ranking.

So, overall score 2 performs the best based on the performance characteristics. However, this does not mean that it performs the best based on all of the performance characteristics. If one requires an SLR system that performs the best based on the leave-one-out method (so a system that has the greatest discriminating power), score function 1 (Overlap) has to be used. So the uniqueness of matching and mismatching characteristics is not taken into account in this case. If one requires an SLR system that performs the best based on the 95% bootstrap confidence interval (so a system that has the highest precision) and performs the best based on misleading evidence (so a system that produces the least number of SLRs that support false hypotheses), score function 2 (Goodall3) has to be used. So only the uniqueness of matching characteristics (so not the mismatching ones) is taken into account in this case. Therefore, **what score function is chosen to be used (either score 1 or 2) for the SLR system depends on the desired qualities of the system**. Note that score 2 performs the best based on 2 out of 3 performance characteristics.

The use of SLR systems has advantages and drawbacks. The advantages are:

1. SLR systems are objective and transparent (in comparison to the subjective opinion of forensic examiners) due to the analysis that is only based on data.

2. The behavior of SLR systems is known (by the evaluation based on the three performance characteristics).

3. The handwriting analysis with an SLR system is more time efficient since the computer does the comparison (after the characteristics of the handwriting are entered into the user interface).

4. Scores of SLR systems can be constructed in such a way that they take the uniqueness of characteristics into account.

5. SLR systems give an insight into the degree of uncertainty of the statement that two writings have the same writer.

The drawbacks are:

1. SLR systems only take a small part of the available information into account, because information is lost since the score function transforms multidimensional data to one dimensional data. Information is also lost when snippets of the handwriting are extracted and when these are transformed into characteristics.
2. Examining the accuracy and performance of SLR systems is complicated, because the LR (which is approximated by the SLR) is unknown.

These drawbacks will be elaborated on in the Discussion in the next chapter.

Lastly, SLRs are best used in combination with the (subjective) opinion of forensic examiners. This way, it is possible to benefit from both the objectivity and transparency of the SLR systems and from the knowledge and expertise of forensic examiners (who take more information into account than the SLR systems).

Thus, the objective of this research (employing score-based likelihood ratio systems for the comparison of handwriting and studying their performance) was achieved.

Chapter 8

Discussion

In this chapter, the limitations of the findings of this research are discussed and notions for further research are proposed.

First off, if more data is used for the research, more accurate results are obtained. For example, for this research, 800 writers (of the 1500 in the CEDAR database) were considered. They each wrote three documents. More accurate results are obtained if more writers are considered and if each writer produces more documents. Moreover, for this research, the letter combination "er" was examined. In the future, a different bigram or a letter combination consisting of more letters can be studied. Furthermore, the bigram "er" has $4 \cdot 6 \cdot 4 \cdot 4 \cdot 3 \cdot 4 \cdot 4 \cdot 5 = 92,160$ possible combinations of characteristics (for this calculation the number of values that each of the eight characteristics can take are multiplied). So, if more characteristics are taken into account (for example the characteristic of the amount of pen pressure), more accurate results are acquired. The same holds if each characteristic can take more values (for example if the characteristic "shape of r" also includes the capital letter "R" as a value it can take). In this way less characteristics are classified under "NSP" (no set pattern).
Sometimes the writers of the CEDAR database were inconsistent with their handwriting within one document. For example, in one document, sometimes the "r" is written in cursive and sometimes it is written in print. So, some characteristics can take multiple values within one document. This can be taken into account for further research. Occasionally the writers were also inconsistent with their handwriting between the documents. For example, in one document the "r" is written in cursive and in another document (of the same writer) it is written in print. This can also be taken into account for further research.
By utilizing the user interface, the handwriting analysis of this research is more time efficient than the analysis as it is done now. However, the characteristics of the bigram still need to be entered into the computer by hand. This is time consuming and can lead to mistakes due to human error. For further research, a computer program can be created (for example with neural networks) that classifies the characteristics of a bigram in handwriting automatically. This would make the analysis less time consuming and less prone to human error.
The code that was used in order to obtain the four SLR systems took a long time to run (a couple of days per SLR system). The same holds for the code

that was used for the calculation of the bootstrap intervals. This is why the process of this calculation of intervals was repeated 50 times and not more (this already leads to $50 \cdot 2 \cdot 2,876,400 = 287,640,000$ samples of different source scores (why it is multiplied by two is explained in chapter 6)). If this research is repeated in the future, a "faster" computer can be used. In this way, the coverage of the 95% bootstrap intervals, which was omitted from this research due to the long run time of the code, can also be studied.
For future research, more performance characteristics of the SLRs can be taken into account. For example, ECE plots can be studied after the PAV (pool adjacent violaters) algorithm is used. This algorithm transforms the SLRs in such a way that the ECE values are minimized, but it preserves the discrimination of the system. If the PAV algorithm does not improve the ECE curve significantly, it indicates that the system is calibrated well. [14]
As explained in subsection 6.3.1, other thresholds (than a threshold of 1) can be applied for the classification of misleading evidence. Further research could investigate what threshold classifies the misleading evidence the best and it could investigate how this performance can be measured.
As explained in section 6.2, the process of acquiring the SLR systems has a sampling uncertainty, because the CEDAR database consists of random samples of handwriting. This means that a new data base of handwriting leads to different results. That is why bootstrap intervals, that quantify this uncertainty, were constructed. For further research, it can be studied what to report in a legal case (so if the SLRs or the SLR bootstrap intervals have to be reported).
As explained in chapter 6, the LR (which is approximated by the SLR) is unknown. Therefore the SLR systems cannot be compared to the LR systems. So, examining the accuracy and performance of SLR systems is complicated. However, their behavior can be investigated. For this research, the leave-one-out method (cross-validation) was executed, the 95% SLR bootstrap confidence intervals were calculated and the misleading evidence was quantified. In the future, it can be researched how the accuracy of SLR systems can be assessed (and therefore if there is a "better" way to compare the SLR systems).
As explained in chapter 2, because the score function transforms multidimensional data to one dimensional data, information is lost. [27] Information is also lost when snippets of the handwriting are extracted and when these are transformed into characteristics. Because of this information loss and other reasons, LRs and SLRs are best used in combination with the (subjective) opinion of forensic examiners. This way, it is possible to benefit from both the objectivity and transparency of the LR (and SLR) systems and from the knowledge and expertise of forensic examiners (who take more information into account than the LR and SLR systems). [14] How the SLRs and opinions of forensic examiners can be combined can be subject for further research. If and how the SLR needs to be implemented in case work can also be studied in the future.

The robustness of the SLR systems can be described as follows. The SLRs that are produced depend on the score functions, the characteristics of the bigram that are analyzed, etc. So, a different setup of the SLR system will lead to different results. But, if the SLR system is set up in the same way (same score functions, same characteristics that are analyzed, etc.), the results will be the same. For this research, the CEDAR database (which consists out of American handwriting samples) was used to obtain the different source scores. Further research could investigate if the use of a different database will result in different SLRs. If this is the case, then, when the SLR system needs to be obtained for handwriting outside of the US, a database consisting out of handwriting samples from that other country has to be used. (For example it is possible that in the Netherlands more people write in cursive than in the US. This affects the uniqueness of the characteristics.)
Lastly, as explained in chapter 2, scores should take both the similarity and the typicality of the evidence into account (anchored approach). Here typicality means that the same and different source scores of the suspect should be used and not those of the general population. [17] It was also explained that, due to the unavailability of some data, the nonanchored approach is applied; scores only take similarity into account (and not typicality). Note that here the nonanchored approach is the common source problem and the anchored approach is the specific source problem. Thus, the SLR is an expression for the degree of uncertainty that a hypothesis is true for the general population, not a specific suspect in the case. The consequences of only taking similarity into account (and not typicality), can be researched in the future. It is expected that the anchored approach will perform better than the nonanchored approach since it takes more information into account (namely the typicality as well). The validity of this hypothesis can be subject to further study as well.

Bibliography

[1] Anderberg, M.R. (1973). *Cluster Analysis for Applications*. Academic Press.

[2] Boriah, S., Chandola, V., & Kumar, V. (2008). Similarity measures for categorical data: a comparative evaluation. *SIAM Data Mining Conference*, 243–254.

[3] CEDAR. (n.d.). *CEDAR Overview*. Retrieved April 27, 2022, from https://cedar.buffalo.edu/about.html

[4] CEDAR. (n.d.). *Research Collaborators*. Retrieved April 27, 2022, from https://cedar.buffalo.edu/Databases/

[5] Corstius, H. (1981). *Opperlandse taal- & letterkunde*. dbnl. https://www.dbnl.org/tekst/bran023oppe01_01/bran023oppe01_01_0008.php

[6] Dekalb Miller. (n.d.). *Forensic Handwriting and Signature Analysis*. Retrieved April 23, 2022, from https://dekalbmiller.com/forensic-handwriting-analysis/

[7] Encyclo NL. (n.d.). *Bigram Definities*. Retrieved April 29, 2022, from https://www.encyclo.nl/begrip/bigram

[8] Garton, N., Ommen, D., Niemi, J., & Carriquiry, A. (2020). Score-based likelihood ratios to evaluate forensic pattern evidence. *arXiv preprint arXiv:2002.09470*.

[9] Gehl, R. & Plecas, D. (2016). *Forensic Sciences. Introduction to Criminal Investigation: Processes, Practices and Thinking*. (pp. 140-158). Justice Institute of British Columbia.

[10] Hayes, A. (2022). *Bayes' Theorem*. Retrieved May 25, 2022, from https://www.investopedia.com/terms/b/bayes-theorem.asp

[11] Hossein, P. (n.d.). *Law of total probability*. Retrieved June 1, 2022, from https://www.probabilitycourse.com/chapter1/1_4_2_total_probability.php

[12] IMGonline. (n.d.). *Creation of the photo collage from multiple pictures online*. Retrieved April 30, 2022, from https://www.imgonline.com.ua/eng/photo-collage.php

[13] Kerkhoff, W., Stoel, R. D., Mattijssen, E. J. A. T., & Hermsen, R. (2013). The likelihood ratio approach in cartridge case and bullet comparison. *AFTE J, 45 (3)*, 284–9.

[14] Leegwater, A. J., Meuwly, D., Sjerps, M., Vergeer, P., & Alberink, I. (2017). Performance study of a score-based likelihood ratio system for forensic fingermark comparison. *Journal of forensic sciences, 62(3)*, 626-640.

[15] Lewandowsky, S. (2018). *Who kidnapped Charles Lindbergh, Jr? Forensic handwriting analysis and expertise*. Psychonomic Society. Retrieved April 23, 2022, from https://featuredcontent.psychonomic.org/who-kidnapped-charles-lindbergh-jr-forensic-handwriting-analysis-and-expertise/

[16] McLeod, S. (2019). *What does a box plot tell you?*. Retrieved May 22, 2022, from https://www.simplypsychology.org/boxplots.html

[17] Morrison, G. S., & Enzinger, E. (2018). Score based procedures for the calculation of forensic likelihood ratios–Scores should take account of both similarity and typicality. *Science & Justice, 58(1)*, 47-58.

[18] Muehlberger, R.J., Newman, K.W., Regent, J., & Wichmann, J.G. (1977). A statistical examination of selected handwriting characteristics. *Journal of Forensic Sciences, 22*, 206–210.

[19] NFI. (2017). Vakbijlage, De reeks waarschijnlijkheidstermen van het NFI en het Bayesiaanse model voor interpretatie van bewijs. *Technical report.*

[20] NIST SEMATECH (n.d.). *What are outliers in the data?*. Retrieved May 24, 2022, from https://www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm

[21] Ommen, D. (2017). Approximate Statistical Solutions to the Forensic Identification of Source Problem. *PhD thesis, South Dakota State University.*

[22] Ommen, D., & Saunders, C. (2018). Building a Unified Statistical Framework for the Forensic Identification of Source Problems. *Law Probability and Risk (17)*, 179–197. doi: 10.1093/lpr/mgy008.

[23] Scheijen, N. (2020). Forensic speaker recognition based on text analysis of transcribed speech fragments.

[24] Schönbrodt, F.D., & Wagenmakers, E. (2018). Bayes factor design analysis: Planning for compelling evidence. *Psychonomic Bulletin & Review, 25*, 128–142.

[25] Srihari, S. (2013). Statistical Examination of Handwriting Characteristics using Automated Tools.

[26] Sullivan, L. (n.d.). *Confidence Intervals*. Retrieved May 22, 2022, from https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_confidence_intervals/bs704_confidence_intervals_print.html

[27] Tang, Y., & Srihari, S. N. (2014). Likelihood ratio estimation in forensic identification using similarity and rarity. *Pattern Recognition, 47(3)*, 945-958.

[28] Van Leeuwen, D.A., & Brümmer, N. (2013). *The distribution of calibrated likelihoodratios in speaker recognition.* Retrieved May 30, 2022, from http://arxiv.org/abs/1304.1199

Appendix A

Detailed calculations, and formulas

The formulas used in this report are listed below. The meaning of the variables can be found in the List of variables.

Section: Calculation of LRs and SLRs

$LR(x, y) = \frac{P(x,y|H_1,I)}{P(x,y|H_2,I)}$

$SLR(x, y) = \frac{P(s(x,y)|H_1,I)}{P(s(x,y)|H_2,I)}$

Posterior Odds $= SLR \cdot$ Prior Odds

$\frac{P(H_1|s(x,y),I)}{P(H_2|s(x,y),I)} = \frac{P(s(x,y)|H_1,I)}{P(s(x,y)|H_2,I)} \cdot \frac{P(H_1|I)}{P(H_2|I)}$

Section: LR and SLR for multinomial features

$s(x, y) = \sum_{i=1}^{n} w_i \cdot s_i(x_i, y_i)$

Chapter: Methods: Construction of SLR systems

$s(x, y) = \sum_{i=1}^{n} w_i \cdot s_i(x_i, y_i)$

$p_i^2(x_i) = \frac{f_i(x_i)\cdot(f_i(x_i)-1)}{N\cdot(N-1)}$

$\hat{p}_i(x_i) = \frac{f_i(x_i)}{N}$

Score 1 (Overlap): $s_i(x_i, y_i) = \begin{cases} 1 & \text{if } x_i = y_i \\ 0 & \text{otherwise} \end{cases}$ and $w_i = \frac{1}{n} = \frac{1}{8}$

Score 2 (Goodall3): $s_i(x_i, y_i) = \begin{cases} 1 - p_i^2(x_i) & \text{if } x_i = y_i \\ 0 & \text{otherwise} \end{cases}$ and $w_i = \frac{1}{n} = \frac{1}{8}$

Score 3 (Burnaby): $s_i(x_i, y_i) = \begin{cases} 1 & \text{if } x_i = y_i \\ \frac{\sum_{q \in A_i} 2\log(1-\hat{p}_i(q))}{\log \frac{\hat{p}_i(x_i)\hat{p}_i(y_i)}{(1-\hat{p}_i(x_i))(1-\hat{p}_i(y_i))} + \sum_{q \in A_i} 2\log(1-\hat{p}_i(q))} & \text{otherwise} \end{cases}$

and $w_i = \frac{1}{n} = \frac{1}{8}$

Score 4 (Anderberg):

$$s(x,y) = \frac{\sum_{i \in \{1 \le i \le n : x_i = y_i\}} \left(\frac{1}{\hat{p}_i(x_i)}\right)^2 \frac{2}{n_i(n_i+1)}}{\sum_{i \in \{1 \le i \le n : x_i = y_i\}} \left(\frac{1}{\hat{p}_i(x_i)}\right)^2 \frac{2}{n_i(n_i+1)} + \sum_{i \in \{1 \le i \le n : x_i \neq y_i\}} \left(\frac{1}{2\hat{p}_i(x_i)\hat{p}_i(y_i)}\right)^2 \frac{2}{n_i(n_i+1)}}$$

Chapter: Evaluation of the quality of performance of SLR systems

$E(SLR|H_2) = 1$

$E(\frac{1}{SLR}|H_1) = 1$

$ECE = -\frac{P(H_1)}{n_1} \sum_{s \in S_1} \log_2 P(H_1|s) - \frac{P(H_2)}{n_2} \sum_{s \in S_2} \log_2 P(H_2|s)$

$ECE = -\frac{10^\Omega}{n_1(1+10^\Omega)} \sum_{s \in S_1} \log_2 \left(\frac{P(s|H_1) \cdot 10^\Omega}{P(s|H_1) \cdot 10^\Omega + P(s|H_2)}\right) - \frac{1}{n_2(1+10^\Omega)} \sum_{s \in S_2} \log_2 \left(\frac{P(s|H_2)}{P(s|H_1) \cdot 10^\Omega + P(s|H_2)}\right)$

$ECE = -\frac{10^\Omega}{1+10^\Omega} \cdot \log_2 \left(\frac{10^\Omega}{10^\Omega+1}\right) - \frac{1}{1+10^\Omega} \cdot \log_2 \left(\frac{1}{10^\Omega+1}\right)$ when $SLR = 1$

Appendix B

Document that was copied by all the writers and a handwritten sample

The document, that was written by each individual three times in order to create the CEDAR data set, is shown below. It contains every letter of the English alphabet at least once. Figure B.1 shows, of the CEDAR data set, a handwritten sample of the text (of the document) provided by a writer.

From

Jim Elder
829 Loop Street, Apt 300
Allentown, New York 14707

Nov 10, 1999

To

Dr. Bob Grant
602 Queensberry Parkway
Omar, West Virginia 25638

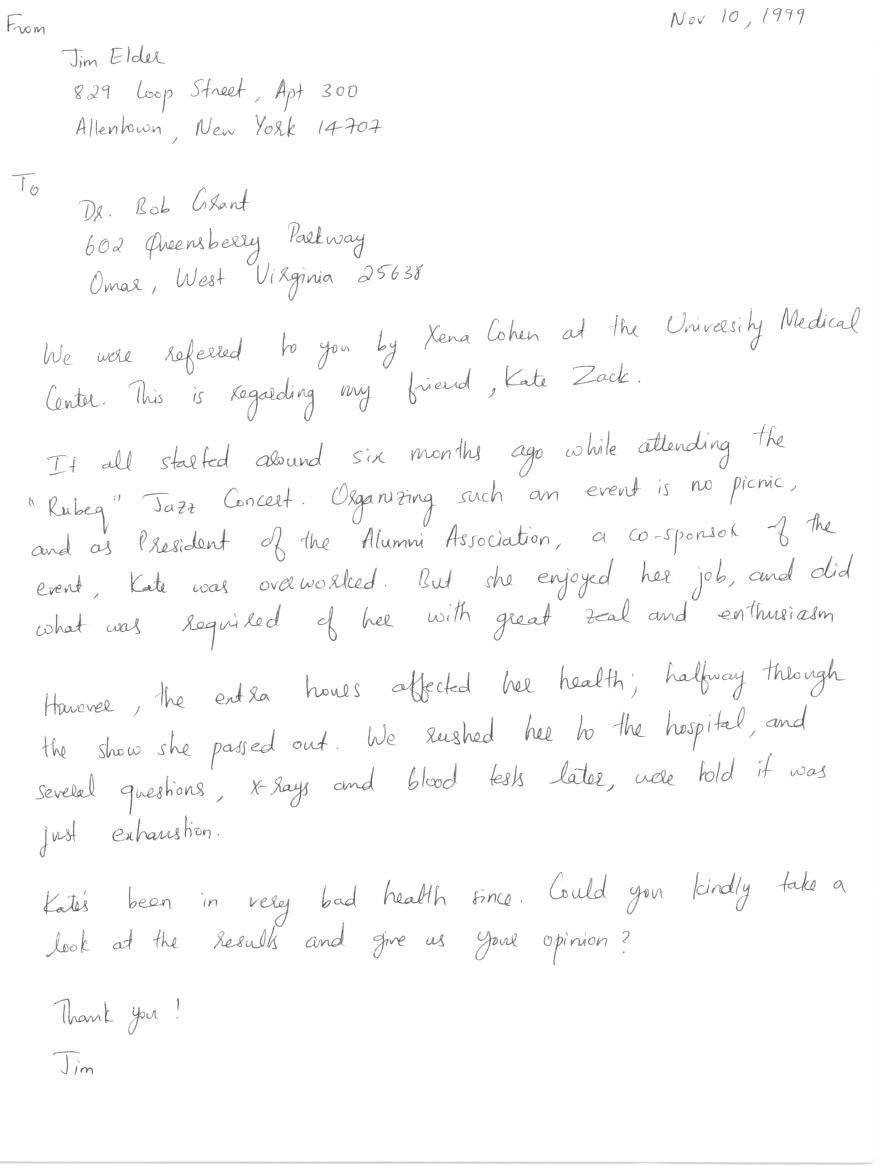
We were referred to you by Xena Cohen at the University Medical Center. This is regarding my friend, Kate Zack.

It all started around six months ago while attending the "Rubeq" Jazz Concert. Organizing such an event is no picnic, and as President of the Alumni Association, a co-sponsor of the event, Kate was overworked. But she enjoyed her job, and did what was required of her with great zeal and enthusiasm.

However, the extra hours affected her health; halfway through the show she passed out. We rushed her to the hospital, and several questions, x-rays and blood test later, were told it was just exhaustion.

Kate's been in very bad health since. Could you kindly take a look at the results and give us your opinion?

Thank you!
Jim [25]


From

Jim Elder
829 Loop Street, Apt 300
Allentown, New York 14707

Nov 10, 1999

To

Dr. Bob Grant
602 Queensberry Parkway
Omar, West Virginia 25638

We were refered to you by Xena Cohen at the University Medical Center. This is regarding my friend, Kate Zack.

It all started around six months ago while attending the "Rubeq" Jazz Concert. Organizing such an event is no picnic, and as President of the Alumni Association, a co-sponsor of the event, Kate was overworked. But she enjoyed her job, and did what was required of her with great zeal and enthusiasm

However, the extra hours affected her health; halfway through the show she passed out. We rushed her to the hospital, and several questions, X-Rays and blood tests later, were told it was just exhaustion.

Kate's been in very bad health since. Could you kindly take a look at the results and give us your opinion?

Thank you!

Jim

Figure B.1: This figure shows, of the CEDAR data set, a handwritten sample of the text (of the document) provided by a writer.

Appendix C

Examples of the different kinds of shapes of "r"

Figures C.1 to C.5 below show examples out of the CEDAR data set of the different kinds of shapes of "r" (so examples of $\{x_2^1, x_2^2 \ldots, x_2^5\}$, because x_2^6 is the case if the feature is "NSP").

Figure C.1: This figure shows an example of feature x_2^1 (Cursive with loop) which is found on document 1 of writer 1.

Figure C.2: This figure shows an example of feature x_2^2 (Cursive without loop) which is found on document 1 of writer 3.

Figure C.3: This figure shows an example of feature x_2^3 (Cursive without horizontal piece) which is found on document 1 of writer 7.

Figure C.4: This figure shows an example of feature x_2^4 (In print (one stroke)) which is found on document 1 of writer 2.

Figure C.5: This figure shows an example of feature x_2^5 (In print (two strokes)) which is found on document 1 of writer 6.

