Albert Ludwig University of Freiburg
Department of Computer Science

Cross-Dating of Intra-Annual Wood Density Series

Master Thesis

Due Date: 5th September 2018

Author:
Alexander Mattheis

Supervisors:
Dr. Martin Raden
PD Dr. Hans-Peter Kahle

Reviewers:
Prof. Dr. Rolf Backofen
PD Dr. Hans-Peter Kahle

A Thesis in the Chair of the
Bioinformatics Group
of Prof. Dr. Rolf Backofen

Selbständigkeitserklärung

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen/Hilfsmittel verwendet habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüber hinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, bereits für eine andere Prüfung angefertigt wurde.

.....	.....
Ort, Datum	Unterschrift

Abstract

The calendar year for a tree sample can be determined by its annual rings. By measuring the widths or maximum densities of the individual rings and aligning them against a master chronology with known dates, one can determine the calendar year of a piece of wood exactly. But it is problematic, if the piece of wood is very short and thus contains only few annual rings. In that case the amount of information derived from series of widths or maximum densities is not sufficient for the determination of a correct date.

This has led to the key question, whether one can determine the calendar year for a small piece of wood with the help of intra-annual data, like ring density-profiles. If yes, then how well works this new method compared to established approaches? The thesis is that density-profile-based approaches have to provide as good or better results since profiles contain much more information than other measured characteristics.

Within this study, this thesis could be confirmed. Several new approaches were found that have outperformed established, earlier methods. The approach producing the highest number of correct solutions is a combination of an established procedure with a newly developed one called the Bucket Approach.

Table of Contents

List of Abbrevations **iii**

1 Introduction **1**
1.1 Structure . . . 1
1.2 Motivation . . . 1
1.3 Introduction to Dendrochronology . . . 1
1.3.1 Dating Techniques . . . 2
1.3.2 Historical Background . . . 3
1.3.3 Dating with Wood Density . . . 4

2 Theoretical Foundations **5**
2.1 Chronology Computation . . . 5
2.1.1 Mean Calculation . . . 5
2.1.2 Multiple Interval-Based Curve Alignment . . . 6
2.2 Measurement Techniques . . . 9
2.2.1 Distance Functions & Normalization . . . 9
2.2.2 Similarity Measurement . . . 12
2.2.3 Quality and Reliability . . . 15

3 Approaches and Evaluation **18**
3.1 Data Acquisition . . . 18
3.1.1 Datasets . . . 18
3.1.2 Generation of Samples & Chronologies . . . 20
3.1.3 Discussion . . . 20
3.2 Points-Based Approaches . . . 21
3.2.1 Methods . . . 21
3.2.2 Results . . . 22
3.2.3 Discussion . . . 24
3.3 Consensus Approach . . . 25
3.3.1 Methods . . . 25
3.3.2 Results . . . 27
3.3.3 Discussion . . . 29
3.4 Per-Tree Approach . . . 31
3.4.1 Methods . . . 31
3.4.2 Results . . . 33
3.4.3 Discussion . . . 36
3.5 Bucket Approach . . . 37
3.5.1 Methods . . . 37
3.5.2 Results . . . 38
3.5.3 Discussion . . . 46
3.6 Two-Step Approach . . . 48
3.6.1 Methods . . . 48

3.6.2 Results . . . 49
3.6.3 Discussion . . . 51
3.7 Voting Approach . . . 53
3.7.1 Methods . . . 53
3.7.2 Results . . . 56
3.7.3 Discussion . . . 61
3.8 Overview of Approaches . . . 63
3.8.1 Length Dependency . . . 63
3.8.2 Runtime Comparison . . . 64

4 Conclusion **66**
4.1 Achievements . . . 66
4.2 Future Work . . . 66

References **70**

Tools **74**
Libraries . . . 75
Hardware . . . 75

Zusammenfassung **76**

5 Appendix **77**
5.1 MICA Parameters . . . 77
5.1.1 Per Tree Consensi . . . 77
5.1.2 Final Consensus . . . 77
5.2 Additional Results . . . 78
5.2.1 Points-Based Approaches . . . 78
5.3 Implementation . . . 79
5.3.1 Modules . . . 79
5.3.2 Classes . . . 80
5.3.3 Processes . . . 84

Acknowledgements **86**

List of Abbrevations

APD	**A**verage **P**oint-**D**istance
APDN	**A**verage **P**oint-**D**istance with **N**ormalized Profiles
SAPD	**S**lope-Based **A**verage **P**oint-**D**istance
SAPDN	**S**lope-Based **A**verage **P**oint-**D**istance with **N**ormalized Profiles

1 Introduction

1.1 Structure

After a general Motivation and Introduction to Dendrochronology in this first chapter, the Theoretical Foundations are formally described. Thereafter, different methods are tested and discussed in the chapter about the Approaches and Evaluation. Achievements, as well as possible future work are finally summarized in the Conclusion. The Implementation is presented at the end, in the Appendix. It contains information about an available R interface for the approaches, as well as the generation of scores.

1.2 Motivation

Assume one has a piece of wood from an old building on which the different annual tree rings are recognizable. Each tree ring corresponds to a year and trees from the same climatic regions often show similar characteristics within their tree rings. Such properties of trees can be used to determine the correct age of a wood sample and thus the construction year of a building. But if the piece of wood is short, there is often no chance to find out the year with the established, ring-width or maximum density based approaches. In order to change this, intra-annual data is used. And that are density-profiles of tree rings which allow for the determination much shorter wood pieces than with given methods used before. Concretely, a famous Biology Professor named Edward Cook wrote 1990 that for meaningful results depending on the quality and type of wood by experience at least 40 rings are necessary [10]. However, in this thesis samples with 10 rings and below are dated!

1.3 Introduction to Dendrochronology

Dendrochronology[1] is the determination of the tree rings corresponding years. The wood properties of different trees from different areas in the same climatic region show the same weather-related characteristics in their annual rings. So for example, the ring-width proportions of different trees are similar. Abrupt cold spells or heatwaves are recognizable in the rings of several trees and all that information can be used to date wood samples. An example, in the summer of 2003 there were major heatwaves and droughts over a bigger period of time in Germany. If now trees are felled 2018, then these droughts will be recognizable in the "same" year ring of multiple trees i.e. the ring-widths turn out smaller in 2003 because of growth slumps caused by water lacks. So if one gets a wood sample from Germany with the year rings from 1992 to 2010 from one of these trees, then the felling year for a tree could be still determined. One has just to compare with the wood patterns of the other trees because the ring-width proportions around 2003 will be similar.

[1] *dendron*: gr. tree, *chronos*: gr. time/year and *logia*: gr. word/teaching/logic,
see `https://translate.google.de/`, so dendrochronology is the tree-year or tree-time teaching

1.3.1 Dating Techniques

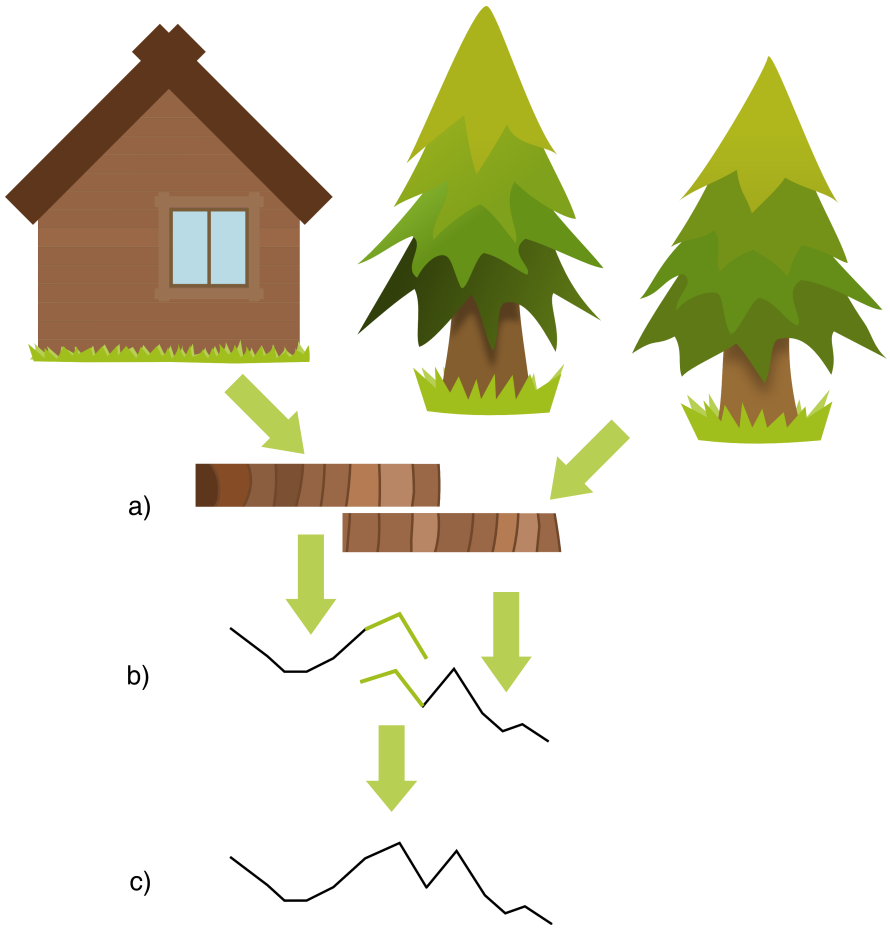

Figure 1.3.1.1 The creation of a master chronology. **a)** take wood samples, **b)** plot ring-width curves, **c)** combine ring-width curves to a master chronology (adapted from [41, p. 85])

Over the past 120 years multiple methods for the dating of organic material were developed or made popular, as examples

- Cross-Dating by Andrew E. Douglass (early 20th century)
- Radiocarbon Dating by Willard F. Libby (1946)
- Amino Acid Dating by Philipp H. Abelson (1954)

which can also be used to determine the age of wood [30; 35]. The dendrochronological main method for dating of wood samples is cross-dating. Therefore, the individual ring-widths of a tree stump were measured from inside the stump to outside under a lens [41, p. 61] or nowadays with computer technology. These ring-widths are then

displayed in a graph over time, more precisely years. Cross-dating is called cross-dating because several ring-width curves are pair-wisely crossed. They are shifted against each other to find matches within the patterns and to build a so-called master chronology (Fig. 1.3.1.1). This chronology can be built by averaging over multiple correctly aligned ring-width curves with known ages. Besides that, maximum densities and other annual parameters of rings can be cross-dated. To find an overlap of years in which a ring-width sample and a subseries from a chronology show very similar shapes, t-values or correlation coefficients between both discrete curves can be computed. So at each position or year within the chronology a window of sample-length i.e. with a certain number of ring-widths has to be cut out and a statistical value has to be measured. By this, the correct sample start-year can be examined under consideration of the shift in the chronology for which the statistical value reaches its maximum. And with the sample start-year, the end-year and thus the potential felling year for a tree can be calculated. Other techniques than the dendrological ones cannot determine the years exactly. That is the main advantage of using dendrochronology!

The current Radiocarbon Dating method from 1977 has a different idea. There are three naturally occurring variants of the carbon atom, so-called isotopes. And one of these isotopes ^{14}C is due to its rapid radioactive decay well suited to date dead organisms [35]. The isotope is formed in the stratosphere i.e. the upper part of the atmosphere by certain chemical processes and included due to photosynthesis in the form of $^{14}CO_2$ in plants. Thus, it is also stored in trees. After the death of an organism no $^{14}CO_2$ and thus no radioactive ^{14}C isotope is included in the plant anymore, the number of ^{14}C atoms decreases as the ^{14}C isotope decays i.e. transforms into other elements. If one measures the ratio to the stable carbon isotope ^{12}C found in the organism, then the age of the organism can be recalculated on the basis of the ^{14}C half-life[1] of about 5730 years [31].

Amino-Acid Dating follows a similar idea as the Radiocarbon Dating method from 1977. It looks at some ratio with respect to a reference value. Here, it is looked on the ratio of certain L-Amino Acid formations to D-Amino Acid formations. After the death of an organism, this ratio changes and based on this, the age of the organism can be determined [30].

1.3.2 Historical Background

The first one who has verifiably described tree rings was Theophrastus of Eresus (c. 372-287 BC) [44], a pupil of Aristotle (384-322 BC [37]), the Greek philosopher and savant. But Theophrastus was presumably also not the first, because he was taught by Aristotle and he had access to the rich literature of that time. Today still manuscript copies and translations of Theophrastus' writings in English and German exist. Most of what he has wrote, is probably knowledge that has been passed many centuries by ancient farmers. It is most likely not from his own research. So it is known that Greeks

[1] the time after which half of the atoms of a sample have decayed to other elements

knew about the existence of tree rings, but it is not known if they have realized that each year a new ring is added [27]. However, from the 12th century, a Chinese story by Hong Mai (1123-1202 AD [24]) is delivered "The Cunninghamia Tree on Chhên's Tomb" [32]. In this story the protagonist had a dream in which he has been warned to fell a 380-year-old tree. And a few years later after the tree was felled, it has been noticed that the tree really had 380 rings. So the importance of tree rings seemed to be already familiar in ancient China in the 12th century. In the 13th century, Albertus Magnus (c. 1200-1280 AD [2]), a German bishop and savant presumably described the layered growth from inside to outside. And the first good evidence that the annual character of tree rings was recognized, can be found in Leonardo da Vinci's (1452-1519 AD) book on panting "Trattato della Pittura"[1] [12; 27; 42]. He has even made the connection between wet and dry years, so he wrote that the wet years led to larger annual rings than the dry years. The development of dendrochronology in the next four centuries took mainly place in Europe [42; 41, pp. 218-222]. After a journey, Michel de Montaignes (1533-1592 AD [5]) reported about his new knowledge from a craftsman; the number of tree rings corresponds to the age of the tree. Later, also connections between annual weather events such as frost and hail were recognized and compared with meteorological data. Thereafter, it was the turn of Andrew E. Douglass (1867-1962), an American astronomer who made dendrochronology popular. With his help ancient archaeological findings were backdated year-exactly on the basis of annual rings. That is how the research in dendrochronology has started, and it proceeds developing up today with this thesis and research in numerous research groups.

1.3.3 Dating with Wood Density

In the year 1970, it was indirectly mentioned by Hubert Polge that density-curves of tree rings could have advantages for wood-dating due to the fact that such curves are encoding lots of parameters like ring-width and cell-wall thickness in a single characteristic [33]. Probably, because of costs and high time consumption, density-based techniques were not extended for a long time [14]. But today, there exist less expensive and fast methods to measure the density of wood samples like the High-Frequency Densitometry [39; 50] by Martin Schinker et al.. So there is an occasion to develop an algorithmic method for cross-dating with density-curves. The approach presented in this work uses therefore among others the landmark approach MICA [4; 29] developed by Martin Raden et al. and Matthias Beck [3] for the computation of time-synchronous, averaged density-series. Apart from that, all here presented methods were newly developed for this master thesis or originating from established, statistical and partially from dendrochronological methods [41].

[1] engl. Treatise on Painting, see `https://translate.google.de/`

2 Theoretical Foundations

2.1 Chronology Computation

2.1.1 Mean Calculation

In points-based chronologies values v_i which correspond to ring-widths or maximum densities are averaged. The mean of the values from a year is represented by $\bar{v}_i$.

Definition 2.1.1.1 (Points Chronology)
A Points Chronology C^v *is a series of annual means:*

$$\mathrm{C}^v = \bar{v}_1 \bar{v}_2 \dots \bar{v}_N$$

where N *is the number of monotonously increasing years.*

But the mean has decisive problems with outliers. If some values are significantly larger than the values of the majority, the whole mean is disturbed. An idea here is therefore to apply a so-called robust mean computation. In the robust method, the outliers are filtered out. How does such a filtering works? There exist different approaches, but a common one, especially in dendrochronology is Tukey's Biweight Robust Mean Estimation [1; 23]. For the estimation preferentially, so-called Maximum Likelihood Type Estimators (M-estimators) are used [23, pp. 43, 146]. But this estimation method has a drawback, several iterations are necessary to compute the robust mean numerically. Instead, in current implementations so-called one-step W-estimators [1; 23, p. 164] are applied to compute the mean in a single pass. W-estimators compute weights using a bisquare function w. These weights are used to weight the individual components with which the mean is computed depending on their distance from the median. By this, components significantly far from the median can get a weight of zero, so they are filtered out. The final mean is now defined as the weighted average of v-components (Def. 2.1.1.2).

Definition 2.1.1.2 (Tukey's Biweight Robust Mean)
Given values from a finite set $\mathbf{V}$*, then the robust mean of the values is*

$$\bar{v} = \frac{\sum_{v \in \mathbf{V}} w(\zeta) \cdot v}{\sum_{v \in \mathbf{V}} w(\zeta)}$$

with standardized values

$$\zeta = \frac{v - \tilde{v}}{c_{tun} \cdot \mathrm{MAD}_v + \varepsilon}$$

and weightings

$$w(\zeta) = \begin{cases} \left(1 - \zeta^2\right)^2 & , |\zeta| \leq 1 \\ 0 & , |\zeta| > 1 \end{cases}$$

whereas $\mathrm{MAD}_v = \underset{v \in \mathbf{V}}{\mathrm{median}} \left\{ v - \tilde{v} \right\}$ *is the median absolute deviation,* $\tilde{v}$ *is the median of all* $v \in \mathbf{V}$*,* ε *and* c_{tun} *are constants, and* ζ *a standardized score.*

In literature Tukey's robust mean is abbreviated with a big T. The constant ε is set to 0.0001 as suggested in [1] to avoid a division by zero. It is recommended to set the tuning parameter c_{tun} to 9, since then values below and above $-/+6$ standard deviations are set to zero [16] i.e. it holds that the expected value $\mathrm{E}[9 \cdot \mathrm{MAD}_v] \approx 6 \cdot \sigma_v$ (6 times standard deviation of v). Usually it is only divided by the dispersion measure (in this case MAD_v) as it is known from z-scores [49, p. 88], but here additionally the tuning parameter c_{tun}, as well as ε are used to fix the mathematical issues with the formula.

2.1.2 Multiple Interval-Based Curve Alignment

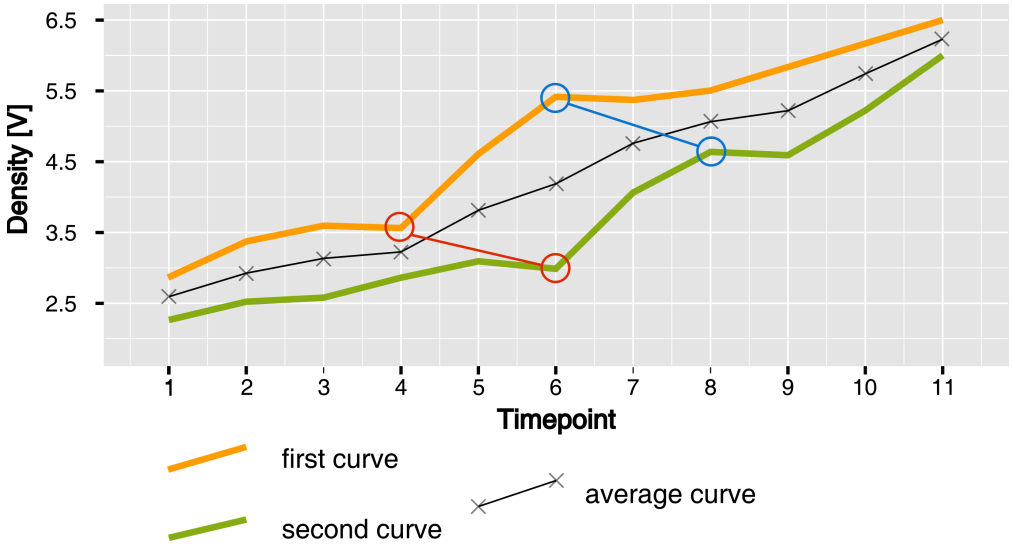

Figure 2.1.2.1 The problem in consensus-computation. Blue and red marked places have to be aligned before building the average (adapted from [3]).

In 2011, an approach for the microstructure alignment of density-profiles i.e. the density-curves of a tree ring was published [4]. For annual rings from stem-discs of Douglas-fir trees, densities $\rho = \frac{m}{V}$ (with mass m and volume V) were measured along eight radial directions and series of wood density-profiles were created from the pith to the bark with High-Frequency Densitometry [39; 50]. Due to the method which was used, the density-profile values were given in volts V. But they also could be transformed due to a linear correlation[1], into volumetric density $\frac{\mathrm{kg}}{\mathrm{m}^3}$ or gravimetric density (GD)[2]. That is the percentage of for example hydrogen in relation to the total mass [46]. The idea was to reconstruct e.g. the growth behaviour and thus, if possible, climatic conditions from a series of tree ring density-profiles. Because density information along several directions is needed to provide accurate inferences about the growth behaviour, that is a difficult task. The stem growth-speed varies in different, horizontal directions.

[1] personal communication with Dr. Martin Raden
[2] *gravitas*: lat. heaviness, *métron*: gr. measure, see `https://translate.google.de/`

So it is possible, that two density-profiles have a similar shape but different lengths. Therefore, it makes no sense to simply average the density-profiles, since the average would not correctly reflect the growth behaviour as it can be recognized by the black curve in Fig. 2.1.2.1. That means characteristic points like inflection points or maxima of different density-profiles have to be aligned and intermediate points must be interpolated. Afterwards, the average curve can be formed, a so-called consensus[1]. For the alignment part, the MICA algorithm (Multiple Interval-based Curve Alignment) was developed. Thereby x-coordinates of characteristic points within wood density-profiles, so-called landmarks or reference points, are aligned to obtain a time-synchronous alignment of the measured values e.g. from a set of trees. Whereby time-synchronous means that the temporal events like droughts, etc. in different profiles are realigned again i.e. they are getting the same relative x-coordinate.

Definition 2.1.2.1 (Density-Profile)
Discrete, finite series with $n > 2$ points:

$$\mathrm{P} = p_1 p_2 ... p_n$$

where p_i^x is the relative x-coordinate with $p_i^x \in [0,1] \subset \mathbb{Q}$, such that $p_1^x = 0$, $p_n^x = 1$, $\forall i < n : p_i^x < p_{i+1}^x$, and the signal $p_i^y \in \mathbb{Q}$ as the y-coordinate.

The number of points in a profile is described with the length-function "len" and sequences of profiles are also called density series. The functions $x(\mathrm{P}) = p_1^x p_2^x ... p_n^x$, $y(\mathrm{P}) = p_1^y p_2^y ... p_n^y$ return the series of x and y coordinates in a profile and for readability these series are abbreviated with $\mathrm{X} = x_1 x_2 ... x_n$, $\mathrm{Y} = y_1 y_2 ... y_n$. Additionally, there is a function $s(\mathrm{P})$ returning the slopes s_i for each point $p_i \in \mathrm{P}$.

Definition 2.1.2.2 (Interpolation)
An Interpolation is a function $\alpha_k(\mathrm{P}) = \hat{\mathrm{P}}$ with profiles $\mathrm{P}, \hat{\mathrm{P}}$ having lengths $\mathrm{len}(\mathrm{P}) = n$, and $\mathrm{len}(\hat{\mathrm{P}}) = k$, where $\hat{\mathrm{P}}$ consists out of k uniformly distributed points that were linearly interpolated into the interval of $x(\mathrm{P})$.

Definition 2.1.2.3 (MICA-Alignment)
A MICA-Alignment[2] is the output of function $\alpha_k^{MICA}(\mathrm{P}^a, \mathrm{P}^b) = (\hat{\mathrm{P}}^a, \hat{\mathrm{P}}^b)$ with two profiles of arbitrary length $\mathrm{P}^a, \mathrm{P}^b$ as inputs returning two x-coordinate-aligned, and linearly interpolated profiles $\hat{\mathrm{P}}^a, \hat{\mathrm{P}}^b$ as outputs, where $\hat{\mathrm{P}}^a$, $\hat{\mathrm{P}}^b$ consist out of k uniformly distributed points in the intervals of $x(\mathrm{P}^a)$ and $x(\mathrm{P}^b)$.

Operations like division by constants and addition of profiles are defined component-wise, as with vectors, on the y-values i.e given profiles $\mathrm{P}^a, \mathrm{P}^b$ linearly interpolated to same length k, so $\mathrm{len}(\mathrm{P}^a) = \mathrm{len}(\mathrm{P}^b) = k$. Then with $\mathrm{P} = \mathrm{P}^a + \mathrm{P}^b$ the y-coordinates of both profiles are added component-wise together. So $\mathrm{Y} = y(\mathrm{P}^a) + y(\mathrm{P}^b) = (y_1^a +$

[1] *consensus*: agreement, see `http://www.thesaurus.com/browse/consensus`
[2] the original MICA approach from [29] does not automatically interpolate to same length

$y_1^b)(y_2^a + y_2^b)...(y_k^a + y_k^b)$ and one of the interpolated series of equidistant x-coordinates is then set as $x(\mathrm{P})$ e.g. $x(\mathrm{P}) = \mathrm{X}^a$, and $y(\mathrm{P}) = \mathrm{Y}$ to create a new profile P. In the following, it holds that profiles having the same length are linearly interpolated (Def. 2.1.2.2). Division i.e. $\frac{\mathrm{P}}{c}$ means that every y-component in P is divided by a constant $c \in \mathbb{Q}$. Further the term "bucket" has to be stated more precisely. This is just a finite, non-empty set. There are buckets of profiles, abbreviated with $\mathbf{B}$ and buckets of corresponding characteristics like ring-widths or maximum-densities for which no symbols are officially introduced.

Definition 2.1.2.4 (Consensus-Profile)
Average profile $\overline{\mathrm{P}}$ *built with a set of profiles interpolated to same length:*

$$\overline{\mathrm{P}} = \frac{1}{|\mathbf{B}|} \sum_{\mathrm{P} \in \mathbf{B}} \mathrm{P}$$

where $\mathbf{B}$ *is a bucket with profiles from the same year.*

Thus, for a Consensus-Profile, the individual y-values of the profiles are added together and divided by the number of profiles.

Definition 2.1.2.5 (Consensus Chronology)
A Consensus Chronology C^{P} *is a series of Consensus-Profiles:*

$$\mathrm{C}^{\mathrm{P}} = \overline{\mathrm{P}}_1 \overline{\mathrm{P}}_2 \dots \overline{\mathrm{P}}_N$$

where N *is the number of monotonously increasing years.*

But chronologies can also be defined on the individual buckets instead of profiles. That allows working on the bucket-profiles itself instead of their consensus i.e. it allows comparisons of some sample-profile against all profiles in a specific year.

Definition 2.1.2.6 (Buckets Chronology)
A Buckets Chronology $\mathrm{C}^{\mathbf{B}}$ *is a series of buckets:*

$$\mathrm{C}^{\mathbf{B}} = \mathbf{B}_1 \mathbf{B}_2 \dots \mathbf{B}_N$$

where N *is the number of monotonously increasing years.*

So a Buckets Chronology is just a series of buckets where each bucket contains the profiles for some year.

2.2 Measurement Techniques

2.2.1 Distance Functions & Normalization

Distance-functions are necessary to evaluate the similarity of two density-profiles. They are used to do cross-dating with density-profiles (Def. 2.1.2.1). So the best match between a series of profiles (the pattern or sample) and a master chronology C is requested. The distance function provides a measure for the quality of a match between the series of profiles and a subseries from the master chronology. To compare two profiles, a distance function measures therefore the distance between profiles that have been interpolated to the same number of points (Def. 2.1.2.2). For patterns with multiple profiles, the distances generated with the individual profiles can e.g. be summed up.

However, if profiles should be compared without multiplicative or additive differences, then the common way is to subtract the mean from values $y_i \in \mathrm{Y}$ and then divide by the standard deviation [49, p. 290]. The subtraction leads to a removal of additive differences i.e. if it there the y-values $\mathrm{Y}^a = y_1^a y_2^a ... y_n^a$, $\mathrm{Y}^b = (y_1^a + \delta)(y_2^a + \delta)...(y_n^a + \delta)$ with $\delta \in \mathbb{Q}$ of two profiles are given, then after subtracting the average y-value mean(Y) from each y_i, both profiles get the same y-coordinates and thus the y-values Y^a, Y^b become identical. Multiplicative differences from profiles can be removed by division with the standard deviation, since by this operation it is also divided by the magnitude $\Big\|\mathrm{Y} - \mathrm{seq}\Big(\mathrm{mean}(\mathrm{Y})\Big)\Big\|$. Consider sequences Y as vectors. So if there are two vectors and it is not known, if they point in the same direction, both can be divided by their magnitudes and it can be proved then whether the resulting vectors are equal. Exactly this idea was applied on the y-values of profiles.

$$
\begin{aligned}
y_i^{std} &= \frac{y_i - \mathrm{mean}(\mathrm{Y})}{\sigma(\mathrm{Y})} \\
&= \frac{y_i - \mathrm{mean}(\mathrm{Y})}{\sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^{n} \Big(y_i - \mathrm{mean}(\mathrm{Y})\Big)^2}} \\
&= \frac{y_i - \mathrm{mean}(\mathrm{Y})}{\sqrt{\frac{1}{n-1}} \cdot \sqrt{\sum_{i=1}^{n} \Big(y_i - \mathrm{mean}(\mathrm{Y})\Big)^2}} \\
&= \frac{y_i - \mathrm{mean}(\mathrm{Y})}{c \cdot \Big\|\mathrm{Y} - \mathrm{seq}\Big(\mathrm{mean}(\mathrm{Y})\Big)\Big\|}
\end{aligned}
\qquad (2.2.1.1)
$$

Normalization by Standard Deviation

where $\mathrm{seq}\Big(\mathrm{mean}(\mathrm{Y})\Big)$ is a series of mean(Y)-entries of the same dimensions as Y.

So the magnitude is weighted by a constant $c = \sqrt{\frac{1}{n-1}}$, where n is the number of y-values in the profile.

Definition 2.2.1.1 (z-Score)
Given a series $\mathrm{Y} = y_1 y_2 ... y_n$ *and a value* $y_i \in \mathrm{Y}$*, then* $y_i^{std} = \frac{y_i - \mathrm{mean(Y)}}{\sigma(\mathrm{Y})}$ *is called a z-score.*

The normalization by computing z-scores is only done on the y-coordinates of profiles $y_i \in \mathrm{Y}$ but it does also effect on the slopes of a profile P. To measure the difference between profiles, the distance on the y-values or on the slopes can be computed. Therefore, one linearly interpolates (Def. 2.1.2.2) two profiles to the same length, sums up all absolute distances e.g. $\left|y_i^a - y_i^b\right|$ and divide by the number of points k i.e. the number of coordinates (Def. 2.2.1.2). Further, it is assumed that profiles for which the distance is measured always have the same length!

The slope for each $p_i \in \mathrm{P}$ can be calculated by computing a linear model i.e. a line is set through $y_{i-2}...y_i...y_{i+2}$ and then the slope of this line is stored as s_i. So a simulated tangent is moved through $y_i \in \mathrm{Y}$ from left to the right and all slopes of that line are stored.

