Statistical modelling of effect sizes and rankings from multiple biomedical studies

Vendula **Švendová**, Bc. Mgr.

Institute for Medical Informatics, Statistics and Documentation

Medical University of Graz

Dissertation submitted for the Degree of Doctor of Medical Science
(Dr. scient. med.)
at the Medical University of Graz

under the supervision of

Prof. Michael G. Schimek, DPhil, PhD
Prof. Peter G. Hall, PhD

2017

Dissertation committee

Prof. Michael G. Schimek, DPhil, PhD
Institute for Medical Informatics, Statistics and Documentation
Medical University of Graz

Prof. Peter G. Hall, PhD (1951–2016)
School of Mathematics and Statistics
University of Melbourne

Prof. Andrea Berghold, PhD
Institute for Medical Informatics, Statistics and Documentation
Medical University of Graz

Věnováno mamince a tatínkovi

Declaration

I hereby declare that this thesis is my own original work and that I have fully acknowledged by name all of those individuals and organisations that have contributed to the research for this thesis. Due acknowledgement has been made in the text to all other material used. Throughout this thesis and in all related publications I followed the "Standards of Good Scientific Practice and Ombuds Committee at the Medical University of Graz".

Vendula Švendová,
June 30, 2017

Statement

Part of this thesis is based on the article:

Švendová, V. and Schimek, M.G. (2017). A novel method for estimating the common signals for consensus across multiple ranked lists. *Computational Statistics and Data Analysis*, Volume 115, November 2017, pp. 122–135. `https://doi.org/10.1016/j.csda.2017.05.010`.

The journal allows re-use of the content in a dissertation.

Acknowledgements

First of all, I would like to thank my main supervisor, Professor Michael G. Schimek, who gave me the chance to do my PhD and lead me through it. Thank you for your guidance, continuous support and motivation whenever I needed it. Great thanks also belong to Professor Peter Hall, a brilliant statistician, who, to the highest regret of all of us, passed away in January 2016. I only had the chance to personally meet him once, but his knowledge, friendliness and modesty all combined immediately amazed me. Thank you for your ideas and inspiration. Many thanks to you, Professor Andrea Berghold, for all the support you gave me during my studies, as well as chances to move forward and work in new research areas of genomics. Thank you, Sereina Herzog, for all the hours spent discussing my work, for reading my manuscripts, for improving them, and for being a dear friend. Thank you, Marcus Bloice, for seemingly endless language corrections, for your encouraging words when things did not go well, and most of all for being there for me.

Thanks to all the others who had a positive influence on my work, in particular Professor Arnoldo Frigessi and his group at the University of Oslo, and Professor Martin Vingron and his group at the Max Planck Institute for Molecular Genetics in Berlin.

Thank you, my brothers Jakub and Petr, for encouraging me to always improve as a scientist. Last but not least, thank you, mum and dad, for always supporting me and unconditionally believing in me.

Abstract

Replicating research experiments or studies can often lead to different conclusions, sometimes even contradicting each other. To resolve potential conflicts, researchers often pool related studies to obtain a combined, more reliable result. This pooling of information is called meta-analysis. Assuming reasonable consistency, one can combine experiments in any research area. A large number of statistical methods for meta-analysis have been developed over the years. Most of these methods are tailored for specific tasks, such as combining clinical trials or genomic experiments, and cannot be immediately applied to other problems. A more general group of meta-analytical methods are based on rank order data. These methods work with ranks instead of the measured values themselves, the latter of which are not always available, and therefore are not limited by the data type and not disturbed by different data transformations, presence of outliers, or requirements regarding their statistical distribution. Nevertheless, the generality of rank-based methods comes at a price: the relative differences between the measured values are lost and as a consequence they could not, until now, estimate the common study signals that have produced the observed ranks.

This thesis proposes a new approach that combines the advantages of rank-based methods, while achieving the ultimate goal of meta-analysis: estimating those signals that are causal for the ranks. Moreover, the standard errors of the signal estimates are estimated by a non-parametric bootstrap, and the stability of the observed rank positions is assessed. The proposed approach is tested on simulated data under various scenarios, as well as applied to real data combining studies and experiments from clinical and genomic research.

The simulations showed that the proposed approach can estimate the underlying signals accurately, as well as estimate the derived rank positions. As expected, better estimates were achieved when the agreement between the studies or experiments was high. The size of the standard errors reflected the uncertainty of the estimated signals, and the amount of overlap of the standard error ranges was indicative of the rank instability. When applying the method to the real-world applications mentioned above, promising results were obtained. Finally it was demonstrated that the proposed method is a useful meta-analytic tool in biomedical research. The main drawback of the method is its computational demands, which, however, could be relaxed by further optimisation of the algorithm.