Appendix D

Python code used in chapter 3

D.1 Python code used for the creation of the user interface

The following Python code was used for the creation of the user interface (with the option menus) as described in section 3.5.

```
from tkinter import *
import pandas as pd

#Create and save Excel sheet

writer = pd.ExcelWriter('Char_Vect.xlsx', engine='xlsxwriter')
writer.save()

#Create first column with the characteristics as rows

df = pd.DataFrame({'Characteristic': ["x1","x2","x3", "x4", "x5",
                                      "x6","x7","x8"]})

#Define Option Menu

class app:
    def __init__(self, root):

        #Define the different options of all the features

        OPTIONSX1=["'e' even with 'r'", "'e' shorter than 'r'",
                   "'e' taller than 'r'", "NSP"]
        OPTIONSX2=["Cursive with loop","Cursive without loop",
                   "Cursive without horizontal piece","In print (
                                                        one stroke)",
                   "In print (two strokes)", "NSP"]
        OPTIONSX3 = ["Leans to the right","Leans to the left",
                     "Stands upright", "NSP"]
        OPTIONSX4 = ["'e' is higher than 'r'","'e' is lower than
                                              'r'",
                     "'e' and 'r' lie on same baseline"]
        OPTIONSX5 = ["Open loop","Closed loop", "NSP"]
        OPTIONSX6 = ["Curved up","Curved down", "Not curved", "
                                             NSP"]
        OPTIONSX7 = ["Leans to the right","Leans to the left",
                     "Stands upright", "NSP"]
        OPTIONSX8 = ["No space between 'e' and 'r'",
```

```
                    "Small space between 'e' and 'r'",
                    "Medium space between 'e' and 'r'",
                    "Large space between 'e' and 'r'", "NSP"]

        win1 = Frame(root)
        win1.grid(row=0,column=0)

        #Create option menu for feature 1

        self.variable1 = StringVar(win1)
        self.variable1.set("'e' even with 'r'")
        self.x1 = OptionMenu(win1, self.variable1,
                        *OPTIONSX1,
                        command = self.varMenu)
        l1 = Label(win1,  text="Height Relationship of 'e' to 'r'
                                        (x1)", width=35 )
        l1.grid(row=5,column=1)
        self.x1.grid(row=5,column=2)

        #Create option menu for feature 2

        self.variable2 = StringVar(win1)
        self.variable2.set("Cursive with loop")
        self.x2 = OptionMenu(win1,
                        self.variable2, *OPTIONSX2)
        l2 = Label(win1,  text="Shape of 'r' (x2)", width=35 )
        l2.grid(row=6,column=1)
        self.x2.grid(row=6,column=2)

        #Create option menu for feature 3

        self.variable3 = StringVar(win1)
        self.variable3.set("Leans to the right")
        self.x3 = OptionMenu(win1,
                        self.variable3, *OPTIONSX3)
        l3 = Label(win1,  text="Angle of 'r' (x3)", width=35 )
        l3.grid(row=7,column=1)
        self.x3.grid(row=7,column=2)

        #Create option menu for feature 4

        self.variable4 = StringVar(win1)
        self.variable4.set("'e' is higher than 'r'")
        self.x4 = OptionMenu(win1,
                        self.variable4, *OPTIONSX4)
        l4 = Label(win1,  text="Position of 'e' relative to 'r' (
                                        alignment) (x4)",
                   width=35 )
        l4.grid(row=8,column=1)
        self.x4.grid(row=8,column=2)

        #Create option menu for feature 5

        self.variable5 = StringVar(win1)
        self.variable5.set("Open loop")
        self.x5 = OptionMenu(win1,
```

```
89                              self.variable5, *OPTIONSX5)
90          l5 = Label(win1,  text="Closed or open loop of 'e' (x5)",
                                          width=35 )
91          l5.grid(row=9,column=1)
92          self.x5.grid(row=9,column=2)
93
94          #Create option menu for feature 6
95
96          self.variable6 = StringVar(win1)
97          self.variable6.set("Curved up")
98          self.x6 = OptionMenu(win1,
99                              self.variable6, *OPTIONSX6)
100         l6 = Label(win1,  text="Shape of the loop of 'e' (x6)",
                                         width=35 )
101         l6.grid(row=10,column=1)
102         self.x6.grid(row=10,column=2)
103
104         #Create option menu for feature 7
105
106         self.variable7 = StringVar(win1)
107         self.variable7.set("Leans to the right")
108         self.x7 = OptionMenu(win1,
109                             self.variable7, *OPTIONSX7)
110         l7 = Label(win1,  text="Angle of 'e' (x7)", width=35 )
111         l7.grid(row=11,column=1)
112         self.x7.grid(row=11,column=2)
113
114         #Create option menu for feature 8
115
116         self.variable8 = StringVar(win1)
117         self.variable8.set("No space between 'e' and 'r'")
118         self.x8 = OptionMenu(win1,
119                             self.variable8, *OPTIONSX8)
120         l8 = Label(win1,  text="Space between the 'e' and 'r' (x8
                                         )", width=35 )
121         l8.grid(row=12,column=1)
122         self.x8.grid(row=12,column=2)
123
124     #This part needs to be added if there's two or more option
                                       menus
125
126     def varMenu(self, selection):
127         if selection == "Heavy":
128             self.variable2.set("colour")
129             self.x2.config(state = DISABLED)
130         else:
131             self.variable2.set("Cursive with loop")
132             self.x2.config(state = NORMAL)
133
134 #For loop for every writer (1-800) and every document for every
                                    writer (1-3)
135 #+ Create x_i binary vector for i=1,...,8 for every document of
                                    every writer
136
137 for j in range(0,800):
138     for d in range(1,4):
```

```
139
140         #Display option menu for every document of every writer
141
142         root = Tk()
143         a = app(root)
144         root.mainloop()
145
146         #Create vectors for every characteristic with the
                                        different features
147
148         Vector_X1=["'e' even with 'r'", "'e' shorter than 'r'",
149                    "'e' taller than 'r'", "NSP"]
150         Vector_X2=["Cursive with loop","Cursive without loop",
151                    "Cursive without horizontal piece","In print (
                                                  one stroke)",
152                    "In print (two strokes)", "NSP"]
153         Vector_X3=["Leans to the right","Leans to the left",
154                      "Stands upright", "NSP"]
155         Vector_X4=["'e' is higher than 'r'","'e' is lower than 'r
                                        '",
156                      "'e' and 'r' lie on same baseline", "NSP"]
157         Vector_X5=["Open loop","Closed loop", "NSP"]
158         Vector_X6=["Curved up","Curved down", "Not curved", "NSP"
                                        ]
159         Vector_X7=["Leans to the right","Leans to the left",
160                      "Stands upright", "NSP"]
161         Vector_X8=["No space between 'e' and 'r'",
162                      "Small space between 'e' and 'r'",
163                      "Medium space between 'e' and 'r'",
164                      "Large space between 'e' and 'r'", "NSP"]
165
166         Vector_X1_bin=[]
167         Vector_X2_bin=[]
168         Vector_X3_bin=[]
169         Vector_X4_bin=[]
170         Vector_X5_bin=[]
171         Vector_X6_bin=[]
172         Vector_X7_bin=[]
173         Vector_X8_bin=[]
174
175         #Transform the vectors of every characteristic into
                                        binary vectors
176         #Where a 1 means the image has that feature and 0 means
                                        that it doesn't
177         #have that feature
178
179         for i in Vector_X1:
180             if i==a.variable1.get():
181                 Vector_X1_bin.append(1)
182             else:
183                 Vector_X1_bin.append(0)
184
185         for i in Vector_X2:
186             if i==a.variable2.get():
187                 Vector_X2_bin.append(1)
188             else:
```

```
                    Vector_X2_bin.append(0)

            for i in Vector_X3:
                if i==a.variable3.get():
                    Vector_X3_bin.append(1)
                else:
                    Vector_X3_bin.append(0)

            for i in Vector_X4:
                if i==a.variable4.get():
                    Vector_X4_bin.append(1)
                else:
                    Vector_X4_bin.append(0)

            for i in Vector_X5:
                if i==a.variable5.get():
                    Vector_X5_bin.append(1)
                else:
                    Vector_X5_bin.append(0)

            for i in Vector_X6:
                if i==a.variable6.get():
                    Vector_X6_bin.append(1)
                else:
                    Vector_X6_bin.append(0)

            for i in Vector_X7:
                if i==a.variable7.get():
                    Vector_X7_bin.append(1)
                else:
                    Vector_X7_bin.append(0)

            for i in Vector_X8:
                if i==a.variable8.get():
                    Vector_X8_bin.append(1)
                else:
                    Vector_X8_bin.append(0)

            #Create the column name for every document of every writer with
            #W (writer number), D (document number) and put the binary vectors
            #of the characteristics in the Excel file

            df['W '+str(j+1)+", D "+str(d)]=[Vector_X1_bin,
                            Vector_X2_bin,Vector_X3_bin,
                                Vector_X4_bin,Vector_X5_bin,
                                Vector_X6_bin, Vector_X7_bin,
                                Vector_X8_bin]

#Save the Excel file with the binary vectors

writer = pd.ExcelWriter('Char_Vect.xlsx', engine='xlsxwriter')
df.to_excel(writer, sheet_name='Sheet1', index=False)
```

D.2 Python code used for displaying the bigrams

The following Python code was used in order to display the bigrams as described in section 3.5.

```
import numpy as np
import pylab as pl
import matplotlib.cm as cm
from PIL import Image

#This definition puts the image on screen with a title (the input is the picture
#itself and a title (string))

def putimageonscreen(mypicture,title):
    pl.imshow(mypicture,cmap=cm.gray)
    pl.title(title)
    pl.show()

#For every document (1-3) of every writer (1-800), the image is opened (called
#(writer number)_(document number).jpg), the image is converted into an array and
#The image is put on screen (with the definition above) (the title is
#Writer (writer number), Document (document number))

for j in range(0,800):
    for d in range(1,4):
        img=Image.open(str(j+1)+"_"+str(d) + ".jpg")
        img_array=np.asarray(img)
        putimageonscreen(img_array,"Writer "+str(j+1)+",Document "+str(d))
```