Definition 2.2.1.2 (Average Point-Distance)
Given two profiles P^a, P^b *with* $k = \mathrm{len}(\mathrm{P}^a) = \mathrm{len}(\mathrm{P}^b)$*, the Average Point-Distance is*

$$\mathrm{dist}_{avg}^{y}\Big(\mathrm{P}^a, \mathrm{P}^b\Big) = \frac{1}{k}\sum_{i=1}^{k}\Big|y(p_i^a) - y(p_i^b)\Big|.$$

The distance function (Def. 2.2.1.2) can also be defined on the slopes as it can be seen in the next definition (Def. 2.2.1.3).

Definition 2.2.1.3 (Slope-Based Average Point-Distance)
Given two profiles P^a, P^b *with* $k = \mathrm{len}(\mathrm{P}^a) = \mathrm{len}(\mathrm{P}^b)$*, the Slope-Based Average Point-Distance is*

$$\mathrm{dist}_{avg}^{s}\Big(\mathrm{P}^a, \mathrm{P}^b\Big) = \frac{1}{k}\sum_{i=1}^{k}\Big|s(p_i^a) - s(p_i^b)\Big|.$$

With the help of the Average Point-Distance it is now possible to define the Mean-Distance (Def. 2.2.1.4) which allows to measure the similarity of a profile within a bucket of profiles.

Definition 2.2.1.4 (Mean-Distance)
Given a bucket $\mathbf{B}$*, the Mean-Distance of a profile* $\hat{\mathrm{P}} \notin \mathbf{B}$ *to a set of profiles* $\mathrm{P} \in \mathbf{B}$ *is*

$$\overline{\mathrm{dist}}_{avg}^{y}\Big(\hat{\mathrm{P}}, \mathbf{B}\Big) = \frac{1}{|\mathbf{B}|}\sum_{\mathrm{P}\in\mathbf{B}}\mathrm{dist}_{avg}^{y}\Big(\hat{\mathrm{P}}, \mathrm{P}\Big).$$

The Mean-Distance can be applied on all profiles within a bucket to divide profiles into two clusters. In this process the distances of a profile to all other profiles are measured and then the sum of distances is divided by the number of measurements.

Definition 2.2.1.5 (Sample Distance)
Given two samples $\mathrm{S}^a = \mathrm{P}_1^a\mathrm{P}_2^a...\mathrm{P}_K^a$ *and* $\mathrm{S}^b = \mathrm{P}_1^b\mathrm{P}_2^b...\mathrm{P}_K^b$*, the Sample Distance is*

$$\mathrm{dist}_{avg}^{\gamma}\Big(\mathrm{S}^a, \mathrm{S}^b\Big) = \sum_{1 \leq i \leq K} \mathrm{dist}_{avg}^{\gamma}\Big(\mathrm{P}_i^a, \mathrm{P}_i^b\Big)$$

with $\mathrm{P}_i^a, \mathrm{P}_i^b$ *as the profiles at sequence position* i*, a coordinate-type* $\gamma \in \{y, s\}$ *and* K *as the number of profiles, as well as a distance function* $\mathrm{dist}_{avg}^{\gamma}\Big(\mathrm{P}_i^a, \mathrm{P}_i^b\Big)$ *(Def. 2.2.1.2 and Def. 2.2.1.3).*

The Sample Distance (Def. 2.2.1.5) computes a result by summing up all pairwise profile-distances from both samples. And instead of computing the distance between two samples, it is also possible to measure the distance between a sample and a sequence of buckets. Instead of just summing up distances, it can be applied a function on the generated profile-wise scores of a sample i.e. the different scores could be for example weighted differently before summing them up. This leads to the Generalized Sample-Bucket Distance from Def. 2.2.1.6.

Definition 2.2.1.6 (Generalized Sample-Bucket Distance)
Given a sample $\mathrm{S} = \mathrm{P}_1\mathrm{P}_2...\mathrm{P}_K$ *and a bucket-series* $\mathrm{S}^{\mathbf{B}} = \mathbf{B}_1...\mathbf{B}_K$*, the Generalized Sample-Bucket Distance is*

$$\mathrm{dist}_{avg}^{\gamma}\Big(\mathrm{S}, \mathrm{S}^{\mathbf{B}}, \mathrm{func}\Big) = \underset{1 \leq i \leq K}{\mathrm{func}}\left\{\min_{\mathrm{P} \in \mathbf{B}_i}\left\{\mathrm{dist}_{avg}^{\gamma}\Big(\mathrm{P}_i, \mathrm{P}\Big)\right\}\right\}$$

with $\mathrm{P}_i, \mathbf{B}_i$ *as profiles or buckets at sequence position* i*, a coordinate-type* $\gamma \in \{y, s\}$*,* K *as the number of profiles in* S *and buckets in* $\mathrm{S}^{\mathbf{B}}$*. The minimum-function* min *returns the minimum out of a set of scores, whereas* func *is an arbitrary function applied on the minimum-scores that were generated by the minimum-function* min*. An abbreviation for this distance-function is* $\mathrm{dist}_{avg}^{\gamma}\Big(\mathrm{S}, \mathrm{S}^{\mathbf{B}}\Big)$*. Distances* $\mathrm{dist}_{avg}^{\gamma}\Big(\mathrm{P}_i, \mathrm{P}\Big)$ *were defined in Def. 2.2.1.2 and Def. 2.2.1.3.*

The Sample-Bucket Distance (Def. 2.2.1.7) is a special case of 2.2.1.6 in which all scores produced with a sample are just summed up. So a distance between a sample S and a bucket-series $\mathrm{S}^{\mathbf{B}}$ can be calculated by the addition of scores created by applying a function "min" on a set of scores. This set of scores is generated by the computation of distances between a profile P_i and profiles P from a bucket $\mathbf{B}_i$.

Definition 2.2.1.7 (Sample-Bucket Distance)
Given a sample $\mathrm{S} = \mathrm{P}_1\mathrm{P}_2...\mathrm{P}_K$ *and a bucket-series* $\mathrm{S}^{\mathbf{B}} = \mathbf{B}_1...\mathbf{B}_K$*, the Sample-Bucket Distance is*

$$\mathrm{dist}_{avg}^{\gamma}\Big(\mathrm{S}, \mathrm{S}^{\mathbf{B}}, \Sigma\Big) = \sum_{1 \leq i \leq K} \min_{\mathrm{P} \in \mathbf{B}_i}\left\{\mathrm{dist}_{avg}^{\gamma}\Big(\mathrm{P}_i, \mathrm{P}\Big)\right\}$$

with $\mathrm{P}_i, \mathbf{B}_i$ *as profiles or buckets at position* i*, a coordinate-type* $\gamma \in \{y, s\}$*,* K *as the number of profiles in* S *and buckets in* $\mathrm{S}^{\mathbf{B}}$*. The minimum-function* min *returns the minimum out of a set of distance-scores. Distances* $\mathrm{dist}^{\gamma}_{avg}\Big(\mathrm{P}_i, \mathrm{P}\Big)$ *were defined in Def. 2.2.1.2 and Def. 2.2.1.3.*

2.2.2 Similarity Measurement

Similarity measurement is necessary within the established approaches. Therefore, the same ideas as before used with distance-functions (see Ch. 2.2.1) are applied. Additive differences are compensable by subtracting the mean from each component in a series of values, and multiplicative differences can be removed by division with the magnitude. A single mathematical construct which already meets these requirements is the formula for the Pearson Correlation Coefficient [38; 48] (Eq. 2.2.2.2). From the values, the means are subtracted and it is divided by the magnitudes. As abbreviations for sequences of values, the uppercase letter V is used.

$$
\begin{aligned}
\rho\Big(\mathrm{V}^a, \mathrm{V}^b\Big) &= \frac{\sigma(\mathrm{V}^a, \mathrm{V}^b)}{\sigma\Big(\mathrm{V}^a\Big) \cdot \sigma\Big(\mathrm{V}^b\Big)} \\
&= \frac{\frac{1}{K-1}\sum_{i=1}^{K}\Big(v_i^a - \mathrm{mean}(\mathrm{V}^a)\Big) \cdot \Big(v_i^b - \mathrm{mean}(\mathrm{V}^b)\Big)}{\sqrt{\frac{1}{K-1}\sum_{i=1}^{K}\Big(v_i^a - \mathrm{mean}(\mathrm{V}^a)\Big)^2} \cdot \sqrt{\frac{1}{K-1}\sum_{i=1}^{K}(\Big(v_i^b - \mathrm{mean}(\mathrm{V}^b)\Big)^2}} \\
&= \frac{\Big(\mathrm{V}^a - \mathrm{seq}\Big(\mathrm{mean}(\mathrm{V}^a)\Big)\Big) \cdot \Big(\mathrm{V}^b - \mathrm{seq}\Big(\mathrm{mean}(\mathrm{V}^b)\Big)\Big)}{\Big\|\mathrm{V}^a - \mathrm{seq}\Big(\mathrm{mean}(\mathrm{V}^a)\Big)\Big\| \cdot \Big\|\mathrm{V}^b - \mathrm{seq}\Big(\mathrm{mean}(\mathrm{V}^b)\Big)\Big\|}
\end{aligned}
$$

(2.2.2.2)
Pearson Correlation
Coefficient Transformation

where $\mathrm{seq}\Big(\mathrm{mean}(\mathrm{V})\Big)$ is a series of entries containing the mean of V.

Definition 2.2.2.1 (Pearson Correlation Coefficient)
Given series of values $\mathrm{V}^a = v_1^a v_2^a ... v_K^a$ *and* $\mathrm{V}^b = v_1^b v_2^b ... v_K^b$*, the Pearson Correlation Coefficient is*

$$\rho\Big(\mathrm{V}^a, \mathrm{V}^b\Big) = \frac{\sigma\Big(\mathrm{V}^a, \mathrm{V}^b\Big)}{\sigma\Big(\mathrm{V}^a\Big) \cdot \sigma\Big(\mathrm{V}^b\Big)}$$

with $\sigma\Big(\mathrm{V}^a, \mathrm{V}^b\Big)$ *as the covariance, and standard deviations* $\sigma\Big(\mathrm{V}^a\Big)$*,* $\sigma\Big(\mathrm{V}^b\Big)$ *of* V^a *and* V^b*.*

Further, the rank has to be defined [6; 47, p. 277]. The rank is the position of an element in a list of elements which was sorted by some property in ascending or descending order. An example, say there is a list of years and each year has a score. Now by sorting the scores in ascending order, one can give each year an order number e.g. the year with the lowest score gets rank 1 and the year with second lowest score gets rank 2. But what would for example happen if there are two elements with same scores? Which of the two elements get which rank? Different approaches exist, but since the elements cannot be distinguished by their score, they could get their maximum position as their rank i.e. all scores within a list are sorted and the two elements get a new position or index in the sorted list. From both consecutive positions, the maximum position[1] is chosen as the rank by assuming that indices start with a 1. Where now ranks are necessary? The Spearman Rank Correlation Coefficient [49, p. 53] as defined in Def. 2.2.2.2 uses ranks. It is working on ranks of scores instead of the scores itself. Apart from that it is analogous to the Pearson Correlation Coefficient.

Definition 2.2.2.2 (Spearman Rank Correlation Coefficient)
Given series of values $\mathrm{V}^a = v_1^a v_2^a ... v_K^a$ *and* $\mathrm{V}^b = v_1^b v_2^b ... v_K^b$*, the Spearman Rank Correlation Coefficient is*

$$\varrho\Big(\mathrm{V}^a, \mathrm{V}^b\Big) = \rho\Big(r(\mathrm{V}^a), r(\mathrm{V}^b)\Big)$$

with $r(\mathrm{V}^a), r(\mathrm{V}^b)$ *as the ranks series of* V^a, V^b *and* ρ *as described in Def. 2.2.2.1.*

Based on this, the Kendall Rank Correlation Coefficient can be defined. It looks how often there is a sort order agreement or disagreement between two series $\mathrm{V}^a, \mathrm{V}^b$ of values. An agreement in the sort order i.e. $v_i^a < v_j^a \wedge v_i^b < v_j^b$ or $v_i^a > v_j^a \wedge v_i^b > v_j^b$ with $i < j$ is called a concordance and a disagreement $v_i^a > v_j^a \wedge v_i^b < v_j^b$ or $v_i^a < v_j^a \wedge v_i^b > v_j^b$ in the sort order is a discordance. Ties $v_i^a = v_j^a \wedge v_i^b \neq v_j^b$ or $v_i^a \neq v_j^a \wedge v_i^b = v_j^b$ are usually not considered. A representation (Def. 2.2.2.3) which takes ties into account can be found in [26]. This general variant is also called the τ_b (Tau-b) correlation coefficient. In literature often only the τ_a (Tau-a) correlation coefficient is specified [18, p. 137] and that does not consider ties i.e. the denominator is $\frac{K \cdot (K-1)}{2}$ with K as the series length of both series. So in the τ_a correlation coefficient, it is just divided by the number of made rank comparisons of ranks $r(v_i^a), r(v_i^b)$ with ranks $r(v_j^a), r(v_j^b)$ and $i < j$.

Definition 2.2.2.3 (Kendall Rank Correlation Coefficient)
Given series of values $\mathrm{V}^a = v_1^a v_2^a ... v_K^a$ *and* $\mathrm{V}^b = v_1^b v_2^b ... v_K^b$*, the Kendall Rank Correlation Coefficient is*

$$\tau\Big(\mathrm{V}^a, \mathrm{V}^b\Big) = \frac{N_{conc} - N_{disc}}{\sqrt{(N_{conc} + N_{disc} + N_{ties}^a) \cdot (N_{conc} + N_{disc} + N_{ties}^b)}}$$

where N_{conc} *is the number of concordant,* N_{disc} *the number of discordant pairs* (v_i^a, v_i^b) *and* N_{ties}^a, N_{ties}^b *the number of ties within series* $\mathrm{V}^a, \mathrm{V}^b$*.*

[1] in the programming language R, the average is taken

Beside correlation coefficients, t-values between two sequences can be measured. A t-value can be interpreted as a kind of non-normalized similarity value. There are multiple variants of statistical tests from which t-values can be used, but they all work similar to the common Student's t-test for two samples [22, pp. 533-534]. Howsoever, for the Hollstein t-Value first the Hollstein Wuchswerte[1] have to be defined. A Wuchswert is the change in the growth from one point (e.g. ring-width) to the next one in a sequence of points [40].

Definition 2.2.2.4 (Hollstein Wuchswerte)
Given a series of values $\mathrm{V} = v_1 v_2 ... v_K$*, the sequence of Hollstein Wuchswerte is*

$$\mathrm{G} = g_1 g_2 ... g_{K-1}$$

with $g_i = \ln\left(\frac{v_i}{v_{i+1}}\right)$ *as a single Hollstein Wuchswert.*

That Wuchswert is necessary to remove growth trends. It can happen that the average ring-width value for multiple sequences is growing over time and such tendencies have to be removed. Given this definition for Hollstein Wuchswerte, the Hollstein t-Value as in Def. 2.2.2.5 can be defined [20].

Definition 2.2.2.5 (Hollstein t-Value)
Given series of Hollstein Wuchswerte $\mathrm{G}^a = g_1^a g_2^a ... g_{K-1}^a$ *and* $\mathrm{G}^b = g_1^b g_2^b ... g_{K-1}^b$ *for series of values with length* K*, the Hollstein t-Value is*

$$t(\mathrm{G}^a, \mathrm{G}^b) = \rho\Big(\mathrm{G}^a, \mathrm{G}^b\Big) \cdot \frac{\sqrt{K-2}}{1 - \rho\Big(\mathrm{G}^a, \mathrm{G}^b\Big)^2}$$

with $\rho\Big(\mathrm{G}^a, \mathrm{G}^b\Big)$ *as the Pearson Correlation Coefficient from Def. 2.2.2.1.*

So that means to compute the Hollstein t-Value, both sequences of points have to be first transformed into sequences of Hollstein Wuchswerte. And then on these Wuchswerte, the t-value has to be calculated. In literature it is stated that in most situations for computation of t-values at least about 25 values within the sequences are necessary [20; 22, p. 534]. However, this is only a recommendation. Further, it has to be mentioned that the formula for the t-value was stated differently in other literature [41, p. 94]. There, also the denominator $1 - \rho\Big(\mathrm{G}^a, \mathrm{G}^b\Big)^2$ is under the root.

[1] *Wuchs*: ger. growth and *-wert*: ger. value, whereas *-werte* is used for the plural

2.2.3 Quality and Reliability

To decide which of the presented approaches in the Results chapter leads to the best results, different quality measures have to be introduced and reliability values have to be defined. So what is the difference between both terms? A reliability value tells the user how sure an algorithm is about a given solution, for example a single predicted date. Whereas, a quality measure helps to evaluate an algorithm e.g. the mean, variance and the median are typical quality measures for a rank distribution. As mentioned before (look Ch. 2.2.2), for generated scores, i.e. distances, ranks can be computed. These are necessary to decide about the quality of an approach. For future work and to avoid a confusion with the existing variants for the variance and median formulas, both were stated below as they were defined for plots in the Results chapter.

Definition 2.2.3.1 (Variance)
The variance for a set of ranks $\mathbf{R} = \{r_1, r_2, ..., r_u\}$ *is*

$$\mathrm{Var}(\mathbf{R}) = \frac{1}{u-1}\sum_{i=1}^{u}(r_i - \bar{r})^2$$

with $\bar{r}$ *as the (non-robust) mean of the ranks in* $\mathbf{R}$.

Definition 2.2.3.2 (Median)
The median for a set of ranks $\mathbf{R} = \{r_1, r_2, ..., r_u\}$ *with* $r_1 \leq r_2 \leq ... \leq r_u$ *is*

$$\mathrm{median}(\mathbf{R}) = \begin{cases} r_{\frac{u+1}{2}} & , u \text{ odd} \\ \frac{1}{2}(r_{\frac{u}{2}} + r_{\frac{u}{2}+1}) & , u \text{ even}. \end{cases}$$

The first reliability measure is the so-called Δ Score. Scores are distances (see Ch. 2.2.1) in this case which are generated between a sample S and a subsequence S_i from a chronology C at a certain position. The subsequence S_i is just a sample extracted from the chronology, such that Def. 2.2.1.5 or Def. 2.2.1.6 can be applied to generate a distance-score.

Definition 2.2.3.3 (Δ Score)
Given the set of generated distance-scores $\mathbf{S}_{\mathrm{S}}^{\mathrm{C}} = \{\varsigma_1, \varsigma_2, ..., \varsigma_{N-K+1}\}$ *between a sample* S *of length* K *and a chronology* C *of length* N, *the lowest score* $\hat{\varsigma}_1 = \min\left(\mathbf{S}_{\mathrm{S}}^{\mathrm{C}}\right)$ *and the second lowest score* $\hat{\varsigma}_2 = \min\left(\mathbf{S}_{\mathrm{S}}^{\mathrm{C}} \setminus \{\hat{\varsigma}_1\}\right)$, *the corresponding* Δ Score *is*

$$\Delta\,\mathrm{Score} = \hat{\varsigma}_2 - \hat{\varsigma}_1.$$

Analogously, a quality measure called Δ Peak can be defined. That is the weighted average distance between the two highest peaks in a histogram. What is the weighted average? A problem in a histogram approach is that there can be multiple bars with the same height. Such bars had to be considered somehow in the calculations. The idea here is to measure first the count-difference i.e. height-difference between the

biggest and the second biggest peak and then divide this value by the number of bars[1] having the same count as the second biggest peak. But what can be done if there is no second-rank-prediction, so a second bar? In this case the difference to zero is measured, since it can be assumed in this case that it exists another bar with a count of zero.

Definition 2.2.3.4 (Δ Peak)
Given a set of histogram-counts $\mathbf{H} = \{\eta_1, \eta_2, ..., \eta_u\}$*, the highest* $\hat{\eta}_1 = \max\left(\mathbf{H}\right)$ *and the second highest count* $\hat{\eta}_2 = \max\left(\mathbf{H} \setminus \{\hat{\eta}_1\}\right)$*, then the corresponding* Δ Peak *is*

$$\Delta \text{ Peak} = \begin{cases} \frac{\hat{\eta}_1 - \hat{\eta}_2}{\text{count}_{\mathbf{H}}(\hat{\eta}_2)} & , u \geq 2 \\ \hat{\eta}_1 & , u = 1 \end{cases}$$

where u *is an arbitrary number of bars and* $\text{count}_{\mathbf{H}}$ *a function which counts how often the passed argument can be found in the set* $\mathbf{H}$*.*

Alternatively, to the first described quality measure, in the literature often so-called p-values [18, pp. 387-389] can be estimated on a distribution generated from a set of scores. The used distributions are not important for understanding and distributions were therefore only generally introduced (Def. 2.2.3.5).

Definition 2.2.3.5 (Cumulative Distribution Function)
The Cumulative Distribution Function $F_Z(\varsigma)$ *under a continuous random variable* Z *is defined by*

$$F_Z(\varsigma) = \Pr[Z \leq \varsigma] = \int_{-\infty}^{\varsigma} f(q)dq$$

where f *is a probability density function and* $\Pr[Z \leq \varsigma]$ *the probability for* $Z \leq \varsigma$*.*

Necessarily, the Empirical Distribution Function [22, p. 208] has to be defined, too. This is a function which returns the number of scores up to a given value ς.

Definition 2.2.3.6 (Empirical Distribution Function)
The Empirical Distribution Function $\hat{F}_M(\varsigma)$ *for scores* $\varsigma_1, \varsigma_2, ..., \varsigma_M$ *is defined by*

$$\hat{F}_M(\varsigma) = \frac{1}{M} \sum_{i=1}^{M} \mathbb{1}_{\varsigma_i \leq \varsigma}$$

where

$$\mathbb{1}_{\varsigma_i \leq \varsigma} = \begin{cases} 1 & , \textit{if } \varsigma_i \leq \varsigma \\ 0 & , \textit{else} \end{cases}$$

is an indicator function [47, p. 482].

[1] each bar represents the number of votes for a single year

Together with the Cumulative Distribution Function, a goodness of fit selection statistic can be defined to check how well a distribution fits to the observed data. The common approach is to look if the Kolmogorov-Smirnov [15, p. 7] statistic $\sup_\varsigma |\hat{F}_M(\varsigma) - F_Z(\varsigma)|$ is below or equal to some critical value $c_{\lambda,M}$.

Definition 2.2.3.7 (Kolmogorov-Smirnov Test)
The Kolmogorov-Smirnov Test is the following test

$$\sup_\varsigma |\hat{F}_M(\varsigma) - F_Z(\varsigma)| \leq c_{\lambda,M}$$

with $\sup_\varsigma |\hat{F}_M(\varsigma) - F_Z(\varsigma)|$ *as the Kolmogorov-Smirnov statistic,* sup *as the supremum, M as the number of scores and $c_{\lambda,M}$ as the so-called critical value, whereas λ is the significance level. $F_Z(\varsigma)$ and $\hat{F}_M(\varsigma)$ were defined in Def. 2.2.3.5 and Def. 2.2.3.6.*

Critical values $c_{\lambda,M}$ can be computed exactly with a procedure found in [17]. However, current implementations do not implement this procedure, but use different approximation techniques. After the right distribution was found, it is possible to compute p-values as defined in Def. 2.2.3.8 for measurement of reliability.

Definition 2.2.3.8 (Left, One-Tailed p-Value)
Given a continuous random variable Z, the left p-value returns the cumulative probability from the left end of a distribution up to a certain score ς and it is defined by

$$p = F_Z(\varsigma) = \int_{-\infty}^{\varsigma} f(q)dq$$

where f is a probability density function and $F_Z(\varsigma)$ the corresponding cumulative distribution function (Def. 2.2.3.5).

So the Left, One-Tailed p-Value returns the probability of having values of the random variable Z that are equal or lower some value ς. So ς is some score, and one asks what is the probability of getting an equal or lower score. If that probability is now under some by experience determined threshold λ, known as the significance level, then one knows that the value is with high probability not the product of chance. At this point often, the null hypothesis H_0 and the alternative hypothesis H_1 are introduced. That means, the p-value is used to support some hypothesis, e.g. the p-value is often defined by $\Pr[Z \leq \varsigma \,|\, H_0]$. But this is not of any importance and was therefore omitted. To work with p-values, a distribution has to be selected and for the lowest ς-value, the p-value has to be calculated, to see if it is a product of chance or not i.e. to check whether it is significant (whether the H_1 hypothesis seems to be true). Density function parameters of the distribution can be calculated by a Maximum Likelihood Estimation. This is an approach to estimate the unknown parameters e.g. here the standard deviation and the expected value of a model, here a distribution. It fits the best model for given data[1]. After model-fitting, p-values can be computed.

[1] this can be done for example with the `optim` command in the programming language R [15, p. 1]

3 Approaches and Evaluation

3.1 Data Acquisition

3.1.1 Datasets

The dataset contained 56 chronologies (Def. 2.1.2.5) extracted from Norway spruces (Picea abies) felled in the Swabian Ostalb in Germany at an average altitude of about 658 meters. The dataset was generated by measuring the density along multiple directions of a stem disc from the pith to the bark [3; 4] with High-Frequency Densitometry [39; 50]. The for example eight measured series of density-profiles were then aligned for each year with MICA (see Ch. 2.1.2) and per year one Consensus-Profile was built. Therefore, before an alignment (Def. 2.1.2.3), a conversion into normalized profiles (Def. 2.2.1.1) was made and afterwards the warping of that normalized profiles was applied on the final profiles. These were interpolated to $k = 100$ x-axis equidistant points and stored as the final dataset in form of a table (`*.csv`-format) similar to the one in Tab. 3.1.1.1. The density values were given in form of gravimetric density (see Ch. 2.1.2). One difficulty was that the years were not consecutive in all files, so there were years without a profile. All MICA-parameters used to compute the Consensus-Profiles together with a short description for the dataset can be found in the Appendix (see Ch. 5.1.1).

year	**GD**
1992	2.016
1992	2.433
1992	2.881
1993	2.043
1993	2.383
⋮	⋮

Table 3.1.1.1 Minimal schematic structure of a density-profiles file. A column for the year and a column for the corresponding gravimetric density-values, whereby a series of density-values for the same year corresponds to a single profile.

The final dataset had in total 2881 profiles available spread over the years 1879-2004. But the first 37 years were deleted since they had only one profile per year[1] what prevented to select samples from these years because then at these positions no data would be available in the chronology. For the analysis, it was also evaluated how much the profiles from the same year are differed and how many outliers were contained within the same year. Therefore, box-plots were created and these were plotted below a histogram for the number of profiles within a year (Fig. 3.1.1.1). The Mean-Distance

[1] there was one exception and that was the year 1914 which had two profiles, so 38 profiles were deleted

(Def. 2.2.1.4) for each profile within a year was measured and these distances were plotted into the lower of both plots. In the beginning, around the year 1916, no outliers were mentioned i.e. there were no red-dots. There were also just a few Mean-Distances measured as can be seen. More profiles have led also to more outliers as it could be observed by comparing the years 1916 up to 1960 with 1960 up to 2004 within both plots. Most measured Mean-Distances were around 0.2 and 0.4, but a few were around 0.6 or had an even higher value.

Ring-widths were given in terms of the number of density measuring points i.e. the per year numbers of density measuring points, from for example eight directions on a stem disc, were added together and then divided by eight. Besides that, a dataset for maximum densities of profiles was created by extraction of the maxima in the given per-tree consensi profiles. The resulting dataset was stored in the same format as the ring-widths on the hard-drive for a fast reloading.

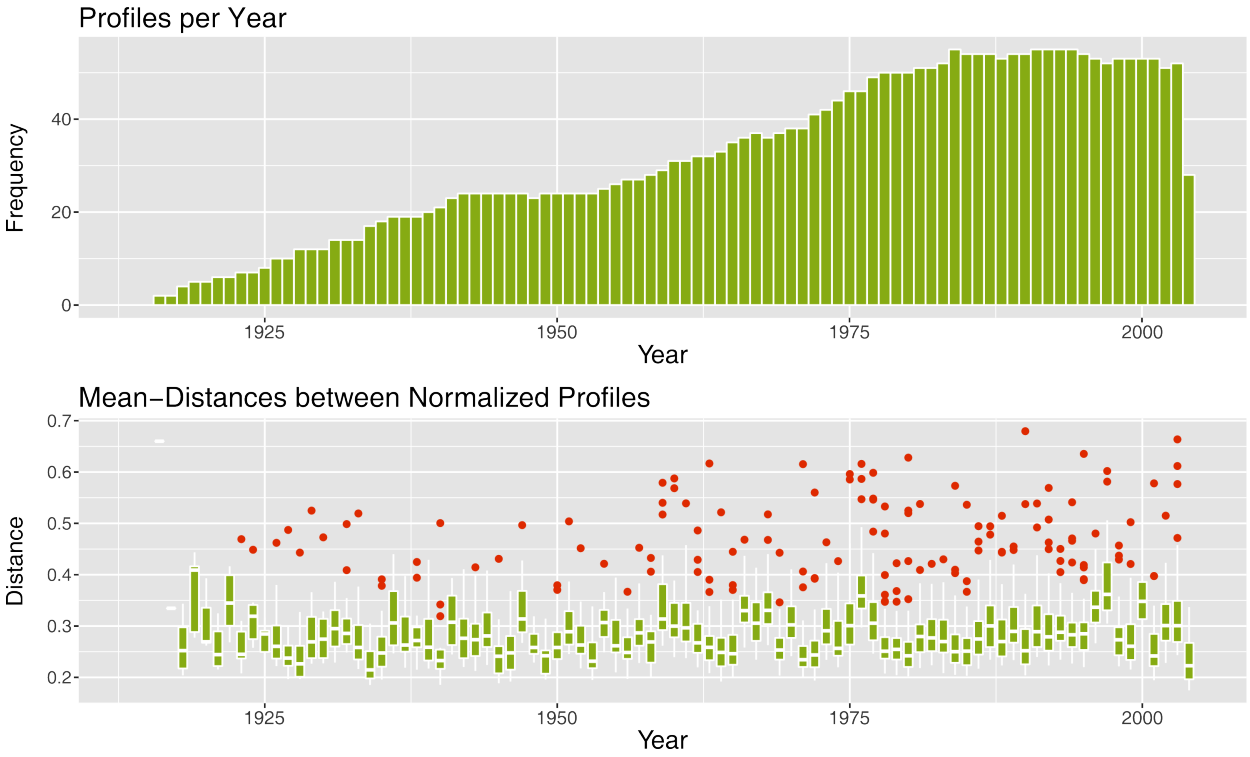

Figure 3.1.1.1 Histogram for the profiles and their Mean-Distances range from 1916 to 2004. In each year the profiles were counted and plotted in a histogram. All pairwise distances between profiles of the same year were calculated for the computation of Mean-Distances (Def. 2.2.1.4). These Mean-Distances can be seen in the lower diagram. The red dots are representing outliers.