The thesis is accompanied by supplementary figures and tables from the simulations, as well as the R source code for the algorithm.

In conclusion, the submitted thesis presents a new, general tool for meta-analysis of ranked lists, which can estimate the underlying signals that inform the observed rankings, the standard errors of the signals, and the involved rank stabilities.

Zusammenfassung

Forschungsexperimente oder Studien zur gleichen Fragestellung führen häufig zu unterschiedlichen Schlussfolgerungen, die einander sogar widersprechen können. Um solche mögliche Konflikte lösen zu können, bündeln Forscherinnen und Forscher oft gleichartige Studien mit dem Ziel verlässlichere Ergebnisse zu erhalten. Diese Art der Informationsbündelung wird Metaanalyse genannt. Unter der Annahme ausreichender Übereinstimmung können so Experimente innerhalb jedes beliebigen Forschungsgebietes kombiniert werden. Zahlreiche statistische Methoden sind über die Jahre hinweg bereits für die Metaanalyse entwickelt worden. Die meisten dieser Methoden sind für spezifische Aufgaben maßgeschneidert, wie zum Beispiel die Kombinierung von klinischen Studien oder von genomischen Experimenten. Daher können diese nicht unmittelbar auf andere Probleme angewandt werden. Eine allgemeinere Gruppe von meta-analytischen Verfahren bilden die rangbasierten Methoden. Diese arbeiten mit Rängen anstelle der eigentlichen Messwerte, welche selbst häufig gar nicht verfügbar sind. Die rangbasierte Methoden sind im Gegensatz zu den eigentlichen Messwerten nicht auf spezifische Datentypen beschränkt, nicht von unterschiedlichen Datentransformationen, dem Vorliegen von Ausreißern oder von spezifischen statistischen Verteilungsannahmen betroffen. Jedoch hat diese Universalität der rangbasierten Methoden ihren Preis: da die relativen Differenzen zwischen den Messwerten verloren gehen, war es bislang nicht möglich, die zugrundeliegenden Signale in den Studien schätzen zu können, welche die beobachteten Ränge erzeugt haben.

In dieser Dissertation wird ein neuer Ansatz vorgestellt, welcher die Vorteile von rang-basierten Methoden mit dem überaus wichtigen Ziel der Metaanalyse verknüpft: die Schätzung jener Signale, welche für die Ränge kausal sind. Darüber hinaus werden die dazugehörenden Standardfehler mit einem nicht-parametrischen Bootstrap Verfahren geschätzt und die sich daraus ergebende Stabilität der beobachteten Rangpositionen evaluiert. Der vorgestellte Ansatz wurde mit simulierten Daten unter verschieden Szenarien getestet und auch auf reale Daten angewandt, in welchen Studien beziehungsweise Experimente in der klinischen und der genomischen Forschung kombiniert wurden.

Die Simulationen haben gezeigt, dass der vorgestellte Ansatz die zugrundeliegenden Signale und die von ihnen abgeleiteten Rangpositionen akkurat schätzen kann. Wie erwartet, wurden bessere Schätzungen erzielt, wenn die Übereinstimmung zwis-

chen den Studien oder den Experimenten hoch war. Die Größe der Standardfehler gab die Unsicherheit der geschätzten Signale wieder. Der Grad der Überlappung der Standardfehlerbereiche zeigte die Ranginstabilität an. Bei den realen Daten konnten vielversprechende Resultate erzielt werden. Schlussendlich konnte gezeigt werden, dass die vorgeschlagene Methode ein brauchbares metaanalytisches Werkzeug für die biomedizinische Forschung ist. Der größte Nachteil der Methode ist der mit ihr verbundene Rechenzeit. Diese könnte durch Optimierungen und Anpassungen des Algorithmus reduziert werden.

Im Anhang der Dissertation sind zusätzliche Grafiken und Tabellen zu den Simulationen sowie der R-Sourcecode für den Algorithmus zu finden.

Zusammenfassend präsentiert die vorliegende Dissertation eine neue, generelle, funktionierende Methode für die Metaanalyse von beobachteten Ranglisten und kann die ihnen zugrundeliegenden Signale, deren Standardfehler, sowie die damit einhergehenden Rangstabilitäten schätzen.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In modern science, enormous amounts of data are being measured, analysed, and stored each day. Irregardless of the scientific field, the purpose of these efforts remains the same: to answer a research question. Every field differs in the ways in which data are gathered and produced, and also the methods for analysing the data often vary across disciplines.