Appendix E

R code used in chapter 5

The following R code was used to calculate the scores and to create the graphs in chapter 5.
The code, that was used to find the parametrization of the scores, is the same for every score (and for the same and different source scores). Therefore, comments were only added for the first score (same source). Furthermore, for the same reason, comments were only added for the first "for loop" (of the same source scores).
The code, that was used to create the histogram, SLR and $\log_{10}$(SLR) plots, has been omitted from this report for succinctness, but is available upon request.

```
library("readxl")
library(fitdistrplus)

All_Char <- read_excel("Char_Vect_All.xlsx")
df_All_Char <-data.frame(All_Char) #Make dataframe of Excel
    file

#Score 1: Overlap (w1 and S1)
#Score 2: Goodall3 (w2 and S2)
#Score 3: Burnaby (w3 and S3)
#Score 4: Anderberg (S4)
N=ncol(All_Char)-1 #Number of documents
w1_k=1/8 #Weight of score 1
w2_k=1/8 #Weight of score 2
w3_k=1/8 #Weight of score 3

#For Score 3 and 4
lst_sums<-vector()
n_k_lst<-vector()
for (i in 1:8){
  lst_log<-vector()
  quant=data.frame(apply(df_All_Char,MARGIN=1,table)[i])
  quant_lst=quant[1:nrow(quant),2] #This is f, so how many
      times each characteristic appears in the data set
  for (j in 1:(length(quant_lst-1)-1)){
    p_khat=quant_lst[j]/N
    log_func=2*log(1-p_khat)
    lst_log<-append(lst_log,log_func)}
```

```
27    lst_sums<-append(lst_sums,sum(lst_log)) #Append the 2*log
          part of score 3
28    n_k_lst<-append(n_k_lst,(length(quant_lst-1)-1)) #Append
          n_k of score 4 (so how many different features each
          characteristic has)
29    }
30  Sum_Char_1=sum(data.frame(apply(df_All_Char,MARGIN=1,table
      )[1])[2])-1 #Total number of characteristics
31
32  ############################################################
33  #Same Source Scores
34  #For 1 writer, document 1 (j) is compared with document 2
      (j+1) and 3 (j+2) and document 2 (j+1) is compared to
      document 3 (j+2)
35  S1s<-vector()
36  S2s<-vector()
37  S3s<-vector()
38  S4s<-vector()
39
40  for (j in seq(2, ncol(All_Char)-1, 3)){
41    w1_k_S1s_k_list<-vector()
42    w2_k_S2s_k_list<-vector()
43    w3_k_S3s_k_list<-vector()
44    parts_sum_equal_lst<-vector() #Part of score 4 (summed
          over equal char.)
45    parts_sum_unequal_lst<-vector() #Part of score 4 (summed
          over unequal char.)
46    for (i in 1:8){
47      df2X<-data.frame(apply(df_All_Char,MARGIN=1,table)[i])
48      rowX= which(df2X == df_All_Char[i,j], arr.ind=TRUE)[1]
49      f_kX=df2X[rowX,2] #How many times characteristic X_k
            appears in data set
50      p_khat_2X=f_kX/N
51      n_k=n_k_lst[i]
52      if (All_Char[i,j]==All_Char[i,j+1]){
53        w1_k_S1s_k_list<-append(w1_k_S1s_k_list,w1_k*1) #
              Append score 1
54        df2<-data.frame(apply(df_All_Char,MARGIN=1,table)[i])
55        row= which(df2 == df_All_Char[i,j], arr.ind=TRUE)[1]
56        f_k=df2[row,2] #How many times char X_k appears in
              data set
57        p_k2=(f_k*(f_k-1))/(N*(N-1))
58        w2_k_S2s_k_list<-append(w2_k_S2s_k_list,w2_k*(1-p_k2)
              ) #Append score 2
59        w3_k_S3s_k_list<-append(w3_k_S3s_k_list,w3_k*1) #
              Append score 3
60        one_part_sum_equal_lst=(1/p_khat_2X)^2 * (2/(n_k*(n_
              k+1)))
61        parts_sum_equal_lst<-append(parts_sum_equal_lst,one_
              part_sum_equal_lst) #Append score 4
62      } else {
```

```
63        df2X<-data.frame(apply(df_All_Char,MARGIN=1,table)[i
              ])
64        rowX= which(df2X == df_All_Char[i,j], arr.ind=TRUE)
              [1]
65        f_kX=df2X[rowX,2]
66        p_khat_2X=f_kX/N #How many times char X_k appears in
               data set
67        df2Y<-data.frame(apply(df_All_Char,MARGIN=1,table)[i
              ])
68        rowY= which(df2Y == df_All_Char[i,j+1], arr.ind=TRUE
              )[1]
69        f_kY=df2Y[rowY,2] #How many times char Y_k appears
              in data set
70        p_khat_2Y=f_kY/N
71        log_func_2=log((p_khat_2X*p_khat_2Y)/((1-p_khat_2X)*
              (1-p_khat_2Y)))
72        tot_func=lst_sums[i]/(log_func_2+lst_sums[i])
73        w3_k_S3s_k_list<-append(w3_k_S3s_k_list,w3_k*tot_func
              )
74        df2Y<-data.frame(apply(df_All_Char,MARGIN=1,table)[i
              ])
75        rowY= which(df2Y == df_All_Char[i,j+1], arr.ind=TRUE
              )[1]
76        f_kY=df2Y[rowY,2] #How many times char Y_k appears
              in data set
77        p_khat_2Y=f_kY/N
78        one_part_sum_unequal_lst=(1/(2*p_khat_2X*p_khat_2Y))
               * (2/(n_k*(n_k+1)))
79        parts_sum_unequal_lst<-append(parts_sum_unequal_lst,
              one_part_sum_unequal_lst)
80      }}
81    S1s<-append(S1s,sum(w1_k_S1s_k_list))
82    S2s<-append(S2s,sum(w2_k_S2s_k_list))
83    S3s<-append(S3s,sum(w3_k_S3s_k_list))
84    S4s<-append(S4s,((sum(parts_sum_equal_lst))/(sum(parts_
          sum_equal_lst)+sum(parts_sum_unequal_lst))))
85
86    w1_k_S1s_k_list<-vector()
87    w2_k_S2s_k_list<-vector()
88    w3_k_S3s_k_list<-vector()
89    parts_sum_equal_lst<-vector()
90    parts_sum_unequal_lst<-vector()
91    for (i in 1:8){
92      df2X<-data.frame(apply(df_All_Char,MARGIN=1,table)[i])
93      rowX= which(df2X == df_All_Char[i,j], arr.ind=TRUE)[1]
94      f_kX=df2X[rowX,2]
95      p_khat_2X=f_kX/N
96      n_k=n_k_lst[i]
97      if (All_Char[i,j]==All_Char[i,j+2]){
98        w1_k_S1s_k_list<-append(w1_k_S1s_k_list,w1_k*1)
99        df2<-data.frame(apply(df_All_Char,MARGIN=1,table)[i])
```

```
100         row= which(df2 == df_All_Char[i,j], arr.ind=TRUE)[1]
101         f_k=df2[row,2]
102         p_k2=(f_k*(f_k-1))/(N*(N-1))
103         w2_k_S2s_k_list<-append(w2_k_S2s_k_list,w2_k*(1-p_k2)
              )
104         w3_k_S3s_k_list<-append(w3_k_S3s_k_list,w3_k*1)
105         one_part_sum_equal_lst=(1/p_khat_2X)^2 * (2/(n_k*(n_
              k+1)))
106         parts_sum_equal_lst<-append(parts_sum_equal_lst,one_
              part_sum_equal_lst)
107       } else {
108         df2X<-data.frame(apply(df_All_Char,MARGIN=1,table)[i
              ])
109         rowX= which(df2X == df_All_Char[i,j], arr.ind=TRUE)
              [1]
110         f_kX=df2X[rowX,2]
111         p_khat_2X=f_kX/N
112         df2Y<-data.frame(apply(df_All_Char,MARGIN=1,table)[i
              ])
113         rowY= which(df2Y == df_All_Char[i,j+2], arr.ind=TRUE
              )[1]
114         f_kY=df2Y[rowY,2]
115         p_khat_2Y=f_kY/N
116         log_func_2=log((p_khat_2X*p_khat_2Y)/((1-p_khat_2X)*
              (1-p_khat_2Y)))
117         tot_func=lst_sums[i]/(log_func_2+lst_sums[i])
118         w3_k_S3s_k_list<-append(w3_k_S3s_k_list,w3_k*tot_func
              )
119         df2Y<-data.frame(apply(df_All_Char,MARGIN=1,table)[i
              ])
120         rowY= which(df2Y == df_All_Char[i,j+2], arr.ind=TRUE
              )[1]
121         f_kY=df2Y[rowY,2]
122         p_khat_2Y=f_kY/N
123         one_part_sum_unequal_lst=(1/(2*p_khat_2X*p_khat_2Y))
                * (2/(n_k*(n_k+1)))
124         parts_sum_unequal_lst<-append(parts_sum_unequal_lst,
              one_part_sum_unequal_lst)
125       }}
126     S1s<-append(S1s,sum(w1_k_S1s_k_list))
127     S2s<-append(S2s,sum(w2_k_S2s_k_list))
128     S3s<-append(S3s,sum(w3_k_S3s_k_list))
129     S4s<-append(S4s,((sum(parts_sum_equal_lst))/(sum(parts_
          sum_equal_lst)+sum(parts_sum_unequal_lst))))
130
131     w1_k_S1s_k_list<-vector()
132     w2_k_S2s_k_list<-vector()
133     w3_k_S3s_k_list<-vector()
134     parts_sum_equal_lst<-vector()
135     parts_sum_unequal_lst<-vector()
136     for (i in 1:8){
```

```
137       df2X<-data.frame(apply(df_All_Char,MARGIN=1,table)[i])
138       rowX= which(df2X == df_All_Char[i,j+1], arr.ind=TRUE)
              [1]
139       f_kX=df2X[rowX,2]
140       p_khat_2X=f_kX/N
141       n_k=n_k_lst[i]
142       if (All_Char[i,j+1]==All_Char[i,j+2]){
143         w1_k_S1s_k_list<-append(w1_k_S1s_k_list,w1_k*1)
144         df2<-data.frame(apply(df_All_Char,MARGIN=1,table)[i])
145         row= which(df2 == df_All_Char[i,j], arr.ind=TRUE)[1]
146         f_k=df2[row,2]
147         p_k2=(f_k*(f_k-1))/(N*(N-1))
148         w2_k_S2s_k_list<-append(w2_k_S2s_k_list,w2_k*(1-p_k2)
              )
149         w3_k_S3s_k_list<-append(w3_k_S3s_k_list,w3_k*1)
150         one_part_sum_equal_lst=(1/p_khat_2X)^2 * (2/(n_k*(n_
              k+1)))
151         parts_sum_equal_lst<-append(parts_sum_equal_lst,one_
              part_sum_equal_lst)
152       } else {
153         df2X<-data.frame(apply(df_All_Char,MARGIN=1,table)[i
              ])
154         rowX= which(df2X == df_All_Char[i,j+1], arr.ind=TRUE
              )[1]
155         f_kX=df2X[rowX,2]
156         p_khat_2X=f_kX/N
157         df2Y<-data.frame(apply(df_All_Char,MARGIN=1,table)[i
              ])
158         rowY= which(df2Y == df_All_Char[i,j+2], arr.ind=TRUE
              )[1]
159         f_kY=df2Y[rowY,2]
160         p_khat_2Y=f_kY/N
161         log_func_2=log((p_khat_2X*p_khat_2Y)/((1-p_khat_2X)*
              (1-p_khat_2Y)))
162         tot_func=lst_sums[i]/(log_func_2+lst_sums[i])
163         w3_k_S3s_k_list<-append(w3_k_S3s_k_list,w3_k*tot_func
              )
164         df2Y<-data.frame(apply(df_All_Char,MARGIN=1,table)[i
              ])
165         rowY= which(df2Y == df_All_Char[i,j+2], arr.ind=TRUE
              )[1]
166         f_kY=df2Y[rowY,2]
167         p_khat_2Y=f_kY/N
168         one_part_sum_unequal_lst=(1/(2*p_khat_2X*p_khat_2Y))
                * (2/(n_k*(n_k+1)))
169         parts_sum_unequal_lst<-append(parts_sum_unequal_lst,
              one_part_sum_unequal_lst)
170       }}
171     S1s<-append(S1s,sum(w1_k_S1s_k_list))
172     S2s<-append(S2s,sum(w2_k_S2s_k_list))
173     S3s<-append(S3s,sum(w3_k_S3s_k_list))
```

```
174   S4s<-append(S4s,((sum(parts_sum_equal_lst))/(sum(parts_
          sum_equal_lst)+sum(parts_sum_unequal_lst))))
175 }
176
177 #Different Source Scores
178 #Document 1 (k) of writer 1 is compared to every document
       (j) of every other writer≥1, the same for document 2 (k
       +1) and 3 (k+2). For writer 2, the documents are
       compared to those of writer≥2
179 S1d<-vector()
180 S2d<-vector()
181 S3d<-vector()
182 S4d<-vector()
183
184 for (k in seq(2,ncol(All_Char)-1,3)){
185   for (j in seq(k+3, ncol(All_Char))){
186     w1_k_S1d_k_list<-vector()
187     w2_k_S2d_k_list<-vector()
188     w3_k_S3d_k_list<-vector()
189     parts_sum_equal_lstd<-vector()
190     parts_sum_unequal_lstd<-vector()
191     for (i in 1:8){
192       df2X<-data.frame(apply(df_All_Char,MARGIN=1,table)[i
            ])
193       rowX= which(df2X == df_All_Char[i,k], arr.ind=TRUE)
            [1]
194       f_kX=df2X[rowX,2]
195       p_khat_2X=f_kX/N
196       n_k=n_k_lst[i]
197       if (All_Char[i,k]==All_Char[i,j]){
198         w1_k_S1d_k_list<-append(w1_k_S1d_k_list,w1_k*1)
199         df2<-data.frame(apply(df_All_Char,MARGIN=1,table)[i
              ])
200         row= which(df2 == df_All_Char[i,j], arr.ind=TRUE)
              [1]
201         f_k=df2[row,2]
202         p_k2=(f_k*(f_k-1))/(N*(N-1))
203         w2_k_S2d_k_list<-append(w2_k_S2d_k_list,w2_k*(1-p_
              k2))
204         w3_k_S3d_k_list<-append(w3_k_S3d_k_list,w3_k*1)
205         one_part_sum_equal_lstd=(1/p_khat_2X)^2 * (2/(n_k*
              (n_k+1)))
206         parts_sum_equal_lstd<-append(parts_sum_equal_lstd,
              one_part_sum_equal_lstd)
207       } else {
208         df2X<-data.frame(apply(df_All_Char,MARGIN=1,table)[
              i])
209         rowX= which(df2X == df_All_Char[i,k], arr.ind=TRUE
              )[1]
210         f_kX=df2X[rowX,2]
211         p_khat_2X=f_kX/N
```

```
212         df2Y<-data.frame(apply(df_All_Char,MARGIN=1,table)[
                i])
213         rowY= which(df2Y == df_All_Char[i,j], arr.ind=TRUE
                )[1]
214         f_kY=df2Y[rowY,2]
215         p_khat_2Y=f_kY/N
216         log_func_2=log((p_khat_2X*p_khat_2Y)/((1-p_khat_2X
                )*(1-p_khat_2Y)))
217         tot_func=lst_sums[i]/(log_func_2+lst_sums[i])
218         w3_k_S3d_k_list<-append(w3_k_S3d_k_list,w3_k*tot_
                func)
219         df2Y<-data.frame(apply(df_All_Char,MARGIN=1,table)[
                i])
220         rowY= which(df2Y == df_All_Char[i,j], arr.ind=TRUE
                )[1]
221         f_kY=df2Y[rowY,2]
222         p_khat_2Y=f_kY/N
223         one_part_sum_unequal_lstd=(1/(2*p_khat_2X*p_khat_2
                Y)) * (2/(n_k*(n_k+1)))
224         parts_sum_unequal_lstd<-append(parts_sum_unequal_
                lstd,one_part_sum_unequal_lstd)
225       }}
226     S1d<-append(S1d,sum(w1_k_S1d_k_list))
227     S2d<-append(S2d,sum(w2_k_S2d_k_list))
228     S3d<-append(S3d,sum(w3_k_S3d_k_list))
229     S4d<-append(S4d,((sum(parts_sum_equal_lstd))/(sum(parts
          _sum_equal_lstd)+sum(parts_sum_unequal_lstd))))
230
231     w1_k_S1d_k_list<-vector()
232     w2_k_S2d_k_list<-vector()
233     w3_k_S3d_k_list<-vector()
234     parts_sum_equal_lstd<-vector()
235     parts_sum_unequal_lstd<-vector()
236     for (i in 1:8){
237       df2X<-data.frame(apply(df_All_Char,MARGIN=1,table)[i
              ])
238       rowX= which(df2X == df_All_Char[i,k+1], arr.ind=TRUE
              )[1]
239       f_kX=df2X[rowX,2]
240       p_khat_2X=f_kX/N
241       n_k=n_k_lst[i]
242       if (All_Char[i,k+1]==All_Char[i,j]){
243         w1_k_S1d_k_list<-append(w1_k_S1d_k_list,w1_k*1)
244         df2<-data.frame(apply(df_All_Char,MARGIN=1,table)[i
                ])
245         row= which(df2 == df_All_Char[i,j], arr.ind=TRUE)
                [1]
246         f_k=df2[row,2]
247         p_k2=(f_k*(f_k-1))/(N*(N-1))
248         w2_k_S2d_k_list<-append(w2_k_S2d_k_list,w2_k*(1-p_
                k2))
```

```
249         w3_k_S3d_k_list<-append(w3_k_S3d_k_list,w3_k*1)
250         one_part_sum_equal_lstd=(1/p_khat_2X)^2 * (2/(n_k*
                (n_k+1)))
251         parts_sum_equal_lstd<-append(parts_sum_equal_lstd,
                one_part_sum_equal_lstd)
252       } else {
253         df2X<-data.frame(apply(df_All_Char,MARGIN=1,table)[
                i])
254         rowX= which(df2X == df_All_Char[i,k+1], arr.ind=
                TRUE)[1]
255         f_kX=df2X[rowX,2]
256         p_khat_2X=f_kX/N
257         df2Y<-data.frame(apply(df_All_Char,MARGIN=1,table)[
                i])
258         rowY= which(df2Y == df_All_Char[i,j], arr.ind=TRUE
                )[1]
259         f_kY=df2Y[rowY,2]
260         p_khat_2Y=f_kY/N
261         log_func_2=log((p_khat_2X*p_khat_2Y)/((1-p_khat_2X
                )*(1-p_khat_2Y)))
262         tot_func=lst_sums[i]/(log_func_2+lst_sums[i])
263         w3_k_S3d_k_list<-append(w3_k_S3d_k_list,w3_k*tot_
                func)
264         df2Y<-data.frame(apply(df_All_Char,MARGIN=1,table)[
                i])
265         rowY= which(df2Y == df_All_Char[i,j], arr.ind=TRUE
                )[1]
266         f_kY=df2Y[rowY,2]
267         p_khat_2Y=f_kY/N
268         one_part_sum_unequal_lstd=(1/(2*p_khat_2X*p_khat_2
                Y)) * (2/(n_k*(n_k+1)))
269         parts_sum_unequal_lstd<-append(parts_sum_unequal_
                lstd,one_part_sum_unequal_lstd)
270       }}
271     S1d<-append(S1d,sum(w1_k_S1d_k_list))
272     S2d<-append(S2d,sum(w2_k_S2d_k_list))
273     S3d<-append(S3d,sum(w3_k_S3d_k_list))
274     S4d<-append(S4d,((sum(parts_sum_equal_lstd))/(sum(parts
          _sum_equal_lstd)+sum(parts_sum_unequal_lstd))))
275
276     w1_k_S1d_k_list<-vector()
277     w2_k_S2d_k_list<-vector()
278     w3_k_S3d_k_list<-vector()
279     parts_sum_equal_lstd<-vector()
280     parts_sum_unequal_lstd<-vector()
281     for (i in 1:8){
282       df2X<-data.frame(apply(df_All_Char,MARGIN=1,table)[i
              ])
283       rowX= which(df2X == df_All_Char[i,k+2], arr.ind=TRUE
              )[1]
284       f_kX=df2X[rowX,2]
```

```
285         p_khat_2X=f_kX/N
286         n_k=n_k_lst[i]
287         if (All_Char[i,k+2]==All_Char[i,j]){
288           w1_k_S1d_k_list<-append(w1_k_S1d_k_list,w1_k*1)
289           df2<-data.frame(apply(df_All_Char,MARGIN=1,table)[i
                ])
290           row= which(df2 == df_All_Char[i,j], arr.ind=TRUE)
                [1]
291           f_k=df2[row,2]
292           p_k2=(f_k*(f_k-1))/(N*(N-1))
293           w2_k_S2d_k_list<-append(w2_k_S2d_k_list,w2_k*(1-p_
                k2))
294           w3_k_S3d_k_list<-append(w3_k_S3d_k_list,w3_k*1)
295           one_part_sum_equal_lstd=(1/p_khat_2X)^2 * (2/(n_k*
                (n_k+1)))
296           parts_sum_equal_lstd<-append(parts_sum_equal_lstd,
                one_part_sum_equal_lstd)
297         } else {
298           df2X<-data.frame(apply(df_All_Char,MARGIN=1,table)[
                i])
299           rowX= which(df2X == df_All_Char[i,k+2], arr.ind=
                TRUE)[1]
300           f_kX=df2X[rowX,2]
301           p_khat_2X=f_kX/N
302           df2Y<-data.frame(apply(df_All_Char,MARGIN=1,table)[
                i])
303           rowY= which(df2Y == df_All_Char[i,j], arr.ind=TRUE
                )[1]
304           f_kY=df2Y[rowY,2]
305           p_khat_2Y=f_kY/N
306           log_func_2=log((p_khat_2X*p_khat_2Y)/((1-p_khat_2X
                )*(1-p_khat_2Y)))
307           tot_func=lst_sums[i]/(log_func_2+lst_sums[i])
308           w3_k_S3d_k_list<-append(w3_k_S3d_k_list,w3_k*tot_
                func)
309           df2Y<-data.frame(apply(df_All_Char,MARGIN=1,table)[
                i])
310           rowY= which(df2Y == df_All_Char[i,j], arr.ind=TRUE
                )[1]
311           f_kY=df2Y[rowY,2]
312           p_khat_2Y=f_kY/N
313           one_part_sum_unequal_lstd=(1/(2*p_khat_2X*p_khat_2
                Y)) * (2/(n_k*(n_k+1)))
314           parts_sum_unequal_lstd<-append(parts_sum_unequal_
                lstd,one_part_sum_unequal_lstd)
315         }}
316       S1d<-append(S1d,sum(w1_k_S1d_k_list))
317       S2d<-append(S2d,sum(w2_k_S2d_k_list))
318       S3d<-append(S3d,sum(w3_k_S3d_k_list))
319       S4d<-append(S4d,((sum(parts_sum_equal_lstd))/(sum(parts
            _sum_equal_lstd)+sum(parts_sum_unequal_lstd))))}}
```

```
320
321 ############################################################
322 #Finding Distributions
323 #Same Source
324 descdist(S1s, discrete = FALSE) #Create Cullen and Frey
      graph
325 S1s<-replace(S1s, S1s==1, 0.999)
326 fit.betaS1s <- fitdist(S1s, "beta") #Fit beta distr. (does
      not work on 0 and 1 so these values are replaced by
      0.001 and 0.999 respectively (the 0 is also replaced
      for the weibull, gamma and lnorm distributions))
327 S1s<-replace(S1s, S1s==0.999, 1)
328 fit.gammaS1s <- fitdist(S1s, "gamma")
329 fit.weibullS1s <- fitdist(S1s, "weibull")
330 fit.lognormalS1s <- fitdist(S1s, "lnorm")
331 plot(fit.betaS1s) #Plot beta distribution (Empirical and
      theoretical density, Q-Q plot, Empirical and
      theoretical CDFs and P-P plot)
332 plot(fit.gammaS1s)
333 plot(fit.weibullS1s)
334 plot(fit.lognormalS1s)
335
336 descdist(S2s, discrete = FALSE)
337 fit.normS2s <- fitdist(S2s, "norm")
338 S2s<-replace(S2s, S2s==0, 0.001)
339 S2s<-replace(S2s, S2s==1, 0.999)
340 fit.betaS2s <- fitdist(S2s, "beta")
341 S2s<-replace(S2s, S2s==0.999, 1)
342 fit.weibullS2s <- fitdist(S2s, "weibull")
343 fit.gammaS2s <- fitdist(S2s, "gamma")
344 fit.lognormalS2s <- fitdist(S2s, "lnorm")
345 S2s<-replace(S2s, S2s==0.001, 0)
346 plot(fit.normS2s)
347 plot(fit.betaS2s)
348 plot(fit.weibullS2s)
349 plot(fit.gammaS2s)
350 plot(fit.lognormalS2s)
351
352 descdist(S3s, discrete = FALSE)
353 S3s<-replace(S3s, S3s==0, 0.001)
354 S3s<-replace(S3s, S3s==1, 0.999)
355 fit.betaS3s <- fitdist(S3s, "beta")
356 S3s<-replace(S3s, S3s==0.999, 1)
357 fit.weibullS3s <- fitdist(S3s, "weibull")
358 fit.gammaS3s <- fitdist(S3s, "gamma")
359 S3s<-replace(S3s, S3s==0.001, 0)
360 plot(fit.betaS3s)
361 plot(fit.weibullS3s)
362 plot(fit.gammaS3s)
363
364 descdist(S4s, discrete = FALSE)
```

```
S4s<-replace(S4s, S4s==0, 0.001)
S4s<-replace(S4s, S4s==1, 0.999)
fit.betaS4s <- fitdist(S4s, "beta")
S4s<-replace(S4s, S4s==0.999, 1)
fit.weibullS4s <- fitdist(S4s, "weibull")
fit.gammaS4s <- fitdist(S4s, "gamma")
S4s<-replace(S4s, S4s==0.001, 0)
plot(fit.betaS4s)
plot(fit.weibullS4s)
plot(fit.gammaS4s)

#Different Source
descdist(S1d, discrete = FALSE)
fit.normS1d <- fitdist(S1d, "norm")
fit.unifS1d <- fitdist(S1d, "unif")
S1d<-replace(S1d, S1d==0, 0.001)
S1d<-replace(S1d, S1d==1, 0.999)
fit.betaS1d <- fitdist(S1d, "beta")
S1d<-replace(S1d, S1d==0.999, 1)
fit.weibullS1d <- fitdist(S1d, "weibull")
fit.gammaS1d <- fitdist(S1d, "gamma")
fit.lognormalS1d <- fitdist(S1d, "lnorm")
S1d<-replace(S1d, S1d==0.001, 0)
plot(fit.normS1d)
plot(fit.unifS1d)
plot(fit.betaS1d)
plot(fit.weibullS1d)
plot(fit.gammaS1d)
plot(fit.lognormalS1d)

descdist(S2d, discrete = FALSE)
fit.normS2d <- fitdist(S2d, "norm")
fit.unifS2d <- fitdist(S2d, "unif")
S2d<-replace(S2d, S2d==0, 0.001)
S2d<-replace(S2d, S2d==1, 0.999)
fit.betaS2d <- fitdist(S2d, "beta")
S2d<-replace(S2d, S2d==0.999, 1)
fit.weibullS2d <- fitdist(S2d, "weibull")
fit.gammaS2d <- fitdist(S2d, "gamma")
fit.lognormalS2d <- fitdist(S2d, "lnorm")
S2d<-replace(S2d, S2d==0.001, 0)
plot(fit.normS2d)
plot(fit.unifS2d)
plot(fit.betaS2d)
plot(fit.weibullS2d)
plot(fit.gammaS2d)
plot(fit.lognormalS2d)

descdist(S3d, discrete = FALSE)
fit.normS3d <- fitdist(S3d, "norm")
fit.unifS3d <- fitdist(S3d, "unif")
```

```
S3d<-replace(S3d, S3d==0, 0.001)
S3d<-replace(S3d, S3d==1, 0.999)
fit.betaS3d <- fitdist(S3d, "beta")
S3d<-replace(S3d, S3d==0.999, 1)
fit.weibullS3d <- fitdist(S3d, "weibull")
fit.gammaS3d <- fitdist(S3d, "gamma")
fit.lognormalS3d <- fitdist(S3d, "lnorm")
S3d<-replace(S3d, S3d==0.001, 0)
plot(fit.normS3d)
plot(fit.unifS3d)
plot(fit.betaS3d)
plot(fit.weibullS3d)
plot(fit.gammaS3d)
plot(fit.lognormalS3d)

descdist(S4d, discrete = FALSE)
fit.normS4d <- fitdist(S4d, "norm")
fit.unifS4d <- fitdist(S4d, "unif")
S4d<-replace(S4d, S4d==0, 0.001)
S4d<-replace(S4d, S4d==1, 0.999)
fit.betaS4d <- fitdist(S4d, "beta")
S4d<-replace(S4d, S4d==0.999, 1)
fit.weibullS4d <- fitdist(S4d, "weibull")
fit.gammaS4d <- fitdist(S4d, "gamma")
fit.lognormalS4d <- fitdist(S4d, "lnorm")
S4d<-replace(S4d, S4d==0.001, 0)
plot(fit.normS4d)
plot(fit.unifS4d)
plot(fit.betaS4d)
plot(fit.weibullS4d)
plot(fit.gammaS4d)
plot(fit.lognormalS4d)
```

Appendix F

Information on the decisions of choosing the distributions in chapter 5

In this appendix it will be explained why the distributions in chapter 5 were chosen for the same and different source scores for each of the four scores. The "R" code, that was used to create the graphs of this appendix, can be found in appendix E.

F.1 Distribution of same source scores (score 1)

Figure F.1 shows the Cullen and Frey graph of the same source scores of score 1. It contains the square skewness and kurtosis of the observed scores and seven theoretical distributions. Using this graph, it was decided that the beta, gamma, Weibull and lognormal distribution could all be the distributions of the observed scores and, therefore, they should be further investigated.

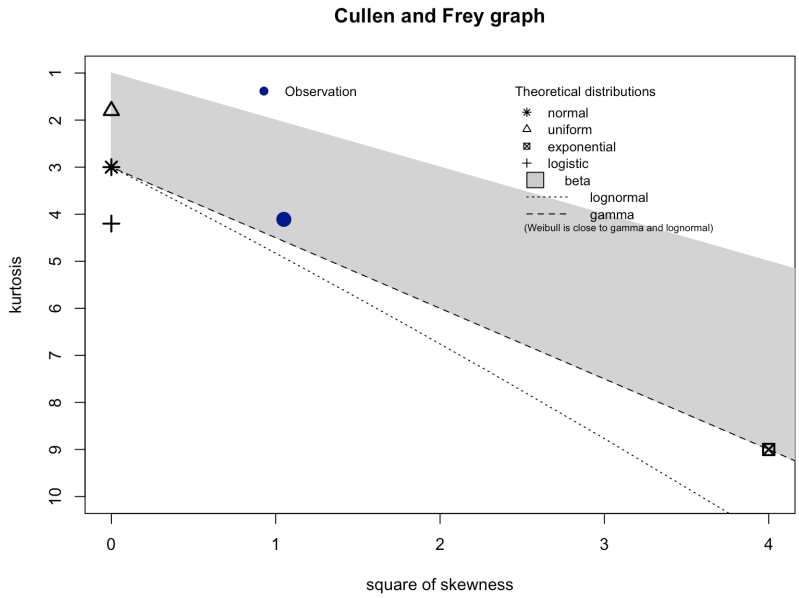


Figure F.1: This figure shows the Cullen and Frey graph of the same source scores of score 1 (Overlap) (it contains the square skewness and kurtosis of the observed scores and seven theoretical distributions).

Figures F.2, F.3, F.4 and F.5 show (of the parametrization with the beta, gamma, Weibull and lognormal distributions (respectively) of the same source scores of score 1) the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots. By mainly looking at the Q-Q plots (and if the theoretical and empirical quantities are on the same line), it can be concluded that the Weibull distribution is the best fit for the same source scores of score 1 (Overlap).

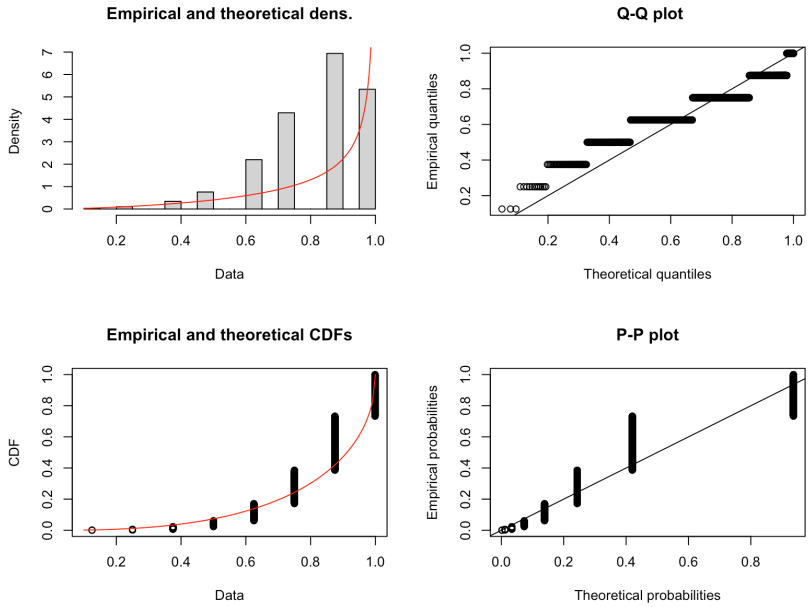


Figure F.2: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **beta** distribution of the same source scores of score 1 (Overlap).

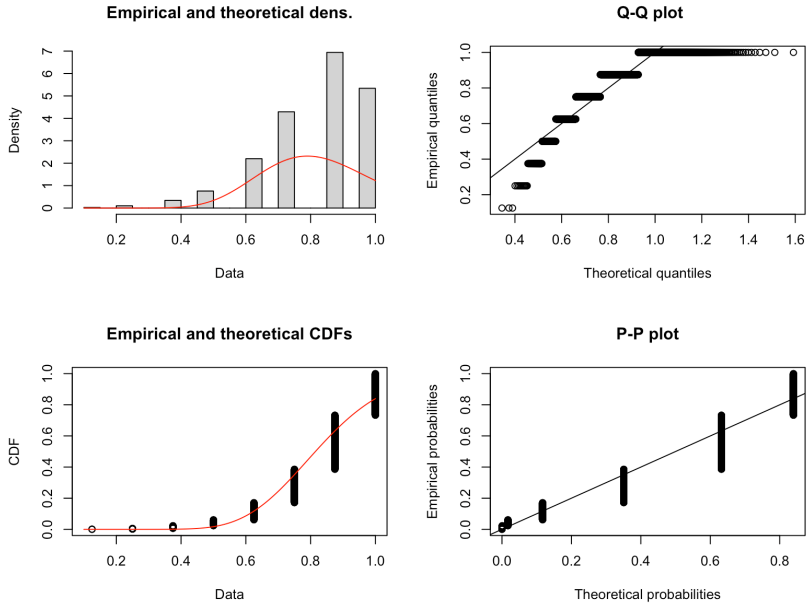


Figure F.3: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **gamma** distribution of the same source scores of score 1 (Overlap).

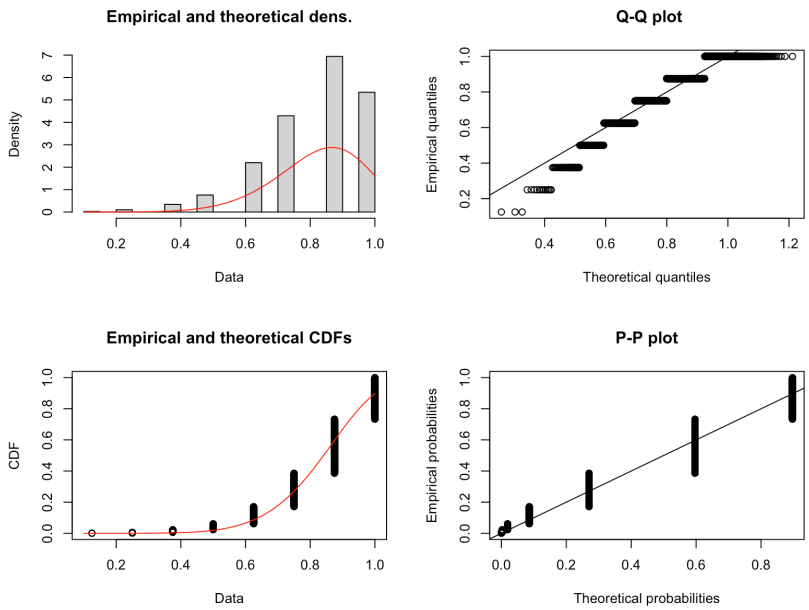


Figure F.4: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **Weibull** distribution of the same source scores of score 1 (Overlap).