3.1.2 Generation of Samples & Chronologies

One goal was to find out if cross-dating with density-profiles could work in general. So chronologies and samples were created. For this purpose, given the per-tree consensi, ring-widths and maximum densities from Ch. 3.1.1, 50 samples of length 10 i.e. 10 consecutive years were uniformly extracted at random using the R programming language. In the trimmed set i.e. without the 37 first years were 2843 profiles or ring-widths (or maximum densities) with their years (Ch. 3.1.1), but only 2329 were available to extract length-10-samples[1]. After an extraction up to 500 profiles, ring-widths and maximum densities from the 2329 available were removed, so about $\frac{50 \cdot 10}{2329} \approx 21.5\%$ of for the extraction available data was used as a test-set, what is a common value for statistical approaches [28]. From the remaining profiles and characteristics, different types of master chronologies were computed (see Ch. 2.1.2). For averaging the ring-widths and maximum-densities of a year, Tukey's Biweight Robust Mean estimation was used (Def. 2.1.1.2). The whole procedure was repeated five times. So new master chronologies together with new samples were extracted and the results of searching these samples within the different types of chronologies were summarized in the following sections. Length-5 and length-1-samples were generated from the length-10-samples by simple sample-splitting to avoid any recomputations and to make results better comparable with each other. So with length 5, the number of samples was nearly doubled (duplicates were removed) and with length 1 the number of samples was about tenfold. Also, new test-sample sets for length 15 with new chronologies were created. For length-15-samples in each pass 33 samples were extracted and 7 passes were done to get more or less the same total number of samples as with length-10-samples using 5 passes.

3.1.3 Discussion

Ring-widths were given as mean-numbers of measuring-points but this was not a problem, since the measuring points were equidistantly distributed and so the mean numbers are in relation to the real ring-widths. Also, the differences in the widths were very high i.e. a ring-width could have the value 69 and the next width could have the value 102. So even if there would be more decimal places, it wouldn't help in any way since neighboured ring-widths were already well distinguishable.

A test-set with about 50 samples is comparatively small. Such that, to make reliable statements about the used methods in the following sections, the extraction of samples and chronology-computation were repeated multiple times. Something similar to a 5-fold cross-validation [21] was applied. Instead of dividing into multiple fair test-sets e.g. sets which were extracted uniformly over all data, it was allowed to have sample test-sets with overlapping elements. This was not a problem since the results for the five test-sets were computed independently on five different chronologies. It was also not possible to just use more samples because the test-set had only a limited number of profiles or years. The given dataset contained the density-profiles for 126 years (1879-2004) from which only 89 were utilizable to extract samples.

[1] last 9 years couldn't be used to extract length-10-samples & not all files had consecutive years

3.2 Points-Based Approaches

3.2.1 Methods

It is of interest how an approach is performing against currently established approaches. Therefore, as reference, correlation coefficient and t-value based approaches have been analyzed first [41]. The computed samples of length 10 i.e. with 10 widths or maximum densities per sample (see Ch. 3.1.2) were shifted along the five chronologies (Def. 2.1.1.1). Schematically the shifting is visualized in Fig. 3.2.1.1. For each shift of the sample a correlation coefficient or t-value was computed and stored. All such values generated with a shifted sample were put in a list which was then sorted in a descending order. Each coefficient or score got by this a rank within the list. The highest score got rank 1, the second highest got rank 2 and so on. The produced score in the correct start-year of a sample was extracted and the calculated rank of that was used to assess the prediction quality. Calculated ranks of all samples are shown in Fig. 3.2.2.1 up to Fig. 3.2.2.4. Tested were different correlation coefficients and t-values which were described under Ch. 2.2.2.

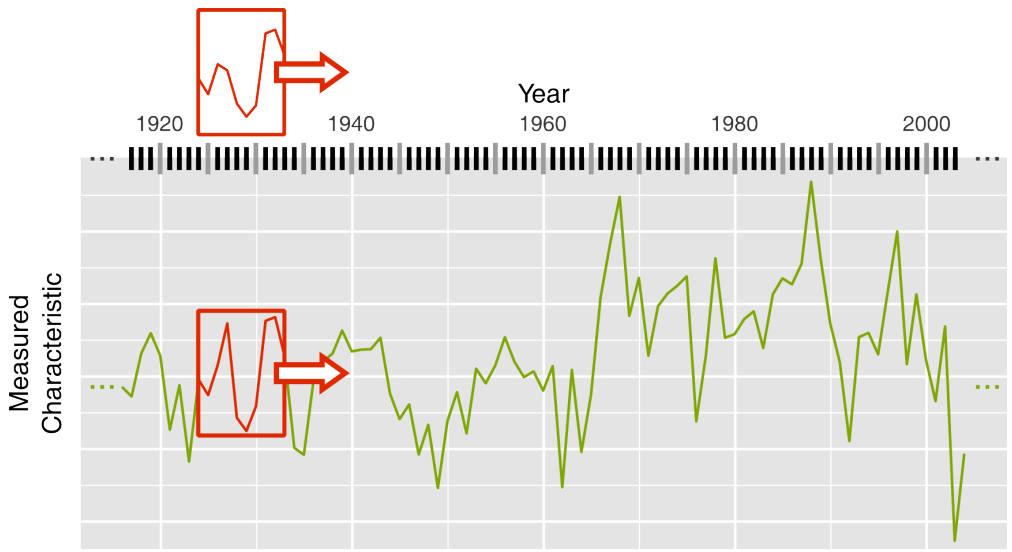

Figure 3.2.1.1 Calculation of correlation coefficients or t-values for each year. A sample is shifted along a chronology to compute a similarity-value for each start-year of a window. The sample and the subsequence from the chronology are highlighted with a red box.

3.2.2 Results

Given the mean, the variance (Def. 2.2.3.1) and the median (Def. 2.2.3.2) of the ranks, it was recognizable that the t-value (t) has led to the best results in Fig. 3.2.2.1, since from all three distributions, its distribution was the one with the highest number of low ranks. In this approach 28% of the test-samples were correctly ranked. For this approach now, the length dependant behaviour was determined which can be recognized in Fig. 3.2.2.2. It was obvious that for longer ring-width sequences, the ranks were decreasing i.e. the prediction got easier. From 231 test-samples under length 15, overall 128 ($\approx 55.4\%$) were correctly ranked, and 184 ($\approx 79.7\%$) were under the top 5.

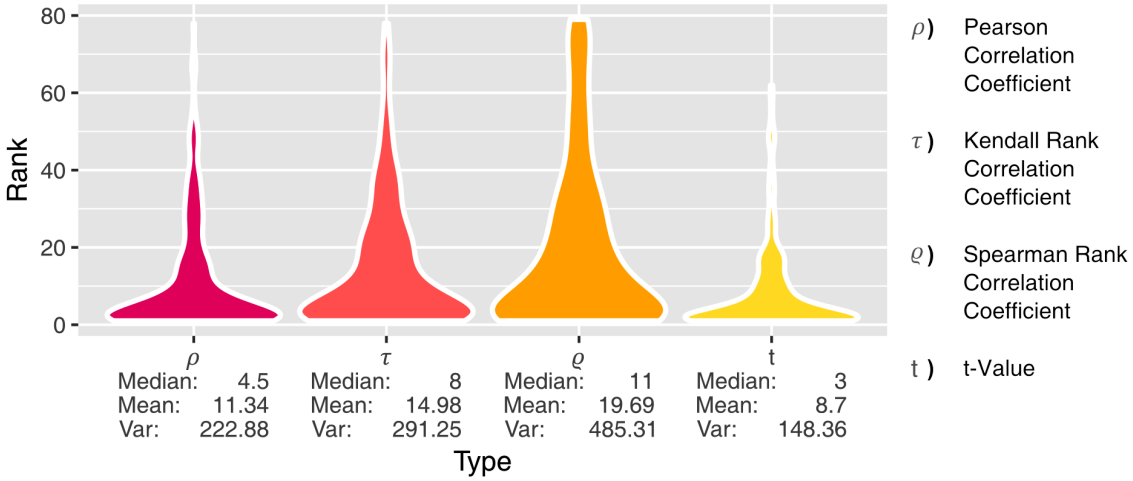

Figure 3.2.2.1 Violin plots of 250 sample-ranks with ring-width series of length 10. The formula for the Pearson Correlation Coefficient (ρ) can be found in Def. 2.2.2.1. For Kendall (τ) it is described in Def. 2.2.2.3, for Spearman (ϱ) in Def. 2.2.2.2 and for t-values (t) in Def. 2.2.2.5.

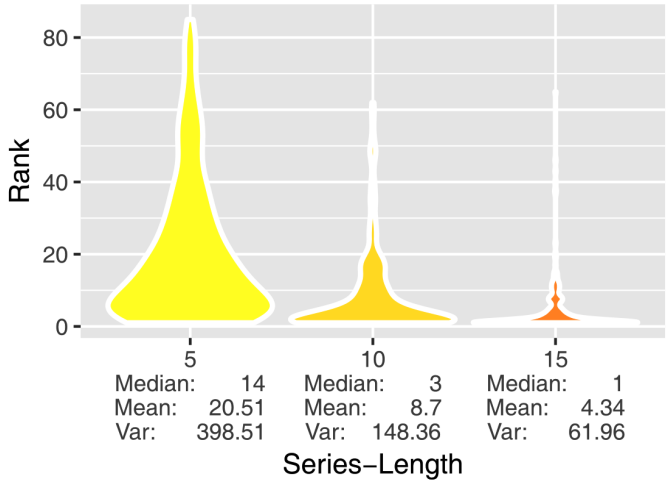

Figure 3.2.2.2 Violin plots of the sample-ranks under ring-widths for the t-value (t) with test-series of lengths 5, 10 and 15.

Afterwards the per sample cross-dating and ranking procedure was repeated with maximum densities of profiles. The results can be seen in Fig. 3.2.2.3. Overall it could be recognized, that all four methods have profited significantly by the usage of maximum density values from profiles. Best results were achieved by the Pearson Correlation Coefficient (ρ). 112 out of 250 test-samples (= 44.8%) were correctly ranked, and 193 (= 77.2%) were in the top 5. For the Pearson Correlation Coefficient again a plot for the length-dependency was created (Fig. 3.2.2.4). The results were very good, especially for length-15-samples. Out of 231 test-samples, 169 ($\approx$ 73.2%) were correctly ranked, and 220 ($\approx$ 95.2%) were in the top 5 of ranks. For the other three methods, the number of top and correctly rated samples was lower (data not shown). Additionally, results under the Gleichläufigkeit-value can be found in the Appendix Ch. 5.2.

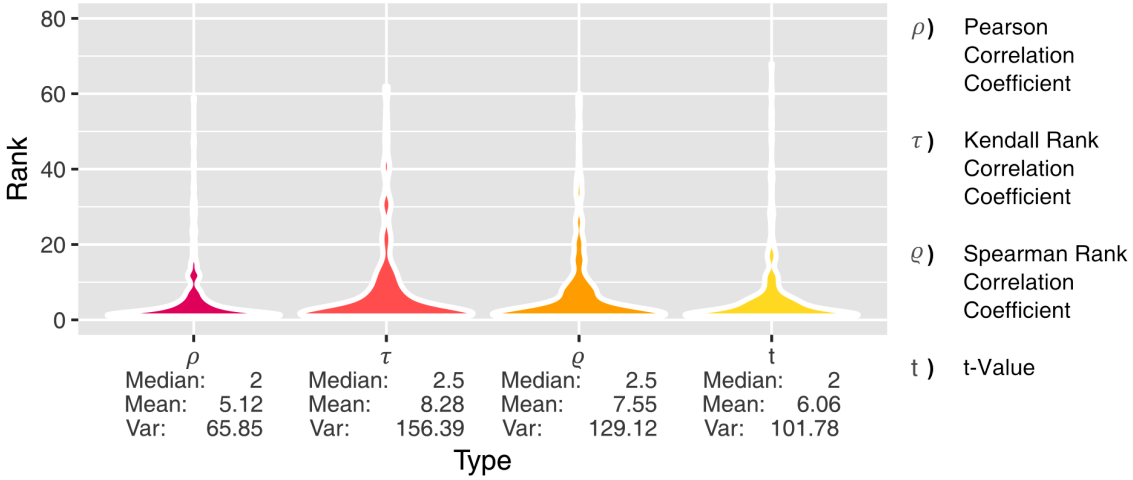

Figure 3.2.2.3 Violin plots of 250 ranks with maximum density series of length 10.

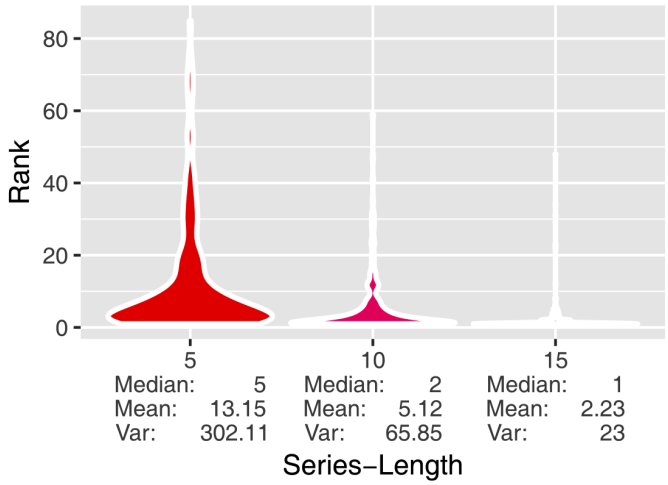

Figure 3.2.2.4 Violin plots of the sample-ranks under maximum densities for the Correlation Coefficient (ρ) with test-series of lengths 5, 10 and 15.

3.2.3 Discussion

The ranking results from the pure ring-width based approach were not bad at all (Fig. 3.2.2.1). This may result from the differences within a sample i.e. the value from one year to the next one varies strongly (see Ch. 3.1.3). Interesting was that the maximum densities had led to the best results under the Pearson Correlation Coefficient (ρ) and not the rank based coefficients. Presumably the reason, therefore, was the length of the chronology (with only 89 years, it was very short). In [11], it was also stated why series of maximum densities perform that good. Maximum Densities of profiles often have a very high correlation to the temperature. That means the computed chronologies should be more stable. So even in this small density-based approach with a single value from the density-profile, ring-widths have been outperformed. How it will look like, if next the whole profiles are compared?

3.3 Consensus Approach

3.3.1 Methods

Instead of applying cross-dating on series of points i.e. on ring-widths and maximum densities, it was now applied on series of density-profiles. The profile-samples of length 10 computed during the Data Acquisition (see Ch. 3.1) were shifted profile-wise along their consensus-chronologies and a score i.e. a distance was computed for every position of a sample within the master chronology. For that distances between every profile of the sample and every profile of a consensus window at the current sample-position i.e. start-year were calculated and summed up as a final score (Def. 2.2.1.5). Schematically this shifting is visualized for a sample of length 3 in Fig. 3.3.1.1.

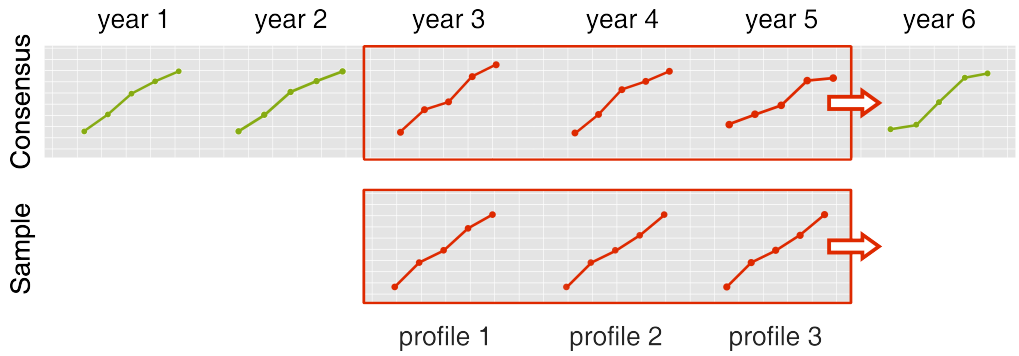

Figure 3.3.1.1 Schematic profile-wise shifting. A sample of length 3 is shifted along a profile Consensus Chronology (Def. 2.1.2.5) of length 6 and at each start-year a distance for a window of profiles from the chronology is computed.

Different distance-calculation techniques were applied to assess the similarity between Consensus-Profiles and sample-profiles. For each test-sample, the distance for the correct position in the master chronology was stored and the rank of that distance within all computed distances between the sample and the master chronology was calculated. That means all distance-scores produced with a sample were put into one list and this list was sorted in ascending order. And the last position of the named distance in the list was then stored as the rank. Practically it did not happen even once that two times the same score appeared within the produced scores, since the probability for that was very low because floating-point numbers have too many digits. Thus, the described procedure was analogous with the determination of the predicted rank of a sample start-year. But to avoid any problems in the first place, it was returned all positions having the correct score and the last position was used as the final rank. The computed final ranks from all five Consensus Chronologies (see Ch. 3.1) were then put together and displayed as violin-plots.

Consensus Chronology How exactly such consensus master chronologies have been computed? To this end, for all profiles from the same year, not contained in one of the about 50 test-samples, the normalized profiles were computed (Def. 2.2.1.1) and aligned. Maximally 24 normalized profiles of a bucket were aligned at the same time for a Consensus-Profile to reduce the computation time and problems with MICA. Because of the huge number of profiles, it could happen that two points got the same x-coordinate in the consensus. If that happened, MICA stopped with an error. This sometimes occurred already with about 30 profiles. So the maximum number of profiles for alignment was set to 24, to avoid such problems in any case. How the 24 profiles were chosen to preserve the variance within a bucket at least to some extent? Therefore, the Mean-Distances (Def. 2.2.1.4) on normalized profiles (see Ch. 2.2.1) within a bucket has been computed. The 12 profiles with the lowest and the 12 profiles with the highest produced Mean-Distance were added to a new bucket of profiles. After an alignment, the computed warping of the aligned normalized profiles' x-coordinates was applied on the final profiles and their Consensus-Profile was calculated as in Def. 2.1.2.4. Individual Consensus-Profiles have been then concatenated to get a consensus-curve for all years. For the computation more or less the same parameters as in the per tree consensus computation were used (see Appendix Ch. 5.1.1). The only real change was that the parameter for the number of samples was increased because more curves have been used for an alignment. It was also looked on the average Mean-Distances of all profiles from a year before and after the alignment. So the Mean-Distances (Def. 2.2.1.4) for all profiles within a year were added up and then the sum was divided by the number of profiles in the given year. Some average distances have fallen, and some have risen after the alignment, fairly independent of the made MICA-settings but that was even the case for the computed per tree consensi from the given dataset. Because of the mentioned error and in the hope to make more distances fall, different MICA-parameters were tested out on a small subset of years to circumvent the problems. That led to the parameters which can be found in the Appendix Ch. 5.1.2.

3.3.2 Results

As a result of different tested distance-calculation techniques, violin plots were created (Fig. 3.3.2.1). The normalized methods, here APDN (c) and SAPDN (d), had the lowest medians and the highest number of top-rated samples around rank 1. Concretely 79 test-samples for APDN (c) and 70 test-samples for SAPDN (d) were correctly ranked. Therefore, they were plotted for different series-lengths 1, 5, 10 and 15 since it was now of major importance to find out which lengths of density-series were datable with the given methods (Fig. 3.3.2.2 and Fig. 3.3.2.3). Therefore, the samples from 3.1 together with their corresponding Consensus Chronologies were used. It was clearly recognizable that both approaches performed better with longer samples. APDN (c) was slightly better compared to SAPDN (d). With length-5 test-samples 94 out of 495 were correctly ranked instead of only 82 correctly rated test-samples. For length 15, the difference was even more better recognizable, 121 out of 231 test-samples compared to 104 correctly rated test-samples. Also, using APDN (c) has led to 159 test-samples within the top 5 using length-15-samples.

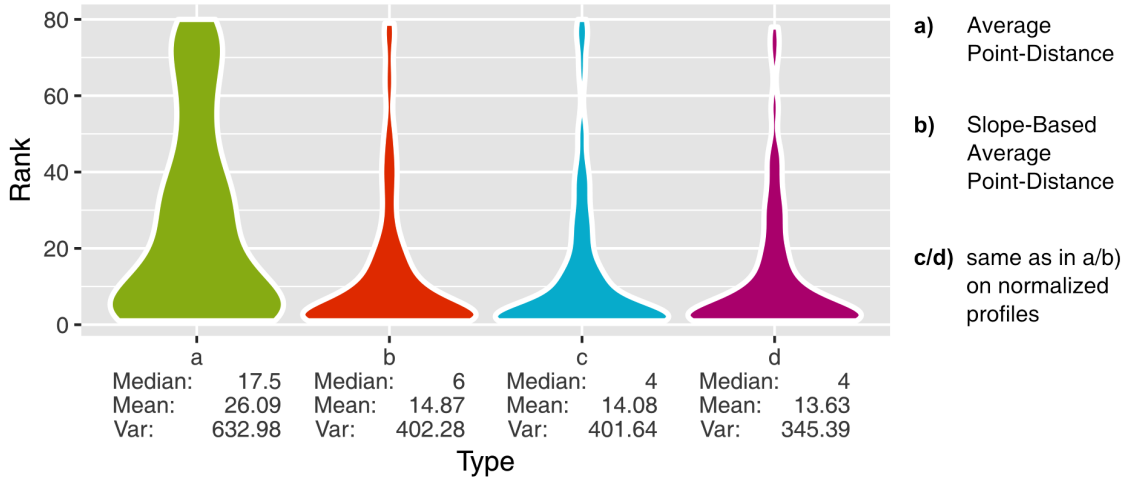

Figure 3.3.2.1 Violin plots of 250 sample-ranks with profile series of length 10. The formula for Average Point-Distance (a) can be found in Def. 2.2.1.2, whereas the Slope-Based Average Point-Distance (b) is described in Def. 2.2.1.3. To transform into normalized profiles Def. 2.2.1.1 was applied on the y-values of the profiles.

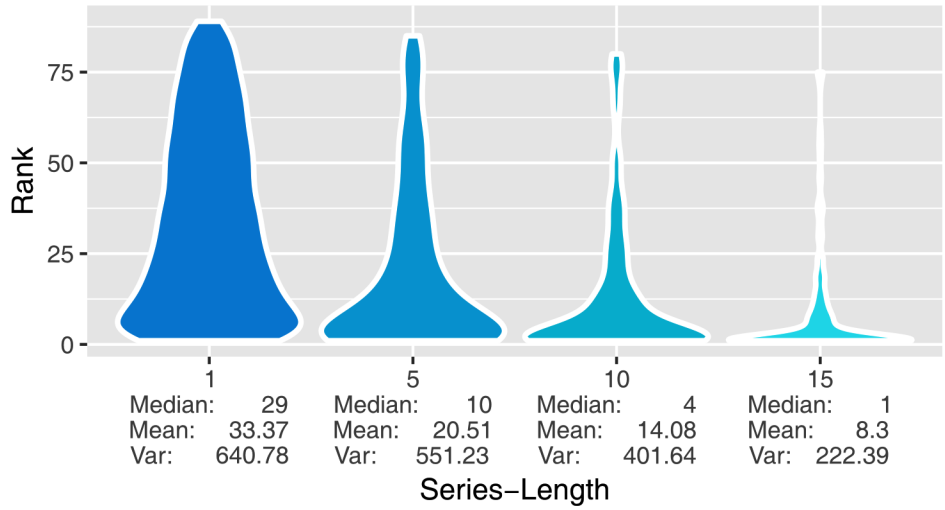

Figure 3.3.2.2 Violin plots of sample-ranks under the Average Point-Distance on normalized profiles (c) with series of lengths 1, 5, 10 and 15.

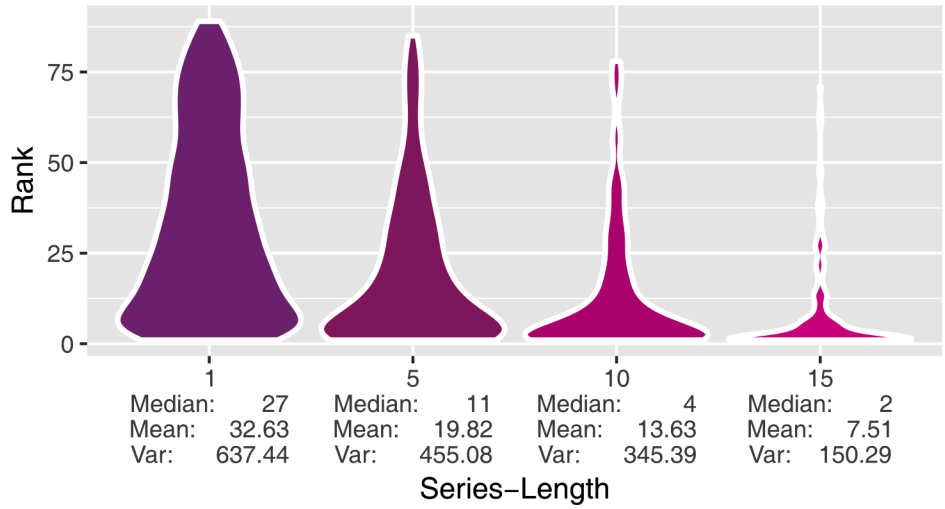

Figure 3.3.2.3 Violin plots of sample-ranks under Slope-Based Average Point-Distance on normalized profiles (d) with series of lengths 1, 5, 10 and 15.

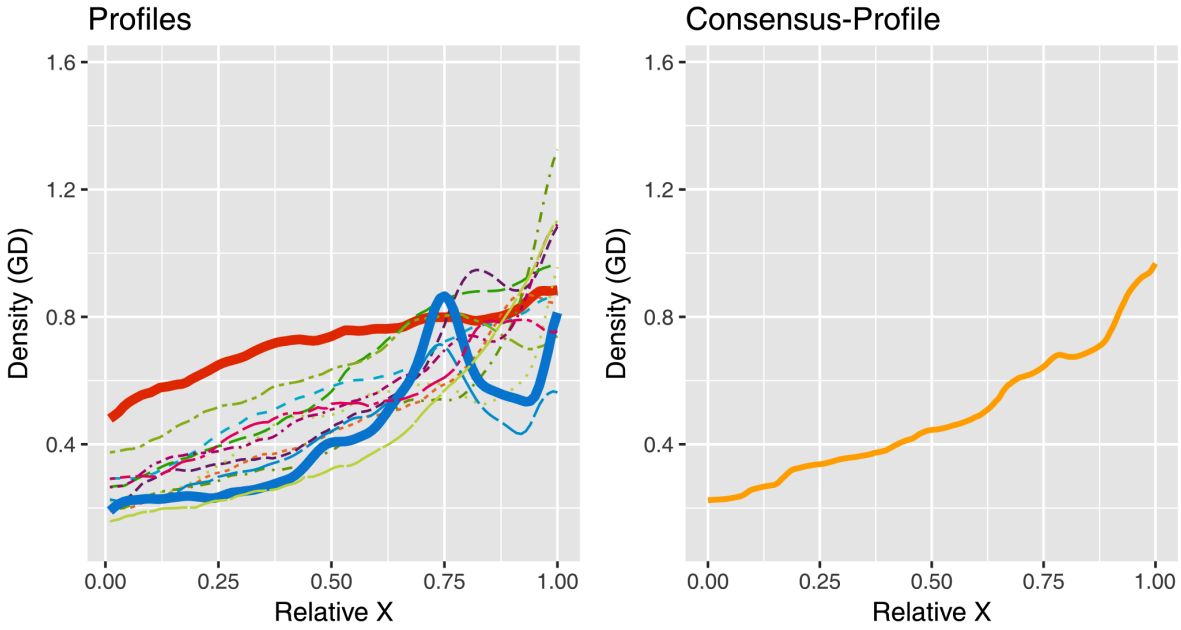

Figure 3.3.2.4 Profiles and their consensus. On the left a subset of different non-aligned profiles and on the right their consensus (see Ch. 2.1.2) computed by MICA, reflecting their shape. Two very different profiles on the left were marked bold.

During the master chronology computation, a problem with the Consensus Approach became recognizable which can be seen in Fig. 3.3.2.4. The consensi did not represent well enough the curves from which they were built.

3.3.3 Discussion

It was observed, that the results from normalized methods APDN (c) and SAPDN (d) were very similar to the results from the ring-width based dating (compare Fig. 3.3.2.1 with Fig. 3.2.2.1). Even the best violin plot or distribution from the Ring-Width Approach was better i.e. the mean and the variance were significantly lower, and the median had a lower rank. For length 15, there was a similar problem, the Ring-Width Approach has led to better results. There, 128 test-samples were correctly ranked using t-values compared to 121 correctly rated test-samples using APDN (c) (see Ch. 3.2.2). The bad performance of the Consensus Approach was even better recognizable by looking on the top 5. In the Consensus Approach, the best method APDN (c) had only 159 test-samples within the top 5 compared to 184 test-samples in the Ring-Width Approach with t-values. And that not enough, cross-dating of maximum densities has led to much better results than cross-dating with ring-widths. There, the Consensus Approach was completely inferior, regardless of the fact that the Consensus Approach works with much more data. In the Points-Based Approach with maximum densities, already under length 10, 112 out of 250 test-samples were correctly ranked.

Compared to only 79 under APDN (c) in the Consensus Approach. For length 15, 169 test-samples were correctly ranked under the Pearson Correlation Coefficient with maximum-densities. How could this be possible? The reason therefore is the variance within a bucket (Fig. 3.3.2.4). So if one looks on the individual profiles from which a single consensus or average profile is built, one can see very different curves. And if now a consensus is built over all these curves, information is lost i.e. the consensus does not represent precisely enough the shape of the curves. Other reasons are presumably the measured noise within profiles, as well as partially the mentioned MICA-bug.

Results computed during the ranking would presumably be better without the MICA-bug. To reduce the problems with that bug as best as possible, the Mean-Distances (Def. 2.2.1.4) of profiles to all remaining profiles in a bucket were measured. By this very similar and very different profiles got into the same final bucket of profiles with maximally 24 profiles. There were used 24 profiles and not for example only 20 profiles. The reason was that with already 30 profiles the bug sometimes happened. And the goal was to avoid for certain the bug but also to use as many profiles as possible for the consensus-computation. First, only the distance from one selected profile to all other profiles was measured. The 12 profiles producing the lowest and the 12 profiles producing the highest distances were inserted into the same bucket, but then the calculations were repeated and all profiles considered because lots of outliers were contained within the profiles as it can be seen in Fig. 3.1.1.1 by the red dots. It was possible that the selected one profile would be an outlier and then it would be very different to all profiles in the bucket, what could possibly lead to a worse representation of the variance within a bucket[1].