In order to verify, broaden, and strengthen previously obtained results, scientific studies can be replicated, when the analysis is re-done on the same data, or repeated, when new data are used to answer the same scientific question. Having access to multiple datasets that try to answer the same research question leads logically to another type of analysis, a type of analysis more powerful than any of the individual analyses, known as *meta-analysis*. Meta-analysis combines the results of conceptually similar studies (for an example see Table 1.1). The ultimate aim is to get a pooled estimate of a common truth underlying all of the studies, a result which is hopefully devoid of biases and errors that are almost certainly present in the individual studies. Recently, other terms for meta-analysis have appeared, such as *data fusion* or *data integration*, but the term *meta-analysis* will be used throughout this thesis.

Drug	Study 1	Study 2	Study 3
A	0.5	0.8	1.2
B	0.1	0.6	1.3
C	0.2	0.5	0.7

Table 1.1: An example eligible for meta-analysis. Effect sizes from three studies assessing three drugs.

1.2 Background

The word *meta* means 'beyond' in Greek and the term *meta-analysis* was chosen by Glass [1976] to refer to 'the analysis of analyses', in other words to an analysis abstracted beyond the individual analyses. Despite coining the term *meta-analysis*, Glass was not the first one to realise the importance of combining studies. An entire century before getting its name, the advantages of meta-analysis were discussed and a method for the 'advantageous combination of measures' was already suggested (Airy [1861]). The first meta-analysis performed on clinical studies was carried out at the beginning of the 20^{th} century (Simpson and Pearson [1904]). Since then, methods for meta-analysis, as well as results from combined studies, began to arise more frequently (see, e.g, O'Rourke [2007] for a historical overview). Such methods have also been used in other fields (for example in agriculture, education, or social sciences), yet medical research remains the field in which the use of meta-analysis is the most frequent. This is no surprise, as diseases have extremely complicated traits and are influenced by a large amount of factors. Hence even when one study is repeated with the same randomisation criteria, the results can be different or even contradicting. Such conflicts between studies are ideally suited to methods that combine the results in order to remove part of the variability. Especially influential in this research was Archie Cochrane (1908–1988) who called for improving the evidence gathered from randomised controlled trials using meta-analysis (Cochrane [1972]). His work eventually led to the foundation of the Cochrane Collaboration (`http://www.cochrane.org/`) in 1993, a network of researchers focusing on gathering evidence from clinical trials, as well as improving the methodology for combining clinical evidence.

As soon as methods for meta-analysis started appearing, criticism emerged. The main critique could be summarised by saying that a meta-analysis is only as good as the studies involved. For example, inconsistent and heterogeneous studies can cause misleading pooled results. Low quality studies produce low quality pooled

estimates. Various biases, that might not be directly apparent, can also influence the credibility of the results. For instance, studies with highly significant results are more likely to be published, and hence used in meta-analyses, than studies with inconclusive results. This might cause an overestimation of the true effect, which is often called *publication bias*.

Despite the criticism, meta-analysis is still a widely used and respected method, and with the emergence of large biological data stored in online databases, this is becoming ever more so. Inclusion criteria and careful examination of the studies is a necessity in order to arrive at reasonable conclusions, however. In fact, one can use knowledge about the studies' differences and investigate how they contribute to the pooled effect.

1.3 Generalising meta-analysis

The general idea behind combining studies is the assumption that there exists an underlying true signal (or effect) of each studied object (for example a treatment, drug, or gene). Existing methods for estimating such signals are limited to combining data of the same type, for instance combining effect sizes from several studies, combining drug response rates from several clinical trials, or combining gene expression values from several experiments. Another problem is that the methods are often tied by strong distributional assumptions.

An elegant way of generalising meta-analysis in order to combine studies with different data types, is to use the rankings of the values instead of the values themselves, i.e. ordering the objects and assigning them a rank position (Table 1.2). This way we can easily combine, for example, effect sizes with response rates. Rankings are robust to outliers and invariant to transformation and normalisation, as long as the relative orderings are preserved. Rank-based methods for meta-analysis (so-called *rank aggregation* methods) have a long history and provide valuable insight into the rank structure of the data. Nevertheless, they also possess a crucial limitation: by ranking the values, the accuracy of the measurements is lost and one can only recover the underlying ranking, rather than the underlying signals themselves.

Our aim, therefore, is to combine the advantages of both approaches: we wish to use the generality of ranking data, but also recover the underlying true signals. In order to make the method applicable on any numeric scale, we aim for the normalised

Drug	Study 1	Study 2	Study 3
A	1	1	2
B	3	2	1
C	2	3	3

Table 1.2: An example of replacing measures from Table 1.1 by their ranks.

values of the underlying signal. We will achieve this by defining a model that determines the ranking procedure, and by quantifying the variability of the rank positions across the studies and objects.