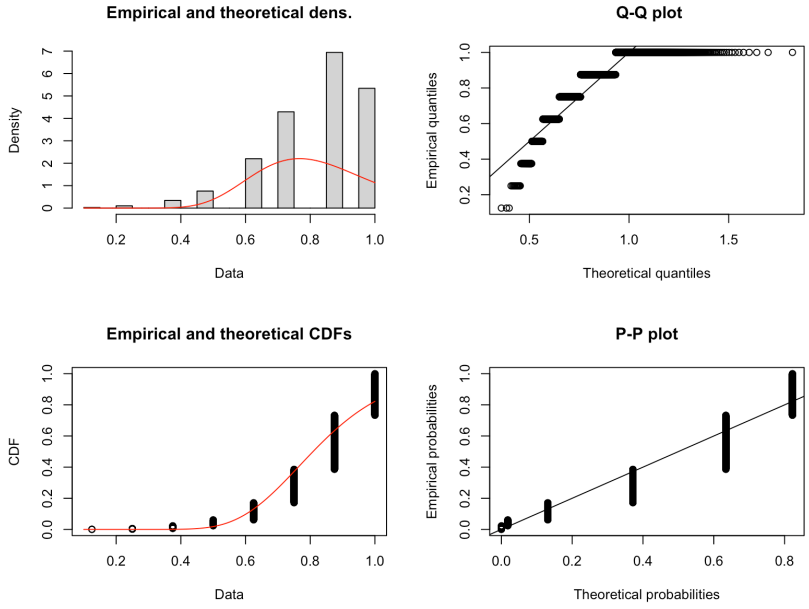


Figure F.5: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **lognormal** distribution of the same source scores of score 1 (Overlap).

F.2 Distribution of different source scores (score 1)

Figure F.6 shows the Cullen and Frey graph of the different source scores of score 1. It contains the square skewness and kurtosis of the observed scores and seven theoretical distributions. Using this graph, it was decided that the normal, uniform, beta, Weibull, gamma and lognormal distribution could all be the distributions of the observed scores and, therefore, they should be further investigated.

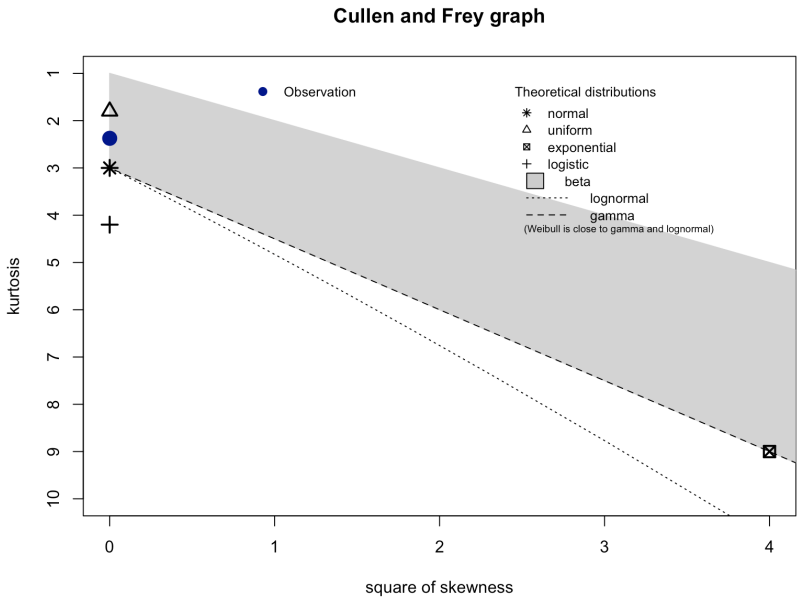


Figure F.6: This figure shows the Cullen and Frey graph of the different source scores of score 1 (Overlap) (it contains the square skewness and kurtosis of the observed scores and seven theoretical distributions).

Figures F.7, F.8, F.9, F.10, F.11 and F.12 show (of the parametrization with the normal, uniform, beta, Weibull, gamma and lognormal distributions (respectively) of the different source scores of score 1) the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots. By mainly looking at the Q-Q plots (and if the theoretical and empirical quantities are on the same line), it can be concluded that the normal distribution is the best fit for the different source scores of score 1 (Overlap).

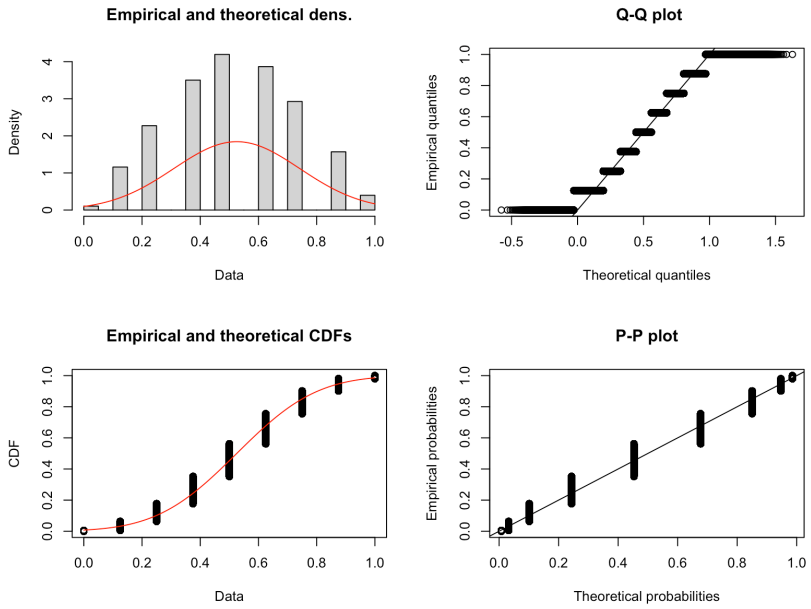


Figure F.7: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **normal** distribution of the different source scores of score 1 (Overlap).

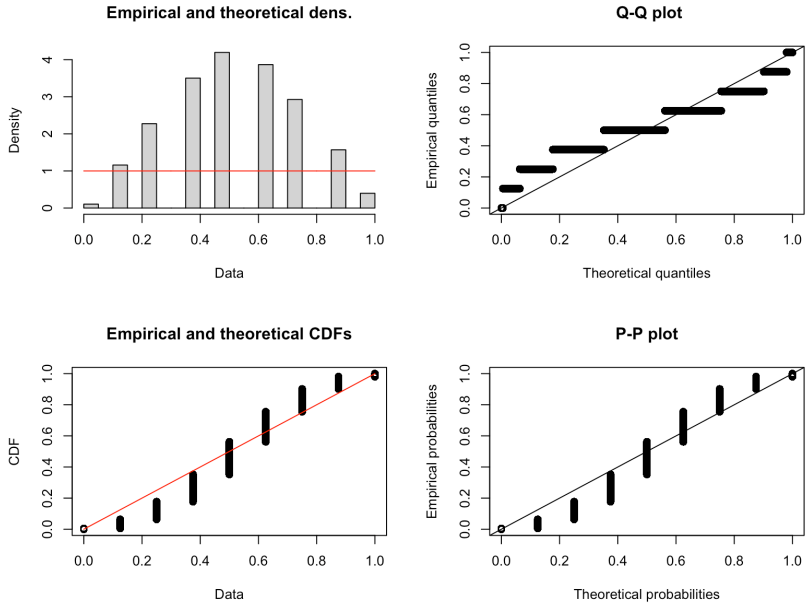


Figure F.8: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **uniform** distribution of the different source scores of score 1 (Overlap).

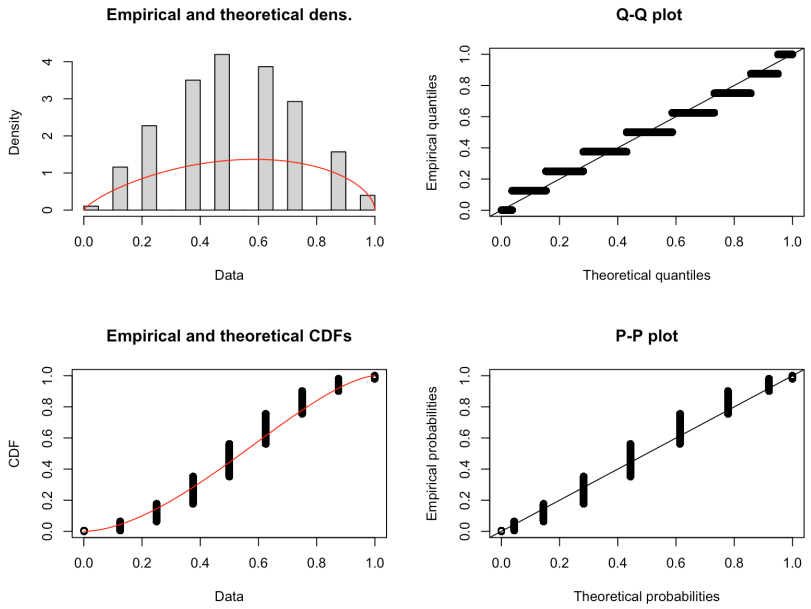


Figure F.9: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **beta** distribution of the different source scores of score 1 (Overlap).

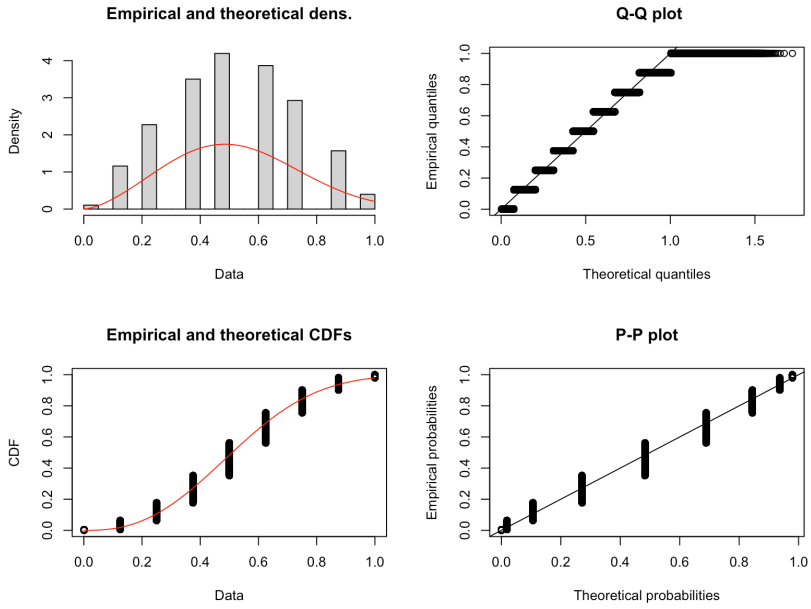


Figure F.10: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **Weibull** distribution of the different source scores of score 1 (Overlap).

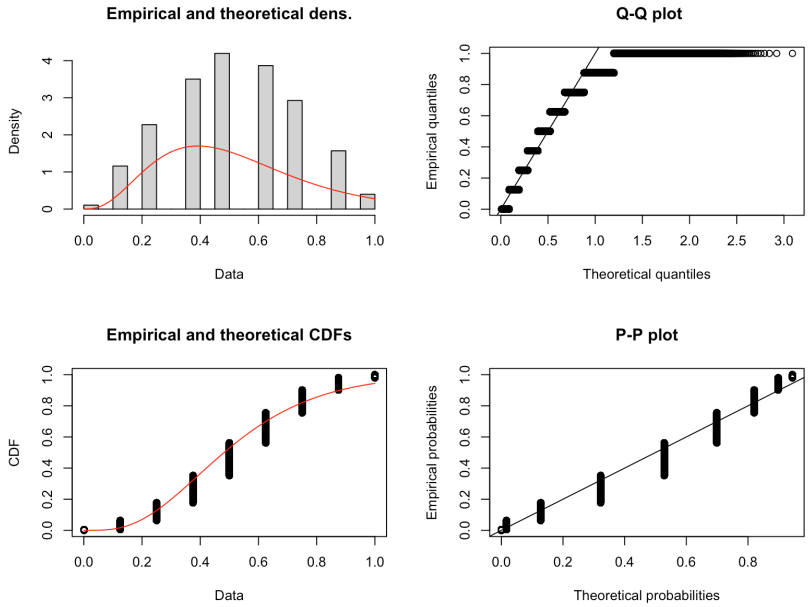


Figure F.11: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **gamma** distribution of the different source scores of score 1 (Overlap).

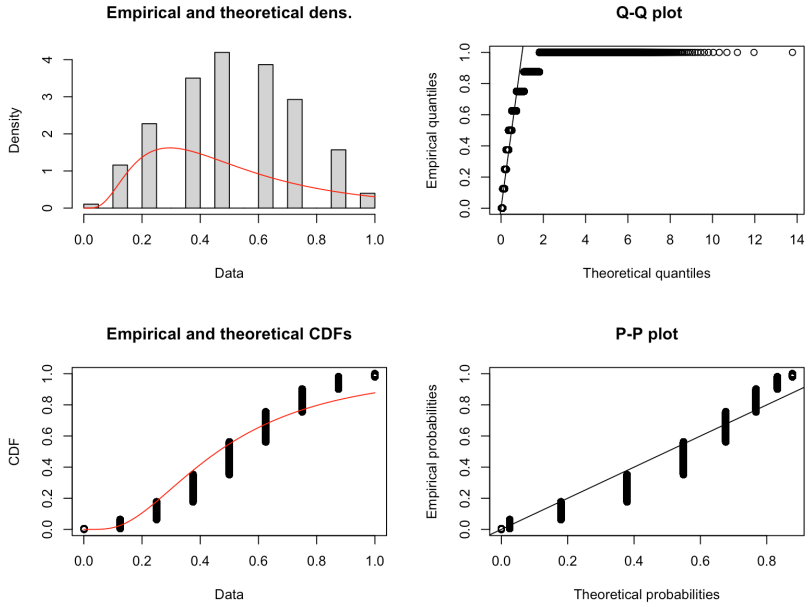


Figure F.12: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **lognormal** distribution of the different source scores of score 1 (Overlap).

F.3 Distribution of same source scores (score 2)

Figure F.13 shows the Cullen and Frey graph of the same source scores of score 2. It contains the square skewness and kurtosis of the observed scores and seven theoretical distributions. Using this graph, it was decided that the normal, beta, Weibull, gamma and lognormal distribution could all be the distributions of the observed scores and, therefore, they should be further investigated.

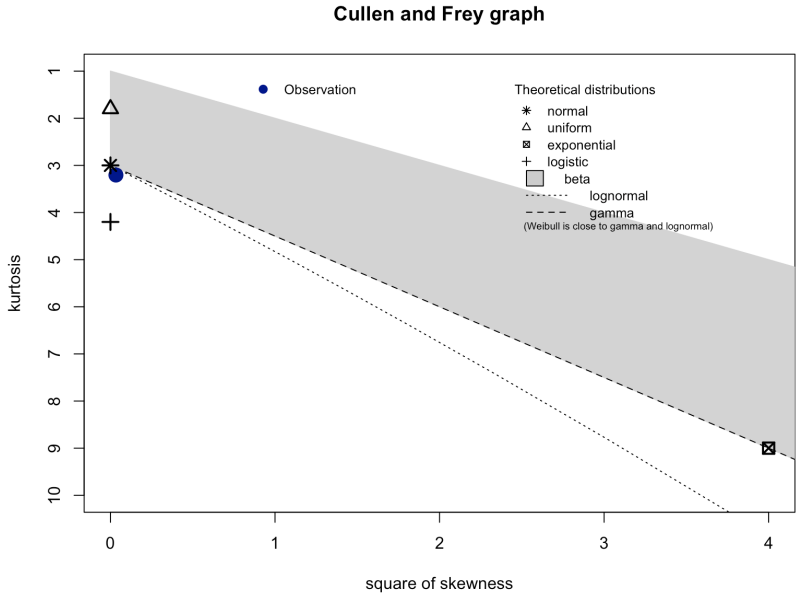


Figure F.13: This figure shows the Cullen and Frey graph of the same source scores of score 2 (Goodall3) (it contains the square skewness and kurtosis of the observed scores and seven theoretical distributions).

Figures F.14, F.15, F.16, F.17 and F.18 show (of the parametrization with the normal, beta, Weibull, gamma and lognormal distributions (respectively) of the same source scores of score 2) the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots. By mainly looking at the Q-Q plots (and if the theoretical and empirical quantities are on the same line), it can be concluded that the normal distribution is the best fit for the same source scores of score 2 (Goodall3).
Note that the Weibull distribution would also be a good fit. However, when looking at the upper right and lower left part of the Q-Q plot, it becomes clear that the normal distribution is a better fit (since the data points are closer to lying on the line).

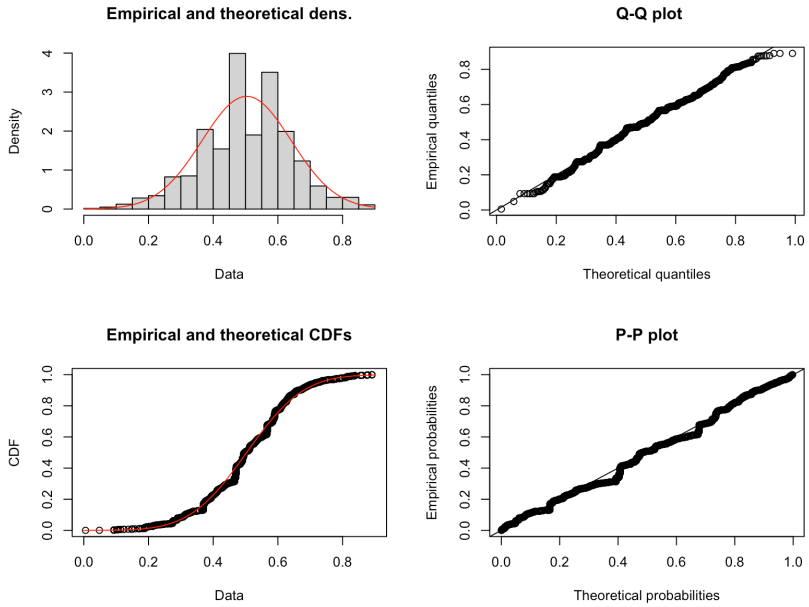


Figure F.14: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **normal** distribution of the same source scores of score 2 (Goodall3).

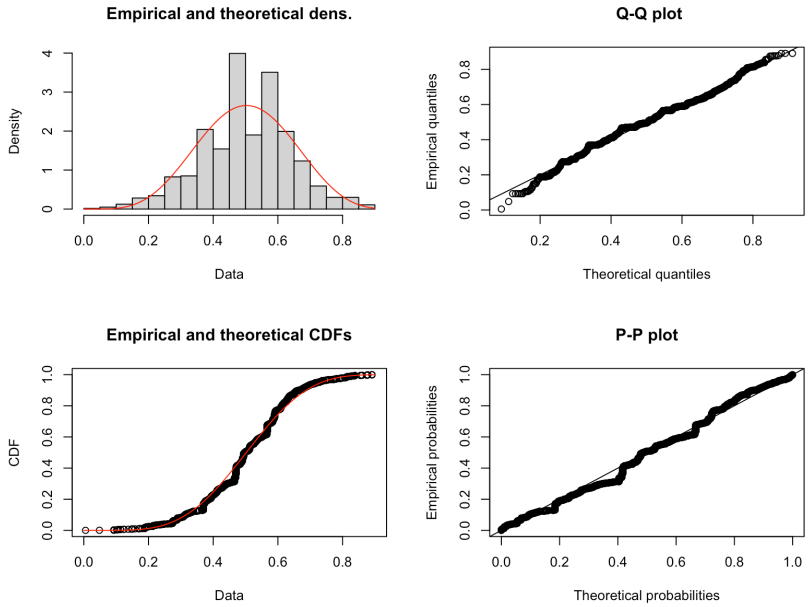


Figure F.15: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **beta** distribution of the same source scores of score 2 (Goodall3).

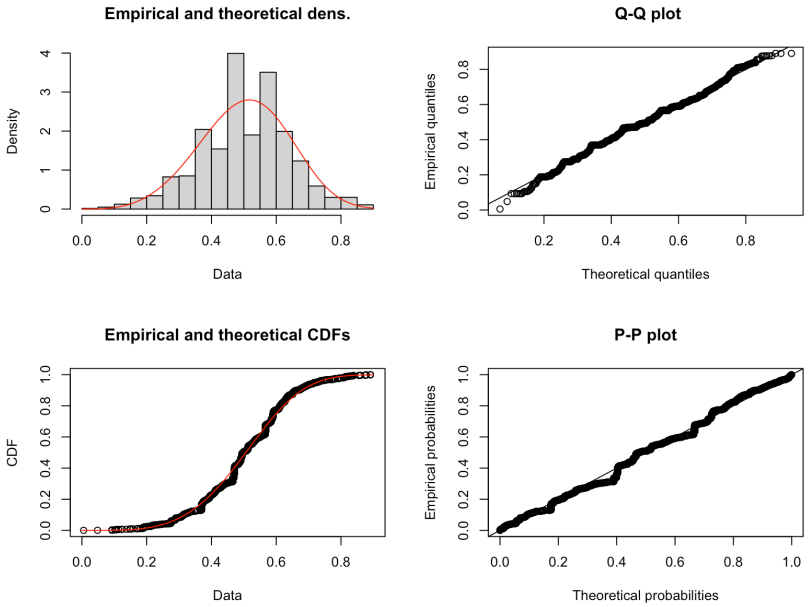


Figure F.16: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **Weibull** distribution of the same source scores of score 2 (Goodall3).

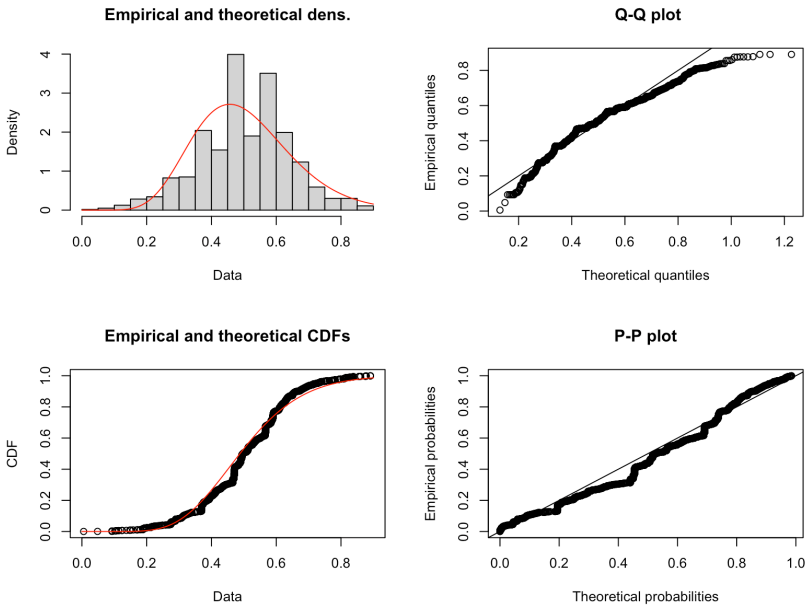


Figure F.17: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **gamma** distribution of the same source scores of score 2 (Goodall3).