It was recognizable in all approaches that the ranks got better, the longer the series was i.e. the rank-concentration around rank 1 became higher as it can be seen by comparing length 1 with length 15 (Fig. 3.3.2.2). A made expectation was that APDN (c) and SAPDN (d) would perform better than the two methods APD (a) and SAPD (b). Since there the same methods have been applied under the usage of normalized profiles (Def. 2.2.1.1). So if the y-ranges of two profiles having the same shape, had different value ranges and thus the slopes too, after the normalization this problem was resolved. That was shown by the violin-plots of APDN (c) and SAPDN (d) which had better medians than APD (a) and SAPD (b) in Fig. 3.3.2.1. Besides that, it was expected that the longer the sample-length is, the lower the produced ranks for the test-samples should be. More correctly matched profiles lead to lower scores for the correctly matched samples with respect to all scores. This assumption seems to be correct at least for samples between lengths 1 and 15 (Fig. 3.3.2.2 and Fig. 3.3.2.3). Howsoever, after a comparison with the distribution for the Pearson Correlation Coefficient in Fig. 3.2.2.4, it can be said that the new Consensus Approach does not lead to any improvement.

[1] personal communication with Dr. Martin Raden

3.4 Per-Tree Approach

3.4.1 Methods

The Consensus Approach has not improved the results with respect to the Points-Based Approaches (see Ch. 3.3.3) presumably due to the mentioned variance problem. So a new approach was tested out which should solve this problem. Therefore, this time, it was not compared against a single Consensus Chronology but against the individual per-tree consensi from which it was build. That means in each year multiple final distances were computed for a sample and from all these distances, the minimum was chosen as the final distance for the given year. So it was shifted not along one chronology, but multiple as it can be recognized in Fig. 3.4.1.1.

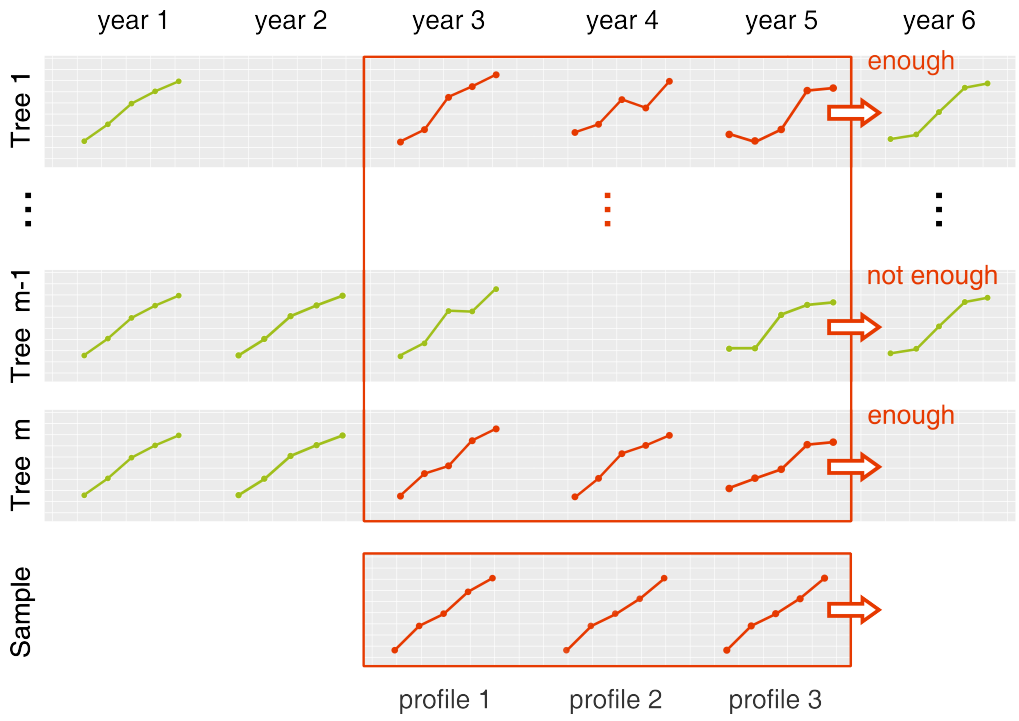

Figure 3.4.1.1 Schematic profile-wise shifting. A sample of length 3 is shifted along the chronologies of multiple trees. Windows of profiles are only extracted from chronologies in which at the given position *enough* consecutive profiles (red marked) exist.

The problem in this approach was to find an appropriate data structure which allows fast access and a low memory consumption. A trivial tactic would be to compute the scores for each chronology separately. But in this case, it would be hard to extend the approach for dynamic creation of subsamples of the test-sample in the case that there are gaps within the chronologies against which a distance is computed. Another data structure for this problem could be to have 56 arrays for the 56 per-tree consensi, each of length 80 for the here shifted length 10-samples. But even this structure would be problematic, since then up to 560 check-ups would be necessary for each length-10-sample at each position in the set of chronologies. The reason is simple, each profile would have to do 56 checks-ups for each year, to test if there is a profile at the given position or a gap. The series as stated in Ch. 3.1 are not always consecutive, especially because of the extracted test-samples. So how the final structure looked like? The first two thoughts were not well applicable for a possible dynamic extension. That is why each chronology got an own start-year and an end-year, but also an index for the current position within the chronology. Non-consecutive years together with their corresponding profiles were stored sequentially as lists. That decreased the number of check-ups significantly. As soon as a sample start-year was within the range of a chronology i.e. within the years in which the chronology contains profiles (between start- and end-year), it was activated and only then the above mentioned check-ups were necessary. The number of check-ups could be reduced, since non-consecutive years were stored sequentially one after another. For example, it was possible that after 1970, directly the profile for 1989 would come. It was just necessary to look if the difference between the extracted start-year in the chronology and the extracted last year plus one, in the case of length-3 test-samples, is three (Fig. 3.4.1.2). Therefore, not more than two check-ups were necessary[1]. The same approach was executed on Points-Based Approaches. So instead of samples with profiles, corresponding maximum-densities and ring-widths were shifted along chronologies.

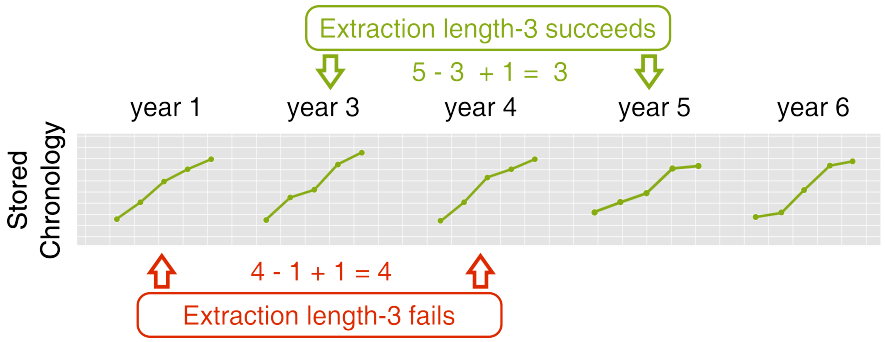

Figure 3.4.1.2 Extraction testing for length-3 samples. A sample of length 3 should be extracted at two positions. At the first test-sample position, an extraction would fail due to a gap, since the calculated year-difference plus one is too high. But at the second position it works since difference plus one is exactly 3.

[1] and a check-up for reaching chronologies' right bounds

Extended Approach The longer a sample, the higher is the probability that for a given year, the score is not computable. Since in a certain range of years it could be the case, that there are not enough consecutive years within the different chronologies i.e. there could be gaps between years as it can be seen in Fig. 3.4.1.2. But if the test-sample would for example be splitted, then one half could be tested against one chronology and the other half could be tested against another chronology. So each length-10 test-sample was subdivided into two, five or ten equally sized subsamples. These subsamples were then shifted one after the other along the chronologies and the corresponding scores reached with the individual parts were summed up for each start-year. That means for each part the minimum-score was computed and then these minimum-scores were summed up. For example, a length-4 test-sample is positioned at the year 1916 and it was divided into two equally sized parts, then a score is computed for the first part and for the second part starting in the year 1918. These two scores are summed to get a final score for the year 1916.

3.4.2 Results

The results with this new approach were very promising as it can be seen in Fig. 3.4.2.1, since the distributions in the violin plots have shown many rank 1 rated samples. Due to the fantastic results for APDN (c), 136 out of 250 samples correct and a median of 1, again violin plots for different lengths of series (Fig. 3.4.2.2) were created. For samples of length 5 around 59.4% were in top 5 ranks, whereas about 36.4% had actually rank 1. And for length 15, already about 84.8% of the samples were in the top 5. The results of executing this Per-Tree Approach on maximum densities can be seen in Fig. 3.4.2.3. All four methods have shown more or less the same results, but the Pearson Correlation Coefficient had the highest number of correct ranked test-samples, 119 out of 250 test-samples were correctly rated.

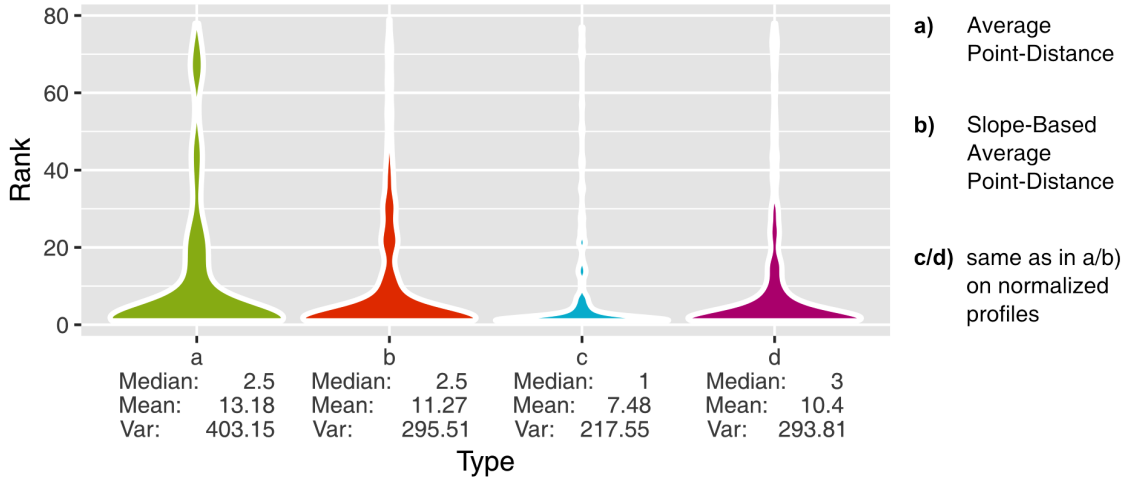

Figure 3.4.2.1 Violin plots of 250 sample-ranks with profile series of length 10.

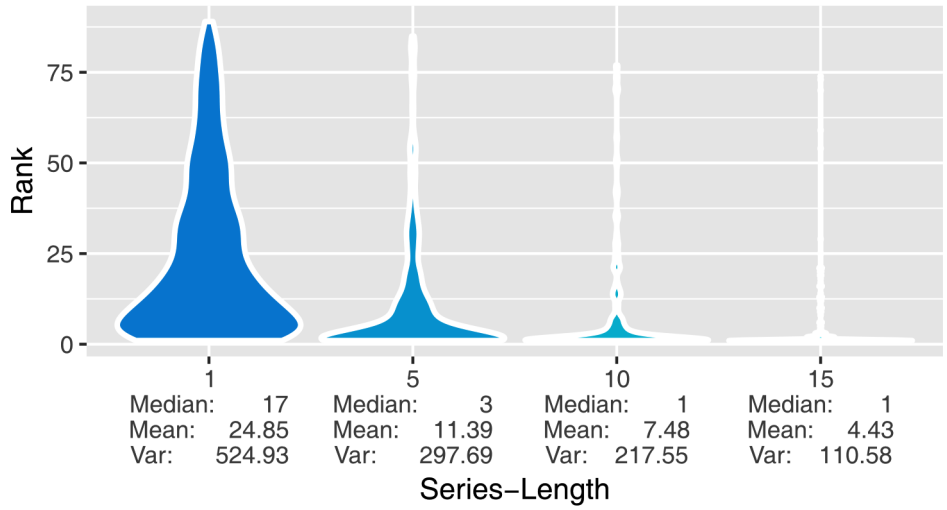

Figure 3.4.2.2 Violin plots of sample-ranks using Average Point-Distance on normalized profiles (c) with series of different length.

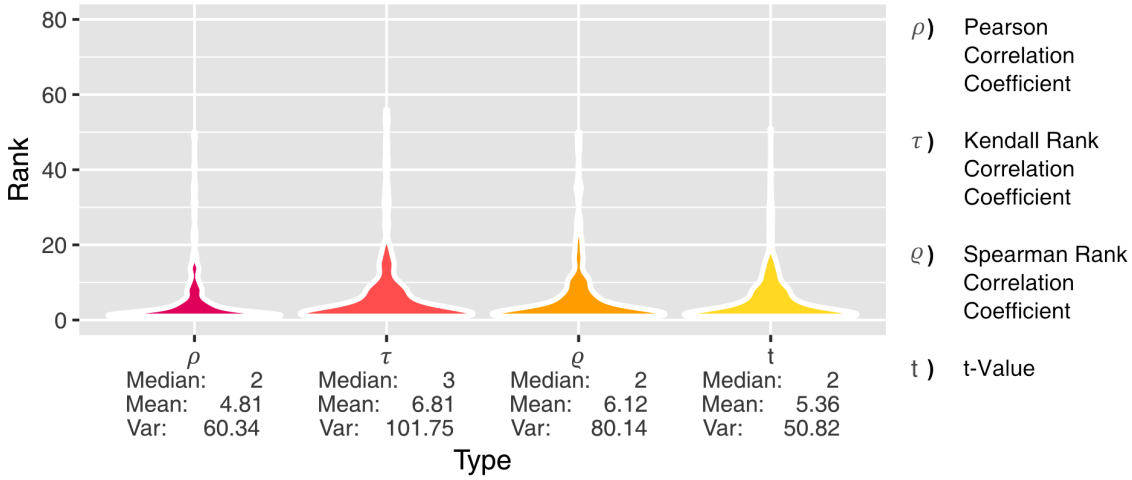

Figure 3.4.2.3 Violin plots of 250 sample-ranks for maximum-density series of length 10.

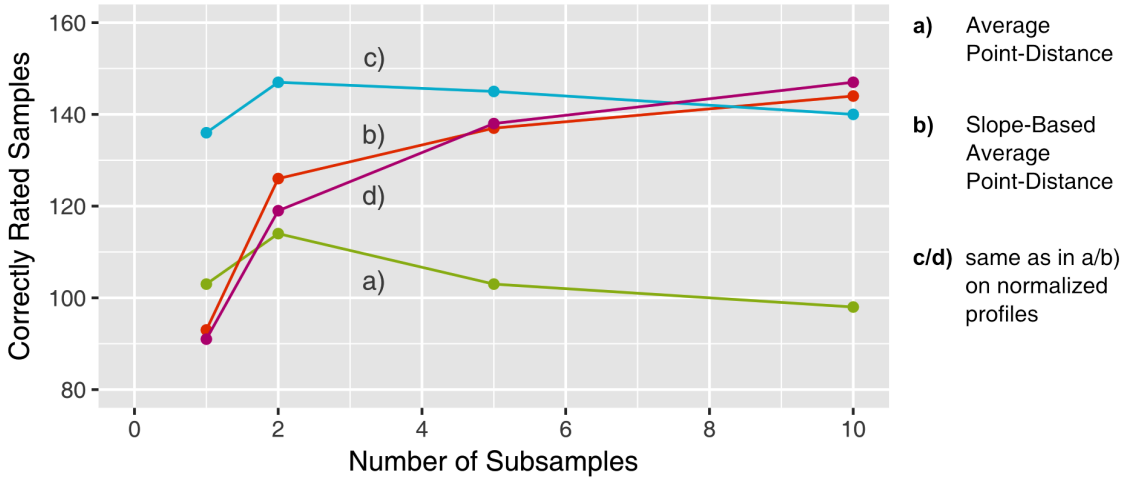

Figure 3.4.2.4 Number of correctly rated samples for 250 test-samples of length 10. It was used different numbers of subsamples in the Per-Tree Approach. No split (whole sample), two subsamples, five and ten subsamples (individual profiles) were considered.

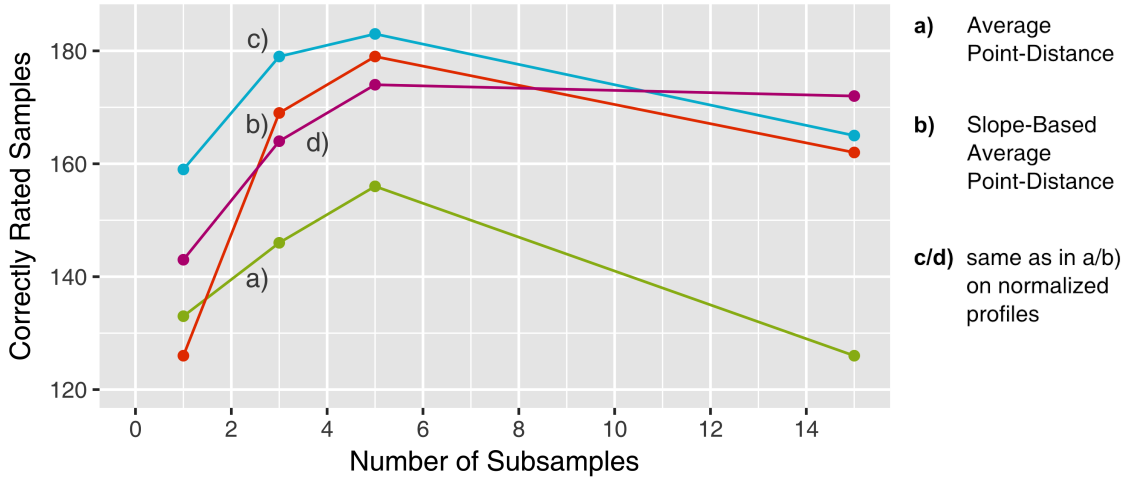

Figure 3.4.2.5 Number of correctly rated samples for 231 test-samples of length 15. No split (whole sample), three subsamples, five and fifteen subsamples (individual profiles) were considered.

The results of splitting length-10-samples into two length-5, five length-2 and ten length-1 subsamples can be seen in Fig. 3.4.2.4. For APDN (c) 147 out of 250 samples were correctly ranked using two subsamples, whereas 172 test-samples were ranked in the top two. For SAPDN (d) similar results were reached using 10 subsamples. Also, 147 were correctly rated and 180 within the first two ranks. In the top 5 were even 194 (= 77.6%). The approach was also executed for length-15-samples using different

numbers of subsamples (Fig. 3.4.2.5). The means and the variance became amazingly low for five subsamples. It was reached a variance of 70.84 and a mean of 3.04 using APDN (c), what has led also to a high number of correct and top-rated samples. 183 ($\approx 79.2\%$) out of 231 samples were correctly ranked, around 85.3% were in the top two and about 90.9% were in the top 5.

3.4.3 Discussion

A new approach was tried out which has led to some improvement. The results for length-10-samples as in Fig. 3.4.2.1 were already promising by comparing with Fig. 3.2.2.3 from the Points-Based Approaches. Instead of only 112 correctly rated test-samples, here in the basic approach even 136 were correctly ranked. Compared to the Consensus Approach in Ch. 3.3, all four distance measuring methods have benefited from the new procedure i.e. the medians were up to 7 times lower and the means about 40% to 50% lower. Especially, the y-based distributions APD (a) and APDN (c) were extraordinarily good compared to earlier results (compare Fig. 3.4.2.1 with Fig. 3.3.2.1). In the Consensus Approach, the method APDN (c) had only 79 (see Ch. 3.3.2) correctly ranked test-samples. But why the Consensus Approach performed that badly compared to the new approach? The problem was the variance within the profiles from the same year as already shown in Fig. 3.3.2.4. This variance had disturbed the computed consensus of the profiles. Determined length-dependent results (Fig. 3.4.2.2) were also compared against the results from the last approach (Fig. 3.3.2.2). The mean and the variance have roughly been halved for lengths 5, 10 and 15. It could also be recognized that the median was overall lower than in the last approach. So the new approach has totally outperformed the first made approach which based on the idea of the Points-Based Approaches. Unfortunately, there was no significant improvement in the per-tree maximum-density approach compared to the previous results of the pure Point-Based Approaches (compare Fig. 3.4.2.3 with Fig. 3.2.2.3), the number of correctly rated test-samples was only slightly improved from 112 to 119. For length-15 maximum-density series and ring-width series this approach was also tested, but showed no improvements with respect to earlier results of pure Points-Based Approaches, such that these results were omitted.

The extension has led to a higher number of correctly and top-rated test-samples. Especially for a subsample-size of around five or below[1], the number of correctly ranked years was optimally (Fig. 3.4.2.4 and Fig. 3.4.2.5). But this Extended Approach was generally as implemented not well applicable, since it was assumed here that samples have a fixed length like 10. In this case the sample could be easily subdivided into equally sized subsamples, but what could be done for samples with non-divisible lengths like length 11? Here, it would be necessary to subdivide into non-equally sized subsamples. The other problem is still the gaps. If there is at least one profile per chronology-year, and one always subdivides test-samples into length-1-samples, a score can be computed. Else it cannot be guaranteed. That finally leads to the next approach called the Bucket Approach which does this length-1-sample subdivision.

[1] two subsamples for length 10 or three subsamples for length 15

3.5 Bucket Approach

3.5.1 Methods

Measuring Quality As seen in the last approach, by comparison with multiple trees it is possible to improve the results significantly due to a better variance handling within a year (see Ch. 3.4.3). Now, instead of subdividing the test-sample, it is done the other way around and the individual chronologies are subdivided into single profiles per year which are all put in buckets (see Ch. 2.1.2). That simply equals the special case of the Per-Tree Approach in which it is subdivided into individual profiles. So for each test-sample profile, it is checked against all profiles from all chronologies within a year. Therefore, a Buckets Chronology (Def. 2.1.2.6) is used. In Fig. 3.5.1.1, it can be seen schematically how the bucket-wise computation works. The shifting is analogous to the previous approaches of profile-wise shifting. Samples are shifted along the chronology and for each position a score is computed. Therefore, sample-profiles are pair-wisely compared with buckets i.e. scores are computed as in Def. 2.2.1.7. That means, between every sample-profile and each corresponding profile in the bucket a distance is computed and then from all these generated distances, the minimum is selected. All profile-wise or bucket-wise minimum distances are then added together to build a final distance for a given start-year.

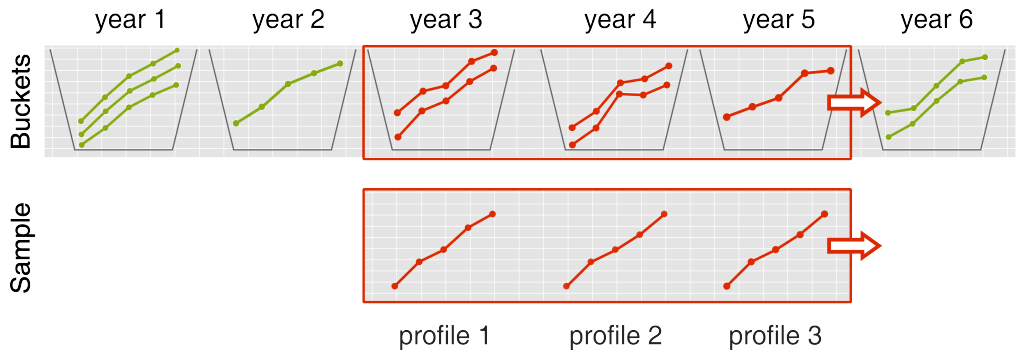

Figure 3.5.1.1 Schematic profile-wise shifting. A sample of length 3 is shifted along a Buckets Chronology of length 6. To compute a distance for a specific start-year, every sample profile is compared with each profile from the corresponding bucket.

So far, the scores created with the individual sample-profiles were simply added up to create a final score for a given year. Now also, alternative strategies were investigated, to learn more about this new procedure and the data. Therefore, the summation-function Σ was replaced (see Def. 2.2.1.6) with the max- and min-function. Why exactly these two functions? The assumption was that the final score is more dependent from the highest subscore or lowest subscore within a list of minimum scores.

Also, it was checked from how many profiles of different trees, the rank-1-rated test-samples constructed their final distance-score. So instead of storing scores for test-samples at each year in the chronology, the corresponding number of trees was stored. By this, for length-10-samples accordingly values between 1 and 10 are derived, since every profile can come from a different tree.

Measuring Reliability In Ch. 2.2.3 different reliability measures were formally described. They tell how sure the algorithm is about a given solution. It was now checked if they are suitable at all for the reliability measurement. First, Δ Scores (Def. 2.2.3.3) were checked. Therefore, the differences in the scores of rank 1 predictions and rank 2 predictions for length-10-samples were examined. For each of the 250 samples from the five passes (see Ch. 3.1.2) the rank 1 prediction scores and the rank 2 prediction scores were computed. Then for these two lowest scores the difference was plotted to recognize how well the Δ Scores work.

As a second approach, p-values were computed on score-distributions (Def. 2.2.3.8). This is an established approach used to measure reliability [7]. The distance-scores from each year generated during sample-shifting were collected. Then, three different distributions were tested concerning their fitting to the scores. Tested were the Log-Normal Distribution, the Gamma Distribution and the Weibull Distribution. Why exactly these distributions were selected? Within the histograms of the score-distributions, the outliers were mostly positive (Fig. 3.5.2.10). So the histograms were asymmetric and with respect to [13], in this case these three distributions should be tried. But to decide which distribution was actually best capturing the empirical score-distribution, the Kolmogorov-Smirnov statistic (Def. 2.2.3.7) as well as other statistics and criteria available in the `fitdistrplus` package were considered for all three distributions.

For the finally chosen distribution, p-values were computed. The lowest distance-score was used as a threshold (Def. 2.2.3.8). That means, the p-value was computed up to the minimum distance-score $\hat{\varsigma}_1$ (Def. 2.2.3.3) to test the significance of that score with respect to the entire distribution.

3.5.2 Results

Quality For length-10-samples the results of this new approach are shown in Fig. 3.5.2.1. It was also looked on the number of correct and top ranked samples (Tab. 3.5.2.1) because visually there were no large differences anymore between the different violin plots. This way, the best method was recognized which was in this case SAPDN (d). For that 147 samples, so 58.8% were correctly ranked. Further, alternative strategies to the simple addition of minimum scores (Fig. 3.5.2.2 and Fig. 3.5.2.3) were investigated. These are choosing the maximum or minimum out of these minimum-scores. Compared with the results from the min-function, the rankings were clearly better by choosing the max-function. The parameters like variance were lower. Nevertheless, the overall performance was worse than by choosing the summation-function.

Method	APD (a)	SAPD (b)	APDN (c)	SAPDN (d)
Total Rated Samples	250			
Rank				
1	98	144	140	147
2	19	26	17	33
3	14	10	10	4
4	11	7	5	5
5	11	3	5	5
> 5	97	60	73	56

Table 3.5.2.1 Values of distribution for the first five ranks of the methods a)-d). The formula for the Average Point-Distance (a) can be found in Def. 2.2.1.2, whereas the Slope-Based Average Point-Distance (b) is described in Def. 2.2.1.3. To transform into normalized profiles Def. 2.2.1.1 was applied on the y-values of the profiles.

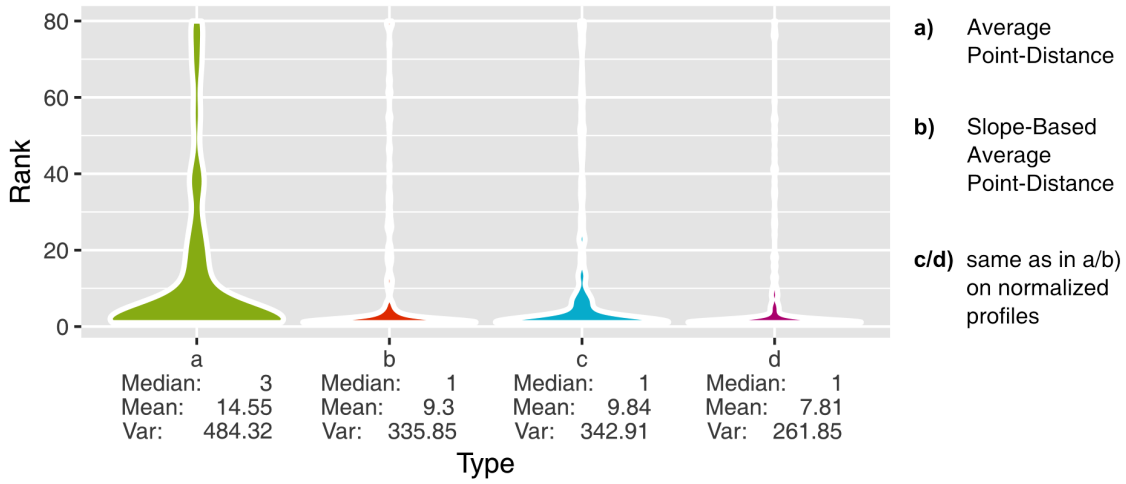

Figure 3.5.2.1 Violin plots of 250 sample-ranks under a Buckets Chronology with series of length 10.

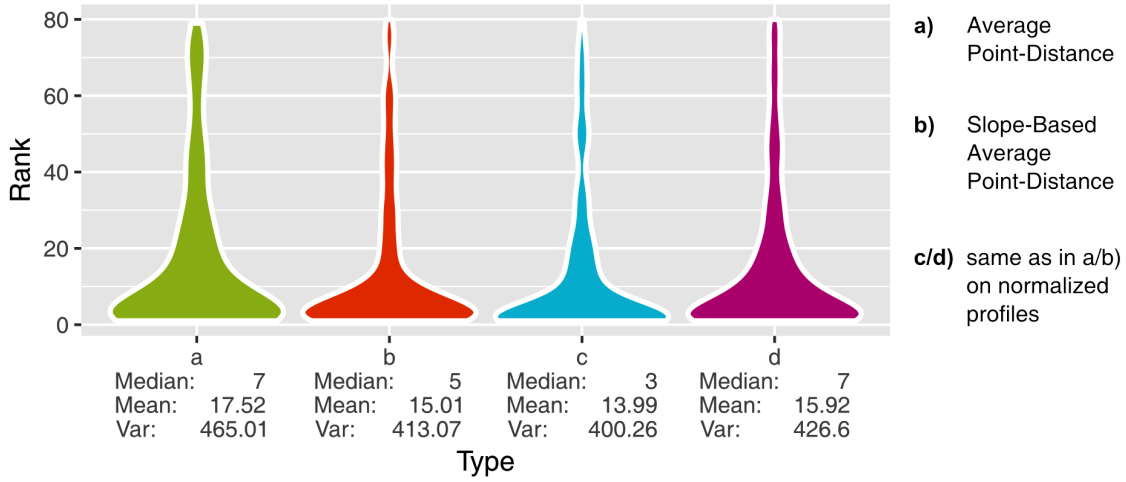

Figure 3.5.2.2 Violin plots of 250 sample-ranks under a Buckets Chronology with series of length 10 and func = max for the computation of the final score (Def. 2.2.1.6).