1.4 R and Bioconductor

Because R and Bioconductor are repeatedly mentioned in this thesis, let us briefly introduce them here. R, or The Comprehensive R Archive Network (CRAN: `https://cran.r-project.org/`), is a programming language, largely used in the community of statisticians and bioinformaticians. R contains a broad range of open source packages, which implement the vast majority of existing statistical techniques. Many of the packages focus on high-throughput biological data, and these packages are developed under the umbrella of the Bioconductor project (`www.bioconductor.org`). All calculations in this thesis were produced using R and its packages.

1.5 Aims and structure of the thesis

This thesis is devoted to (i) understanding the concept of meta-analysis, (ii) describing its usage in medical and genomic research, (iii) building a novel model for meta-analysis, applicable to studies or experiments of any numerical data type, and (iv) testing the model on simulated and real-world datasets.

The following two chapters serve as a background for understanding our model and its relevance to other methods: Chapter 2 introduces the methods for meta-analysis in biomedical research, and Chapter 3 is dedicated to ranking data and their modelling. The main contribution of this work, a novel method for estimating the normalised underlying true signals from ranked data, is defined, explained, and tested on simulated data in Chapter 4. The graphical results of the simulations are provided in the appendix. The method is used as a meta-analytical tool on real-world data in Chapter 5. Chapter 6 discusses the results and draws conclusions. The

R code for the functions written for this thesis, together with a running example, are provided in the appendix.

Chapter 2

Meta-analysis

Meta-analysis, also called *pooling*, *quantitative synthesis* or *data fusion*, is "[...]the statistical analysis of a large collection of analysis results from individual studies for the purpose of integrating the findings.[...]", as defined by Glass [1976]. Each individual study provides an estimate of the unknown truth, disrupted by an unknown error. The goal of meta-analysis is to obtain a pooled estimate of the same unknown truth and reduce the errors associated with the individual studies. For example, studies with small samples have very low statistical power (Freiman et al. [1978]), and almost a century ago Simpson and Pearson [1904] had already pointed out that studies need to be grouped together in order to improve the estimate. Studies involved in meta-analysis must be sufficiently homogeneous in order to be grouped and must address sufficiently similar target populations.

Meta-analysis can be used in any field where multiple studies report on the same research question, whether it is medicine, biology, physics, economics, psychology or sociology. Our main interest is meta-analysis in clinical and genomics research, and therefore this chapter focuses on combining data from clinical trials (section 2.1) and genomics experiments (section 2.2).

2.1 Meta-analysis in clinical research

Meta-analysis in clinical research has a long history mainly starting in 1970s, and its usage has been growing exponentially since then (Haidich [2011]). This type of meta-analysis combines and integrates the results of several independent randomised clinical trials (RCT), or observational studies. It is crucial that the trials or studies

have to be 'combinable', i.e. they have to address the same research question, and have the same characteristics. More on the choice of studies for meta-analysis can be found in the handbook[1] of Cochrane. The main purposes of meta-analysis in clinical research are (Chalmers [1988]):

- To resolve conflicting conclusions found in previous studies by examining the quality, subjects and interventions of these studies.
- To increase statistical power of small randomised trials, which are prone to Type II error (Freiman et al. [1978]).
- To add more certainty and precision to the effect size estimate, as an individual study might over- or under-estimate the effect.
- To answer new questions that could not be answered in an individual study, for example by comparing multiple treatments (see Section 2.1.2.2).

2.1.1 Clinical data for meta-analysis

Most of the outcome data in clinical trials are either categorical (e.g. binary failure vs. success), continuous (e.g. cholesterol level) or count data (e.g. days to remission). Depending on the available outcome, the data that enter meta-analysis can be either *contrast-level*, describing differences between pairs of treatments (e.g. using an odds ratio), or *arm-level*, describing the outcomes for each individual treatment (e.g. number of patients that responded).

2.1.2 Methods

The meta-analytical methods of clinical data can be divided into two groups: *pairwise meta-analysis*, combining studies with two specific treatments, or *network meta-analysis* (NMA), combining studies with multiple treatments. We introduce both groups below and focus on NMA, because our algorithm is used as a NMA tool in Chapter 5.

[1] `http://handbook.cochrane.org/part_2_general_methods_for_cochrane_reviews.htm`

2.1.2.1 Pairwise meta-analysis

Pairwise meta-analysis is a standard meta-analysis that combines studies, which compare two specific treatments (Sutton et al. [2000]). Each study reports an effect estimate for this pair of treatments, whether it is an odds ratio, relative risk or difference in change from the baseline. The result of pairwise meta-analysis is a pooled effect estimate for this pair of treatments, which is supposedly less biased and has lower degree of uncertainty, compared to the individual study estimates.