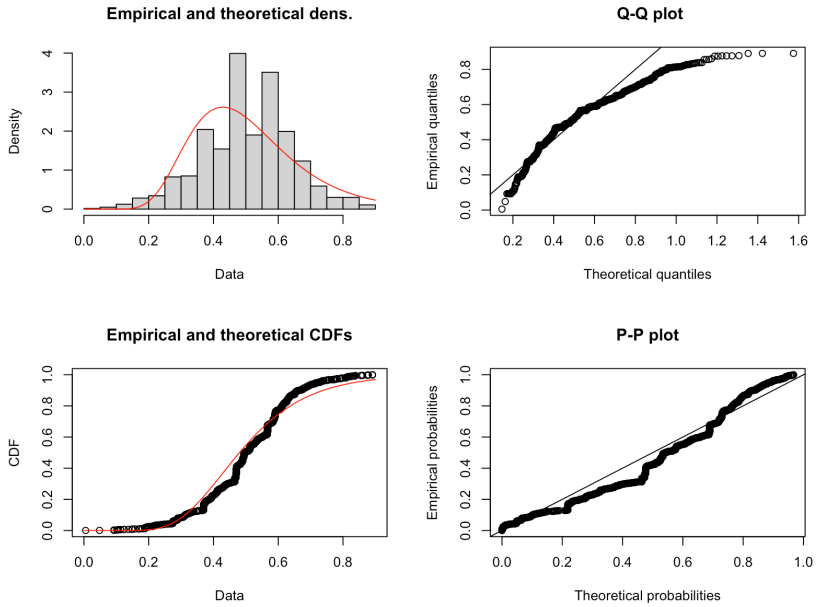


Figure F.18: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **lognormal** distribution of the same source scores of score 2 (Goodall3).

F.4 Distribution of different source scores (score 2)

Figure F.19 shows the Cullen and Frey graph of the different source scores of score 2. It contains the square skewness and kurtosis of the observed scores and seven theoretical distributions. Using this graph, it was decided that the normal, uniform, beta, Weibull, gamma and lognormal distribution could all be the distributions of the observed scores and, therefore, they should be further investigated.

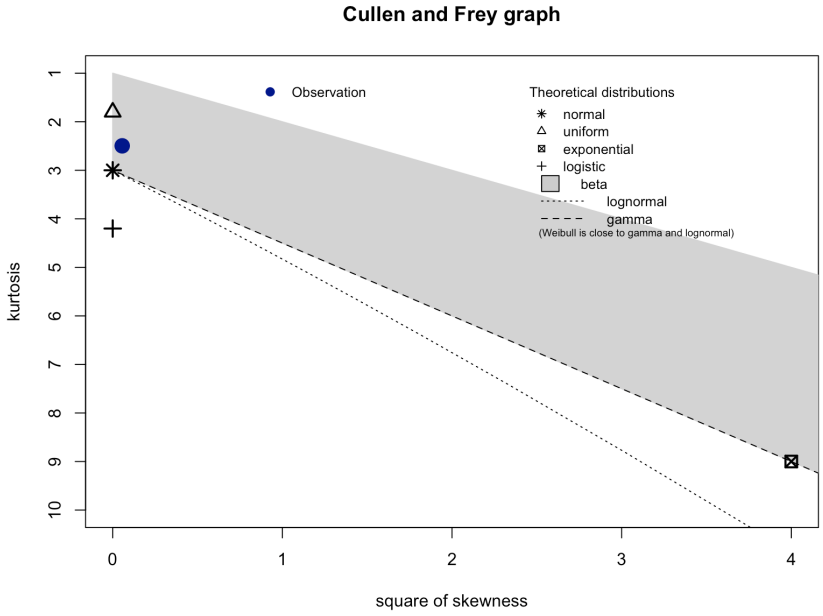


Figure F.19: This figure shows the Cullen and Frey graph of the different source scores of score 2 (Goodall3) (it contains the square skewness and kurtosis of the observed scores and seven theoretical distributions).

Figures F.20, F.21, F.22, F.23, F.24 and F.25 show (of the parametrization with the normal, uniform, beta, Weibull, gamma and lognormal distributions (respectively) of the different source scores of score 1) the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots. By mainly looking at the Q-Q plots (and if the theoretical and empirical quantities are on the same line), it can be concluded that the normal distribution is the best fit for the different source scores of score 2 (Goodall3).

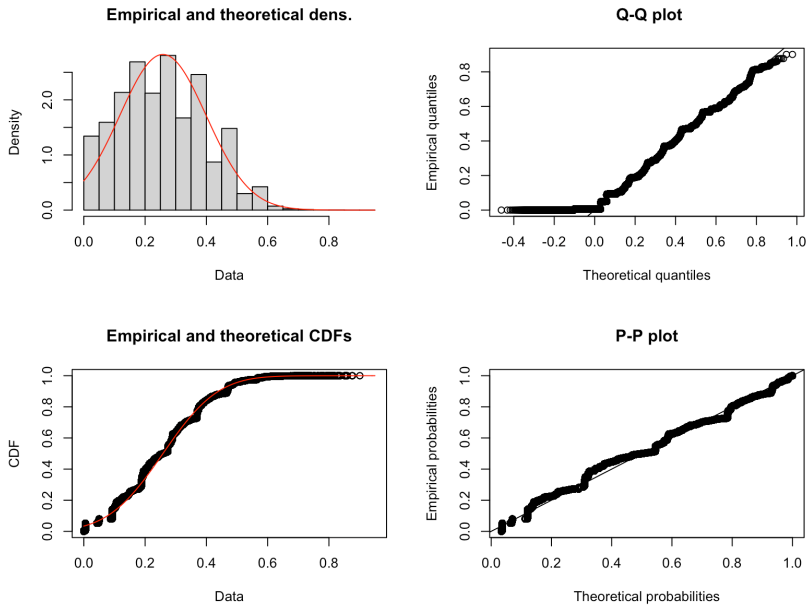


Figure F.20: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **normal** distribution of the different source scores of score 2 (Goodall3).

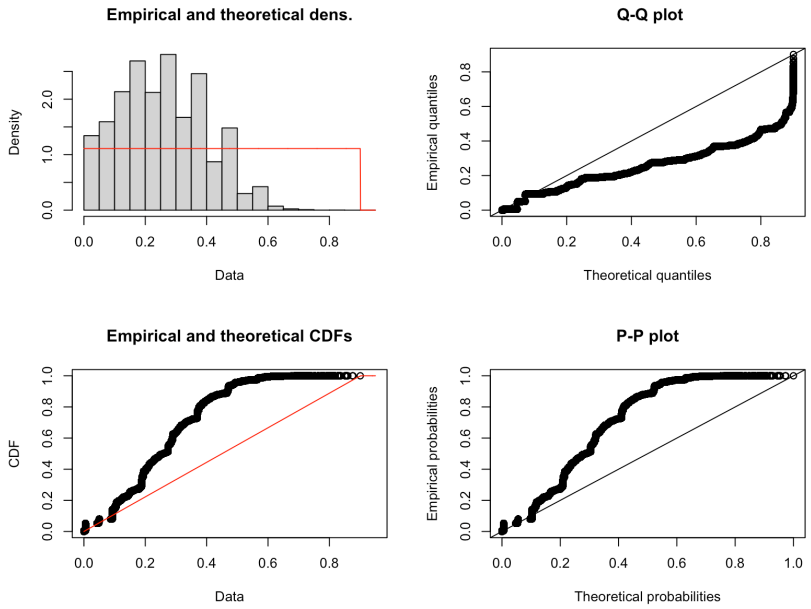


Figure F.21: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **uniform** distribution of the different source scores of score 2 (Goodall3).

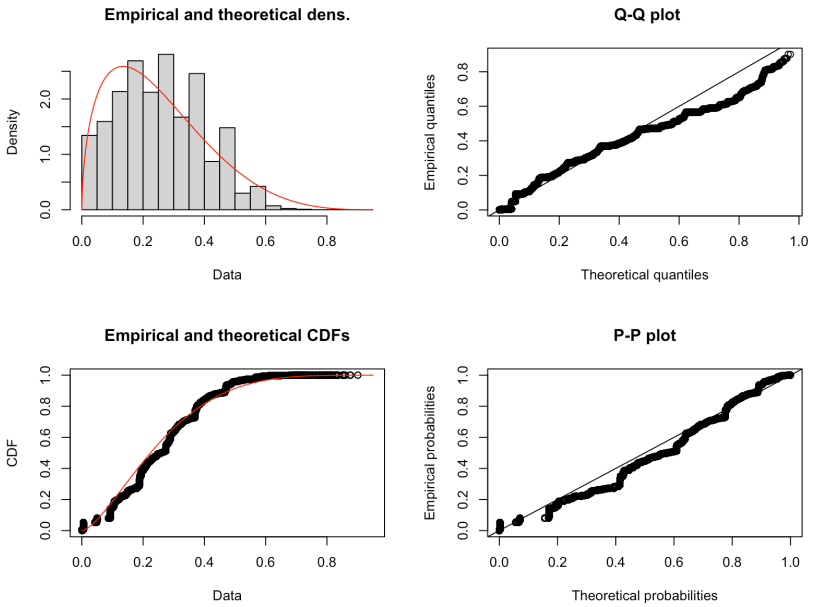


Figure F.22: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **beta** distribution of the different source scores of score 2 (Goodall3).

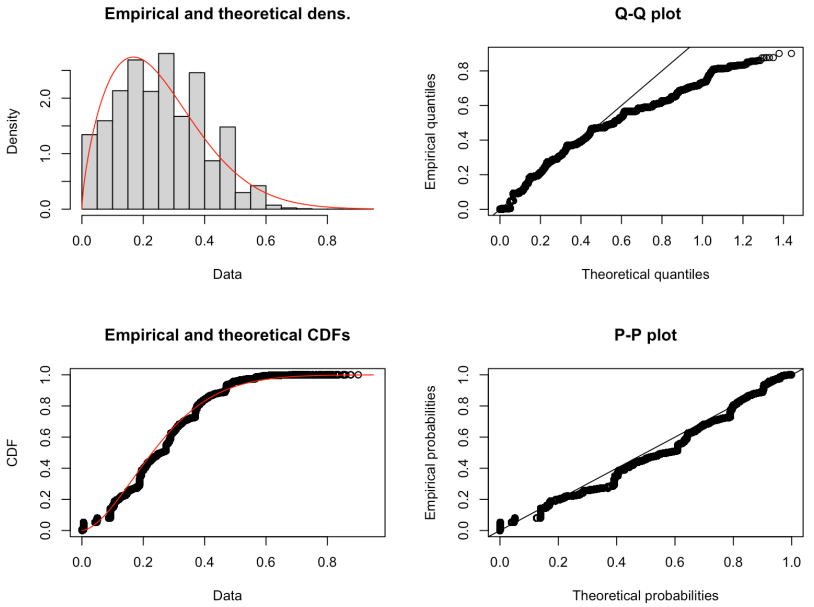


Figure F.23: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **Weibull** distribution of the different source scores of score 2 (Goodall3).

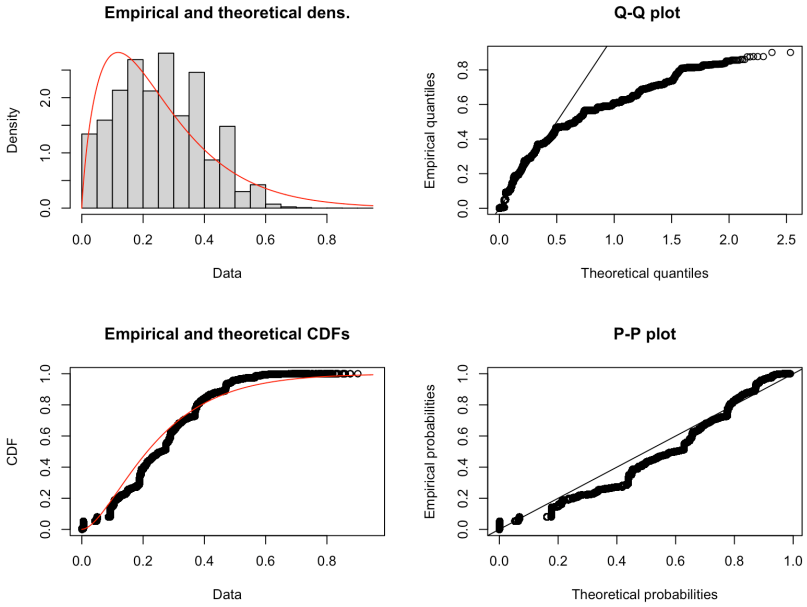


Figure F.24: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **gamma** distribution of the different source scores of score 2 (Goodall3).

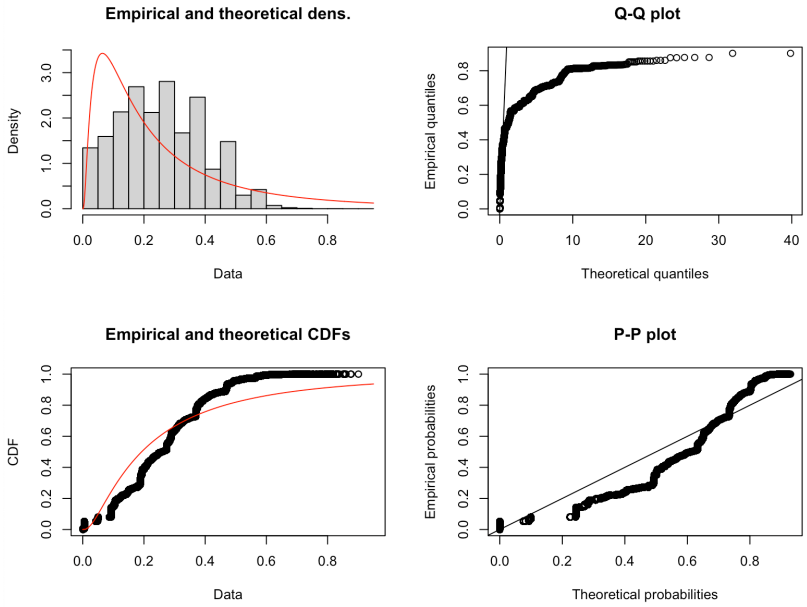


Figure F.25: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **lognormal** distribution of the different source scores of score 2 (Goodall3).

F.5 Distribution of same source scores (score 3)

Figure F.26 shows the Cullen and Frey graph of the same source scores of score 3. It contains the square skewness and kurtosis of the observed scores and seven theoretical distributions. Using this graph, it was decided that the beta, Weibull and gamma distribution could all be the distributions of the observed scores and, therefore, they should be further investigated.

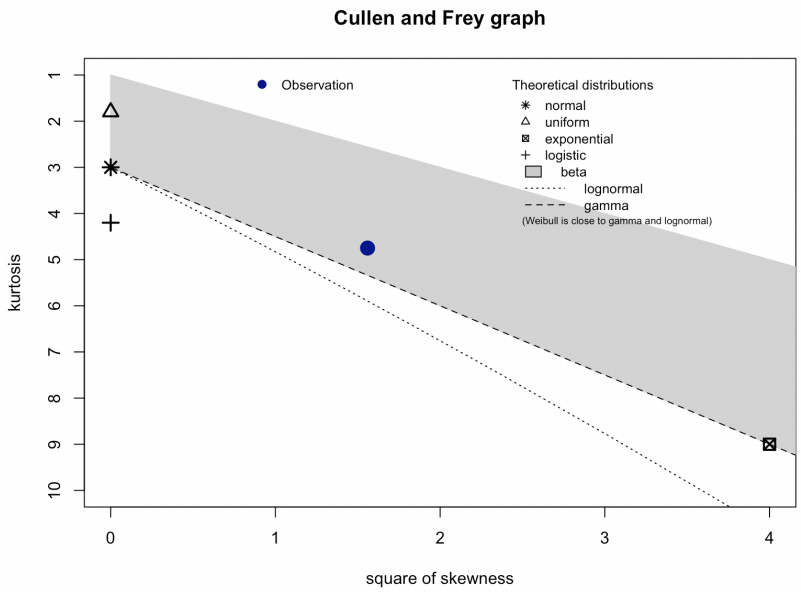


Figure F.26: This figure shows the Cullen and Frey graph of the same source scores of score 3 (Burnaby) (it contains the square skewness and kurtosis of the observed scores and seven theoretical distributions).

Figures F.27, F.28 and F.29 show (of the parametrization with the beta, Weibull and gamma distributions (respectively) of the same source scores of score 3) the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots. By mainly looking at the Q-Q plots (and if the theoretical and empirical quantities are on the same line), it can be concluded that the Weibull distribution is the best fit for the same source scores of score 3 (Burnaby) (and the gamma distribution is the second best fit).

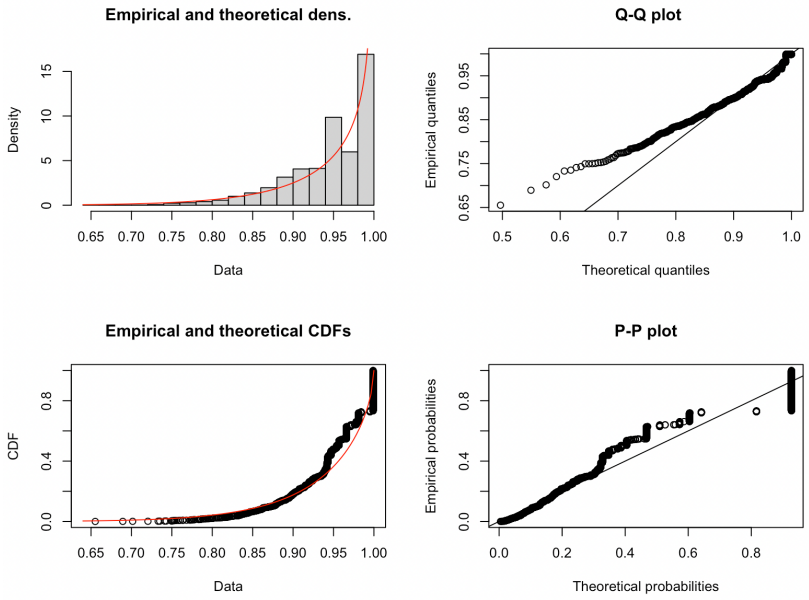


Figure F.27: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **beta** distribution of the same source scores of score 3 (Burnaby).

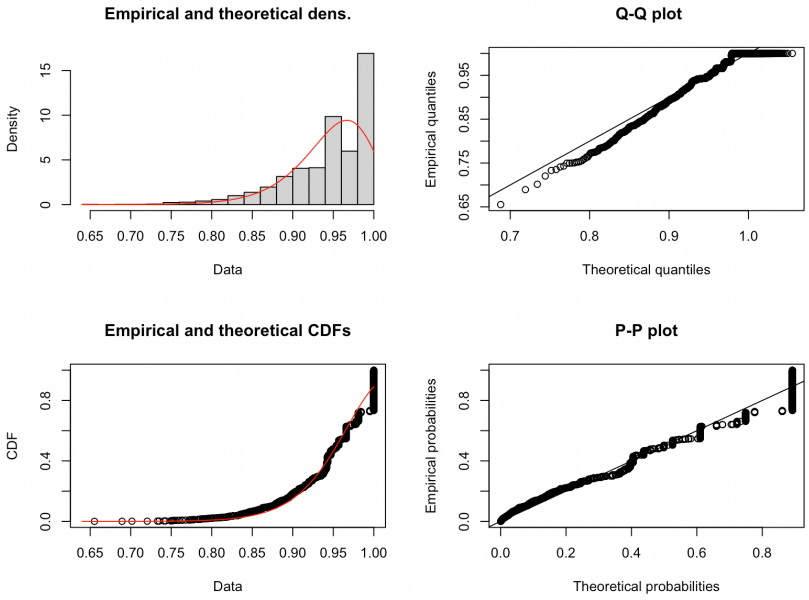


Figure F.28: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **Weibull** distribution of the same source scores of score 3 (Burnaby).

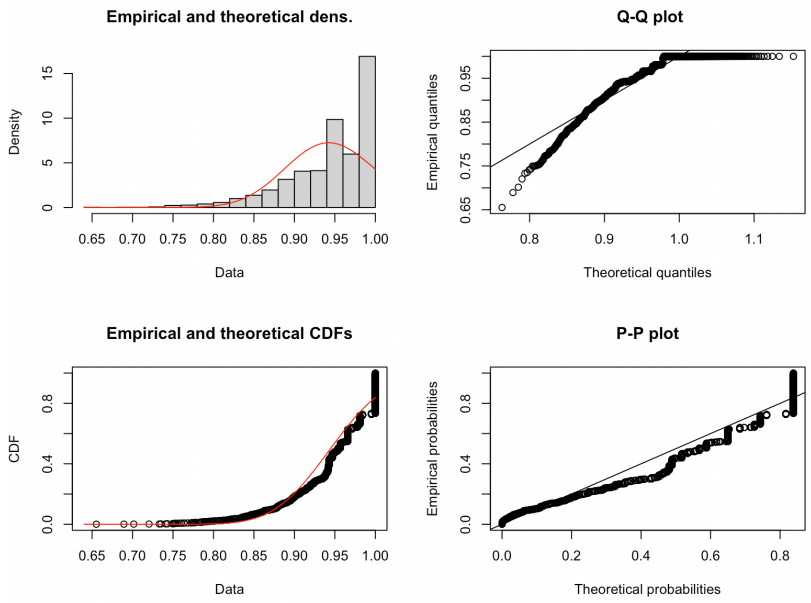


Figure F.29: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **gamma** distribution of the same source scores of score 3 (Burnaby).

F.6 Distribution of different source scores (score 3)

Figure F.30 shows the Cullen and Frey graph of the different source scores of score 3. It contains the square skewness and kurtosis of the observed scores and seven theoretical distributions. Using this graph, it was decided that the normal, uniform, beta, Weibull, gamma and lognormal distribution could all be the distributions of the observed scores and, therefore, they should be further investigated.

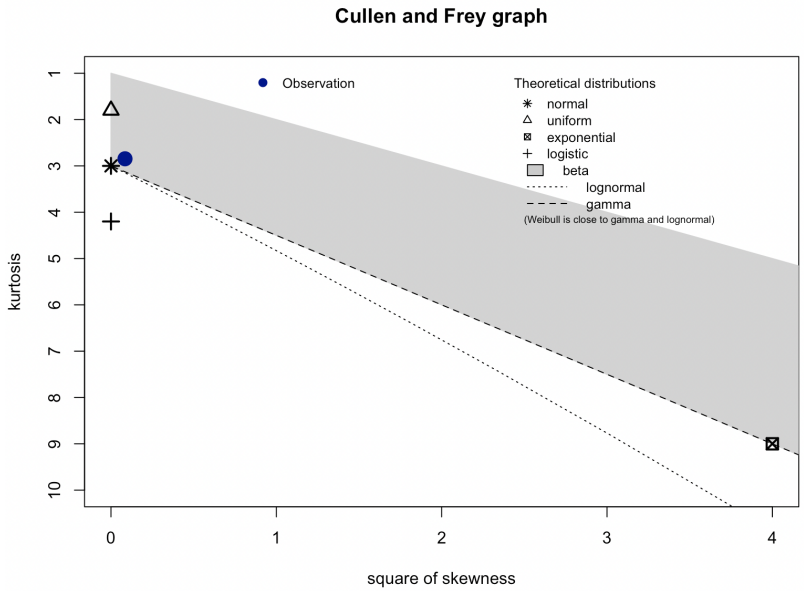


Figure F.30: This figure shows the Cullen and Frey graph of the different source scores of score 3 (Burnaby) (it contains the square skewness and kurtosis of the observed scores and seven theoretical distributions).

Figures F.31, F.32, F.33, F.34, F.35 and F.36 show (of the parametrization with the normal, uniform, beta, Weibull, gamma and lognormal distributions (respectively) of the different source scores of score 1) the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots. By mainly looking at the Q-Q plots (and if the theoretical and empirical quantities are on the same line), it can be concluded that the normal distribution is the best fit for the different source scores of score 3 (Burnaby).