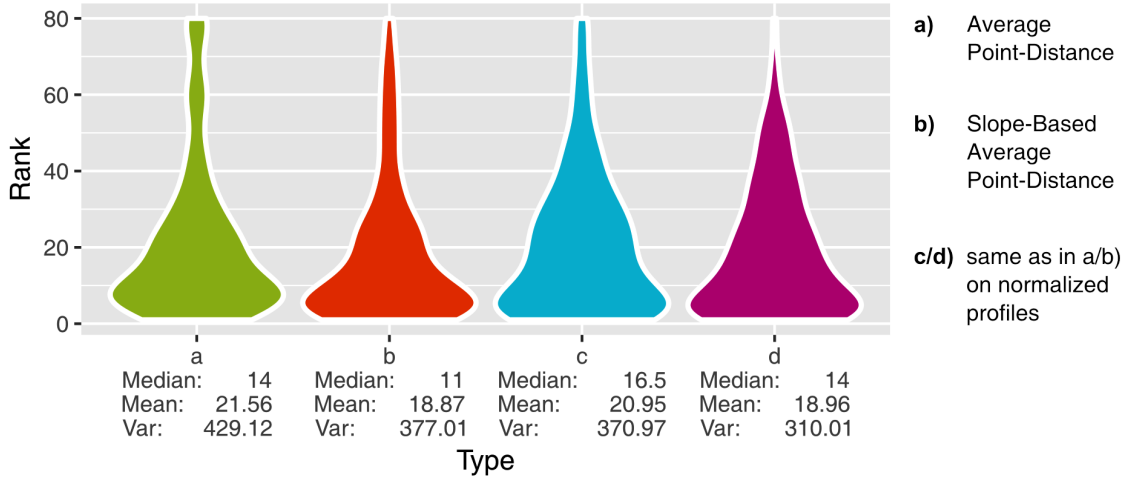

Figure 3.5.2.3 Violin plots of 250 sample-ranks under a Buckets Chronology with series of length 10 and func = min for the computation of the final score.

Finally, length-dependency of this approach was examined, length-5 test-samples were ranked (Fig. 3.5.2.4). The median-ranks were around the value 3 for three of the four methods. The method SAPD (b) has shown the highest number of top-rated samples, concretely 317 ($\approx 64\%$) out of 495 were in the top 5 from which 168 had rank 1. For samples with length 15, the results can be seen in Fig. 3.5.2.5. From 231 test-samples, 172 ($\approx 74.5\%$) were correctly ranked and overall 203 ($\approx 87.9\%$) were under the best five ranks.

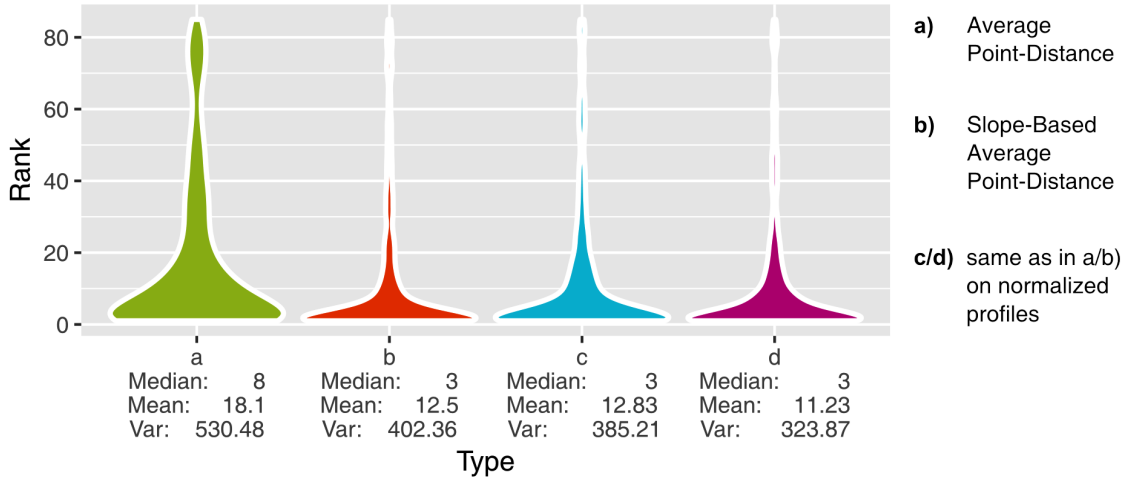

Figure 3.5.2.4 Violin plots of 495 sample-ranks under a Buckets Chronology with series of length 5.

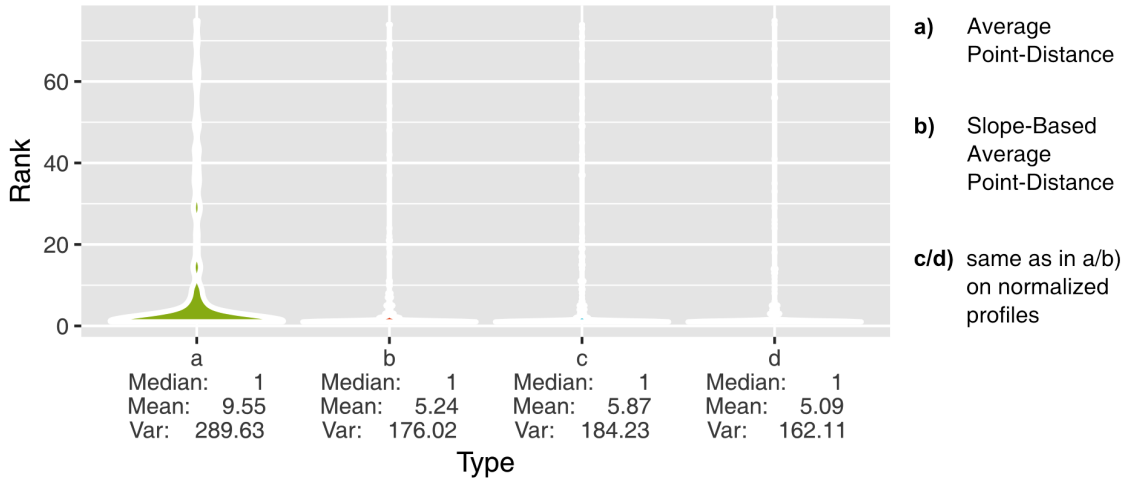

Figure 3.5.2.5 Violin plots of 231 sample-ranks under a Buckets Chronology with series of length 15.

The number of trees from which the minimum-score for rank-1-rated test-samples was build using method SAPDN (d) can be seen in Fig. 3.5.2.6. In the figure, it was clearly recognizable that most scores were generated with the profiles from seven or eight trees. But, did this also hold for test-samples which were rated not with rank 1 (Fig. 3.5.2.7)? For this samples, it was a little more flattened. However, there was still a significant increase around seven and eight trees.

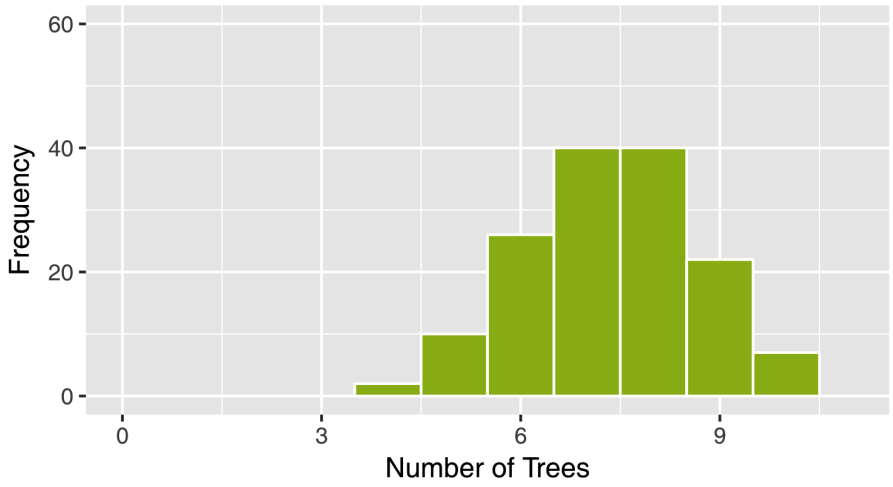

Figure 3.5.2.6 Number of profiles from different trees in the rank-1-rated samples. 250 length-10-samples were shifted along a master chronology and with the Slope-Based Average Point-Distance (d) on normalized profiles, scores were computed. But it was stored the corresponding number of trees from which the scores were generated.

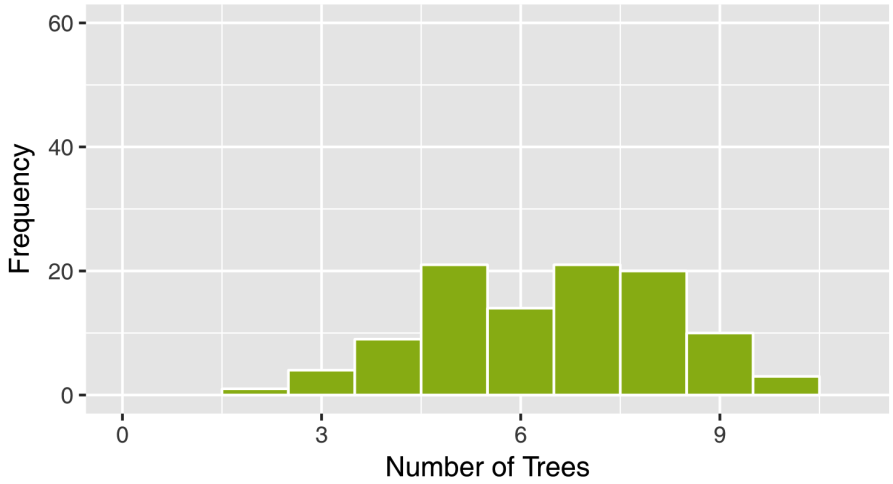

Figure 3.5.2.7 Number of profiles from different trees in the non-rank-1-rated samples.

Reliability To see, how well Δ Scores (Def. 2.2.3.3) are applicable as a reliability measure, the Δ Scores of the 250 length-10 test-samples were mapped on the x-axis of a scatter-plot (Fig. 3.5.2.8). On the y-axis, the corresponding correct rank of the sample was mapped. The higher the Δ Score, the lower were the ranks. An example, starting at a Δ Score of about 1, most ranks to the right of 1 were rated with rank 1. This proves, Δ Scores can be used as a reliability measure. Such plots were computed

for all four tested distance methods and their results were very similar, but SAPDN (d) has shown the best results i.e. the bad rated samples were not spread that much along all available Δ Scores and this was also measurable. Therefore, the hundred samples having the highest Δ Scores were examined. From this hundred samples, 91 had rank 1 using SAPDN (d), whereas for the other methods this number was lower (Tab. 3.5.2.2). Actually, SAPDN (d) was also the method, which has shown the best ranks in Tab. 3.5.2.1. A second plot was created which only indicated how often rank 1 predictions had actually rank 1 for different Δ Scores. Therefore, on the y-axis the answer on the question *Is rank 1?* was plotted, so `TRUE` and `FALSE` were plotted on this axis and on the x-axis once more the Δ Score was plotted (Fig. 3.5.2.9). There, in the lower plot, the trend was recognizable that with higher Δ Scores, the tendency to have a rank-1-rated test-sample increases.

Method	APD (a)	SAPD (b)	APDN (c)	SAPDN d)
Rank 1 Rated	66	86	89	91

Table 3.5.2.2 Number of rank-1-rated samples within the samples having the hundred highest Δ Scores. The formula for Average Point-Distance (a) can be found in Def. 2.2.1.2, the Slope-Based Average Point-Distance (b) is described in Def. 2.2.1.3.

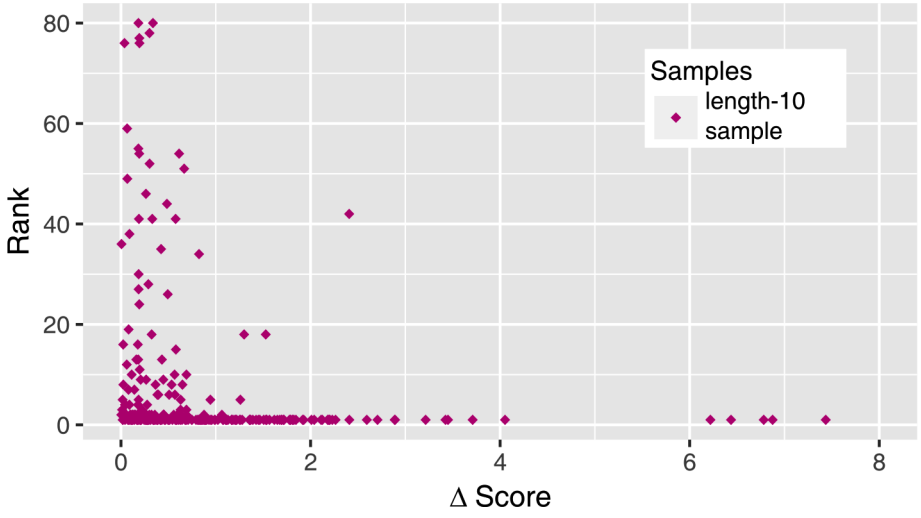

Figure 3.5.2.8 Ranks for the Slope-Based Average Point-Distance (d) and their corresponding differences in the scores. The second lowest and lowest generated score were selected and the Δ Score (Def. 2.2.3.3) between both were computed and mapped on the horizontal axis. The corresponding predicted rank for the sample was plotted on the vertical axis.

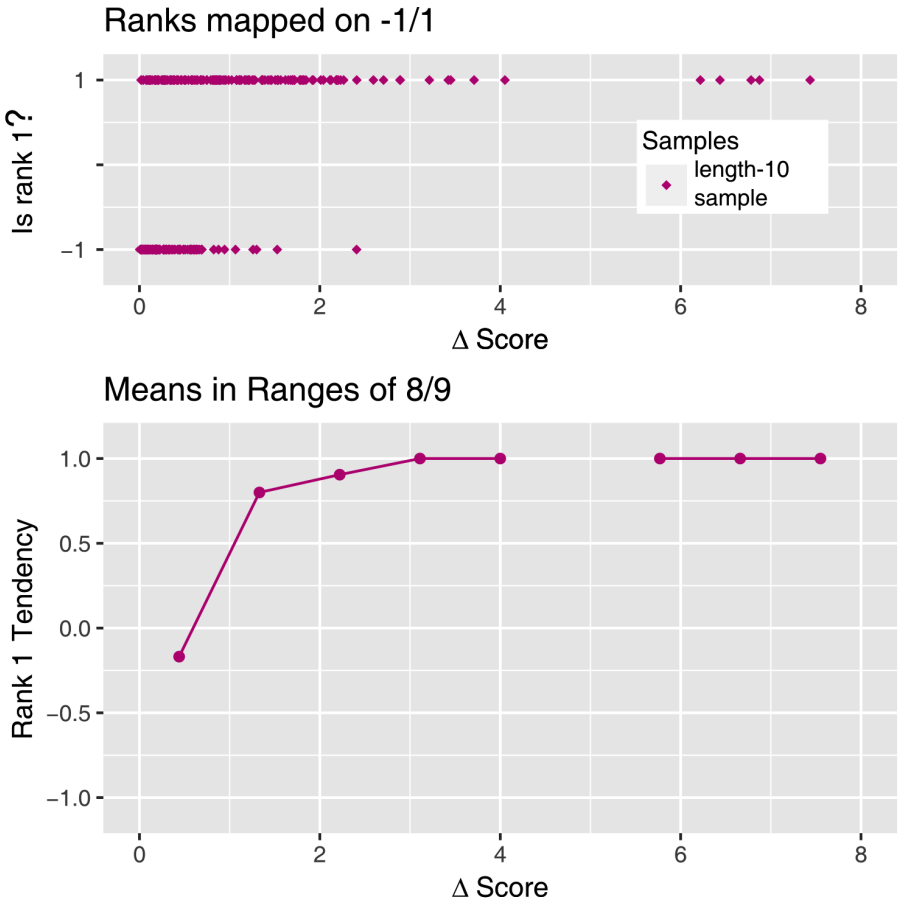

Figure 3.5.2.9 Measuring the trend for rank 1. The value -1 equals `FALSE` and the value 1 equals `TRUE`. The upper plot shows how the length-10-samples were rated. In the second plot below the truth-values from the first plot were averaged in ranges of $\frac{8}{9}$ Δ Score-units and corresponding averages were set into ranges' centres.

Furthermore, p-values were tested as an established reliability measurement. Therefore, first different distributions were tested for their suitability to the per-year generated distance-scores of a sample. The Weibull Distribution has led to the worst results here because it couldn't capture the maxima correctly (Fig. 3.5.2.10). And the Gamma Distribution and the Log-Normal Distribution led to nearly identical fittings. The Log-Normal fitted always better than the Gamma and the Weibull Distribution. On the Log-Normal Distribution the Kolmogorov-Smirnov Test was applied and from 250 length-10-samples, 89.6% have passed the test under a significance level λ of 5%. The Gamma Distribution has reached 84.4% in the same test and the Weibull Distribution only 32.8%. So the test said that the Log-Normal Distribution is the best model for the given scores. This Log-Normal Distribution was then used to derive p-values. The results can be seen in Fig. 3.5.2.11. In that plot, it was recognized, that p-values around 1% and below are significant enough to increase the confidence for a solution. To compare with the first approach (Fig. 3.5.2.8), it was again looked on the hundred best ranked samples. From the ones with the lowest hundred p-values, 88 had rank 1.

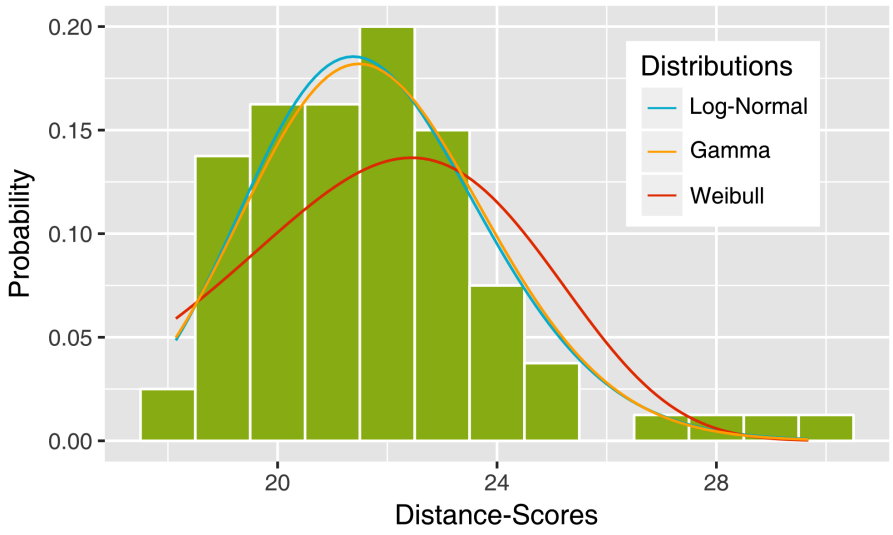

Figure 3.5.2.10 Fitting of different distributions to the generated per-year distance-scores.

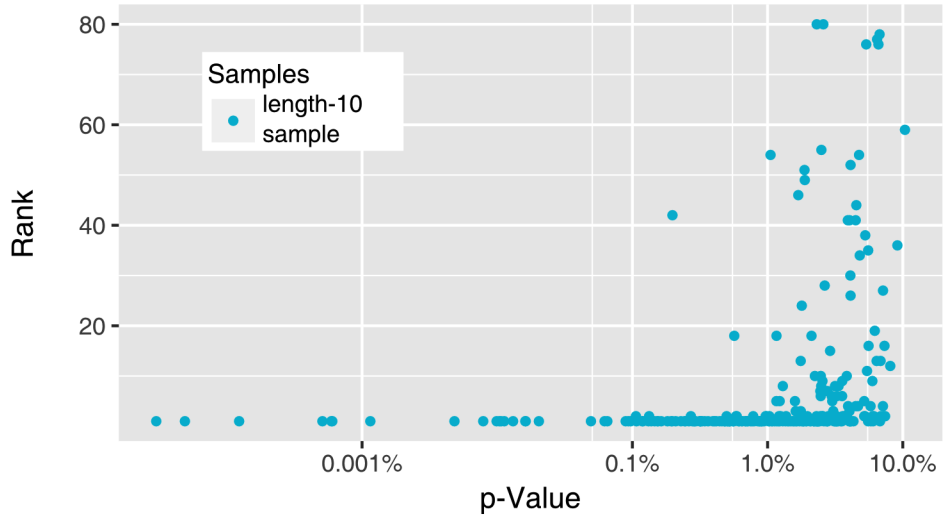

Figure 3.5.2.11 The ranks for the Slope-Based Average Point-Distance (d) and the corresponding p-values. It was measured the left-sided p-values (Def. 2.2.3.8) from minus infinity up to the minimum score $\hat{\varsigma}_1$. These were mapped on the horizontal axis, whereas the ranks were mapped on the vertical axis (compare with Fig. 3.5.2.8).

3.5.3 Discussion

The mentioned variance problem discussed in Ch. 3.3.3 (see Fig. 3.3.2.4) was solved by selecting the minimum-distance in each bucket. So for each profile in a bucket, a distance-score was computed by comparison with a test-sample profile and then from all that scores, the minimum was selected. The idea behind this operation is to select the most similar profile in a bucket and to compute with that the distance, so in order to avoid selecting an inappropriate profile that does not fit to the test-sample profile. Alternatives to the summation of these minimum-scores were also tested. By this, it was shown that especially the minimum function func = min (Def. 2.2.1.6) leads to very bad results compared to the results of the maximum function func = max. So the highest values in a list of minimum-scores were more important than the low values. But how this knowledge could be used to improve the results was another question. Different approaches were tested, however, nothing has led to any improvement.

Some results of the Bucket Approach were also discussed in Ch. 3.4.3. The performance for length 1 can already be recognized in the length-dependency comparison from Fig. 3.4.2.2 of the Per-Tree Approach and method APDN (c). For length-5-samples in the Per-Tree Approach around 59.4% were in top 5 ranks with the best method APDN (c) (Fig. 3.4.2.2). The best method from the new approach SAPD (b) had for length 5 around 64% top 5 rated samples. However, APDN (c) had more correctly rated samples in the Per-Tree Approach, concretely about 36.4% instead of around 33.9%. Probably, this was just the case because of the variance i.e. the Per-Tree Approach should not have generally more correctly rated samples, it also depends on the used data. Comparing with APDN (c) in the Per-Tree Approach for length 10, there was an improvement of 4.4% (Tab. 3.5.2.1), since in the Per-Tree Approach only 136 test-samples were correctly ranked. For length 15, there is Fig. 3.5.2.5. SAPDN (d) had there about 87.9% samples ranked in the top 5 compared to 84.8% with the Per-Tree Approach best method APDN (c) (see Ch. 3.4.2). But with subdivision, even the Bucket Approach has been outperformed on length-15-samples (Fig. 3.4.2.5). By division into five length-3 samples, the Per-Tree Approach reached about 79.2% correctly rated samples instead of only ~74.5%. But the difference in the top 5 was not high, the Per-Tree Approach reached about 90.9%. The reason, therefore, was a lower mean and a lower variance for the rank distributions i.e. the violin plots. The mean was about 40.3% below the mean of the Bucket Approach and the variance was about 56.3% lower by comparison with subdivision into length-3-subsamples. But additionally, the Bucket Approach has also solved the problem mentioned in Ch. 3.4.3 with gaps. That means, if one gets a score for a start-year with the Consensus Approach, then it is also possible to generate a score with the Bucket Approach, but this is not generally true for the Per-Tree Approach. Howsoever, the Bucket Approach was also compared to Points-Based Approaches. For length-15-samples the Points-Based Approaches using maximum-densities were superior to the Bucket Approach. In the top 5, the Points-Based Approach reached 95.2%. But for length-10-samples, the results were again inferior to the ones of the Bucket Approach. There, the Points-Based Approach maximally reached 112 (= 44.8%) instead of 147 (= 58.8%) correctly rated test-samples.

Two approaches for a reliability measurement were tested, but which one was better now? The statistics i.e. the applied rule of thumb to look on the hundred best reliability values and measure how many samples have rank 1 showed very similar results for both approaches. The p-value approach showed 88 rank-1-rated samples by applying the rule, whereas the Δ Scores approach showed 91 rank 1 samples within the hundred best. But with more generated scores i.e. longer chronologies the p-value approach would probably outperform the Δ Scores approach, since it contains also information about the distribution. So with more per-year generated scores, one should presumably trust more the p-values. But due to the short chronology, it made sense to output both Δ Scores and p-values, since both deliver a criterion with which one can decide to trust or not to trust a solution.

What are now the drawbacks and benefits of the new method with buckets? The new method is slower compared to earlier methods, since instead of comparing a sample-profile only against e.g. one Consensus-Profile, the sample-profile has to be compared against every profile in a bucket. But the advantage, compared to the Consensus Approach in Ch. 3.3, is that no time-consuming computation for a Consensus Chronology is necessary anymore in a pre-processing. A second drawback is the memory consumption, which is of course higher since the chronology contains buckets of profiles instead of single profiles. Also, it could be a problem how the variance problem is handled now. Probably if there will be too many profiles in the buckets or if the chronology much longer then the approach could become unstable i.e. the results could get worse. Why should this be the case? The reason is, if there are more profiles in a bucket than the probability is high to get a very similar profile in the bucket for a profile in the test-sample.

3.6 Two-Step Approach

3.6.1 Methods

The Points-Based Approaches are very fast, but the results are behind those of the slow Bucket Approach. The following Two-Step Approach combines both approaches to get an overall fast and precise algorithm (Fig. 3.6.1.1).

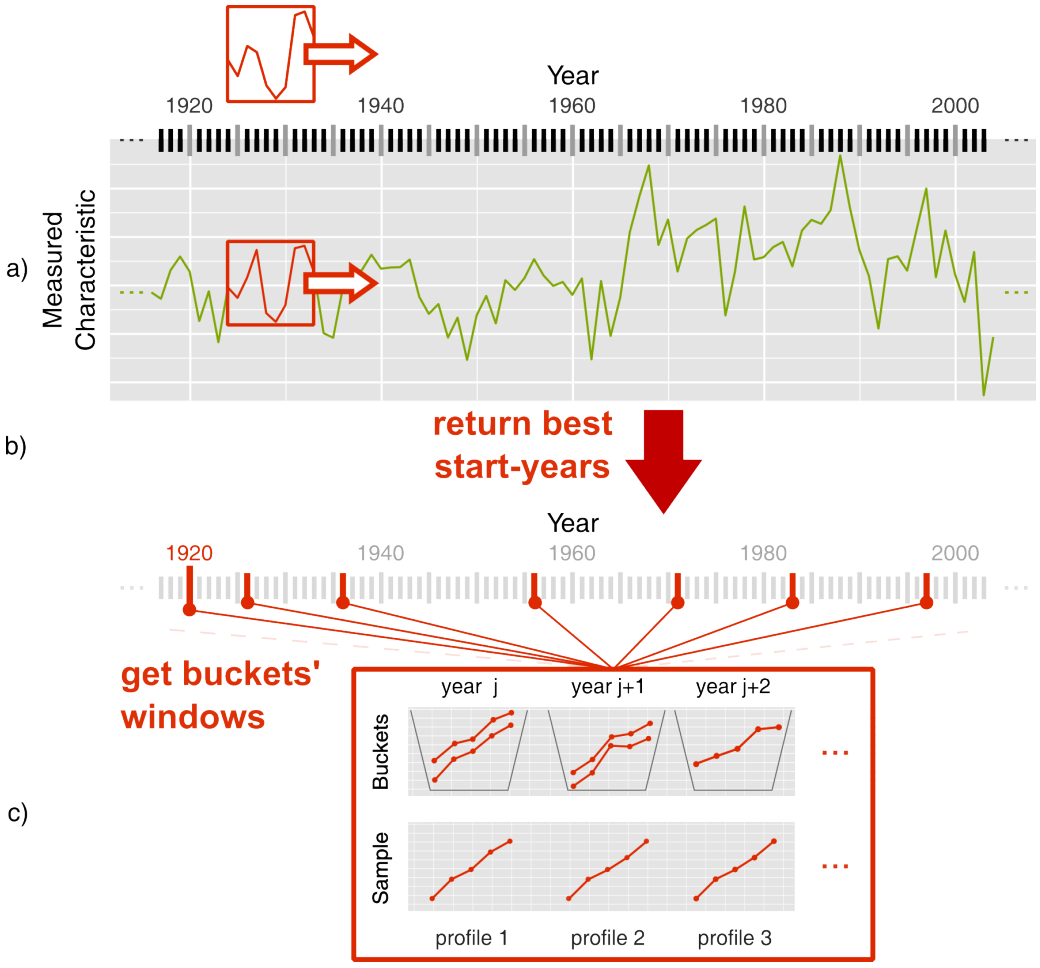

Figure 3.6.1.1 Combination of Points-Based Approaches with the Bucket Approach. **a)** a sample of points is shifted along a chronology to compute correlation coefficients, or t-values for each start-year (see Ch. 2.2.2), **b)** multiple start-years with the highest correlation coefficients / t-values are returned, **c)** the corresponding windows of buckets are extracted given the start-years and only on these windows the Bucket Approach is executed.

First a classical Points-Based Approach is applied i.e. ring-width and maximum-density samples are searched within a chronology. Then, a number of top rated positions i.e. years with the highest correlation coefficients, or t-values, is stored. Then for these top-years, the Bucket Approach is applied by extracting the corresponding windows of buckets at the given positions. So in the Buckets Chronology only these top rated start-years are checked. Again, the ranks of the correct scores of samples are evaluated (see Ch. 3.3.1). The found ranks are plotted into a histogram. Violin plots were not used since the number of years is limited and a direct comparison with other approaches is not possible.

3.6.2 Results

The number of years that should be passed by the Bucket Approach was limited. Therefore, different numbers of years returned by the Points-Based Approaches were tested and finally the number was fixed to twenty years. But why exactly twenty years were checked? With twenty years the highest number of top 5 rated test-samples was reached as can be seen in Fig. 3.6.2.1. 83.6% of the test-samples were in the top 5 and there was also a high number of correctly rated samples, here 69.6%. The number of correctly rated samples could be increased up to 70.8% by selecting to return 17 years, but then the number of top rated years was lower. 25 returned years were not an option since in that case the number of correct years has significantly decreased (data not shown).