There are two ways of thinking of the effect the treatments have: fixed-effect model (FEM) and random-effect model (REM). FEM assumes that all studies estimate the same true underlying effect, while REM assumes that the true underlying effects may differ for each study.

Let us assume that n studies report an effect estimate, say $y_j, j = 1, \ldots, n$, for treatment A versus treatment B. A REM for pairwise treatment meta-analysis can be then written as

$$\begin{aligned} y_j &\sim \mathbb{N}(\delta_j, \sigma_j^2), \\ \delta_j &\sim \mathbb{N}(\theta, \epsilon^2), \end{aligned} \tag{2.1}$$

where $y_j \in \mathbb{R}$ is the effect estimated by j^{th} study, $\delta_j \in \mathbb{R}$ is the underlying, unknown, true effect in study j, σ_j^2 is the within-study variance, θ is the pooled estimate, and ϵ^2 is the between-study variability. When $\epsilon^2 = 0$, i.e. all variability is assumed to be caused purely by the within-study variance, then the model becomes a FEM. When comparing the estimate of θ in FEM and θ in REM, the standard error, variance and confidence interval will always be larger for θ in REM, because additional variability is involved.

The parameters can be estimated in both, Bayesian and frequentist way. An often heard critique of the Bayesian approach is that the prior distribution needs to be assumed for θ and ϵ. In order to avoid bias caused by the choice of the prior distribution, a noninformative 'flat' distribution is often used.

2.1.2.2 Network meta-analysis (NMA)

A disadvantage of pairwise treatment meta-analysis is the restriction to two treatments only. When more than two treatments are available to treat a health condition, a pairwise treatment comparison does not have the ability to capture the rela-

tive differences among multiple treatments. Commonly, clinicians are interested in multiple treatments comparison, where the treatments are ordered according to the efficacy or acceptability. This can then help to decide which treatment is suitable for a specific group of patients. Such comparisons can be achieved by constructing large RCTs involving all existing treatments, which is, nevertheless, financially and logistically difficult. Therefore most of the RCTs compare only two or three interventions, one of them often being placebo, with the aim to show that a new treatment is more effective than placebo or a treatment already implemented in regular care (a 'gold standard' treatment). *Network meta-analysis*, or *multiple treatment meta-analysis*, offers a way to compare multiple treatments, without planning large and expensive clinical trials. NMA cumulates direct comparisons, so-called *direct evidence*, from contrast-based (two treatments) or multi-arm (several treatments) clinical trials, in order to obtain comparisons between treatments that were not directly compared in any of the trials, so-called *indirect evidence* (see an illustration on Figure 2.1).

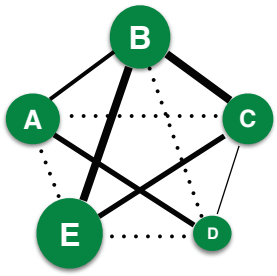


Figure 2.1: Network of comparisons of treatments A,B,C,D,E for NMA. Solid lines illustrate direct and dotted lines indirect evidence. The width of the lines is proportional to the number of trials comparing the pair of treatments that it is connecting, the size of each node is proportional to the number of patients.

Bucher et al. [1997] and Hasselblad [1998] were first to suggest the use of indirect comparisons in multiple treatment meta-analysis. Lumley [2002] proposed the term *network meta-analysis*. The currently most used Bayesian approach (see formula 2.3), implementing the Markov chain Monte Carlo algorithm, was suggested by Lu and Ades [2004]. More recently, some frequentist approaches have been suggested: the multivariate meta-analysis model by White et al. [2012], or the confidence distribution approach by Yang et al. [2014]. Detailed summaries and reviews of the methods can be found in Salanti et al. [2008], Efthimiou et al. [2016], or Madden et al. [2016].

Methods for NMA can be classified in different ways, whether it is the type

of model, the type of data used (contrast-based or multi-arm) or the ways of implementing it (Bayesian or frequentist). Here the methods are divided based on the model type, according to The Cochrane Collaboration[1]. All the models can be fit in both Bayesian and frequentist ways and differ mainly in the complexity of implementation and available software.