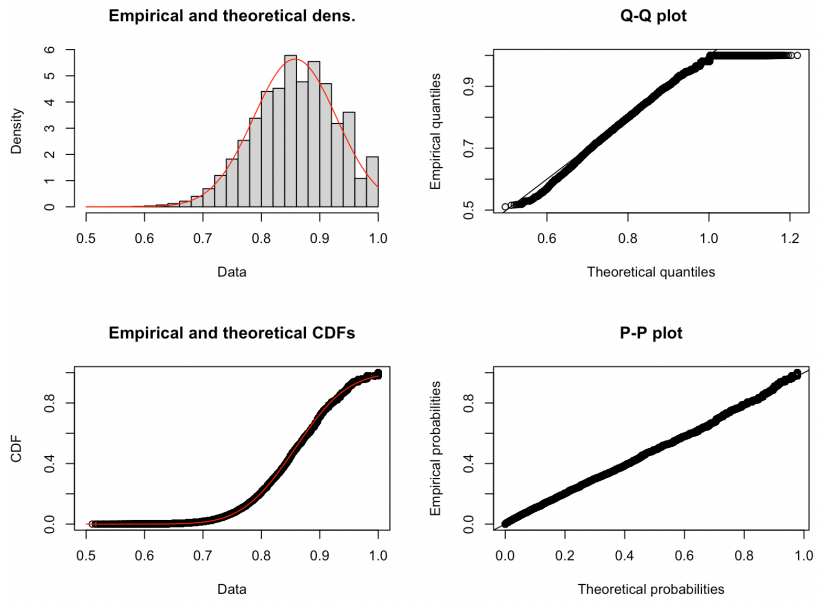


Figure F.31: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **normal** distribution of the different source scores of score 3 (Burnaby).

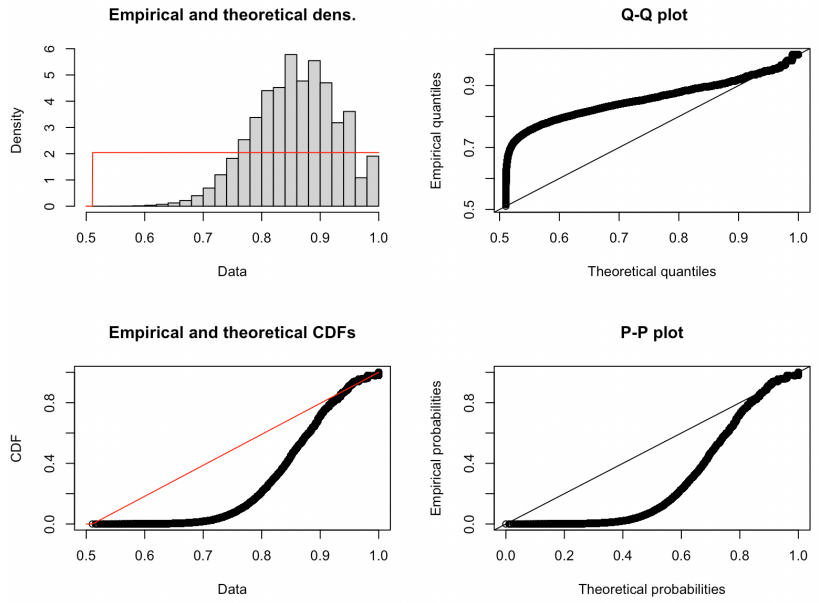


Figure F.32: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **uniform** distribution of the different source scores of score 3 (Burnaby).

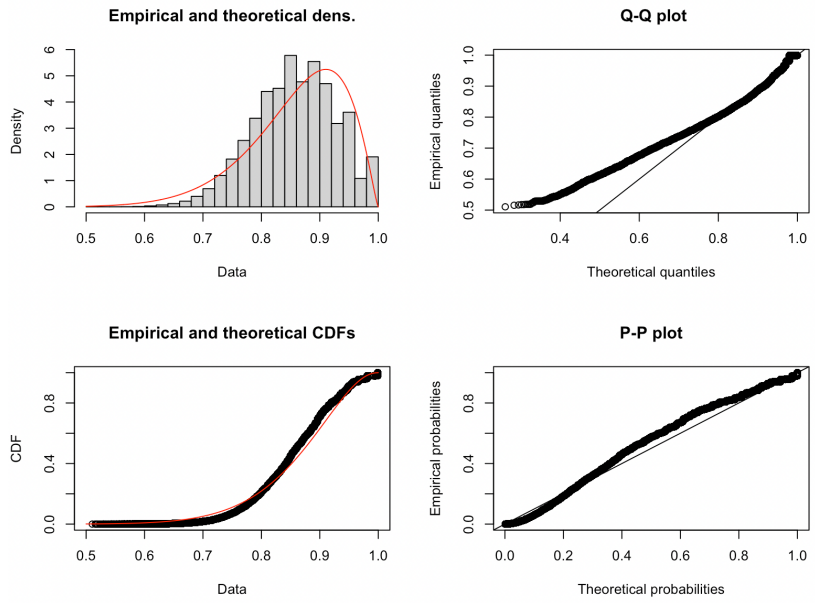


Figure F.33: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **beta** distribution of the different source scores of score 3 (Burnaby).

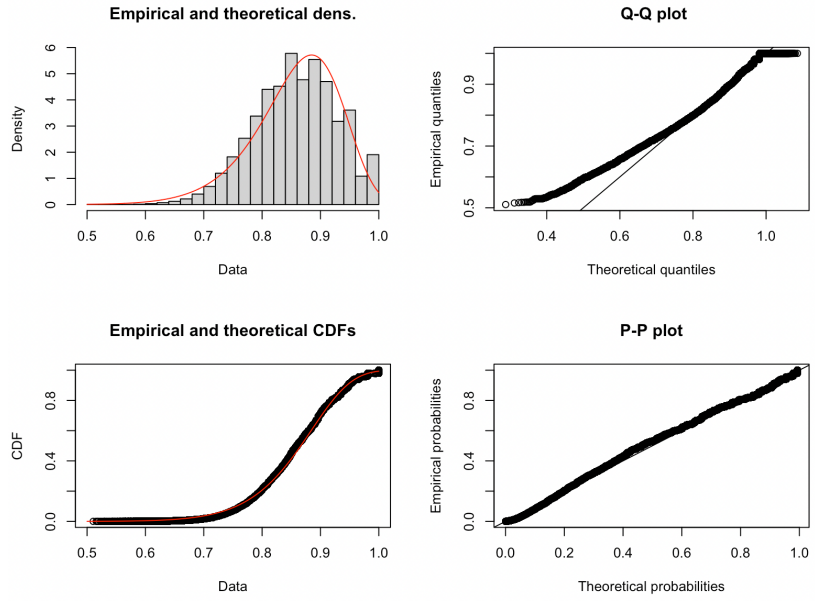


Figure F.34: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **Weibull** distribution of the different source scores of score 3 (Burnaby).

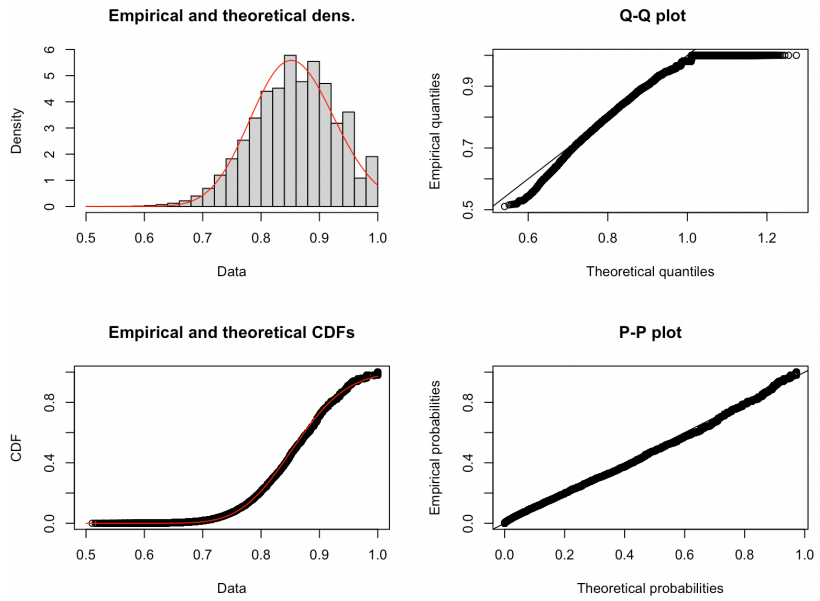


Figure F.35: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **gamma** distribution of the different source scores of score 3 (Burnaby).

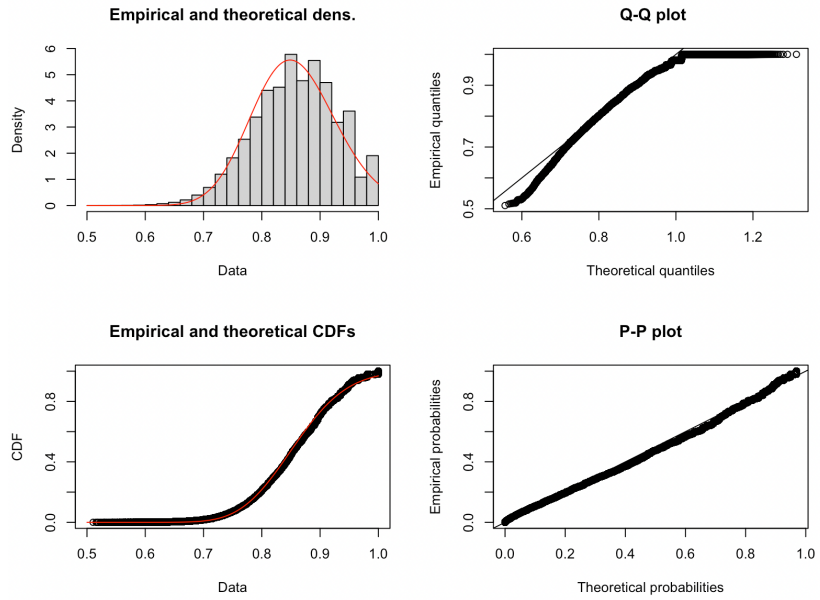


Figure F.36: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **lognormal** distribution of the different source scores of score 3 (Burnaby).

F.7 Distribution of same source scores (score 4)

Figure F.37 shows the Cullen and Frey graph of the same source scores of score 4. It contains the square skewness and kurtosis of the observed scores and seven theoretical distributions. Using this graph, it was decided that the beta, Weibull and gamma distribution could all be the distributions of the observed scores and, therefore, they should be further investigated.

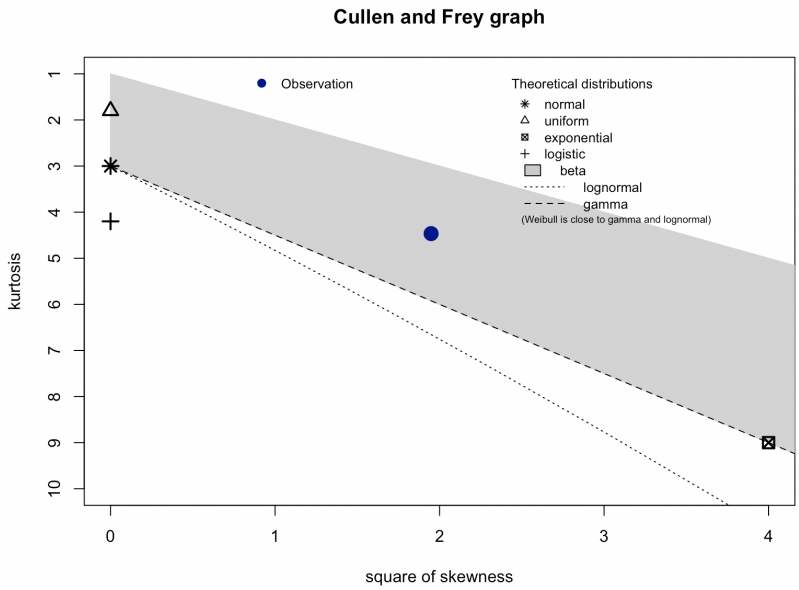


Figure F.37: This figure shows the Cullen and Frey graph of the same source scores of score 4 (Anderberg) (it contains the square skewness and kurtosis of the observed scores and seven theoretical distributions).

Figures F.38, F.39 and F.40 show (of the parametrization with the beta, Weibull and gamma distributions (respectively) of the same source scores of score 4) the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots. By mainly looking at the Q-Q plots (and if the theoretical and empirical quantities are on the same line), it can be concluded that the beta distribution is the best fit for the same source scores of score 4 (Anderberg).

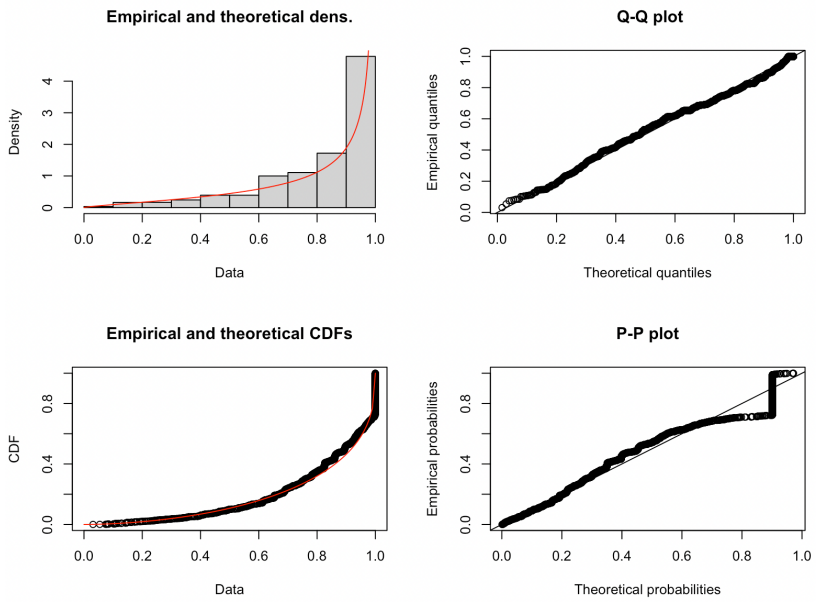


Figure F.38: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **beta** distribution of the same source scores of score 4 (Anderberg).

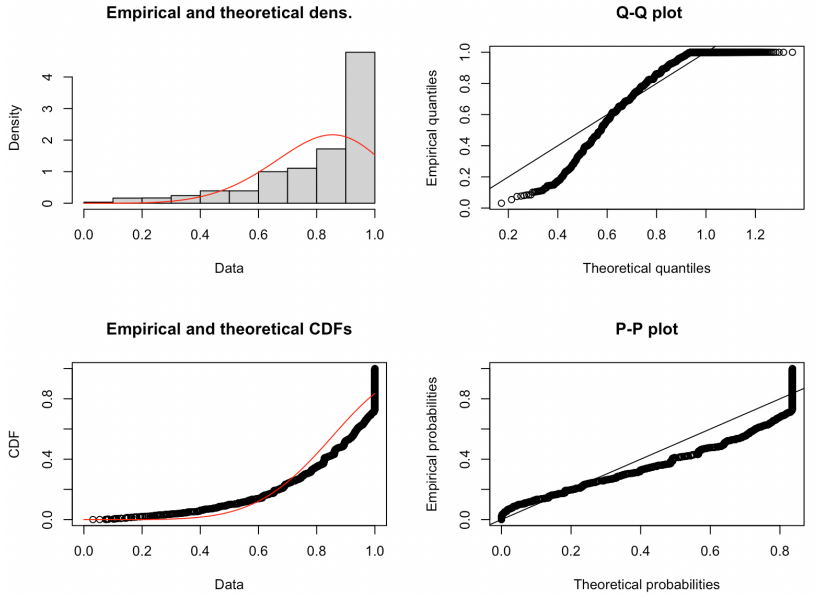


Figure F.39: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **Weibull** distribution of the same source scores of score 4 (Anderberg).

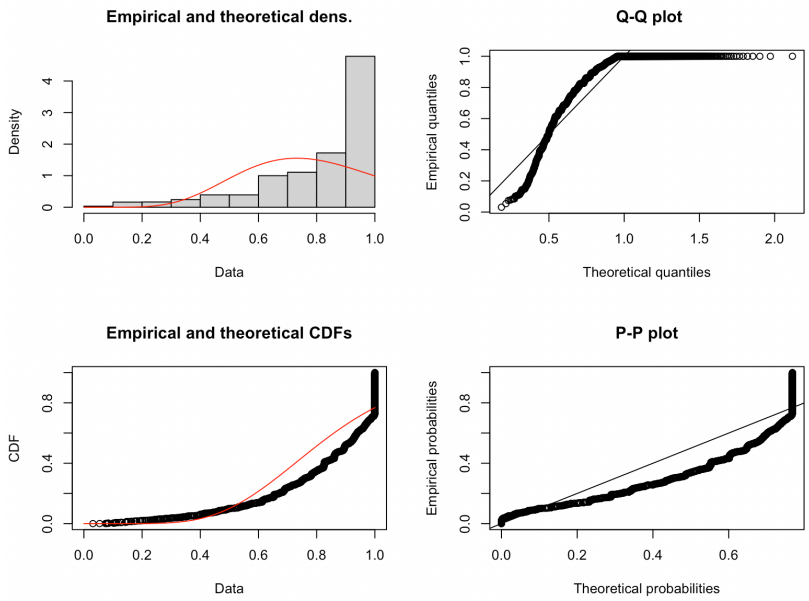


Figure F.40: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **gamma** distribution of the same source scores of score 4 (Anderberg).

F.8 Distribution of different source scores (score 4)

Figure F.41 shows the Cullen and Frey graph of the different source scores of score 4. It contains the square skewness and kurtosis of the observed scores and seven theoretical distributions. Using this graph, it was decided that the normal, uniform, beta, Weibull, gamma and lognormal distribution could all be the distributions of the observed scores and, therefore, they should be further investigated.

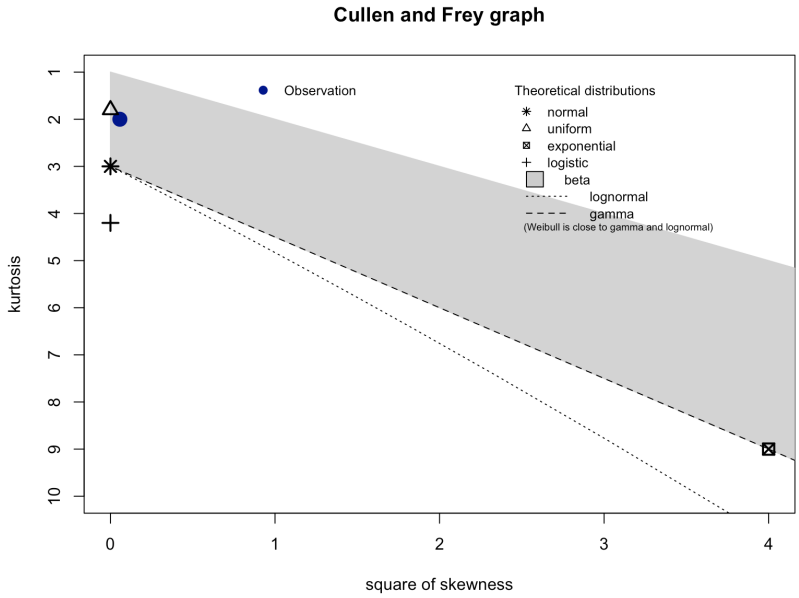


Figure F.41: This figure shows the Cullen and Frey graph of the different source scores of score 4 (Anderberg) (it contains the square skewness and kurtosis of the observed scores and seven theoretical distributions).

Figures F.42, F.43, F.44, F.45, F.46 and F.47 show (of the parametrization with the normal, uniform, beta, Weibull, gamma and lognormal distributions (respectively) of the different source scores of score 1) the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots. By mainly looking at the Q-Q plots (and if the theoretical and empirical quantities are on the same line), it can be concluded that the beta distribution is the best fit for the different source scores of score 4 (Anderberg).

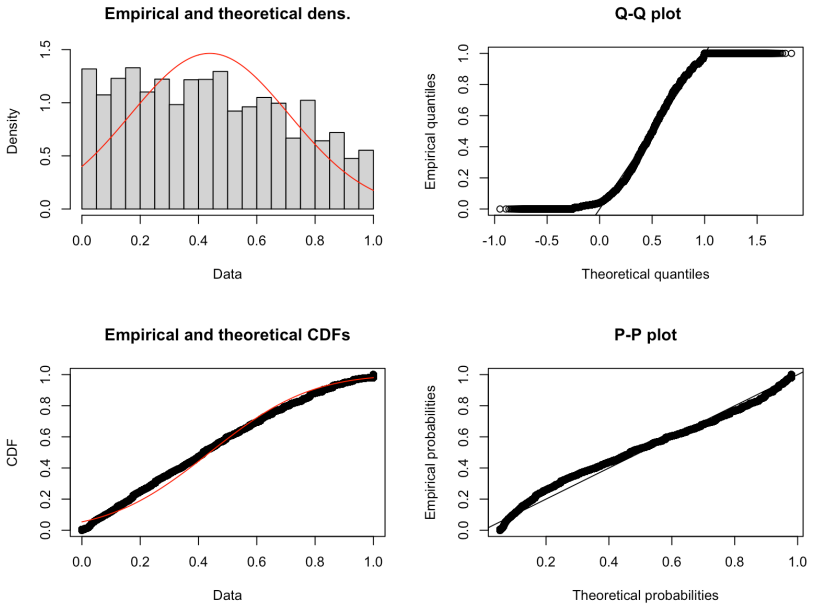


Figure F.42: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **normal** distribution of the different source scores of score 4 (Anderberg).

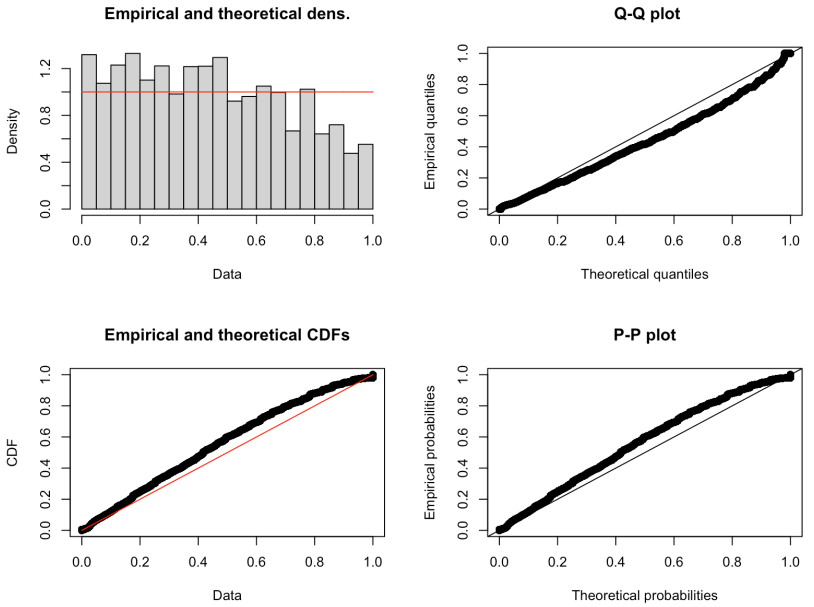


Figure F.43: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **uniform** distribution of the different source scores of score 4 (Anderberg).