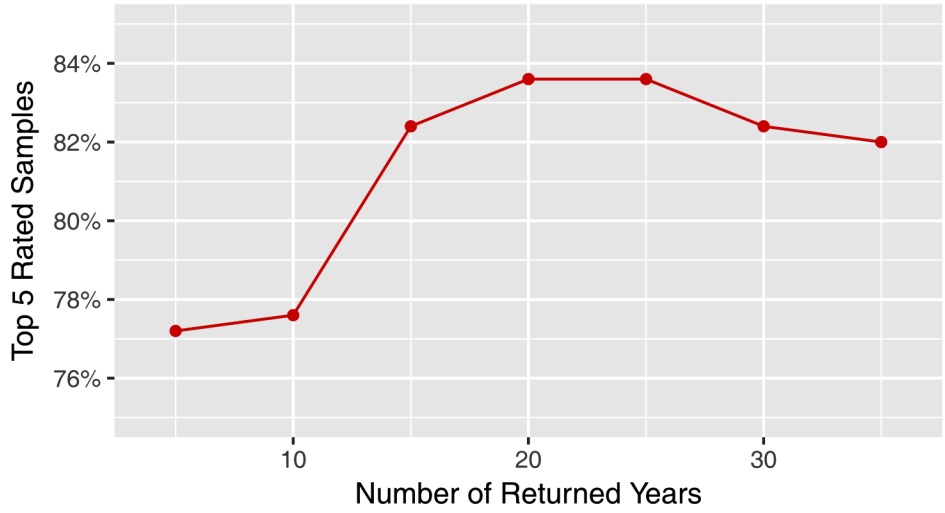

Figure 3.6.2.1 The different number of returned years by the Points-Based Approach (using maximum densities) and the corresponding number of top 5 rated test-samples in the Two-Step Approach. There has been used length-10-samples to create this diagram and the Slope-Based Average Point-Distance on normalized profiles (d) as the distance method in the second step.

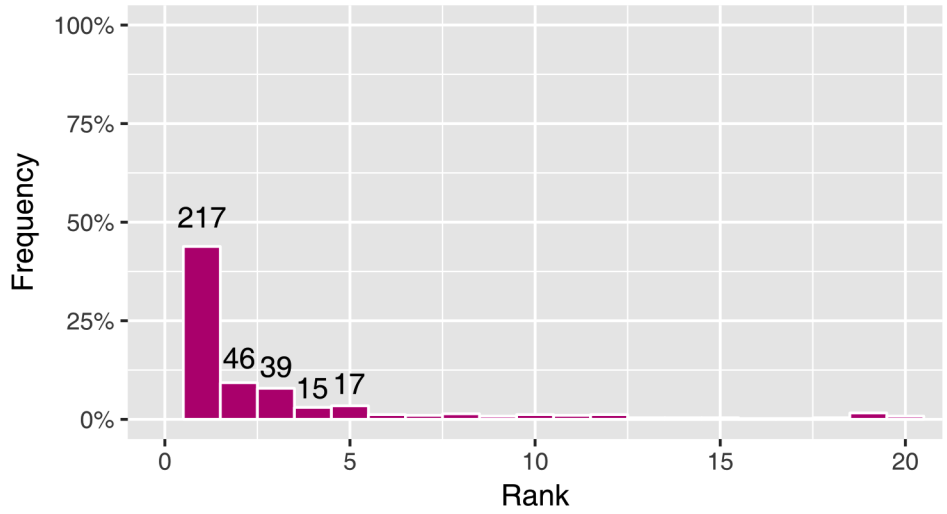

Figure 3.6.2.2 Two-Step Approach rank-histogram for 495 length-5-samples. After applying the Points-Based Approach under the Pearson Correlation Coefficient (ρ) (Def. 2.2.2.1) and storing the 20 best years per sample, the Slope-Based Average Point-Distance on normalized profiles (d) was applied only on these 20 years to create a histogram of the ranks. On the first five bars, the absolute numbers were written.

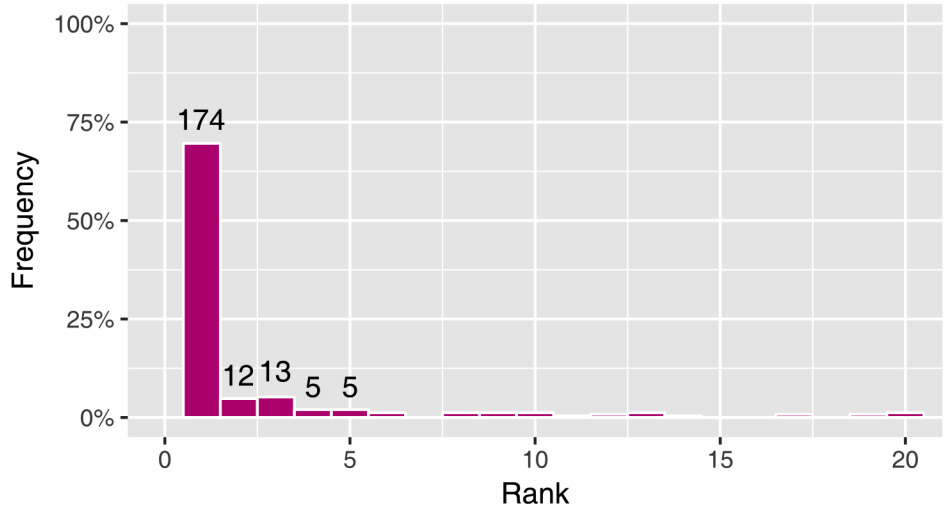

Figure 3.6.2.3 Two-Step Approach rank-histogram for 250 length-10-samples. In the first step the Pearson Correlation Coefficient (ρ) and in the second step the Slope-Based Average Point-Distance on normalized profiles (d) was applied. On the first five bars, the absolute numbers were written.

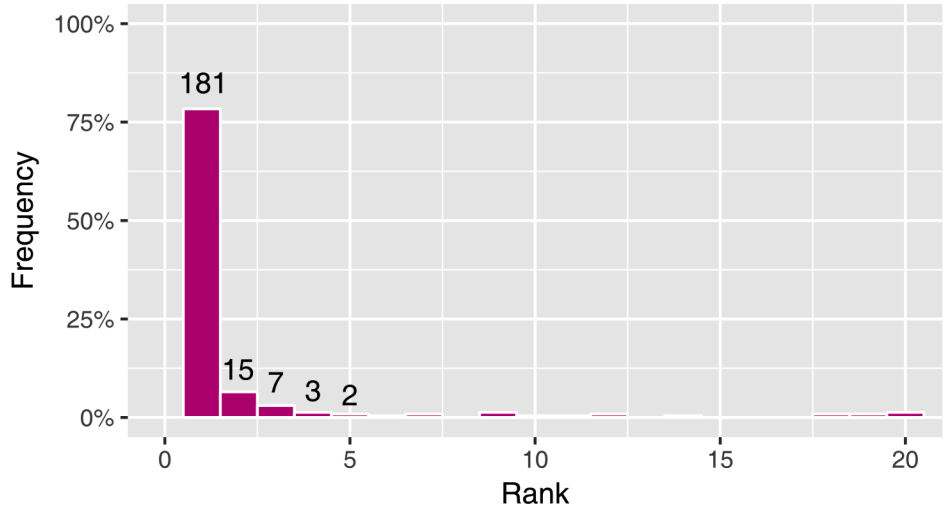

Figure 3.6.2.4 Two-Step Approach rank-histogram for 231 length-15-samples. In the first step the Pearson Correlation Coefficient (ρ), in the second step the Slope-Based Average Point-Distance on normalized profiles (d) was applied.

Using maximum densities in the first step to limit the number of years for the Bucket Approach has led to about 43.8% correctly rated length-5-samples and about 67.5% test-samples in the top 5 (Fig. 3.6.2.2). Therefore, the Slope-Based Average Point-Distance on normalized profiles was used which has already shown good results with the Bucket Approach (see Ch. 3.5.2). For length-10-samples, the results were even better (Fig. 3.6.2.3). Here, 69.6% were correctly ranked and 83.6% were in the top 5 ranks. With length-15-samples the results were also very good (Fig. 3.6.2.4). 208 ($\approx$ 90%) out of 231 test-samples were under the top 5. From which 181 ($\approx$ 78.4%) were correct. Using ring-widths with the t-value method in the first step has led to similar results. 65.6% were correct and 81.6% were in the top 5 ranks using length-10-samples (data not shown). The approach was also much faster than the Bucket Approach, 50 length-10-samples were processed in about 1 minute and 17 seconds. Therefore, the runtime was measured five times with length-10-samples to get a representative average value. The same was repeated using the Bucket Approach. On average, it needed 4 minutes and 39 seconds.

3.6.3 Discussion

The Two-Step Approach was compared with earlier results from the Bucket Approach (see Ch. 3.5.2). For length-1-samples an execution of the approach was not possible since the correlation coefficients need at least two values i.e. length-2 samples. With length-5 test-samples, 43.8% were correctly rated, whereas with the best method SAPD (b) of the Bucket Approach, 33.9% of the test-samples were correctly ranked. The number of correctly rated test-samples with the Bucket Approach using length-10-samples was 58.8%, so 10.8% lower than with this new approach. For length-15-samples,

the difference was not that big anymore, only an improvement by about 4% was reached. But why the results got better compared to the Bucket Approach? The first step in the Two-Step Approach can be seen as the application of a heuristic. That heuristic provides an approximate solution which is then further refined by applying the Bucket Approach. The returned dates or years from a Points-Based Approach contain with a high probability the correct date since means and variance in such approaches are very low (see Fig. 3.2.2.4). If now the Bucket Approach is applied on the filtered subset of potential candidates of correct dates, it can find the correct date due to a high accuracy. Howsoever, this technique with applying a fast approach in the first step and a slow procedure in the second step, is very common in computer science. So far, it can be said that the Bucket Approach was superseded by the Two-Step Approach which is significantly faster and more precise. Thus, it is up to now from all tested approaches the most accurate one.

3.7 Voting Approach

3.7.1 Methods

Measuring Quality Finally, also, a statistical approach was tested which based on the procedure with buckets. First a very simple algorithm is introduced for this purpose (Fig. 3.7.1.1) that doesn't promise to show any improvement. But it is the basis for a more complex one which could even be able to outperform the Bucket Approach. Here, again a sample is shifted along the Buckets Chronology, but instead of computing a single score for the whole sample for a specific year, all scores created with the sample were stored separately. This is similar with setting the function "func" in Def. 2.2.1.6 to an order preserving identity function i.e. the scores should be returned in the order they've been created from left to right to get the depicted score-table (Tab. 3.7.1.1).

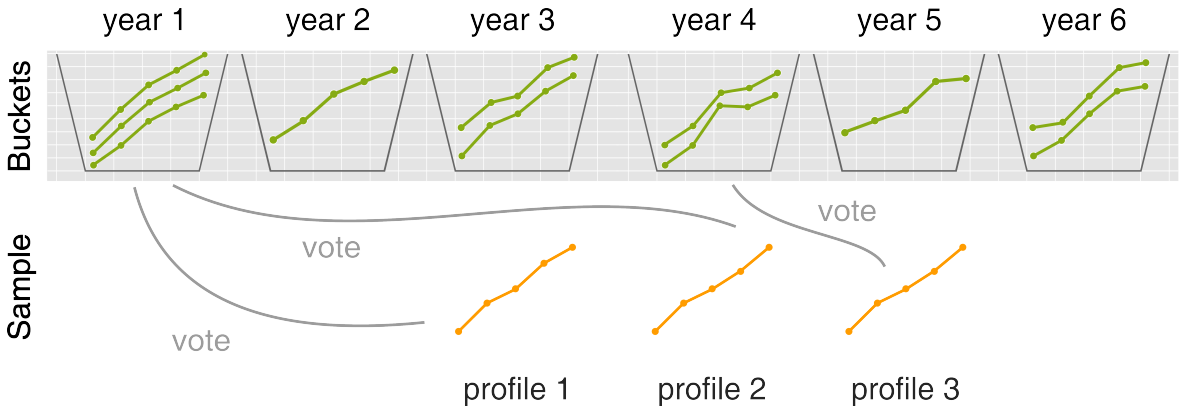

Figure 3.7.1.1 Profile-wise start-year voting for a sample of length 3 and a Buckets Chronology of length 6. Each sample-profile votes for a start-year to build afterwards a histogram and select a winner, i.e. the most promising year for the whole sample.

Start-Year of Sample	Profile 1	Profile 2	Profile 3
1	*0.17*	*0.16*	0.21
2	0.22	0.18	0.34
3	0.19	0.32	0.32
4	0.3	0.33	*0.2*

Table 3.7.1.1 Corresponding exemplary score-table for the sample of length 3 and the chronology of length 6 in Fig. 3.7.1.1. Each line represents the produced scores from a single shift i.e. the scores generated with individual profiles from a sample S and a bucket subsequence $S_i^{\mathbf{B}}$ in the start-year i. Lowest scores which led to vote for a start-year were italicized.

In order to avoid any misinterpretation, Tab. 3.7.1.1 represents in each line a single shift! So the italicized value 0.16 for the second profile was created when sample-profile 2 was compared against the profiles from year 2 in the chronology. And analogous the value 0.21 for the third profile was created. So the third profile in the sample was compared against the year 3 in the chronology. In the four shifts, sample-profile 2 is only compared against years 2 up to 5 and profile 3 is compared against years 3 up to 6. The scores from these comparisons are then stored one below the other to simplify voting for a start-year. As can be seen in Fig. 3.7.1.1, it was voted two times for year 1 and ones for the year 4 by profile 3. That was the case because the profiles have reached their lowest distance-scores in that start-years (compare with Tab. 3.7.1.1). Howsoever, the samples which were used for comparisons had all the length 10. The tables stored for each sample had eleven columns (ten score-columns plus the year-column). These tables were then used to compute histograms. Therefore, the ten best years from each of the ten score-columns were calculated i.e. the column-values were sorted in ascending order and then the sorting was applied on the unsorted years. The ten first years in each of these sortings were then used to compute a histogram. So in this case all 100 years were taken and a histogram was calculated for each test-sample. Thus, with respect to figure 3.7.1.1 each profile has voted for ten and not only for one year.

Afterwards, for each generated histogram a so-called peak-rank for the correct sample start-year was computed. So analogously to the last approaches, now instead of the correct score, the correct start-year of a sample was taken and its peak-rank was looked up in the histogram. How was the peak-rank calculated? The frequency of the correct sample start-year was identified. Then all histogram bars were sorted depending on their frequency value in a descending order and the index[1] of the last bar with the correct start-year-frequency was used as the peak-rank. Sometimes the correct year of a sample was not contained in the corresponding histogram, such a sample is called an outcast. The peak-rank-distributions are not directly comparable with the rank-distributions that were computed in the last approaches due to the outcasts, such that now histogram-plots were created instead of violin-plots.

Extended Approach Information given by the individual profiles and their combination to small subsets shouldn't get lost. How could this be incorporated? Given the per year scores as in table 3.7.1.1, for every subset of sample-scores in a row, a new column with the sum was computed. They idea behind this was that profiles should also be able to decide together. For them it should be possible to give together a single vote, and this was realized by building common scores. So for a length-5-sample, one gets a new table with $2^5 - 1$ different sums per row. For each of the 31 columns, the year showing the minimum sum was identified. But these 31 years were not all equally important. The score or sum build by a length 5 subset, i.e. a score from 5 summands, is more important than the value from a single profile. So the selected year with the minimum distance-score from a length 5 subset was counted 5 times. Length 4 subset years were counted 4 times and so on. By this the corresponding histogram altogether gets 80 votes (Tab. 3.7.1.2).

[1] indices in programming language R start with 1

Subset-Length i	1	2	3	4	5
Subset-Length Frequency $\binom{5}{i}$	5	10	10	5	1
Total Votes $\sum_{i=1}^{5} \binom{5}{i} \cdot i$			80		

Table 3.7.1.2 Table for the computation of the final number of votes within a histogram in the Extended Voting Approach under a length-5-sample. The entries of both rows have to be multiplied and then the products have to be summed up to get a final number of votes which equals 80.

But the problem with looking only at the best year in each column has led again to a loss of data e.g. the rank 2 predictions were not considered for each column. Such that as in the first, simple Voting Approach multiple years were considered. Per column between one and four years were tested. And another, so-called double weighted variant of the approach was created. That means, dependent on the predicted rank, years were counted differently i.e. length 5, five times if it is a rank 1, and if it is rank 2, four times and so on. So the years were now also weighted by their ranks within a column and not only by their (sub-)sample-length.

The Extended Approach was also checked for length-10-samples, but due to the assumed high computation time for length 10 there was a problem. For length 10 there were namely $2^{10} - 1 = 1023$ score-columns and from each the best years had to be selected, the ones with the lowest scores. So a minimum length had to be introduced to omit an exponential runtime i.e. all 1023 columns could not be looked through, so the number of columns was reduced to 56 columns. How were these 56 columns chosen? Only column-combinations for length 8, length 9 and subsamples of length 10 were built. This has led to $\binom{10}{8} + \binom{10}{9} + \binom{10}{10} = 56$ columns. Similarly, this procedure was repeated on length 15 test-samples. So again, only the three longest subsample-lengths were considered and this has led to 121 columns.

Measuring Reliability Next it was checked how suitable the new approach is for reliability measurement. Therefore, Δ Peaks (Def. 2.2.3.4) were computed with the highest and second highest peaks. It could happen that all votes have gone to the same year, especially in the extended variant which considered combinations of scores. If that happened, one had to measure the difference to zero (see Ch. 2.2.3) for the singleton bar i.e. the second case in the Δ Peak-formula had to be applied (Def. 2.2.3.4). But due to visual aesthetics, these singleton bars or counts were considered separately and not included into the plot.

3.7.2 Results

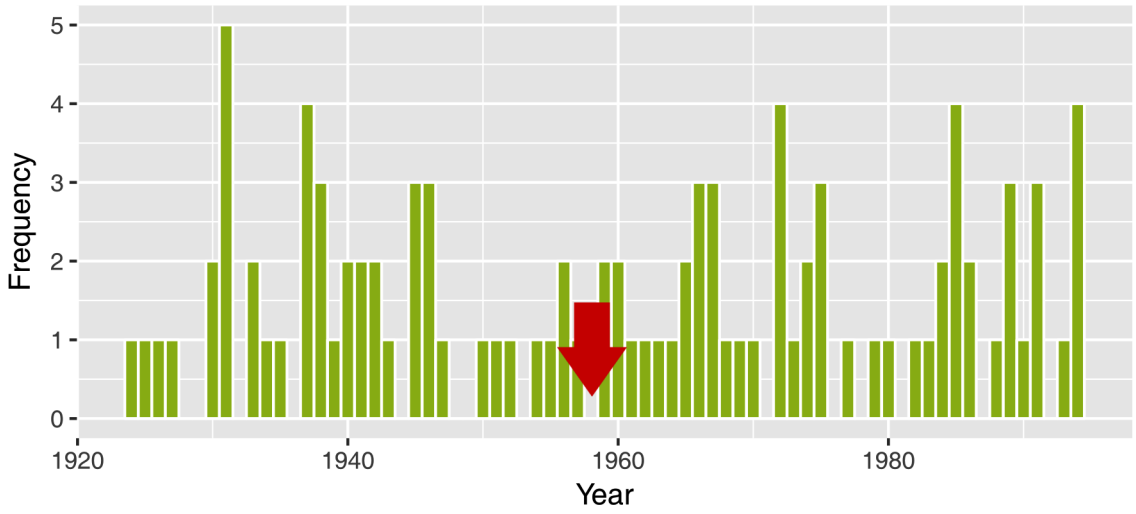

Figure 3.7.2.1 Outcast example for a sample with the length 10. The histogram was computed for the tree sample `721 / 1958-1967` under the usage of a Buckets Chronology (Def. 2.1.2.6). The start-year 1958 (marked with a red arrow) was not in the histogram.

Quality One of the histograms used for rating in the simple, first approach can be seen in Fig. 3.7.2.1. Such a histogram contained exactly 100 votes or counts, since 100 (10×10) values were taken to create it. Per profile of a length-10-sample, the ten years with the lowest produced distance-scores were selected to create that histogram. And as it can be seen in that plot, the test-sample was an outcast since the correct start-year (here 1958) of the sample was not contained in the histogram.

In figure 3.7.2.2 the results for the simple Voting Approach are shown. Here, SAPD (b) outperformed the other methods, since for all three top ranks it showed higher values and also for rank 5. APDN (c) was the second best, and the method APD (a) performed worst. Afterwards, the results from the extended variant were evaluated. First, the length-10-samples (Fig. 3.7.2.3) were tested. This time SAPDN (d) became the best. In the top two ranks were more test-samples than in the other three methods. Concretely, 147 had rank 1 and there were 24 test-samples for rank 2.

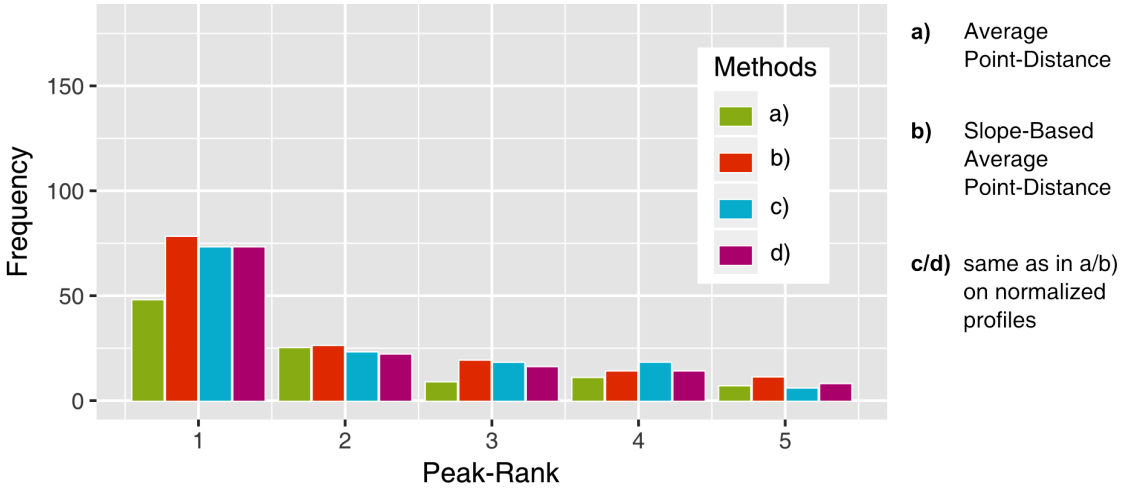

Figure 3.7.2.2 Histograms of sample-ranks with the simple Voting Approach using series of length 10. The formula for the Average Point-Distance (a) can be found in Def. 2.2.1.2. For the Slope-Based Average Point-Distance (b), the formula is described in Def. 2.2.1.3. The transformation into normalized profiles can be done using Def. 2.2.1.1.

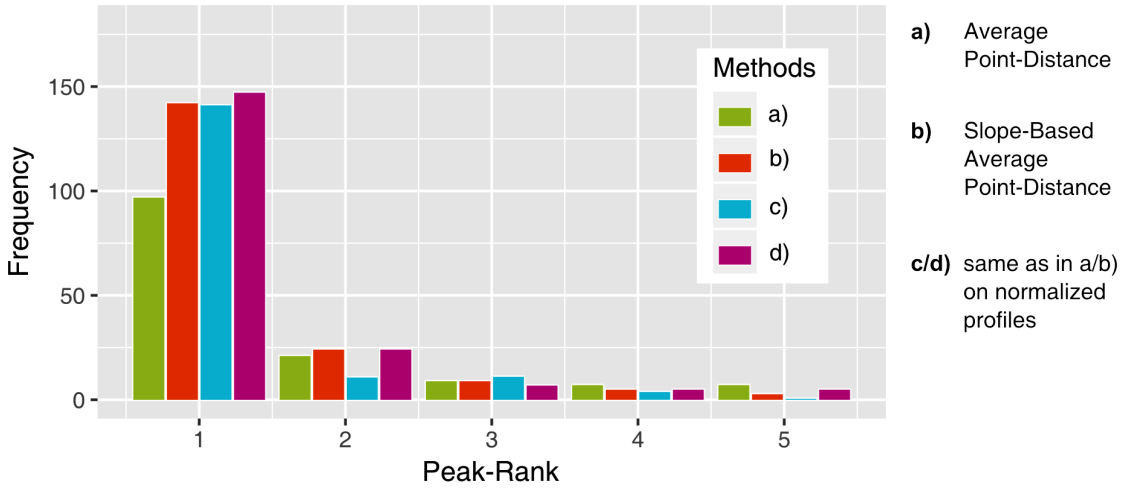

Figure 3.7.2.3 The ranks in the top 5 for length-10 test-samples with the Extended Voting Approach using $l = 1$ selected years per column and minimum subsample-length 8. In total, the ranks for 250 test-samples were determined, exactly 56.4% up to 75.2%, dependent on the used method, were in the top 5.

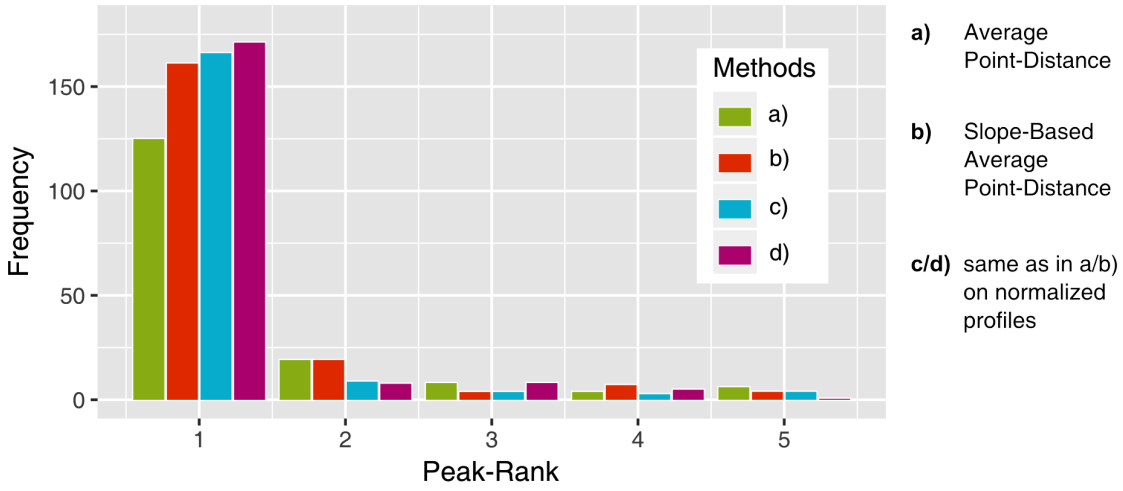

Figure 3.7.2.4 The ranks in the top 5 for length-15-samples under the Extended Voting Approach with $l = 1$ selected years per column and a minimum subsample-length of 13. In total, the ranks for 231 test-samples were determined, about 70.1% up to ~84.4%, dependent on the used method were in the top 5.

The tests were repeated for length-15 test-samples. The results with the best parameter l were plotted (Fig. 3.7.2.4). Here, SAPDN (d) showed overall the highest number of correctly rated samples, 171 had rank 1 ($\approx 74\%$).

The next results were generated for length-5 test-samples. It was first used $l = 1$ top-years i.e. one year per column was selected. SAPD (b) has delivered the best results, about 32.9% of the solutions were correct and about 54.9% within the top 5 ranks. And so for method SAPD (b), different numbers of top- years were tested. Higher numbers have led for length-10 and length-15 to a worsening in the results, such that a plot was only created for length-5-samples (Fig. 3.7.2.5). That plot was made to recognize which parameters deliver the best results. By this an optimum for $l = 2$ selected top-years could be recognized. However, the results were below the ones of the pure Bucket Approach as can be recognized by the green line.

As stated before, also a double weighted variant was implemented. So again, different values for the top-years were tried (Fig. 3.7.2.6). The method SAPD (b) was again the best one for two selected top-years as in the first improved approach without double weighting. But overall the results have shown no improvement with double weighting.

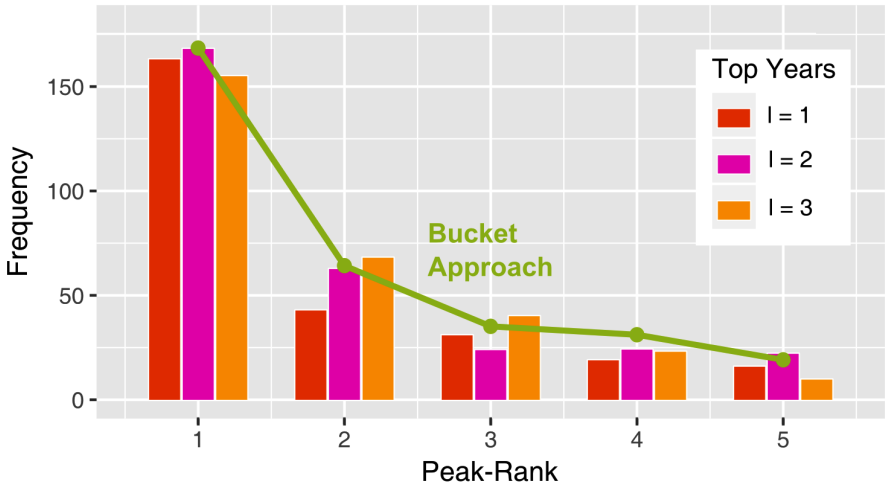

Figure 3.7.2.5 The method Slope-Based Average Point-Distance (b) in the first five ranks for different numbers l of top-years and series-length 5. The green line shows the corresponding ranks with the Bucket Approach (see Ch. 3.5) which was tested before.

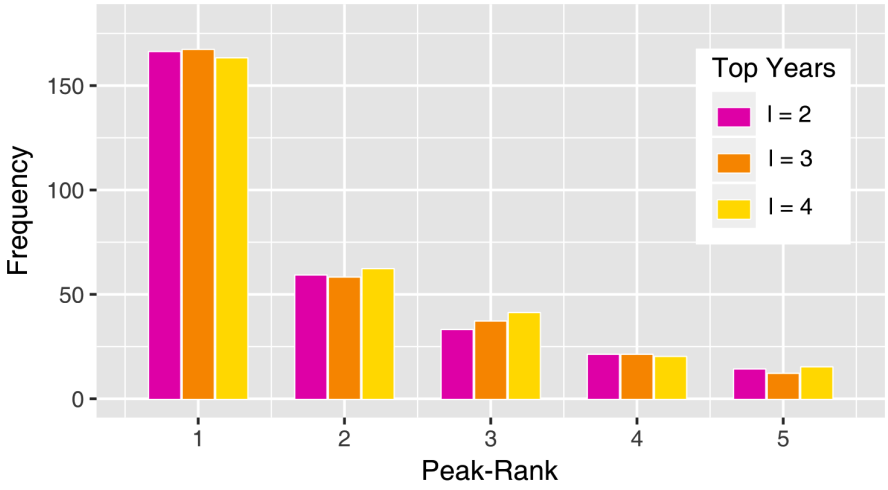

Figure 3.7.2.6 The Slope-Based Average Point-Distance (b) drawn for the first five ranks using different numbers l of top-years, series-length 5 and double weighting.