Most of the models are based on the so-called *consistency equation*. Let us assume we want to mutually compare treatments A, B and C, i.e. we want to know the treatment differences between AB, AC and BC. Imagine, we only have access to studies that compared A with B, and A with C. The consistency equation says that we can derive the missing BC comparison as

$$\theta_{BC} = \theta_{AB} - \theta_{AC}. \tag{2.2}$$

Let us assume there are p treatments compared, say $\{A, B, C, D, \ldots\}$, but the majority of the studies compared only a couple out of these treatments. Nevertheless, the aim of NMA is to describe mutual comparisons of all possible pairs of treatments, both direct and indirect ones. There are $c = \frac{p!}{2(p-2)!}$ such pairs. In reality, we do not need to estimate all c comparisons, but only a subset of them, called *basic parameters*, and derive the remaining ones using the consistency equation 2.2. For example, if there are 4 treatments, say A, B, C, D, involved in NMA, then there are 6 pairs of treatments (AB, AC, AD, BC, BD, CD) we need to know the comparison of. But, in fact, we only need to calculate the comparisons of 3 pairs, e.g. AB, AC, AD (these are called *basic parameters*) because the remaining 3 can be easily derived from them using the consistency equation: $\theta_{BC} = \theta_{AB} - \theta_{AC}$, $\theta_{BD} = \theta_{AB} - \theta_{AD}$ and $\theta_{CD} = \theta_{AC} - \theta_{AD}$.

The most common models for NMA are hierarchical, meta-regression, multivariate and two-stage linear models. We introduce them below and mention how are they implemented.

Hierarchical models (Lu and Ades [2004]; Salanti et al. [2008]) assume the same hierarchy as in the pairwise REM (formula 2.1), but are more complex, because there are more than 2 treatments involved. Because there are multiple comparisons, the effect estimate y_j from formula 2.1 becomes a vector $\mathbf{y}_j = (y_{jAB}, y_{jAC}, y_{jBC}, \ldots) \in$

[1] `http://methods.cochrane.org/cmi/sites/methods.cochrane.org.cmi/files/uploads/CMIMGStream2document20131001.pdf`

$\mathbb{R}^c$, where c is number of treatment pairs, and similarly δ_j, σ_j^2 become vectors in $\mathbb{R}^c$. Let us denote $y_{j..}$ an arbitrary element of $\mathbf{y}_j$, $\delta_{j..}$ the underlying effect in study j, $\sigma^2_{j..}$ the within-study variance, and $\epsilon^2_{..}$ the between-study variance. Two dots always denote a pair of treatments from the set $\{A, B, C, D, \ldots\}$. The NMA hierarchical model can then be written as

$$\begin{aligned} y_{j..} &\sim \mathbb{N}(\delta_{j..}, \sigma^2_{j..}), j = 1, \ldots, n \\ \delta_{j..} &\sim \mathbb{N}(\theta_{..}, \epsilon^2_{..}). \end{aligned} \tag{2.3}$$

Hierarchical models seem to be implemented most often (Efthimiou et al. [2016]), as they are flexible in modelling the underlying assumptions and are easy to extend, although more difficult to fit using the frequentist approach.

Meta-regression models (Lumley [2002]) treat different treatment comparisons as covariates in a meta-regression model. The general meta-regression model for estimating the underlying pairwise effects θ can be written as (Salanti et al. [2008], Efthimiou et al. [2016]):

$$\mathbf{y} = \mathbf{X}\theta + \epsilon + \sigma, \tag{2.4}$$

where $\mathbf{y}$ is the vector of individual relative treatment effects, as observed in each study, ϵ is the vector of random effects (between-study variance) and σ is the vector of random errors in $\mathbf{y}$ (within-study variance). $\mathbf{X}$ is the design matrix, taking values 0, -1, 1, and describing the structure of the network, i.e. which comparison was performed in which study. For example, if the network consisted of three studies: Study 1 comparing treatments AB, Study 2 comparing AC and CD, and Study 3 comparing AD, then the model (2.4) would be written as

$$\begin{pmatrix} y_{1AB} \\ y_{2AC} \\ y_{2CD} \\ y_{3AD} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_{AB} \\ \theta_{AC} \\ \theta_{AD} \end{pmatrix} + \begin{pmatrix} \epsilon_{1AB} \\ \epsilon_{2AC} \\ \epsilon_{2CD} \\ \epsilon_{3AD} \end{pmatrix} + \begin{pmatrix} \sigma_{1AB} \\ \sigma_{2AC} \\ \sigma_{2CD} \\ \sigma_{3AD} \end{pmatrix}. \tag{2.5}$$

Remember, that not all the comparisons need to enter the model. The omitted comparisons can be written as a linear combination of the calculated ones, following the consistency equation (2.2). Hence in the example (2.5) only $\theta_{AB}, \theta_{AC}, \theta_{AD}$ appear in the model, because the missing effect θ_{CD} can be calculated as $\theta_{CD} = \theta_{AC} - \theta_{AD}$.