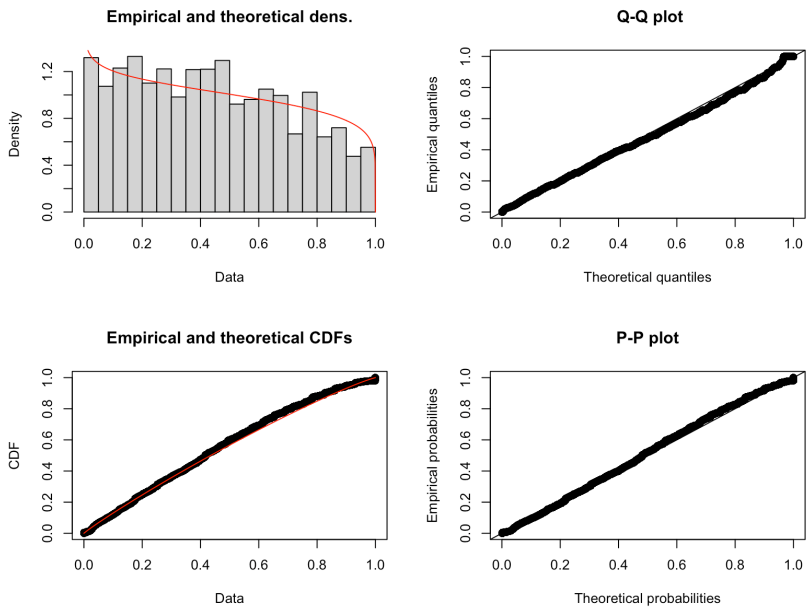


Figure F.44: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **beta** distribution of the different source scores of score 4 (Anderberg).

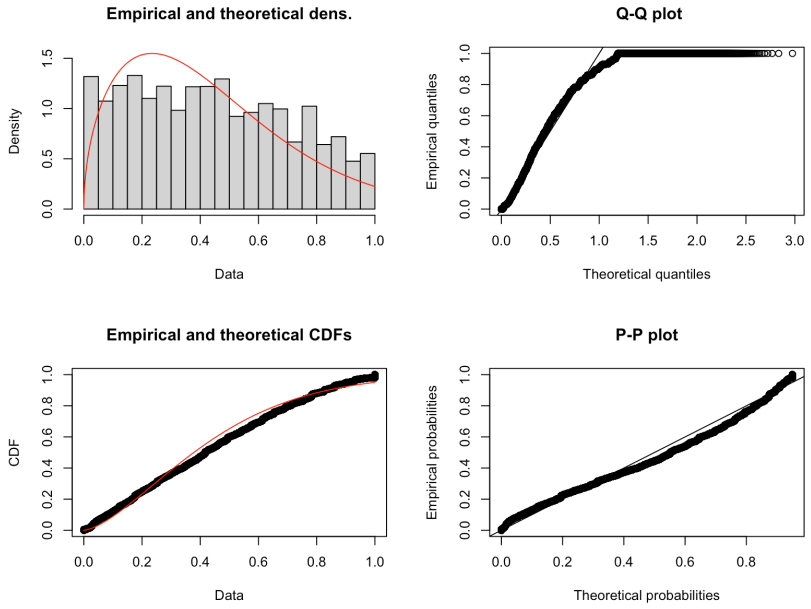


Figure F.45: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **Weibull** distribution of the different source scores of score 4 (Anderberg).

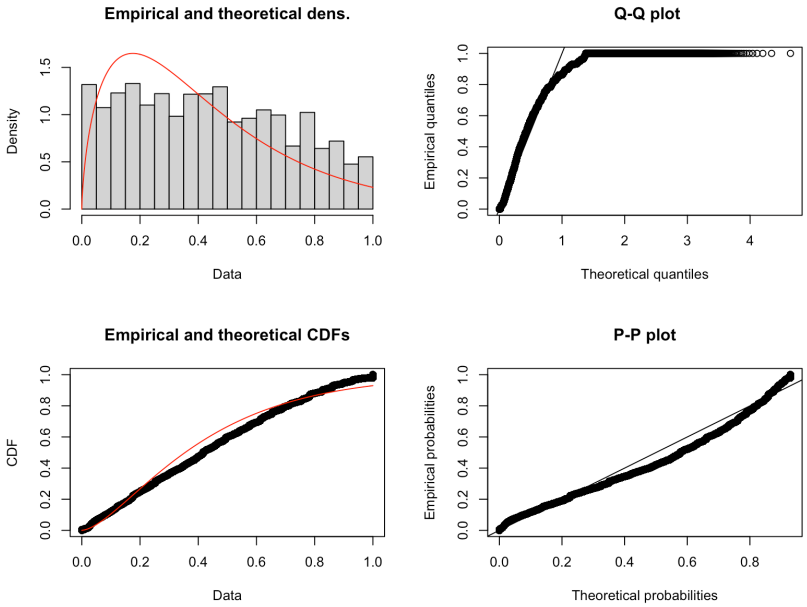


Figure F.46: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **gamma** distribution of the different source scores of score 4 (Anderberg).

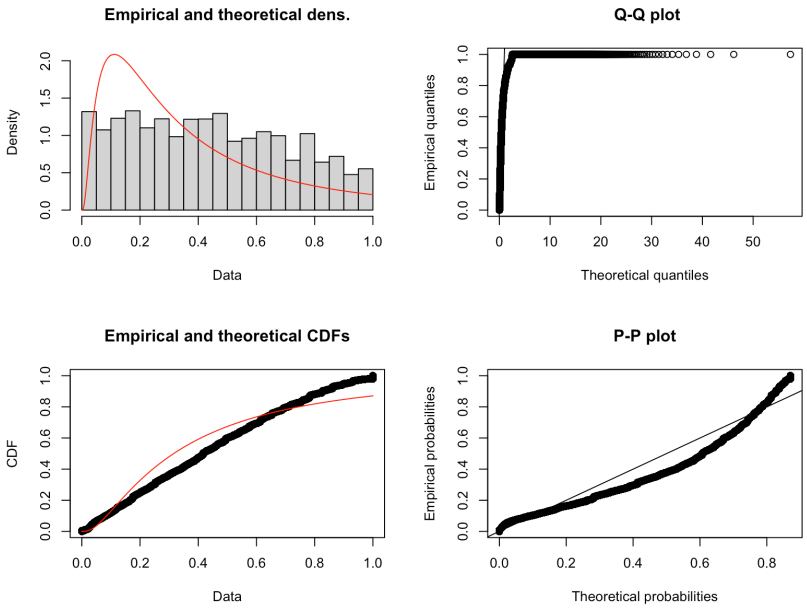


Figure F.47: This figure shows the empirical and theoretical density and CDF plots, the Q-Q and the P-P plots of the parametrization with the **lognormal** distribution of the different source scores of score 4 (Anderberg).

Appendix G

R code used in chapter 6

Note that, in order to run the codes in this appendix, the R code of appendix E needs to be ran first.

G.1 R code used in section 6.1

The following R code was used to carry out the leave-one-out method (cross validation) in section 6.1.

```
library("readxl")
library(fitdistrplus)

All_Char <- read_excel("Char_Vect_All.xlsx")
df_All_Char <-data.frame(All_Char) #Make dataframe of Excel
    file

#Same Source

S1s_Comp<-vector()
S2s_Comp<-vector()
S3s_Comp<-vector()
S4s_Comp<-vector()
#For loop to (for each document): remove the same source
   scores, to
#perform the parametrization again and to calculate the
   SLRs for that document
#(for scores 1, 2, 3 and 4)
for (i in 1:length(S1s)){
  S1s_without<-S1s[-i]
  fit.weibullS1s_without <- fitdist(S1s_without, "weibull")
  Log_SLR1<-log10(dweibull(seq(0,1,0.001), fit.weibullS1s_
     without$estimate[1], fit.weibullS1s_without$estimate
     [2])/dnorm(seq(0,1,0.001), fit.normS1d$estimate[1],
     fit.normS1d$estimate[2]))
  S1s_Comp<-append(S1s_Comp,Log_SLR1[S1s[i]/0.001+1])
  S2s_without<-S2s[-i]
  fit.normS2s_without <- fitdist(S2s_without, "norm")
  Log_SLR2<-log10(dnorm(seq(0,1,0.001), fit.normS2s_without
     $estimate[1], fit.normS2s_without$estimate[2])/dnorm(
     seq(0,1,0.001), fit.normS2d$estimate[1], fit.normS2d$
     estimate[2]))
```

```
24    S2s_Comp<-append(S2s_Comp,Log_SLR2[S2s[i]/0.001+1])
25    S3s_without<-S3s[-i]
26    fit.gammaS3s_without <- fitdist(S3s_without, "gamma")
27    Log_SLR3<-log10(dgamma(seq(0,1,0.001), fit.gammaS3s_
          without$estimate[1], fit.gammaS3s_without$estimate
          [2])/dnorm(seq(0,1,0.001), fit.normS3d$estimate[1],
          fit.normS3d$estimate[2]))
28    S3s_Comp<-append(S3s_Comp,Log_SLR3[S3s[i]/0.001+1])
29    S4s_without<-S4s[-i]
30    fit.betaS4s_without <- fitdist(S4s_without, "beta")
31    Log_SLR4<-log10(dbeta(seq(0,1,0.001), fit.betaS4s_without
          $estimate[1], fit.betaS4s_without$estimate[2])/dbeta(
          seq(0,1,0.001), fit.betaS4d$estimate[1], fit.betaS4d$
          estimate[2]))
32    S4s_Comp<-append(S4s_Comp,Log_SLR4[S4s[i]/0.001])
33  }
34  #Make data frame of the SLRs of each document
35  #(of scores 1, 2, 3 and 4 which form the columns)
36  df_Comp_s<-data.frame(Score1=S1s_Comp, Score2=S2s_Comp,
        Score3=S3s_Comp, Score4=S4s_Comp)
37
38  #Make boxplots of the same source comparisons
39  boxplot(df_Comp_s, xlab="Score System", ylab=expression("
        Log"[10]* "SLR"),
40          main="Boxplots of the Log SLRs of the same source
                comparisons for the four
41          different score systems using the leave-one-out
                method (cross-validation)",
42          names=c("Score 1 (Overlap)","Score 2 (Goodall3)",
                "Score 3 (Burnaby)", "Score 4 (Anderberg)"),
                col=rgb(0,1,0,0.25))
43  abline(h=0,lty=2,lwd=1)
44
45  #Different Source
46
47  S1d_Comp<-vector()
48  S2d_Comp<-vector()
49  S3d_Comp<-vector()
50  S4d_Comp<-vector()
51  #For loop to (for each document): remove the different
        source scores, to
52  #perform the parametrization again and to calculate the
        SLRs for that document
53  #(for scores 1, 2, 3 and 4)
54  for (i in 1:length(S1d)){
55    S1d_without<-S1d[-i]
56    fit.normS1d_without <- fitdist(S1d_without, "norm")
57    Log_SLR1d<-log10(dweibull(seq(0,1,0.001), fit.weibullS1s$
          estimate[1], fit.weibullS1s$estimate[2])/dnorm(seq
          (0,1,0.001), fit.normS1d_without$estimate[1], fit.
          normS1d_without$estimate[2]))
```

```
  S1d_Comp<-append(S1d_Comp,Log_SLR1d[S1d[i]/0.001+1])
  S2d_without<-S2d[-i]
  fit.normS2d_without <- fitdist(S2d_without, "norm")
  Log_SLR2d<-log10(dnorm(seq(0,1,0.001), fit.normS2s$
      estimate[1], fit.normS2s$estimate[2])/dnorm(seq
      (0,1,0.001), fit.normS2d_without$estimate[1], fit.
      normS2d_without$estimate[2]))
  S2d_Comp<-append(S2d_Comp,Log_SLR2d[S2d[i]/0.001+1])
  S3d_without<-S3d[-i]
  fit.normS3d_without <- fitdist(S3d_without, "norm")
  Log_SLR3d<-log10(dgamma(seq(0,1,0.001), fit.gammaS3s$
      estimate[1], fit.gammaS3s$estimate[2])/dnorm(seq
      (0,1,0.001), fit.normS3d_without$estimate[1], fit.
      normS3d_without$estimate[2]))
  S3d_Comp<-append(S3d_Comp,Log_SLR3d[S3d[i]/0.001+1])
  S4d_without<-S4d[-i]
  fit.betaS4d_without <- fitdist(S4d_without, "beta")
  Log_SLR4d<-log10(dbeta(seq(0,1,0.001), fit.betaS4s$
      estimate[1], fit.betaS4s$estimate[2])/dbeta(seq
      (0,1,0.001), fit.betaS4d_without$estimate[1], fit.
      betaS4d_without$estimate[2]))
  S4d_Comp<-append(S4d_Comp,Log_SLR4d[S4d[i]/0.001])
}
#Make data frame of the SLRs of each document
#(of scores 1, 2, 3 and 4 which form the columns)
df_Comp_d<-data.frame(Score1=S1d_Comp, Score2=S2d_Comp,
    Score3=S3d_Comp, Score4=S4d_Comp)

#Make boxplots of the different source comparisons
boxplot(df_Comp_d, xlab="Score System", ylab=expression("
    Log"[10]* "SLR"),
        main="Boxplots of the Log SLRs of the different
            source comparisons for the four
        different score systems using the leave-one-out
            method (cross-validation)",
        names=c("Score 1 (Overlap)","Score 2 (Goodall3)",
            "Score 3 (Burnaby)", "Score 4 (Anderberg)"),
            col=rgb(0,1,0,0.25))
abline(h=0,lty=2,lwd=1)
```

G.2 R code used in section 6.2

The following R code was used to calculate the 95% bootstrap confidence intervals in section 6.2. Only the code for the intervals of score 1 (Overlap) is shown for succinctness, but the code for the intervals of the other scores is available upon request.

```
library("readxl")
library(fitdistrplus)

```

```
All_Char <- read_excel("Char_Vect_All.xlsx")
df_All_Char <-data.frame(All_Char) #Make dataframe of Excel
    file

#Score 1
n.s=length(S1s) #Sample size same source scores
n.d=2*length(S1d) #Sample size different source scores
B<-50 #Number of times the process is repeated
names(S1s) <- seq_along(S1s) #Give scores indices

#Put each boot-sample in a column
Sample.1s=sample(S1s, size=B*n.s,replace=TRUE)
#Create a matrix of the same source scores samples
#(50 rows, 2400 columns)
Boot.1s<-matrix(Sample.1s, ncol=B,nrow=n.s)

#Find the 2397 different source scores belonging to each
   document
Document_Sample.1s=names(Sample.1s)
Number_Scores_D=rep(seq(2397,0,-3),each=3)
Sample.1d<-vector()
for (i in Document_Sample.1s){
  Number_Scores_D_i=Number_Scores_D[strtoi(i)]
  Number_Scores_D_Before_i=Number_Scores_D[1:(strtoi(i)-1)
     ]
  Before_i=sum(Number_Scores_D_Before_i)
  Scores_of_Interest2<-vector()
  Counter=0
  for (j in Number_Scores_D_Before_i){
    if (j!=Number_Scores_D_i){
      Scores_of_Interest2<-append(Scores_of_Interest2,S1d[(
         Counter+j-strtoi(i))])
      Counter<-Counter+j}}
  Scores_of_Interest_i1=S1d[(Before_i+1):(Before_i+Number_
     Scores_D_i)]
  Scores_of_Interest_i=c(Scores_of_Interest_i1,Scores_of_
     Interest2)
  Sample.1d_i=sample(Scores_of_Interest_i,size=n.s,
     replace=TRUE)
  Sample.1d<-append(Sample.1d,Sample.1d_i)}

#Create a matrix of the different source scores samples
#(50 rows, 2*2,876,400=5,752,800 columns)
Boot.1d<-matrix(Sample.1d, ncol=B,nrow=n.d)

SLR_1_ALL<-vector()
#Perform the parametrization
for (i in 1:B){
  Column_1s=Boot.1s[,i]
  Column_1d=Boot.1d[,i]
  Dist_1s<-fitdist(Column_1s, "weibull")
```

```
48    Dist_1d<-fitdist(Column_1d, "norm")
49    SLR_1_ALL<-append(SLR_1_ALL,log10(dweibull(seq(0,1,0.001)
          , Dist_1s$estimate[1],Dist_1s$estimate[2])/dnorm(seq
          (0,1,0.001), Dist_1d$estimate[1], Dist_1d$estimate
          [2])))}
50
51  #Calculate the median and upper and lower quantile (in
        order to obtain the 95% SLR bootstrap confidence
        interval)
52  means_1<-vector()
53  lq_1<-vector()
54  hq_1<-vector()
55  for (j in 1:1001){
56    SLR_1<-vector()
57    for (i in seq(j,length(SLR_1_ALL),1001)){
58      SLR_1<-append(SLR_1,SLR_1_ALL[i])}
59    means_1<-append(means_1,mean(SLR_1))
60    lq_1<-append(lq_1,quantile(SLR_1,0.025))
61    hq_1<-append(hq_1,quantile(SLR_1,0.975))}
62
63  #Plot the median and the 95% bootstrap confidence interval
64  #for score 1 (Overlap)
65  plot(seq(0,1,0.001),means_1, type="l",xlab="Score 1 (
        Overlap)",
66       ylab=expression("Log"[10]* "(SLR)"),
67       main="Median (50% point) of the SLR bootstrap results
             with the boundaries of the 95% SLR bootrap
            interval of score (Overlap)",lty=4)
68  lines(seq(0,1,0.001),lq_1, col = "springgreen4")
69  lines(seq(0,1,0.001),hq_1, col = "springgreen4")
70  abline(h=0,lty=2, lwd=1.5)
71  grid()
72  legend("bottomright", legend=c("Median", "95% SLR
        bootstrap interval"),
73         col=c("black", "springgreen4"), lty=c(4,1),cex=0.8)
74
75  #Maximum and mean width of Interval
76  width<-hq_1-lq_1
77  width<-na.omit(width)
78  max(width)
79  mean(width)
```