Reliability Next, reliabilities were measured as in Ch. 3.5.2. As usual in machine learning [28], `TRUE` was encoded with value 1 and `FALSE` with value -1. The question if a sample has rank 1 was mapped on the y-axis, and the weighted average (Def. 2.2.3.4) of the differences between first and second rank bars was mapped on the x-axis. The whole mapping is analogous to the mappings in the Δ Score-plots (Fig. 3.5.2.9). Important to mention is that outcasts get in this plot a Rank 1 Tendency of -1. The Extended Approach with length-10 test-samples and method SAPDN (d) (one top-year) was used. As said, there were also cases in which all sample-lengths have voted for the same year. These samples were not mapped into the plot due to a high Δ Peak-value. The Rank 1 Tendency for these test-samples was around 0.47, 28 from 38 samples had rank 1 (compare with Fig. 3.7.2.7).

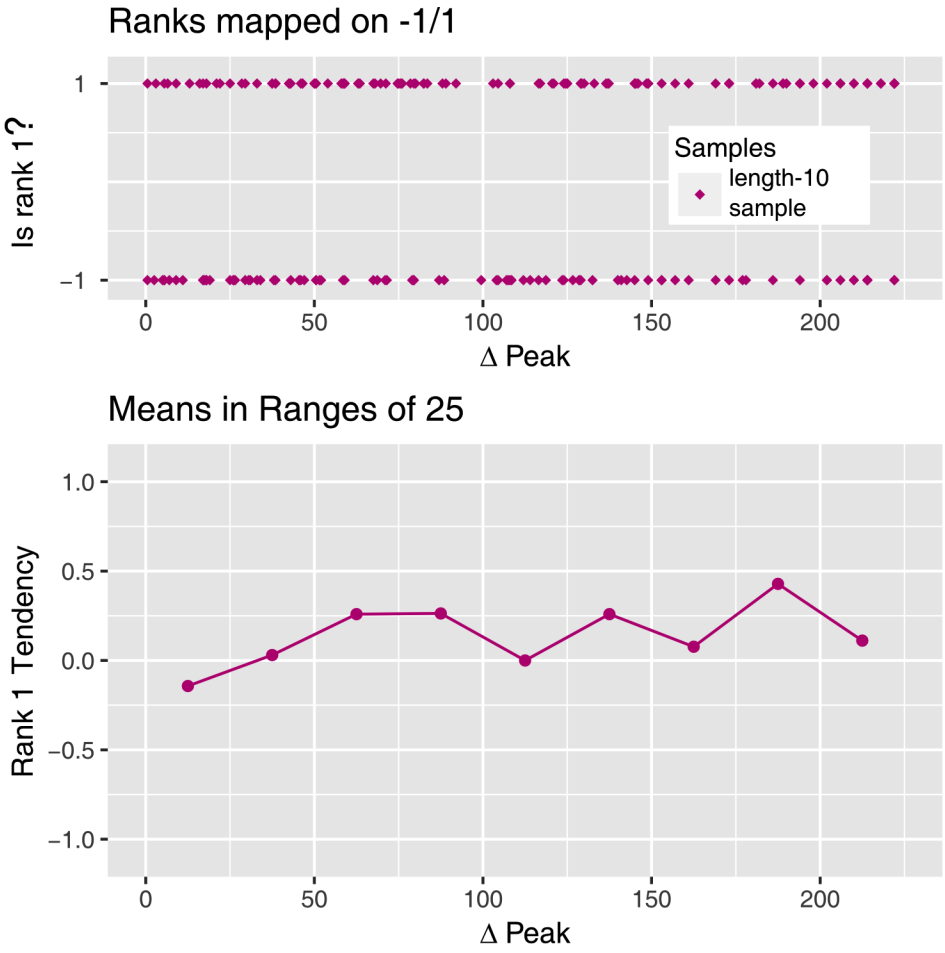

Figure 3.7.2.7 Measuring the trend for rank 1. The value -1 equals `FALSE` and the value 1 equals `TRUE`. The weighted difference i.e. Δ Peak was measured. This difference was plotted on the x-axis. On the upper plot it can be seen how length-10-samples (represented as rectangles or dots) were rated and on the second plot below the trends within ranges of 25 Δ Peak-units can be recognized.

3.7.3 Discussion

First the results between the simple and extended variant (see Fig. 3.7.2.2 and Fig. 3.7.2.3) are discussed. In the extended variant, method SAPDN (d) became the best, same as in the Bucket Approach (see Ch. 3.5.3), whereas in the simple approach, method SAPD (b) has shown best results. Overall the extension has led to a significant improvement. The question was why the first approach performed badly and the extension made it so much better? There are two main-reasons, one is that with consideration of several top-years many wrong years were added into the histogram as noise. And the other reason is that it can suddenly happen that there appear multiple bars in the histogram of same height which are not distinct (compare with Fig. 3.7.2.1). So how can the right one be chosen? This is not generally possible. It was chosen always the last one. That means, the correct start-year was identified and its corresponding count i.e. the bar height, was stored. Then all bars were sorted in descending order and the last position of this named correct height within all sorted bars was used as the rank. The reason, therefore is that it cannot be distinguished between the different bars of same height. This problem with scores did not exist in the pure Bucket Approach since the probability for two equal scores was fairly low i.e. in the previous approaches the year with the lowest scores could be selected because all scores were unique.

The basic idea of the improved version was now to avoid any information loss. And the only real difference to the first simple approach was just that a transformed score-table was used. So for every subset of scores in the original score-table, the column with the sum was built which led e.g. for length-5-samples to 31 score-columns as has been shown before. Now there might arise the question why don't build for example the summed scores of all 31 columns and select there the best year? It is simple, for a set of score-columns $\{c_1, c_2, ..., c_K\}$ of the original score-table, the corresponding summed scores column would have the following form $2^{K-1}(c_1 + c_2 + ... + c_K)$ since every column element c_i appears 2^{K-1} in the transformed score-table with 31 columns. That means, the only thing which would happen is that the individual columns of the original table would be added up and multiplied by some factor. This would again lead to a Bucket Approach similar algorithm, since the best years from one column of scores would be selected, as before.

For length-10-samples the Extended Approach was too slow, such that the approach had to be restricted, only the most promising three lengths for samples or subsets were considered. But the results were behind those of the Bucket Approach. With the new approach, the number of rank 1 rated samples was more or less equal, but the number of top rated samples within the best five was about 5% lower (compare Fig. 3.7.2.3 with Tab. 3.5.2.1). The Bucket Approach was even slightly better for length-15 test-samples i.e. there were more top ranks, for method SAPDN (d), 203 ($\approx 87.9\%$) from 231 ranked samples were in the top 5 and with the new approach only 193 ($\approx 83.5\%$) samples. So there was a decrease by about 4%. The Bucket and thus also the Two-Step Approach have outperformed this new approach.

It was also tried to find a reliability measure, but the trend was not particularly evident in the curve (look Fig. 3.7.2.7) unlike with the Δ Scores approach (Fig. 3.5.2.9). There was only a slight increase with higher Δ Peaks which made it hard to make reliable statements. The curve was first below zero and then it rose up and continued around 0.125, until it was finally around 0.25. This was expectable since the majority of the samples in the plot had rank 1, exactly 119 from 212 shown samples. But there were also 38 not represented samples from which 28 had rank 1. So the difference within the second and the first prediction helped in the case in which it was voted for a single year. In all other cases the difference was not helpful. The computation of p-values as in Ch. 3.5.1 did not make sense, since the number of different values was often too low i.e. the number different counts was not high enough to estimate reliable distributions. Also, the critical value in the applied Kolmogorov-Smirnov Test (Def. 2.2.3.7) was approximated, but the approximation did only work with thirty or more values [19]. That means a correct fitting of a distribution to the given values was not possible.

3.8 Overview of Approaches

In previous chapters new algorithms were developed and extended. Now, the results from these approaches should finally be compared side by side against each other to get a better overview on the strengths and weaknesses of these algorithms.

3.8.1 Length Dependency

The number of correctly rated samples was evaluated, so test-samples which were rated with rank 1 during the sample shifting along the chronologies were counted (Fig. 3.8.1.1). In all methods, it was apparent that with a higher test-sample size, the number of correctly rated samples has increased. And it was recognized that using maximum densities, the Points-Based Approach could even outperform the Two-Step Approach for higher sample-lengths, since its curve is growing faster. Curves from only five and not all seven approaches were plotted, since the not plotted curves intersected with the curves from the Bucket Approach. That means, it was difficult to distinguish the curves from each other. But it was also created a second plot in which the top-five ranks for all approaches were shown (Fig. 3.8.1.2). So it was counted how many test-samples got a rank of five or below. Here now again using maximum densities, the Points-Based Approach attracted the attention, about 95.2% of the samples were rated within the best five ranks for length-15-samples. That was about 5.2% above the results of the Two-Step Approach. So overall it can be said that the Two-Step Approach is the best approach for short sample lengths, among others. However, using maximum densities the Points-Based Approaches provide good alternatives for length 15 and above.

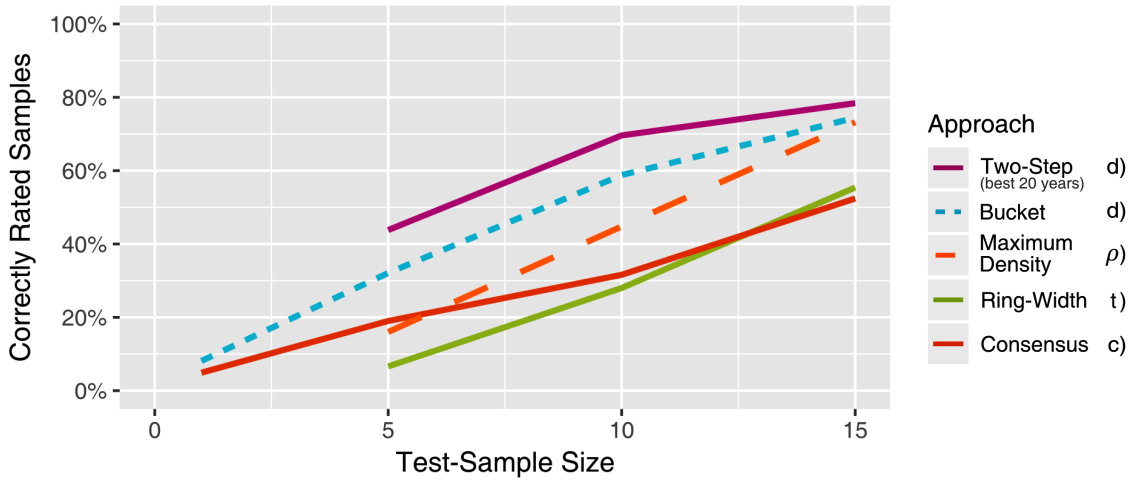

Figure 3.8.1.1 Number of correctly rated samples summarized for different test-sample-lengths. The Slope-Based Average Point-Distance on normalized profiles (d) is defined with the help of Def. 2.2.1.1 and Def. 2.2.1.3. ADPN (c) is defined with Def. 2.2.1.1 and Def. 2.2.1.2. The Pearson Correlation Coefficient (ρ) is defined in Def. 2.2.2.1. The t-value (t) is described in Def. 2.2.2.5.

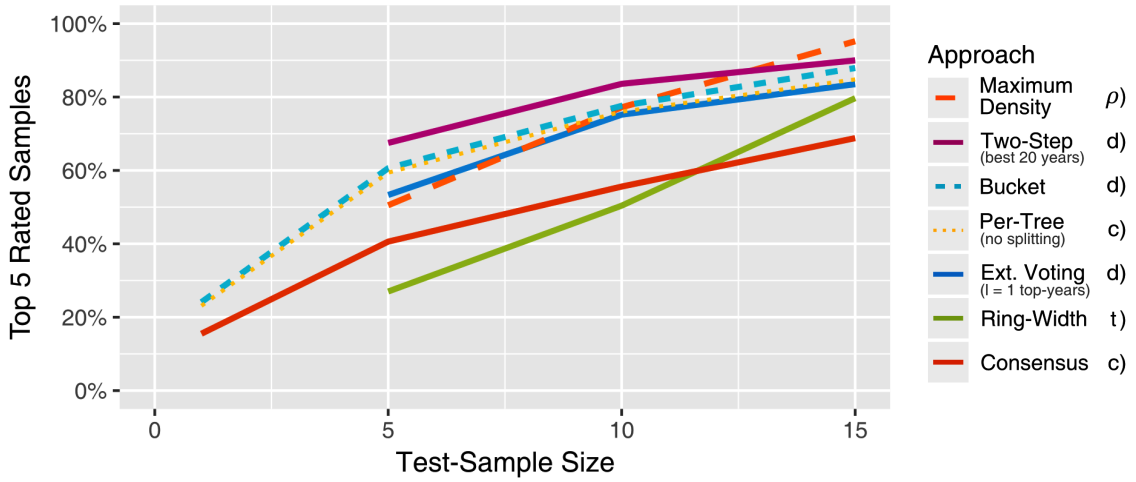

Figure 3.8.1.2 Number of top 5 rated samples summarized for different lengths of test-samples. The Slope-Based Average Point-Distance on normalized profiles (d) was used, as well as the Average Point-Distance on normalized profiles (c). But also, the Pearson Correlation Coefficient (ρ) and t-values (t). The Two-Step Approach was executed using maximum densities in the first step and SAPDN (d) in the second step.

3.8.2 Runtime Comparison

Further, computation times for the different approaches were compared to identify the approach with the best performance i.e. the one which has the most satisfactory runtime with respect to an increase of top rated samples (see Ch. 3.8.1). Therefore, the same 50 length-10 test-samples were used for each of the approaches, namely the samples extracted in Ch. 3.1. For the Points-Based Approaches, the corresponding ring-widths and maximum densities of length 10 were used instead of profiles. Each approach was executed five times and the average runtime of all five test-runs was then displayed in Fig. 3.8.2.1. The runtime for the shifting of samples along the chronology and the runtime for the extraction of predicted dates from score-tables were mapped into the same plot. This splitting was important to make the bottlenecks better recognizable within the different approaches.

It could be shown that the extraction of dates has only played a role in the Extended Voting Approach. For other algorithms this extraction needed far below one second and was not even recognizable in the overall runtime. The Extended Voting Approach had the problem of generating a big score-table which contained a column with the sum for every subset of scores from the original score-table. From each of these columns, then the best year had to be extracted for a histogram. But this is unfortunately a very slow procedure as can be recognized (Fig. 3.8.2.1). Thus, the whole procedure is about 2900 times slower than Points-Based Approaches. However, the first part of the algorithm

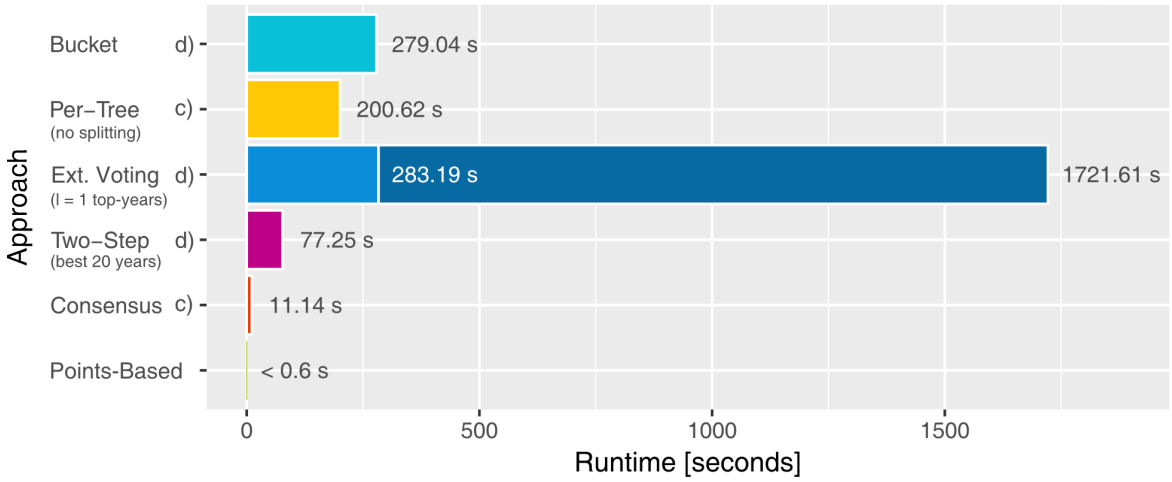

Figure 3.8.2.1 The runtimes of the different approaches for dating 50 length-10-samples. Darker colored is the extraction of the right dates (only visible in the Voting Approach). The Slope-Based Average Point-Distance on normalized profiles (d) was used, as well as the Average Point-Distance on normalized profiles (c). The Two-Step Approach was executed using maximum densities in the first step and SAPDN (d) in the second step.

is pretty similar to the Bucket Approach. Only the function "func" had to be replaced with an order preserving identity-function (Def. 2.2.1.6). That explains, the similar runtimes of both approaches in the first part i.e. the sample-shifting. Overall also comparing with Fig. 3.8.1.2, it can be said that the Two-Step Approach is considering the advantages over using density-series, the best approach, since it is about four times faster than the Bucket Approach and its results up to 10% better (see Fig. 3.8.1.1). Also, it is apparent that presumably the Two-Step Approach won't be necessary for lengths greater than 15. The reason for this are the Points-Based Approaches which should reach similar results for that lengths using maximum densities.

4 Conclusion

4.1 Achievements

With this thesis five new algorithms for cross-dating of density-profiles were introduced and finally summarized in an overview. The overview has shown the runtimes and the efficiency, i.e. the number of top and correctly ranked test-samples of the approaches. Also available with this thesis is a R interface that allows to test each of these approaches (see Appendix Ch. 5.3). The new Bucket Approach showed some advantages and disadvantages compared to the earlier Points-Based Approaches. A first advantage compared to Points-Based Approaches is that with new available per-tree data, the master-chronology has not to be recomputed. But this was no problem at all with the amount of data in the given chronology, so it can only be considered as a small advantage. A more important advantage was the higher hitrate of the already good Bucket Approach, roughly 59% were correct for length-10-samples instead of only 44.8%, due to the usage of intra-annual profile-data. Both approaches combined to the Two-Step Approach even outperformed the Bucket Approach. This approach executes a fast Points-Based Approach in the first step to reduce the runtime. It returns the most promising years for a sample and then the Bucket Approach is executed on these years. Therefore, the corresponding windows of buckets are extracted. So the number of tested years is limited. A positive side effect is a higher precision, namely 69.6% correctly rated test-samples.

After all, the key-question whether density-profiles can be used for cross-dating under short test-sample lengths can be clearly answered with a "yes"! It is possible to use density-profiles for cross-dating of short test-samples and it helps to improve the results. Especially the number of correct solutions could be improved by about 25% for length-10-samples! However, the extraction of profiles is in contrast more complex compared to the simple approach of extracting ring-widths, but as mentioned at the beginning, today automated approaches like the High-Frequency Densitometry [39; 50] exist. So there shouldn't be any problems using this new procedure, the Two-Step Approach, in current research. Additionally, two different reliability measuring methods are available, an established one, using p-values and a new approach with so-called Δ Scores. These two reliability methods were thoroughly tested and showed similar good results. Overall for each new algorithm data was gathered, results were documented and discussed. When possible, comparisons were made against available, established approaches. The task to develop a new, more accurate method which also works precise enough on short samples could be accomplished and the introduced approaches can already be practically applied by using the R interface introduced in the Appendix!

4.2 Future Work

At this point it should be clarified how this thesis could be continued? Therefore, in the next step it is going to be summarized what has been tried so far and what has not been working. Especially assumptions are going to be made, why something has not worked as hoped and how it could now potentially be led to a success.

Boundary Detection in Tree Rings A general problem is the boundary detection of a tree ring. In this thesis, it was assumed that the data was filtered correctly i.e. that the boundaries of tree rings were detected correctly. This is a general problem in such tree ring based approaches. False rings, so mistakenly formed annual rings created due to certain environmental factors have to be manually recognized and filtered out. That means such positions with duplicated years have to be deleted somehow before a density-profile based approach is applied. Also, it can happen that for a position no density is calculated due to wood damage, for example by parasites. So there is still research necessary to fully automatize the extraction of density-profiles which are used for the introduced cross-dating approaches.

Characteristic Years Alternatively to cross-dating with density-profiles or ring-widths, so-called pointer-years can be checked. That are years in which for most trees in a certain climatic region, the ring-widths are significantly increased or decreased e.g. a certain peak within the tree ring widths of multiple trees. As a good example, for a pointer-year in Germany, the year 2003 could be stated. The hot summer in 2003 has led in many trees to a significant change within their ring-widths i.e. the corresponding rings were relatively small. And thus, it was checked, if such years also show distinct density-profiles. But this was not the case, such that there was unfortunately no benefit from such years for the new approach possible. Also, it can be said that the Bucket Approach has not profited at all from any type of characteristic years. Therefore, it was looked for best and worst ranked test-samples on the maximum-intersection subsets of the used distance-methods. The test-samples with the ten lowest and the ten highest distances per method were intersected and it was looked on the correct years of such samples. So, a back-to-back plot was created, on the one side good rated years and on the other side bad rated years were plotted. What could be recognized is that the years with many available profiles were better ranked than the ones with only a few profiles. Thus, it was concluded that the performance or quality of the Bucket Approach is depending only by the number of profiles in each year. Such that most likely, one can only profit from characteristic years if it is gone over ring-widths. Maybe, it is possible to find a distance function which is considering the profiles and ring-widths at the same time. Different approaches were considered but nothing was found so far. The problem is that e.g. the Two-Step Approach can profit from ring-widths and from profiles only separately i.e. both are not considered at the same time in a single value. But if the consideration in a single value would be possible than probably also the Bucket Approach could profit from characteristic years.

Clustering Furthermore, it was checked how buckets are affected by clustering. The Buckets Chronology has two disadvantages compared to an ordinary chronology. First the buckets can become huge and thus the runtime can become very high, since every profile of a bucket is compared with a sample profile. A second problem can be the variance. For example, if there would be hundreds of profiles in each bucket, then the probability for a single sample-profile to find a pretty similar profile in each bucket becomes high due to the high variance between the curves in the bucket. That can

actually destabilize the approach i.e. make the approach useless. To avoid such problems, there was the idea to compute clusterings with the profiles in a bucket. Different clustering algorithms, which were all available in the NbClust package [9] were tested. Also, the Gap Statistic [45] was implemented to select the right number of clusters. But the approach in connection with a clustering algorithm was not stable. Presumably, the problem was primarily the clustering algorithm which was finally used, here k-means. This algorithm is partially randomized and so executing the entire approach twice leads to different results. About 20% up to 40% of the buckets get the same or a similar number of clusters and for the remaining buckets the numbers were completely different, once the profiles were divided into five and the next time into ten clusters. So what was the problem? One problem was to set the parameters for the clustering algorithm correctly. The data was relatively high dimensional with a hundred density-points per profile and accordingly the parameters for the k-means algorithm were set very high. Multiple restarts and many iteration-steps were used to make the algorithm converge to a global minimum. But there was a limit, the parameters could not be set too high, because else the clustering-process took too much time. And overall the whole procedure was not stable. So it can happen that the Bucket Approach becomes unstable with more profiles per year or with longer chronologies as already discussed in 3.5.3. In this case clustering would be necessary. How could this work? A bucket is clustered and from each cluster one profile can be selected as content for a new bucket with a limited number of profiles. This was also done with the described unstable approach and has led to about eight profiles per bucket. The results were pretty similar to the results in the Consensus Approach, but the runtime was very high. Since the results with Two-Step were much better anyway, it was not further time investigated. Presumably a stabilizing effect is already obtained with the Two-Step Approach due to the usage of an established Points-Based Approach in the first step. Unfortunately, this can only be proven on longer chronologies and more different data that was not available for this thesis.

Data The different approaches were tested only with short, gap-free samples. Not all approaches can guarantee to work correctly if too many samples are extracted. As an example, the Per-Tree Approach always needs enough data i.e. profiles in the chronologies or the score for a year cannot be computed. Only about 20% up to 30% of the profiles can be extracted for testing and the rest has to be used for the chronology. Already with about 50 to 60 length-10-samples this percentage-number is reached. This is also the reason why multiple chronologies were computed. The chronologies were very short and it has to be still looked how these new approaches will behave with longer test-sample-sizes, longer chronologies, and test-samples that contain gaps. On the given set it was reasonably no real option to test with longer test-samples, also due to gaps in the data i.e. the years where not always consecutive within the per-tree data. Nevertheless, the goal was reached, a new, more precise procedure for samples of short length has been created. How a procedure could be established which is also working with not gap-free test-samples? In this case, the first and second piece of a sample could be sent multiple times as a single, fragmented test-sample through the chronology using different gap-

sizes. So gap sizes between two up to ten could be tested and the gap-size under which the test-sample is reaching its minimum could be returned together with the predicted date. That means, the approaches are executed as usual, but for the positions of the gap, the score has to become zero. To avoid a significant runtime increase, the first and the second piece could be sent individually through the chronology. The corresponding profile-wise scores could be stored separately to avoid a recomputation for different gap-sizes. If the pieces were sent once through the chronology, and the scores were stored profile-wise, one can just add up the profile-scores to get scores for different gap sizes. From the resulting final scores under different gap-sizes one has then just to select the minima to find the optimal gap-size. This should be the most promising algorithm. Alternatively, if the sample-pieces are very long, the first piece could be sent through the chronology and the minimum position could be stored. The same has then to be repeated with the second piece for all positions right of the first found minimum position to find an optimal position for the second piece. By this an overall position for the incomplete sample is found.

Closing Remarks As shown for very short lengths like length 10 and below there is currently no approach that leads to better results than the Two-Step Approach i.e. it reaches the highest number of correctly ranked test-samples. Further, it is assumed, that the Two-Step Approach may not be the best approach for longer chronologies and longer test-samples. By comparison with the results from the Points-Based Approaches using length-15-samples (Fig. 3.8.1.2), it can be recognized that with maximum densities similar or even better results were reached. So the Points-Based Approaches should outperform the Two-Step on longer test-samples. Also, it could be thought about extending the Per-Tree Approach to a Two-Step Approach. This would make sense because for length-15-samples, the mean was about 40.3% lower and the variance about 56.3% lower, compared to the Bucket Approach (see Ch. 3.5.3). That means there is a chance to improve the results even more. Another question that comes up, is whether it is necessary to create a consensus for each tree. What would happen by using the individual profiles of a tree? All profiles from the same year could be put into the same bucket. Thereby, precomputations wouldn't be necessary anymore. But could this actually lead to an improvement? Presumably yes, because more data for comparisons should also lead to lower scores. Lastly, more types of normalizations and more distance-functions should be considered e.g. the Euclidean Distance could be checked for the distance-computation. Also, the samples used for the creation of a chronology could be filtered. Presumably bad measured data did disrupt the cross-dating. Where does this assumption come from? When checking for characteristic years, one per-tree consensus has attracted the attention. In particular, this was the consensus from the oldest tree. Extracted test-samples from that tree were significantly more often ranked badly than test-samples coming from other trees.

So it had to be developed a new method for cross-dating with density-profiles. There it has been succeeded! The key questions were answered, but also new questions have arisen and there will still be more research necessary to develop a more stable, faster or more precise procedure. That thesis will hopefully only be the beginning for a new kind of cross-dating using intra-annual data!