The random effects ϵ, as well as the random errors σ are assumed to follow a multivariate normal distribution, so $\epsilon \sim N(\mathbf{0}, \mathbf{S})$ and $\sigma \sim N(\mathbf{0}, \Delta)$, where $\mathbf{S}$ is the between-study variance-covariance matrix and Δ is the within-study variance-covariance matrix. There are also various approaches for estimating Δ (Franchini et al. [2012]; Higgins and Whitehead [1996]) and $\mathbf{S}$ (Berkey et al. [1998]; Jackson et al. [2010]; van Houwelingen et al. [2002]).

The model parameters θ can then be estimated in a Bayesian way using Markov Chain Monte Carlo (MCMC) simulations, or in a classical frequentist way, for example by generalised least squares (GLS) as $\hat{\theta} = \mathbf{X^T W X}^{-1}\mathbf{X^T W y}$, where $W = (\mathbf{S} + \Delta)^{-1}$.

Setting up the variance-covariance matrices $\mathbf{S}$ and Δ can be a challenging task (Higgins et al. [2012]; Lu and Ades [2004]), especially when the reference treatment is not the same for all studies. According to Madden et al. [2016], there is a "considerable danger that the analyst could make mistakes in the model formulation or in the construction of the covariance matrix".

Multivariate models (White et al. [2012]) are models where each comparison is treated as one outcome. It is required that all studies have the same reference treatment. If there is a study that does not include the reference treatment, it can be imputed using a data-augmentation technique. The model can be written as (Efthimiou et al. [2016]):

$$\mathbf{y} = \mathbf{X}^*\theta + \epsilon + \sigma. \tag{2.6}$$

In this model the design matrix $\mathbf{X}^*$ takes only 0 or 1 values, according to the "outcomes" reported. Here "outcomes" have quotation marks, because some of them might be imputed, hence not being real outcomes. For instance, say Study 1 compares AB, Study 2 compares BC, and Study 3 compares AC. If A is considered the reference treatment, then Study 2 cannot enter the model. One could exclude this study from the analysis, but loose a valid direct evidence. Instead, the treatment A can be imputed and Study 2 becomes a three-arm study and can be used in the model (in blue):

$$\begin{pmatrix} y_{1AB} \\ \color{blue}{y_{2AB}} \\ \color{blue}{y_{2AC}} \\ y_{3AC} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \color{blue}{1} & \color{blue}{0} \\ \color{blue}{0} & \color{blue}{1} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_{AB} \\ \theta_{AC} \end{pmatrix} + \begin{pmatrix} \epsilon_{1AB} \\ \color{blue}{\epsilon_{2AB}} \\ \color{blue}{\epsilon_{2AC}} \\ \epsilon_{3AC} \end{pmatrix} + \begin{pmatrix} \sigma_{1AB} \\ \color{blue}{\sigma_{2AB}} \\ \color{blue}{\sigma_{2AC}} \\ \sigma_{3AC} \end{pmatrix}. \tag{2.7}$$

Two-stage linear models (Lu et al. [2011]) combine classical pairwise meta-analysis and meta-regression model in two stages: (i) pairwise meta-analysis to obtain pooled estimates, (ii) meta regression on the pooled estimates. This method can be used to investigate how the direct evidence influences the overall network estimates.

2.1.2.3 Heterogeneity and inconsistency

Despite many advantages, the NMA methodology is often criticised, mainly because it assumes that different sources of evidence are consistent and in agreement. If this assumption is violated, the results from NMA become less reliable and difficult to interpret (Efthimiou et al. [2016]). Specifically, NMA models assume that the direct evidence is in agreement across trials and that the indirect evidence does not contradict the direct evidence. Hence, before applying any of the above mentioned methods, these assumptions should be verified. *Heterogeneity*, i.e. level of disagreement between studies that compare the same treatments, should be assessed for each pair of treatments included in the NMA for which there is direct evidence (Donegan et al. [2013]). *Inconsistency*, i.e. level of disagreement between the direct and indirect evidence, should be assessed for all treatments with both direct and indirect evidence, i.e. for all closed loops in the network (Efthimiou et al. [2016]). Another problem with combining different trials is that the indirect comparisons do not preserve the within-trial randomisation and hence the combined evidence is observational rather than randomised (Bucher et al. [1997]). Other factors influencing the quality of NMA can be *publication bias* or *selective reporting bias*, where some comparisons are generally more focused on than others (Salanti et al. [2008]).