G.3 R code used in section 6.3

The following R code was used to quantify the misleading evidence in section 6.3. So it was used in order to calculate the percentages of misleading evidence, to calculate the indications of the strength of evidence, to create the Tippett plot and to calculate the expected values.

```
1  library("readxl")
```

```
library(fitdistrplus)

All_Char <- read_excel("Char_Vect_All.xlsx")
df_All_Char <-data.frame(All_Char) #Make dataframe of Excel
    file

SLR1<-dweibull(seq(0,1,0.001), fit.weibullS1s$estimate[1],
   fit.weibullS1s$estimate[2])/dnorm(seq(0,1,0.001), fit.
   normS1d$estimate[1], fit.normS1d$estimate[2])
SLR2<-dnorm(seq(0,1,0.001), fit.normS2s$estimate[1], fit.
   normS2s$estimate[2])/dnorm(seq(0,1,0.001), fit.normS2d$
   estimate[1], fit.normS2d$estimate[2])
SLR3<-dgamma(seq(0,1,0.001), fit.gammaS3s$estimate[1], fit.
   gammaS3s$estimate[2])/dnorm(seq(0,1,0.001), fit.normS3d
   $estimate[1], fit.normS3d$estimate[2])
SLR4<-dbeta(seq(0,1,0.001), fit.betaS4s$estimate[1], fit.
   betaS4s$estimate[2])/dbeta(seq(0,1,0.001), fit.betaS4d$
   estimate[1], fit.betaS4d$estimate[2])

#Calculate the SLR values for the same and different
   source scores
S1s_CompM<-vector()
S2s_CompM<-vector()
S3s_CompM<-vector()
S4s_CompM<-vector()
S1d_CompM<-vector()
S2d_CompM<-vector()
S3d_CompM<-vector()
S4d_CompM<-vector()
for (i in 1:length(S1s)){
  S1s_CompM<-append(S1s_CompM,SLR1[S1s[i]/0.001+1])
  S2s_CompM<-append(S2s_CompM,SLR2[S2s[i]/0.001+1])
  S3s_CompM<-append(S3s_CompM,SLR3[S3s[i]/0.001+1])
  S4s_CompM<-append(S4s_CompM,SLR4[S4s[i]/0.001+1])}
for (i in 1:length(S1d)){
  S1d_CompM<-append(S1d_CompM,SLR1[S1d[i]/0.001+1])
  S2d_CompM<-append(S2d_CompM,SLR2[S2d[i]/0.001+1])
  S3d_CompM<-append(S3d_CompM,SLR3[S3d[i]/0.001+1])
  S4d_CompM<-append(S4d_CompM,SLR4[S4d[i]/0.001+1])}

#Indication of the strength of evidence
Interval_vect<-c(1/10000,1/1000,1/100,1/
   10,1,10,100,1000,10000)

#Same Source
Perc_1s<-vector()
Perc_2s<-vector()
Perc_3s<-vector()
Perc_4s<-vector()
Perc_1d<-vector()
Perc_2d<-vector()
```

```
Perc_3d<-vector()
Perc_4d<-vector()
Perc_1s<-append(Perc_1s,sum(S1s_CompM < Interval_vect[1])/
    length(S1s))
Perc_2s<-append(Perc_2s,sum(S2s_CompM < Interval_vect[1])/
    length(S1s))
Perc_3s<-append(Perc_3s,sum(S3s_CompM < Interval_vect[1])/
    length(S1s))
Perc_4s<-append(Perc_4s,sum(S4s_CompM < Interval_vect[1])/
    length(S1s))
Perc_1d<-append(Perc_1d,sum(S1d_CompM < Interval_vect[1])/
    length(S1d))
Perc_2d<-append(Perc_2d,sum(S2d_CompM < Interval_vect[1])/
    length(S1d))
Perc_3d<-append(Perc_3d,sum(S3d_CompM < Interval_vect[1])/
    length(S1d))
Perc_4d<-append(Perc_4d,sum(S4d_CompM < Interval_vect[1])/
    length(S1d))
for (i in 1:(length(Interval_vect)-1)){
  Perc_1s<-append(Perc_1s,sum(S1s_CompM > Interval_vect[i]
     & S1s_CompM < Interval_vect[i+1])/length(S1s))
  Perc_2s<-append(Perc_2s,sum(S2s_CompM > Interval_vect[i]
     & S2s_CompM < Interval_vect[i+1])/length(S1s))
  Perc_3s<-append(Perc_3s,sum(S3s_CompM > Interval_vect[i]
     & S3s_CompM < Interval_vect[i+1])/length(S1s))
  Perc_4s<-append(Perc_4s,sum(S4s_CompM > Interval_vect[i]
     & S4s_CompM < Interval_vect[i+1])/length(S1s))
  Perc_1d<-append(Perc_1d,sum(S1d_CompM > Interval_vect[i]
     & S1d_CompM < Interval_vect[i+1])/length(S1d))
  Perc_2d<-append(Perc_2d,sum(S2d_CompM > Interval_vect[i]
     & S2d_CompM < Interval_vect[i+1])/length(S1d))
  Perc_3d<-append(Perc_3d,sum(S3d_CompM > Interval_vect[i]
     & S3d_CompM < Interval_vect[i+1])/length(S1d))
  Perc_4d<-append(Perc_4d,sum(S4d_CompM > Interval_vect[i]
     & S4d_CompM < Interval_vect[i+1])/length(S1d))}
Perc_1s<-append(Perc_1s,sum(S1s_CompM > Interval_vect[
    length(Interval_vect)])/length(S1s))
Perc_2s<-append(Perc_2s,sum(S2s_CompM > Interval_vect[
    length(Interval_vect)])/length(S1s))
Perc_3s<-append(Perc_3s,sum(S3s_CompM > Interval_vect[
    length(Interval_vect)])/length(S1s))
Perc_4s<-append(Perc_4s,sum(S4s_CompM > Interval_vect[
    length(Interval_vect)])/length(S1s))
Perc_1d<-append(Perc_1d,sum(S1d_CompM > Interval_vect[
    length(Interval_vect)])/length(S1d))
Perc_2d<-append(Perc_2d,sum(S2d_CompM > Interval_vect[
    length(Interval_vect)])/length(S1d))
Perc_3d<-append(Perc_3d,sum(S3d_CompM > Interval_vect[
    length(Interval_vect)])/length(S1d))
Perc_4d<-append(Perc_4d,sum(S4d_CompM > Interval_vect[
    length(Interval_vect)])/length(S1d))
```

```
69
70 #Calculate the percentages of misleading evidence
71 Perc_M_1s=sum(Perc_1s[1:5])
72 Perc_M_2s=sum(Perc_2s[1:5])
73 Perc_M_3s=sum(Perc_3s[1:5])
74 Perc_M_4s=sum(Perc_4s[1:5])
75 Perc_M_1d=sum(Perc_1d[6:10])
76 Perc_M_2d=sum(Perc_2d[6:10])
77 Perc_M_3d=sum(Perc_3d[6:10])
78 Perc_M_4d=sum(Perc_4d[6:10])
79
80 #Calculate the expected values of 1/SLR given H1 and SLR
      given H2
81 Exp_1LR_1s=mean(1/S1s_CompM)
82 Exp_1LR_2s=mean(1/S2s_CompM)
83 Exp_1LR_3s=mean(1/S3s_CompM)
84 Exp_1LR_4s=mean(1/S4s_CompM)
85 Exp_LR_1d=mean(S1d_CompM)
86 Exp_LR_2d=mean(S2d_CompM)
87 Exp_LR_3d=mean(S3d_CompM)
88 Exp_LR_4d=mean(S4d_CompM)
89
90 #Creation of the Tippett plot
91 logS1s_CompM=log10(S1s_CompM)
92 logS2s_CompM=log10(S2s_CompM)
93 logS3s_CompM=log10(S3s_CompM)
94 logS4s_CompM=log10(S4s_CompM)
95 logS1d_CompM=log10(S1d_CompM)
96 logS2d_CompM=log10(S2d_CompM)
97 logS3d_CompM=log10(S3d_CompM)
98 logS4d_CompM=log10(S4d_CompM)
99 Freq_1s<-table(logS1s_CompM)
100 Freq_2s<-table(logS2s_CompM)
101 Freq_3s<-table(logS3s_CompM)
102 Freq_4s<-table(logS4s_CompM)
103 Freq_1d<-table(logS1d_CompM)
104 Freq_2d<-table(logS2d_CompM)
105 Freq_3d<-table(logS3d_CompM)
106 Freq_4d<-table(logS4d_CompM)
107
108 #Same source, Score 1
109 lst_y_t_1s<-vector()
110 lst_y_t_1s<-append(lst_y_t_1s,1)
111 counter=1
112 for (i in Freq_1s){
113   counter=counter-i/length(S1s)
114   lst_y_t_1s<-append(lst_y_t_1s,counter)}
115 lst_y_t_1s<-append(lst_y_t_1s,0)
116 logS1s_CompM_dup1<-logS1s_CompM[!duplicated(logS1s_CompM)]
117 logS1s_CompM_dup1<-append(logS1s_CompM_dup1,c(-5,5))
118 logS1s_CompM_dup<-sort(logS1s_CompM_dup1)
```

```
119
120 #Same source, Score 2
121 lst_y_t_2s<-vector()
122 lst_y_t_2s<-append(lst_y_t_2s,1)
123 counter=1
124 for (i in Freq_2s){
125   counter=counter-i/length(S1s)
126   lst_y_t_2s<-append(lst_y_t_2s,counter)}
127 lst_y_t_2s<-append(lst_y_t_2s,0)
128 logS2s_CompM_dup1<-logS2s_CompM[!duplicated(logS2s_CompM)]
129 logS2s_CompM_dup1<-append(logS2s_CompM_dup1,c(-5,5))
130 logS2s_CompM_dup<-sort(logS2s_CompM_dup1)
131
132 #Same source, Score 3
133 lst_y_t_3s<-vector()
134 lst_y_t_3s<-append(lst_y_t_3s,1)
135 counter=1
136 for (i in Freq_3s){
137   counter=counter-i/length(S1s)
138   lst_y_t_3s<-append(lst_y_t_3s,counter)}
139 lst_y_t_3s<-append(lst_y_t_3s,0)
140 logS3s_CompM_dup1<-logS3s_CompM[!duplicated(logS3s_CompM)]
141 logS3s_CompM_dup1<-append(logS3s_CompM_dup1,c(-5,5))
142 logS3s_CompM_dup<-sort(logS3s_CompM_dup1)
143
144 #Same source, Score 4
145 lst_y_t_4s<-vector()
146 lst_y_t_4s<-append(lst_y_t_4s,1)
147 counter=1
148 for (i in Freq_4s){
149   counter=counter-i/length(S1s)
150   lst_y_t_4s<-append(lst_y_t_4s,counter)}
151 lst_y_t_4s<-append(lst_y_t_4s,0)
152 logS4s_CompM_dup1<-logS4s_CompM[!duplicated(logS4s_CompM)]
153 logS4s_CompM_dup1<-append(logS4s_CompM_dup1,c(-5,5))
154 logS4s_CompM_dup<-sort(logS4s_CompM_dup1)
155
156 #Different source, Score 1
157 lst_y_t_1d<-vector()
158 lst_y_t_1d<-append(lst_y_t_1d,1)
159 counter=1
160 for (i in Freq_1d){
161   counter=counter-i/length(S1d)
162   lst_y_t_1d<-append(lst_y_t_1d,counter)}
163 lst_y_t_1d<-append(lst_y_t_1d,0)
164 logS1d_CompM_dup1<-logS1d_CompM[!duplicated(logS1d_CompM)]
165 logS1d_CompM_dup1<-append(logS1d_CompM_dup1,c(-5,5))
166 logS1d_CompM_dup<-sort(logS1d_CompM_dup1)
167
168 #Different source, Score 2
169 lst_y_t_2d<-vector()
```

```
lst_y_t_2d<-append(lst_y_t_2d,1)
counter=1
for (i in Freq_2d){
  counter=counter-i/length(S1d)
  lst_y_t_2d<-append(lst_y_t_2d,counter)}
lst_y_t_2d<-append(lst_y_t_2d,0)
logS2d_CompM_dup1<-logS2d_CompM[!duplicated(logS2d_CompM)]
logS2d_CompM_dup1<-append(logS2d_CompM_dup1,c(-5,5))
logS2d_CompM_dup<-sort(logS2d_CompM_dup1)

#Different source, Score 3
lst_y_t_3d<-vector()
lst_y_t_3d<-append(lst_y_t_3d,1)
counter=1
for (i in Freq_3d){
  counter=counter-i/length(S1d)
  lst_y_t_3d<-append(lst_y_t_3d,counter)}
lst_y_t_3d<-append(lst_y_t_3d,0)
logS3d_CompM_dup1<-logS3d_CompM[!duplicated(logS3d_CompM)]
logS3d_CompM_dup1<-append(logS3d_CompM_dup1,c(-5,5))
logS3d_CompM_dup<-sort(logS3d_CompM_dup1)

#Different source, Score 4
lst_y_t_4d<-vector()
lst_y_t_4d<-append(lst_y_t_4d,1)
counter=1
for (i in Freq_4d){
  counter=counter-i/length(S1d)
  lst_y_t_4d<-append(lst_y_t_4d,counter)}
lst_y_t_4d<-append(lst_y_t_4d,0)
logS4d_CompM_dup1<-logS4d_CompM[!duplicated(logS4d_CompM)]
logS4d_CompM_dup1<-append(logS4d_CompM_dup1,c(-5,5))
logS4d_CompM_dup<-sort(logS4d_CompM_dup1)

#Plotting the Tippett plot
plot(logS1s_CompM_dup,lst_y_t_1s,type="s",ylim=c(0,1),xlim
   =c(-5,5),
     xlab=expression("SLR greater than (in Log"[10]*")"),
        ylab="Proportion of cases",
     main="Tippett plot showing the proportion of cases in
         which the SLR
     (given H1 or H2) exceeds a certain value for the four
         different score systems",
     col="springgreen4",lwd=2)
lines(logS2s_CompM_dup,lst_y_t_2s,type="s",col="
   yellowgreen",lwd=2)
lines(logS3s_CompM_dup,lst_y_t_3s,type="s",col="royalblue3
   ",lwd=2)
lines(logS4s_CompM_dup,lst_y_t_4s,type="s",col="steelblue1
   ",lwd=2)
```

```
lines(logS1d_CompM_dup,lst_y_t_1d,type="s",col="
    springgreen4",lty=2,lwd=2)
lines(logS2d_CompM_dup,lst_y_t_2d,type="s",col="
    yellowgreen",lty=2,lwd=2)
lines(logS3d_CompM_dup,lst_y_t_3d,type="s",col="royalblue3
    ",lty=2,lwd=2)
lines(logS4d_CompM_dup,lst_y_t_4d,type="s",col="steelblue1
    ",lty=2,lwd=2)
abline(v=0,lty=2,lwd=2)
grid()
legend("bottomleft", legend=c("Score 1 (Overlap)", "Score
    2 (Goodall3)", "Score 3 (Burnaby)","Score 4 (Anderberg)
    "), col=c("springgreen4", "yellowgreen","royalblue3","
    steelblue1"), cex=0.8,lwd=c(2,2))
```

G.4 R code used in subsection 6.3.4

The following R code was used to obtain the ECE plots in subsection 6.3.4.

```
library("readxl")
library(fitdistrplus)

All_Char <- read_excel("Char_Vect_All.xlsx")
df_All_Char <-data.frame(All_Char) #Make dataframe of Excel
    file

Log_Odds=seq(-4,4,0.001)

#ECE of the noninformative system (with SLR=1 always)
ECE_SLR_1<-vector()
for (j in Log_Odds){
  ECE_SLR_1<-append(ECE_SLR_1,-(10^j)/(1+10^j)*log2((10^j)/
      (1+10^j))-1/(1+10^j)*log2(1/(1+10^j)))}

Dist_1s<-dweibull(seq(0,1,0.001), fit.weibullS1s$estimate
    [1], fit.weibullS1s$estimate[2])
Dist_1d<-dnorm(seq(0,1,0.001), fit.normS1d$estimate[1], fit
    .normS1d$estimate[2])
Dist_2s<-dnorm(seq(0,1,0.001), fit.normS2s$estimate[1], fit
    .normS2s$estimate[2])
Dist_2d<-dnorm(seq(0,1,0.001), fit.normS2d$estimate[1], fit
    .normS2d$estimate[2])
Dist_3s<-dgamma(seq(0,1,0.001), fit.gammaS3s$estimate[1],
    fit.gammaS3s$estimate[2])
Dist_3d<-dnorm(seq(0,1,0.001), fit.normS3d$estimate[1], fit
    .normS3d$estimate[2])
Dist_4s<-dbeta(seq(0,1,0.001), fit.betaS4s$estimate[1], fit
    .betaS4s$estimate[2])
Dist_4d<-dbeta(seq(0,1,0.001), fit.betaS4d$estimate[1], fit
    .betaS4d$estimate[2])
```

```
22
23 #Calculate the ECEs for the four SLR systems
24 ECE_1<-vector()
25 ECE_2<-vector()
26 ECE_3<-vector()
27 ECE_4<-vector()
28 for (j in Log_Odds){
29   sum_S1_1=0
30   sum_S1_2=0
31   sum_S1_3=0
32   sum_S1_4=0
33   for (i in 1:length(S1s)){
34     P_s1_h1_1=Dist_1s[S1s[i]/0.001+1]
35     P_s1_h2_1=Dist_1d[S1s[i]/0.001+1]
36     sum_S1_1=sum_S1_1+log2((P_s1_h1_1*10^j)/(P_s1_h2_1+P_
         s1_h1_1*10^j))
37     P_s1_h1_2=Dist_2s[S2s[i]/0.001+1]
38     P_s1_h2_2=Dist_2d[S2s[i]/0.001+1]
39     sum_S1_2=sum_S1_2+log2((P_s1_h1_2*10^j)/(P_s1_h2_2+P_
         s1_h1_2*10^j))
40     P_s1_h1_3=Dist_3s[S3s[i]/0.001+1]
41     P_s1_h2_3=Dist_3d[S3s[i]/0.001+1]
42     sum_S1_3=sum_S1_3+log2((P_s1_h1_3*10^j)/(P_s1_h2_3+P_
         s1_h1_3*10^j))
43     P_s1_h1_4=Dist_4s[S4s[i]/0.001+1]
44     P_s1_h2_4=Dist_4d[S4s[i]/0.001+1]
45     sum_S1_4=sum_S1_4+log2((P_s1_h1_4*10^j)/(P_s1_h2_4+P_
         s1_h1_4*10^j))}
46   sum_S2_1=0
47   sum_S2_2=0
48   sum_S2_3=0
49   sum_S2_4=0
50   for (i in 1:length(S1d)){
51     P_s2_h1_1=Dist_1s[S1d[i]/0.001+1]
52     P_s2_h2_1=Dist_1d[S1d[i]/0.001+1]
53     sum_S2_1=sum_S2_1+log2(P_s2_h2_1/(P_s2_h2_1+P_s2_h1_1*
         10^j))
54     P_s2_h1_2=Dist_2s[S2d[i]/0.001+1]
55     P_s2_h2_2=Dist_2d[S2d[i]/0.001+1]
56     sum_S2_2=sum_S2_2+log2(P_s2_h2_2/(P_s2_h2_2+P_s2_h1_2*
         10^j))
57     P_s2_h1_3=Dist_3s[S3d[i]/0.001+1]
58     P_s2_h2_3=Dist_3d[S3d[i]/0.001+1]
59     sum_S2_3=sum_S2_3+log2(P_s2_h2_3/(P_s2_h2_3+P_s2_h1_3*
         10^j))
60     P_s2_h1_4=Dist_4s[S4d[i]/0.001+1]
61     P_s2_h2_4=Dist_4d[S4d[i]/0.001+1]
62     sum_S2_4=sum_S2_4+log2(P_s2_h2_4/(P_s2_h2_4+P_s2_h1_4*
         10^j))}
63   ECE_1<-append(ECE_1,-(10^j)/(length(S1s)*(1+10^j))*sum_S1
       _1-1/(length(S1d)*(1+10^j))*sum_S2_1)
```

```
64    ECE_2<-append(ECE_2,-(10^j)/(length(S1s)*(1+10^j))*sum_S1
          _2-1/(length(S1d)*(1+10^j))*sum_S2_2)
65    ECE_3<-append(ECE_3,-(10^j)/(length(S1s)*(1+10^j))*sum_S1
          _3-1/(length(S1d)*(1+10^j))*sum_S2_3)
66    ECE_4<-append(ECE_4,-(10^j)/(length(S1s)*(1+10^j))*sum_S1
          _4-1/(length(S1d)*(1+10^j))*sum_S2_4)}
67
68  #Plot the ECE Plots for the four SLR systems
69  plot(Log_Odds,ECE_1,col="springgreen4",type="l",xlab=
        expression("Log"[10]*"(Prior odds)"), ylab="Empirical
        Cross-Entropy", ylim=c(0,1),main="ECE plot of the four
        SLR systems",lwd=2)
70  lines(Log_Odds,ECE_2,col="yellowgreen",lwd=2)
71  lines(Log_Odds,ECE_3,col="royalblue3",lwd=2)
72  lines(Log_Odds,ECE_4,col="steelblue1",lwd=2)
73  lines(Log_Odds,ECE_SLR_1,col="black",lwd=2,lty=2)
74  grid()
75  legend("bottomleft",
76         legend=c("Score 1 (Overlap)", "Score 2 (Goodall3)",
                "Score 3 (Burnaby)","Score 4 (Anderberg)","
                Noninformative SLR system"),
77         col=c("springgreen4", "yellowgreen", "royalblue3",
                "steelblue1", "black"),cex=0.8,lwd=c(2,2,2,2,2),
                lty=c(1,1,1,1,2))
```

Appendix H

Derivations of the formulas in subsection 6.3.4

H.1 Derivation of the formula for the ECE

Prior odds $= \frac{P(H_1)}{P(H_2)} = \frac{P(H_1)}{1-P(H_1)}$ (because $P(H_1)+P(H_2) = 1$ (since H_1 and H_2 are two mutually exclusive hypothesis by definition), thus $P(H_2) = 1-P(H_1)$)

So; $P(H_1) = \text{Prior Odds} \cdot (1 - P(H_1)) = \text{Prior Odds} - P(H_1) \cdot \text{Prior Odds}$

Thus; $\text{Prior Odds} = P(H_1) + P(H_1) \cdot \text{Prior Odds} = P(H_1)(1 + \text{Prior Odds})$

Therefore; $P(H_1) = \frac{\text{Prior Odds}}{1+\text{Prior Odds}} = \frac{10^{\Omega}}{1+10^{\Omega}}$ with $\Omega = \log_{10}\left(\frac{P(H_1)}{P(H_2)}\right)$ (so the logarithm with base 10 of the prior odds)

$P(H_2) = 1-P(H_1) = 1-\frac{\text{Prior Odds}}{1+\text{Prior Odds}} = \frac{1+\text{Prior Odds}-\text{Prior Odds}}{1+\text{Prior Odds}} = \frac{1}{1+\text{Prior Odds}} = \frac{1}{1+10^{\Omega}}$

So;

$$P(H_1) = \frac{10^{\Omega}}{1+10^{\Omega}} \tag{H.1}$$

$$P(H_2) = \frac{1}{1+10^{\Omega}} \tag{H.2}$$

Now; $P(H_1|s) = \frac{P(s|H_1)P(H_1)}{P(s)}$ (by Bayes' Theorem that states that $P(A \mid B) = \frac{P(B|A)\cdot P(A)}{P(B)}$ [10]) $= \frac{P(s|H_1)P(H_1)}{P(s|H_1)P(H_1)+P(s|H_2)P(H_2)}$ (by the law of total probability [11])

So; $P(H_1|s) = \frac{P(s|H_1)\cdot\frac{10^{\Omega}}{1+10^{\Omega}}}{P(s|H_1)\cdot\frac{10^{\Omega}}{1+10^{\Omega}}+P(s|H_2)\cdot\frac{1}{1+10^{\Omega}}}$ (by equations (H.1) and (H.2))

So;

$$P(H_1|s) = \frac{P(s|H_1)\cdot 10^{\Omega}}{P(s|H_1)\cdot 10^{\Omega} + P(s|H_2)} \tag{H.3}$$

And in the same way;

$$P(H_2|s) = \frac{P(s|H_2)}{P(s|H_1) \cdot 10^{\Omega} + P(s|H_2)} \tag{H.4}$$

Therefore; $ECE = -\frac{P(H_1)}{n_1} \sum_{s \in S_1} \log_2 P(H_1|s) - \frac{P(H_2)}{n_2} \sum_{s \in S_2} \log_2 P(H_2|s) =$
$-\frac{10^{\Omega}}{n_1(1+10^{\Omega})} \sum_{s \in S_1} \log_2 \left(\frac{P(s|H_1) \cdot 10^{\Omega}}{P(s|H_1) \cdot 10^{\Omega} + P(s|H_2)} \right) - \frac{1}{n_2(1+10^{\Omega})} \sum_{s \in S_2} \log_2 \left(\frac{P(s|H_2)}{P(s|H_1) \cdot 10^{\Omega} + P(s|H_2)} \right)$
(by equations (H.1), (H.2), (H.3) and (H.4))

So;

$$ECE = -\frac{10^{\Omega}}{n_1(1+10^{\Omega})} \sum_{s \in S_1} \log_2 \left(\frac{P(s|H_1) \cdot 10^{\Omega}}{P(s|H_1) \cdot 10^{\Omega} + P(s|H_2)} \right) - \frac{1}{n_2(1+10^{\Omega})} \sum_{s \in S_2} \log_2 \left(\frac{P(s|H_2)}{P(s|H_1) \cdot 10^{\Omega} + P(s|H_2)} \right)$$

H.2 Derivation of the formula for the ECE of a noninformative SLR system

In the case of a noninformative SLR system, the SLR is always equal to 1. So, $SLR = \frac{P(s|H_1)}{P(s|H_2)} = 1$ for all s (this formula was given in chapter 2). Thus, $P(s \mid H_1) = P(s \mid H_2)$.

So; $ECE = -\frac{10^{\Omega}}{n_1(1+10^{\Omega})} \sum_{s \in S_1} \log_2 \left(\frac{P(s|H_1) \cdot 10^{\Omega}}{P(s|H_1) \cdot 10^{\Omega} + P(s|H_2)} \right) - \frac{1}{n_2(1+10^{\Omega})} \sum_{s \in S_2} \log_2 \left(\frac{P(s|H_2)}{P(s|H_1) \cdot 10^{\Omega} + P(s|H_2)} \right)$
(See previous section)
$= -\frac{10^{\Omega}}{n_1(1+10^{\Omega})} \sum_{s \in S_1} \log_2 \left(\frac{P(s|H_1) \cdot 10^{\Omega}}{P(s|H_1) \cdot 10^{\Omega} + P(s|H_1)} \right) - \frac{1}{n_2(1+10^{\Omega})} \sum_{s \in S_2} \log_2 \left(\frac{P(s|H_2)}{P(s|H_2) \cdot 10^{\Omega} + P(s|H_2)} \right)$
$= -\frac{10^{\Omega}}{n_1(1+10^{\Omega})} \sum_{s \in S_1} \log_2 \left(\frac{P(s|H_1) \cdot 10^{\Omega}}{P(s|H_1) \cdot (10^{\Omega}+1)} \right) - \frac{1}{n_2(1+10^{\Omega})} \sum_{s \in S_2} \log_2 \left(\frac{P(s|H_2)}{P(s|H_2) \cdot (10^{\Omega}+1)} \right)$
$= -\frac{10^{\Omega}}{n_1(1+10^{\Omega})} \sum_{s \in S_1} \log_2 \left(\frac{10^{\Omega}}{10^{\Omega}+1} \right) - \frac{1}{n_2(1+10^{\Omega})} \sum_{s \in S_2} \log_2 \left(\frac{1}{10^{\Omega}+1} \right)$
$= -\frac{10^{\Omega}}{n_1(1+10^{\Omega})} \cdot n_1 \cdot \log_2 \left(\frac{10^{\Omega}}{10^{\Omega}+1} \right) - \frac{1}{n_2(1+10^{\Omega})} \cdot n_2 \cdot \log_2 \left(\frac{1}{10^{\Omega}+1} \right)$ (because $\log_2 \left(\frac{10^{\Omega}}{10^{\Omega}+1} \right)$ and $\log_2 \left(\frac{1}{10^{\Omega}+1} \right)$ do not depend on s and because the sizes of S_1 and S_2 are n_1 and n_2 respectively (this was defined in subsection 6.3.4))

Therefore;

$$ECE = -\frac{10^{\Omega}}{1+10^{\Omega}} \cdot \log_2 \left(\frac{10^{\Omega}}{10^{\Omega}+1} \right) - \frac{1}{1+10^{\Omega}} \cdot \log_2 \left(\frac{1}{10^{\Omega}+1} \right)$$

(Note that this is equal to $ECE = -P(H_1) \cdot \log_2 (P(H_1)) - P(H_2) \cdot \log_2 (P(H_2))$)