References

Articles

[1] **Affymetrix I**. «Statistical algorithms description document». In: *Technical paper* (2002), pp. 22–23

[4] **Bender Bela Johannes et al.** «Microstructure alignment of wood density profiles: an approach to equalize radial differences in growth rate». In: *Trees* 26.4 (2012), pp. 1267–1274. DOI: 10.1007/s00468-012-0702-y

[7] **Browner W. S.** «The Reliability of P Values». In: *Science* 301 (2003), pp. 167–168. DOI: 10.1126/science.301.5630.167c

[8] **Buras Allan** and **Wilmking Martin**. «Correcting the calculation of Gleichläufigkeit». In: *Dendrochronologia* 34 (2015), pp. 29–30. DOI: 10.1016/j.dendro.2015.03.003

[9] **Charrad Malika et al.** «NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set». In: *Journal of Statistical Software* 61.6 (2014). DOI: 10.18637/jss.v061.i06

[13] **Damodaran Aswath**. «Probabilistic Approaches: Scenario analysis, decision trees, and simulations». In: *Research Paper* (2007), p. 61

[14] **de Mil Tom et al.** «A field-to-desktop toolchain for X-ray CT densitometry enables tree ring analysis». In: *Annals of Botany* 117 (2016), p. 1188. DOI: 10.1093/aob/mcw063

[15] **Delignette-Muller Marie Laure**, **Dutang Christophe et al.** «fitdistrplus: An R Package for Fitting Distributions». In: (2014), pp. 1, 7

[17] **Facchinetti Silvia**. «A procedure to find exact critical values of Kolmogorov-Smirnov test». In: *Statistica Applicata* 21 (2009), pp. 337–359

[29] **Mann Martin et al.** «MICA: Multiple interval-based curve alignment». In: *SoftwareX* 7 (2018), pp. 53–58. DOI: 10.1016/j.softx.2018.02.003

[30] **Miller G.H.**, **Kaufman D.S.** and **Clarke S.J.** «Amino Acid Dating». In: (2013), pp. 37–48. DOI: 10.1016/b978-0-444-53643-3.00054-6

[31] **Muller Richard A.** «Radioisotope Dating with a Cyclotron». In: *Science* 196.4289 (1977), p. 490. DOI: 10.1126/science.196.4289.489

[33] **Polge Hubert**. «The use of X-ray densitometric methods in dendrochronology». In: *Tree-Ring Bulletin* (1970), p. 1. URL: http://hdl.handle.net/10150/259942

[39] **Schinker Martin G.**, **Hansen Norbert** and **Spiecker Heinrich**. «High-frequency densitometry - A new method for the rapid evaluation of wood density variations». In: *IAWA Journal* 24.3 (2003), pp. 231–239. DOI: 10.1163/22941932-90001592

[44] **Studhalter R. A.** «Tree Growth». In: *The Botanical Review* 21 (1955), p. 3. DOI: 10.1007/bf02872376

[45] **Tibshirani Robert**, **Walther Guenther** and **Hastie Trevor**. «Estimating the number of clusters in a data set via the gap statistic». In: *Journal of the Royal Statistical Society* 63.2 (2001), pp. 411–423. DOI: 10.1111/1467-9868.00293

[46] **Tozzini Valentina** and **Pellegrini Vittorio**. «Prospects for hydrogen storage in graphene». In: (2012), p. 1. DOI: 10.1039/c2cp42538f

[50] **Wassenberg Marc et al.** «Exploring high frequency densitometry calibration functions for different tree species». In: *Dendrochronologia* 32 (2014), pp. 273–281. DOI: 10.1016/j.dendro.2014.07.001

Books

[6] **Brich Stefanie** and **Hasenbalg Claudia**. *Kompakt-Lexikon Wirtschaftsmathematik und Statistik*. Springer Fachmedien Wiesbaden, 2013, pp. 109–110. DOI: 10.1007/978-3-658-03181-7

[10] **Cook E. R.** and **Kairiukstis L. A.**, eds. *Methods of Dendrochronology*. Springer Netherlands, 1990, p. 46. DOI: 10.1007/978-94-015-7879-0

[11] **Council National Research**. *Surface Temperature Reconstructions for the Last 2,000 Years*. National Academies, 2007, p. 47. ISBN: 0-309-10225-1

[12] **da Vinci Leonardo**. *Leonardo on Painting: An Anthology of Writings by Leonardo da Vinci with a Selection of Documents Relating to His Career*. Yale University Press, 2001, p. 178. ISBN: 0-300-09095-1

[18] **Fahrmeir Ludwig et al.** *Statistik*. Springer Berlin Heidelberg, 2016, pp. 137, 387–389. DOI: 10.1007/978-3-662-50372-0

[21] **Hastie Trevor**, **Tibshirani Robert** and **Friedman Jerome**. *The Elements of Statistical Learning*. Springer, 2009, pp. 241–249. DOI: 10.1007/978-0-387-84858-7

[22] **Hedderich Jürgen** and **Sachs Lothar**. *Angewandte Statistik*. Springer Berlin Heidelberg, 2018, pp. 208, 533–534. DOI: 10.1007/978-3-662-56657-2

[23] **Huber Peter**. *Robust Statistics*. John Wiley, 1981, pp. 43, 146, 164. ISBN: 0-471-41805-6

[24] **Inglis Alister David**. *Hong Mai's Record of the Listener And Its Song Dynasty Context (Suny Series in Chinese Philosophy & Culture)*. State University of New York, 2006, p. 4. ISBN: 0-7914-6821-6

[25] **Kaiser Richard**. *C++ mit Visual Studio 2017*. Springer, 2018, p. 320. DOI: 10.1007/978-3-662-49793-7

[27] **Krojer Franz**. *Chronologie der Dendrochronologie*. Differenz-Verlag, 2014, pp. 9–38

[32] **Needham Joseph**, **Daniels Christian** and **Menzies Nicholas K.** *Science and Civilisation in China Volume 6: Biology and Biological Technology, Part 3, Agro-Industries and Forestry*. Cambridge University Press, 1996, pp. 631–632. ISBN: 0-521-41999-9

[37] **Reisch Heiko**. *Kleine Geschichte der Philosophie*. Springer Fachmedien Wiesbaden, 2018, p. 37. DOI: 10.1007/978-3-658-16237-5

[38] **Riede Adolf**. *Mathematik für Biowissenschaftler*. Springer Fachmedien Wiesbaden, 2015, pp. 218–219. DOI: 10.1007/978-3-658-03687-4

[40] **Schmidt Burghart** and **Gruhle Wolfgang**. *Mensch und Umwelt - Baumwachstum und Klima*. Austrian Academy of Sciences Press, 2015, p. 24. ISBN: 978-3-7001-7670-1

[41] **Schweingruber Fritz Hans**. *Der Jahrring: Standort, Methodik, Zeit und Klima in der Dendrochronologie*. Haupt, 1983, pp. 61, 85, 93–94, 218–222. ISBN: 3-258-03120-7

[42] **Speer James H.** *Fundamentals of Tree Ring Research*. University of Arizona Press, 2010, pp. 29–31. ISBN: 978-0-8165-2684-0

[43] **Steyer Ralph**. *Programmierung in Python*. Springer Fachmedien Wiesbaden, 2018, p. 161. DOI: 10.1007/978-3-658-20705-2

[47] **Walz Guido**, ed. *Lexikon der Mathematik: Band 2*. Springer, 2017, pp. 277, 482. DOI: 10.1007/978-3-662-53504-2

[48] **Walz Guido**, ed. *Lexikon der Mathematik: Band 4*. Springer, 2017, p. 162. DOI: 10.1007/978-3-662-53500-4

[49] **Walz Guido**, ed. *Lexikon der Mathematik: Band 5*. Springer, 2017, pp. 53, 88, 290. DOI: 10.1007/978-3-662-53506-6

[51] **Wickham Hadley**. *ggplot2*. Springer International Publishing, 2016. DOI: 10.1007/978-3-319-24277-4

[52] **Wilkinson Leland**. *The Grammar of Graphics (Statistics and Computing)*. Springer, 2005. ISBN: 978-0387-24544-7

Thesis

[3] **Beck Matthias**. «Multiple Interval-based Curve Alignment (MICA)». Master Thesis. Albert Ludwig University of Freiburg, 2014, pp. 2–3. URL: http://www.bioinf.uni-freiburg.de/Lehre/Theses/MA_Matthias_Beck.pdf (visited on 03/06/2018)

Unpublished

[20] **Forest Growth Chair of** and **Dendroecology**. «Statistical Tests». Technical paper. pp. 1-3

Web

[2] *Albertus Magnus*. URL: http://www.albertus-magnus-institut.de/ (visited on 06/19/2018)

[5] *Birthday of Michel de Montaigne*. URL: https://geboren.am/person/michel-de-montaigne (visited on 07/31/2018)

[16] *Dendrochronology Program Library in R*. p. 105. URL: https://cran.r-project.org/web/packages/dplR/dplR.pdf (visited on 05/18/2018)

[19] *Fitting of a Parametric Distributions*. p. 42. URL: https://cran.r-project.org/web/packages/fitdistrplus/fitdistrplus.pdf (visited on 06/14/2018)

[26] *Kendall General*. URL: https://newonlinecourses.science.psu.edu/stat509/node/158/ (visited on 05/16/2018)

[28] *Lecture, Machine Learning in Life Sciences*. URL: http://www.bioinf.uni-freiburg.de/Lehre/Courses/2016_WS/V_ML/ (visited on 03/27/2018)

[34] *R6 classes*. URL: https://cran.r-project.org/web/packages/R6/vignettes/Introduction.html (visited on 05/27/2018)

[35] *Radiokarbonmethode*. URL: https://www.leibniz.uni-kiel.de/de/ams-14c-labor/radiokarbonmethode (visited on 03/06/2018)

[36] *Reference classes in R*. URL: http://adv-r.had.co.nz/R5.html (visited on 05/27/2018)

Tools

Tools used for writing the thesis, research, experiments and development are listed below. For the experiments and programming different libraries like ggplot2 from `http://ggplot2.org/` listed here under Libraries has been used. The grammar and spelling were checked manually, by the usage of Scribens `https://www.scribens.com/` and the one grammar checker listed below.

Bitbucket	`https://bitbucket.org/` version control system
Inkscape 0.48.5	`https://inkscape.org/` creation and editing of vector graphics
JabRef 4.1	`http://www.jabref.org/` references management software
Java 8 Update 152	`https://www.java.com/` object-oriented programming language
LanguageTool 4.1	`https://www.languagetool.org/` offline grammar checker written in Java
MICA 2.02	`https://github.com/BackofenLab/MICA` multiple interval-based curve alignment
Netbeans 8.2	`https://netbeans.org/` integrated development environment
pdf2jpg 6.00 (free)	`http://www.lotapps.com/` converter for pdf-files (especially plots)
R 3.4.3	`https://www.r-project.org/` statistical programming language
R-Studio 1.1.383	`https://www.rstudio.com/` integrated development environment for R
TexMaker 4.4.1	`https://www.rstudio.com/` cross-platform LaTeX editor
Windows 10 16299.192 - 17134.165	`https://www.microsoft.com/` commercial computer operating system

Libraries

R programming language libraries for development are listed below.

ggplot2 2.2.1	`http://ggplot2.org/` grammar of graphics based plotting system
gridExtra 2.3	`http://www.rdocumentation.org/packages/gridExtra` to arrange ggplots in grids
fitdistrplus 1.0.9	`https://github.com/cran/fitdistrplus` to find correct distribution parameters for data
mica-functions 2.02	`https://github.com/BackofenLab/MICA` R-script containing an interface to MICA 2.02
plyr 1.8.4	`https://github.com/hadley/plyr` to split and combine data
reshape2 1.4.3	`https://github.com/hadley/reshape` to reshape R datatypes
rlist 0.4.6.1	`https://github.com/renkun-ken/rlist` additional operations for lists
scales 0.4.1	`https://github.com/hadley/scales` extension for ggplot2 allowing custom axes
stringr 1.2.0	`https://github.com/tidyverse/stringr` additional operations for strings
testthat 2.0.0	`http://testthat.r-lib.org/` to write unit-tests in R

Hardware

The computations were done on the following computer-hardware:

Processor	Intel Core i5-4460 (Haswell) @ 3.2 GHz
Memory (RAM)	8 GB DDR3 @ 1600 MHz

Zusammenfassung

Die Bestimmung des exakten kalendarischen Jahres für eine Baumprobe geschieht mit Hilfe der Jahrringe, indem die Breiten oder Maximaldichten der einzelnen Jahrringe gemessen werden und die entstehenden Kurven gegen eine Masterchronologie mit bekannten Datierungen ausgerichtet werden. Problematisch ist das Ganze, wenn das Holzstück, dessen Alter bestimmt werden soll, sehr kurz ist und somit nur wenige Jahrringe enthält. In dem Fall reicht die Menge an Informationen gewonnen aus einer Ringbreiten- oder Maximaldichten-Serie zur korrekten Datierung in der Regel nicht mehr aus.

Das führte schließlich zur Leitfrage, ob im Falle eines kurzen Holzstücks das exakte Kalenderjahr unter zu Hilfenahme von intra-annualen Daten, der Jahrring-Dichte-Profile bestimmt werden kann. Wenn ja, wie gut funktioniert das Verfahren im Vergleich zu einem klassischen Ansatz? Die These ist, dass Dichte-Profile-basierte Verfahren mindestens genauso gute oder bessere Ergebnisse liefern sollten, da Profile sehr viel mehr Daten enthalten als andere gemessene Charakteristiken.

Innerhalb dieser Studie konnte diese These wurde bestätigt. Mehrere neue Ansätze wurden gefunden, die etablierte, frühere Methoden übertroffen haben. Der Ansatz mit der höchsten Anzahl korrekter Lösungen ist hierbei eine Kombination eines etablierten Ansatzes mit einem neuentwickelten, sogenannten Bucket-Ansatz.

5 Appendix

5.1 MICA Parameters

5.1.1 Per Tree Consensi

Property	Value
distFunc	3
distSample	500
distWarpScaling	0
maxWarpingFactor	2.5
maxRelXShift	0.1
minRelIntervalLength	0.05
minRelMinMaxDist	0.02
minRelSlopeHeight	0.02
reference	0
outslope	FALSE

Table 5.1.1.1 MICA parameters for per-tree consensus computation.

The parameters from Tab. 5.1.1.1 were applied by Dr. Martin Raden for the dataset `PCAB_Ostalb_GD` to compute the per tree consensi. Per tree and year, it was minimally used three density-profiles for a Consensus-Profile and regularly about eight density-profiles. These profiles had about 60 up to 400 points and regularly between 100 to 200 points. The MICA-parameters[1] were chosen appropriately to fit the dataset.

5.1.2 Final Consensus

The given per-tree consensi dataset of `PCAB_Ostalb_GD` consisted out of 56 consensi given as `*.csv`-files (Tab. 3.1.1.1). Maximum number of profiles used for the computation of a Consensus-Profile was set to 24, to avoid problems with the MICA-Alignment procedure. These profiles were selected such that the variance was partially maintained. Due to the used alignment algorithm, it could happen that two points get the same x-coordinate, in which case the MICA-Alignment procedure stopped without a result (MICA 2.02). For the computation of the final consensus, the parameters from Tab. 5.1.2.1 were used.

[1] official GitHub repository `https://github.com/BackofenLab/MICA`

Property	Value
distFunc	3
distSample	1000
distWarpScaling	0
maxWarpingFactor	3
maxRelXShift	0.1
minRelIntervalLength	0.06
minRelMinMaxDist	0.04
minRelSlopeHeight	0.04
reference	0
outslope	FALSE

Table 5.1.2.1 MICA parameters for final consensus computation.

5.2 Additional Results

5.2.1 Points-Based Approaches

Methods Given two sequences of points i.e. curves as discrete ring-widths. For these two sequences it is computed how often they grow in the same way. Whenever the difference between points at a specific position in the first and in the second sequence show the same trend, so an increasement, decreasement or even no change, this is rewarded with a 1. And if one curve is growing, whereas the other does not show any change, this is rewarded with ½. If both curves show contrary trends, the one curve an increasement, the other a decreasement, the reward is even 0. At the end the sum of rewards is divided by the number of points minus one, to get an average value in percent which tells how similar the growth-behaviour of the curves was. This procedure as stated here, is called the (corrected) Gleichläufigkeit[1] [8].

Results The results of the Points-Based Approach using ring-widths and the Gleichläufigkeit as a "correlation coefficient" can be seen in Fig. 5.2.1.1. Only 18 out of 250 test-samples were correctly ranked, that's why the median was that bad. 60 test-samples (= 24%) were ranked in the top 5.

Discussion The results have not been shown before due the fact that for the Gleichläufigkeit, it was not expected to get good results. The reason was how the Gleichläufigkeit is defined. For short lengths like length 10, and length 5, only 9 or 4 difference-trends are considered. That is not enough to make reliable statements, since for many positions the same score is computed due to limited amount of considered trends i.e. the same trends can be a result of chance.

[1] there is small mathematical mistake in Tree-Rings [41], Schweingruber did not reward the case in which both curves show no growth-change

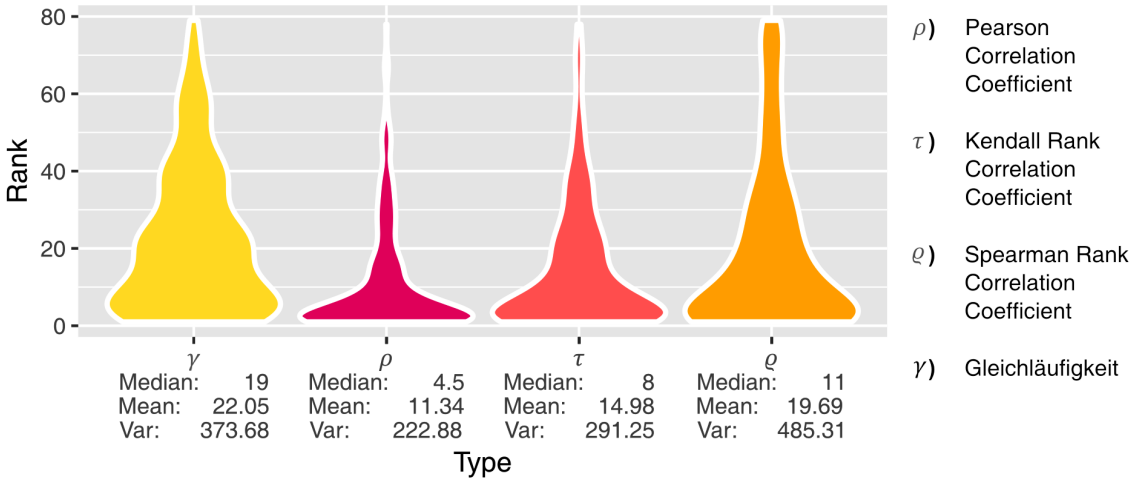

Figure 5.2.1.1 Violin plots of 250 sample-ranks with ring-width series of length 10. The γ stands for the results under the corrected Gleichläufigkeit-value stated before. For Pearson Correlation Coefficient (ρ), the formula can be found in Def. 2.2.2.1. For Kendall (τ) it is described in Def. 2.2.2.3, for Spearman (ϱ) in Def. 2.2.2.2.

5.3 Implementation

5.3.1 Modules

Module	Description
`experiments`	contains scripts executing tasks that were made within different approaches i.e. computation of scores
`maths`	contains mathematical classes
`system`	loading, storing and access on approaches via an `Interface`-class
`tests`	contains unit-tests and describing PDF-files
`visuals`	contains functions to encode and finally visualize the data

Table 5.3.1.1 Modules of the cross-dating project.

The newly found algorithms were first tested and implemented in the programming language R. Since R is a functional programming language with only a simple integrated development environment, complex changes on the architecture e.g. renaming of functions over multiple files is not easy to achieve without breaking the code. Therefore, the architecture was kept simple. It was divided into multiple modules i.e. folders with R-scripts that build together a unit. For example, the `maths`-module stores different R-scripts, so-called classes, containing mathematical operations, whereas the `visuals`-module contains classes which are responsible for the visualization. A summary of the different modules with their classes can be found in Tab. 5.3.1.1. Beside these modules there are also two folders `input` and `output`. All data is loaded from the `input`, so the datasets or pre-computed scores can be found there. Rendered or computed data, everything that is generated, is stored in the `output`.

5.3.2 Classes

Within the `experiments`-module are eight classes. The class `DataAnalysis` allows checking the data, creating histograms and other visualizations of the dataset. It was necessary for the chapter about Data Acquisition. Also, contained is a class `Presentation` which was used to generate the visualizations from the master-colloquium. All remaining classes within this module are so-called experiments whereas the class `Experiment0` contained small programming tests made with an artificial dataset. The other experiments correspond to the contents within the different approaches in the Results chapter. The module `maths` has four classes, the `Math`-class for simple mathematical operations like computation of normalized profiles (Def. 2.2.1.1) or computation of derivatives. It also contains a class called `CurvesMiner`. This class allows to work with the profile i.e. to add curves together, to filter curve data or to equalize profiles to the same length. Profiles can also be combined to clusters, and this is done in the `Clustering` class. The class `Analyzer`, however, allows evaluating data i.e. to compute statistical information like scores, ranks and many more. But such statistical data cannot be directly used for visualization. Therefore, the module `visuals` is important, it contains two classes, the `Encoder` and the `Plotter`. The `Encoder` can use the information from the `Analyzer` and encode data in a way it can easily be plotted by the `Plotter` class i.e. by the ggplot2 library (see Tools). That `Plotter` class creates a plot by using the grammar of graphics [51; 52] which is implemented within that named library. That grammar allows creating plots layer-wise and intuitively what results in beautiful plots.

Another module that has not been described yet is the module `system`. It contains five classes, the `Defaults` class storing all constants like paths or strings except functional strings e.g. grammar of graphics related strings like `dotted` or `histogram` are not stored. The reason for not storing functional strings in this class was a slowdown in the development. So it makes sense to store paths in one place since paths can change during the development but functional strings won't definitely be changed during development. Loading and storing is done with the `Loader` and `Storer`. These classes have a wealth

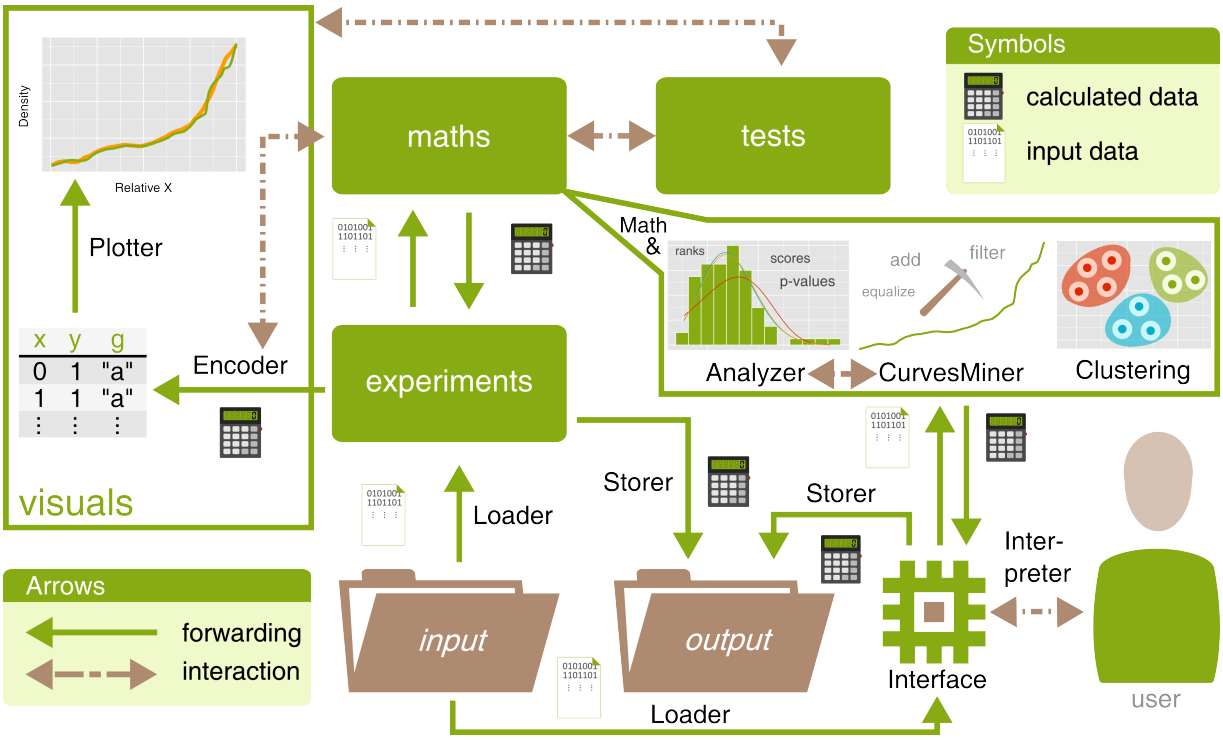

Figure 5.3.2.1 Schematic representation of interactions between modules, classes and user. Modules are green boxes with a label in lowercase letters and classes start with an uppercase letter and usually represented along an arrow. In the left lower corner is the legend for arrows and on the upper top corner for symbols.

of functions to load and save numerous formats and special encoded data like score-files. Due to very-time consuming computations, all data like scores were stored on the hard-drive and reloaded whenever some kind of visualization was necessary. Also, the `system`-module contains an `Interface`-class for communication with the user, together with an `Interpreter`-class that processes user entered strings from the `Interface`. The `Interface`-class allows trying out all approaches on therefore prepared files. It was encoded some files in a new format which is described detailed in the following. As last not described module, there is the `tests`-module which simply stores some unit-tests that can be executed by running the `run_tests` script. In this module-folder, there are also some PDF-files contained describing in a formal, mathematical way the extensive precalculations done for these unit-tests. It was tested the loading and storing of scores, implementations of different formulas in R and the correctness of visualizations. Also, some examples for the applications of formulas from this thesis are found there. The full interaction between the modules, the `Interface` and the user were summarized in Fig. 5.3.2.1. That should help programmers who want extend or work with that project in the future.

Interface Class The `Interface`-class allows executing all approaches. It receives the data in a special format which can be recognized in Tab. 5.3.2.1. That is the format for the per-tree consensi. Samples which should be dated, so for which a start-year has to be predicted, have a similar format. Only the column named `year` is replaced with a column called `part`, since the years for the sample-parts are unknown. The `part` numbers e.g. `1,2,3, ...` tell which density-value belongs to which profile like in Tab. 5.3.2.1 with the years. Each year has multiple density-values which together build a profile. Files with that encoding were precomputed, and they can be found within the project `input`-folder. The output from the `Interface` is a matrix which can also be automatically stored as a `*.csv`-file. There is an option for each `Interface`-function allowing to store the computed data in the `output` folder. That matrix has a similar format to the one described in Tab. 5.3.2.2.

year	**density**	**characteristic**
1992	2.016	166
1992	2.433	166
1992	2.881	166
1993	2.043	128
1993	2.383	128
⋮	⋮	⋮

Table 5.3.2.1 Schematic structure of a file. A column for the year, a column for the density value and a column for the corresponding characteristic i.e. a ring-width or a maximum-density value. A sequence of density-values for the same year corresponds to a profile. For an individual year the characteristic value is constant.

sample	**pValue**	**rank1**	**score1**	**rank2**	···
1041_MICA-cons	0.06209708	1960	41.6288	1947	···
1051_MICA-cons	0.01127934	1987	14.9864	1940	···
1201_MICA-cons_1	0.06737666	1957	24.6861	1955	···
1201_MICA-cons_2	0.066093	1941	24.6395	1962	···
⋮	⋮	⋮	⋮	⋮	···

Table 5.3.2.2 Structure from the output matrix of the Bucket Approach. In the first column the name of the loaded sample and beside the reliability for the solution in form of a p-value. Then the rank 1 prediction with its corresponding score to the right. Ranks are presented along with the achieved scores side by side. The number of predicted ranks as well as the used reliability measure can be adjusted by the user.

The advantage of the `*.csv`-format is that it can be read in within any spreadsheet program as a table. Such that the data can be easily further analyzed. In the table description of Tab. 5.3.2.2, it was stated that it is the table structure one gets by the Bucket Approach. So due to mathematical reasons p-values are not available in all approaches. Δ Scores were not output, but therefore the corresponding predicted scores of the ranks. That means, Δ Scores have to be calculated manually. By this decision, it can be compared with several scores and probably this can also help to increase the confidence for a solution. A complete documentation of the `Interface` and its possibilities is described in detail on the official Bioinformatics GitHub repository of Prof. Dr. Rolf Backofen[1] and on the attached CD in the back cover.

Conventions It was tested different in R language available object-oriented systems like R6 [34] and Reference classes formerly known as R5 [36] in the beginning. But it made either debugging more difficult or the code did not work anymore by changing to a newer version of R. Because of this something simpler was done which is known from C++. In C++, the class name is simply written in front of a function e.g. `void Plotter::createHistogram` [25]. This idea was reused in this project, what made every function name unique and avoided unforeseen problems.

To make recognize a later developer about used R libraries, all libraries were imported with the `library`-function in the `Main`-class which is used for executing the project. That was not necessary since within the project all non-internal functions except the ggplot2 functions were accessed with `::` operator. But it has the advantage that developers can immediately recognize what has to be installed before executing the project.

Also, the debugging was simplified by using import-booleans. In every class e.g. `Exercise1` a constant named like the class was initialized with `TRUE` i.e. `EXERCISE_1_IMPORTED <- TRUE;`. This boolean is now set, when the class is sourced for example after setting a breakpoint in R-Studio (see Tools). If now the `Main`-class is executed, then the class `Exercise1` won't be resourced due to a check-up `if(!exists("EXERCISE_1_IMPORTED")) source("Exercise1.R");` in `Main`. So the breakpoint is maintained i.e. not removed since the class `Exercise1.R` is not resourced by `Main`.

Access modifiers were simulated i.e. a technique, and a convention were used which are known from the Python programming language [43]. Two underscores in front of a function were used for a private function i.e. `Exercise1.__getSubpatterns` and one underscore for protected functions e.g. `Exercise1._createAlignment`, whereas public functions do not have underscores.

For assignments generally the `<-` operator was used instead of an equal-sign `=`. With these named conventions a simple extension of the project should be guaranteed.

[1] see `https://github.com/BackofenLab/Cross-Dating`

5.3.3 Processes

Sample Processing How test-samples were processed? At the beginning they were generated as it was stated in Ch. 3.1. Then these test-samples were put in the respective folders under `input` of the approaches i.e. folders like the `passes`-folder were created. That was the folder which was used for the Consensus Approach (see Ch. 3.3) under length-10-samples. In this folder five subfolders were contained `pass_1`, `...`, `pass_5` for the different passes that had to be done as simulation for the mentioned 5-fold cross-validation (see Ch. 3.1.3). In such `pass`-subfolders now folders were contained that stored the scores of the given type with which they have ended i.e. `a`, `b`, `c`, `d`. Also, there was contained the folder with the test-samples as well as a Consensus Chronology file. That was the general storing scheme for scores and test-samples. Also, `passes`-folder for other lengths and other approaches have existed. It was used always the same storing-scheme, only the scoretype, pass and the `passes`-folder had to be passed to a samples-processing function. Such functions which were stored in `Experiment`-classes has then started the given approach under the given method, here e.g. methods `a` to `d`. So for example, normalization of profiles or computation of slopes was activated.

Shifting of Samples Given the results from the previous chapters, a general procedure for the computation of scores is now described (Alg. 5.3.3.1). The input (line 1) is a sample S and a chronology C of arbitrary type. For the length of the series or sample, the character K is used, whereas the length of a chronology is described with a N. The algorithm iterates over possible positions (of the sample S) within the chronology C (line 6 - line 11) and for each iteration it cuts out a series S_i^{i+K-1} of profiles (line 7) or buckets between position i and $i+K-1$. For each subsequence S_i^{i+K-1} a distance-score ς to the sample S is computed. It is still assumed since chapter 2.2.1 that the subsequence S_i^{i+K-1} from C and the sample S have the same number of points in their profiles when a distance is calculated (line 8). In reality, an interpolation to the same length would possibly be necessary. The distance ς is stored (line 9), but the position at which the score was produced has not to be stored since the positions or years can be read out in the chronology again. It depends on the arguments which function for the computation of $\varsigma := \mathrm{dist}_{avg}^{\gamma}(\mathrm{S}, \mathrm{S}_i^{i+K-1})$ is used. If S_i^{i+K-1} is a sequence of buckets, the generalized sample to bucket distance is used (Def. 2.2.1.6) and if it is a sequence of profiles, the definition for the Sample Distance is used for the computation (Def. 2.2.1.5). Then the current position is incremented by one (line 10). As soon as $N-K-1$ iteration have been done i.e. the full chronology has been gone through with the sample, all scores are finally returned (line 12). Problematically in this algorithm is the runtime, there are $N-K+1$ comparisons between chronology-windows and S. And thus $(N-K+1)\cdot K \approx N \cdot K$ many profile- or bucket-comparisons. In each comparison $2k$ profile-points are gone through to compute a distance. Under the assumption that every bucket $\mathbf{B}$ has the same amount of profiles, this leads to an overall runtime of $\mathrm{O}(N \cdot K \cdot |\mathbf{B}| \cdot k)$ for the Bucket Approach.

Algorithm 5.3.3.1 Shifting of sample within the chronology.

procedure computeSampleScores(S, C)
    $K := |\mathrm{S}|$
    $N := |\mathrm{C}|$
    $\mathbf{S}_{\mathrm{S}}^{\mathrm{C}} := \emptyset$
    $i := 1$
    while $i \leq N - K + 1$ **do**
        $\mathrm{S}_i^{i+K-1} \sqsubseteq \mathrm{C}$
        $\varsigma := \mathrm{dist}_{avg}^{\gamma}(\mathrm{S}, \mathrm{S}_i^{i+K-1})$
        $\mathbf{S}_{\mathrm{S}}^{\mathrm{C}} := \mathbf{S}_{\mathrm{S}}^{\mathrm{C}} \cup \{\varsigma\}$
        $i := i + 1$
    end while
    return $\mathbf{S}_{\mathrm{S}}^{\mathrm{C}}$
end procedure