2.1.3 Tools and packages

Most authors apply a Bayesian approach to estimate the parameters (Bafeta et al. [2014] reviewed 121 NMA studies, out of which 91 chose a Bayesian approach). Carlin et al. [2013] argue that Bayesian methods are more flexible than frequentist, as they can incorporate various sources of information; and also that Bayesian methods provide probability of the individual treatments to be ranked first, second, third, and so on, and hence are more practical for interpretations. However, Rücker et al. [2015], for example, suggested a way to rank the treatments also in frequentist NMA. Salanti et al. [2008] claim that the frequentist software is not straightforward and for

complex situations Bayesian estimation procedures are more convenient. Another fact, whether it is a reason or consequence of higher usage of Bayesian methods, is that the majority of user-friendly software implement Bayesian methods.

There are several software tools that have implemented the models listed above. Among the most used is a Bayesian software **WinBUGS** (Lunn et al. [2000] - cited 4114 times, as of January 13th 2017), and its follower **OpenBUGS**, both developed under the BUGS (Bayesian Inference Using Gibbs Sampler) project (Lunn et al. [2009]) by Medical Research Council Biostatistics Unit, Cambridge, UK, jointly with the Imperial College School of Medicine at St Mary's, London, UK. A similar software is **JAGS** (Just Another Gibbs Sampler) by Plummer et al. [2003], which was designed to have a cross-platform engine for the BUGS language, but is independent of the BUGS project. Another Bayesian software is **Stan** (Carpenter et al. [2016]), which is based on Hamiltonian Monte Carlo sampling, unlike the above ones, which are based on Gibbs sampling. Stan is written in C++ and hence should be very fast. All four programs can be run from R software environment, using packages `R2WinBUGS` for WinBUGS, `BRugs` for OpenBUGS, `rjags, R2jags` for JAGS, and `RStan` for Stan. R itself offers around 70 packages for meta-analysis[1], out of which three were designed specifically for network meta-analysis (Neupane et al. [2014]):

- `gemtc` (van Valkenhoef and Kuiper [2014]) - Bayesian hierarchical NMA modelling, uses JAGS
- `pcnetmeta` (Zhang et al. [2014]) - Bayesian hierarchical NMA modelling, uses JAGS
- `netmeta` (Rücker et al. [2015]) - Fixed-effect meta-regression modelling (model 2.4 with $\epsilon = 0$), uses frequentist methods

2.2 Meta-analysis in genomic research

Genomics is a research discipline that applies sequencing and bioinformatics methods in order to analyse the function and structure of an organism's genome. Such research can find genetic variants that increase risk or severity of specific diseases, such as cancer, or, for example, susceptibility to a certain disease in a population.

[1] `https://cran.r-project.org/web/views/MetaAnalysis.html`

Meta-analysis in genomic research refers to methods that combine results from different genomic studies. Such studies measure gene expression of thousands of genes, and, by combining them, one can identify common patterns and improve the signal in the data (Rung and Brazma [2013]). Meta-analyses of gene expression data are performed with similar aims as those in clinical research:

- To improve the power of detecting weak signals, which could not be found in any of the individual data sets.
- To increase the amount of information without costly and time consuming additional experiments in a single laboratory.
- To combine data from several different array platforms.

An example of use of genomic meta-analysis can be found in Chen et al. [2011b], where several microarray datasets from prostate, bladder and renal cancers were combined in order to detect urinary biomarkers specific for prostate cancer. In another study, Fortney et al. [2015] combined several lung cancer gene signatures and compared them to genomic response to various drugs, in order to identify new lung cancer therapeutics.

A detailed summary of gene expression meta-analysis can be found in Rung and Brazma [2013]. Here the genomic data are introduced, and the most used models and software are briefly described.

2.2.1 Gene expression data

The gene expression data in a genomic study are typically provided as a $p \times n$ matrix, where p is the number of genes, which can vary from anywhere between one hundred to several tens of thousands, and n is the number of samples or studies, usually numbering in the single or double digits. The matrix contains gene expression values (for example color ratios or read counts, as described below).

There are two leading data technologies: *microarray* and more modern *sequencing*, whose main differences are in design and resolution quality (a comparison can be found in Mantione et al. [2014]). Both technologies started as very expensive and therefore sharing the analysed data among other researchers was not always a matter of course. Today, however, almost all published studies are required to make the raw, as well as processed data publicly available. Below, both technologies are introduced and some larger databases listed.

Microarray is a hybridisation technique started already by Grunstein and Hogness [1975]. Microscopic DNA spots, also called *probes*, which represent genes, are attached to a solid surface, called *chip*. The DNA sequences of interest (say, from a patient and a healthy control) are color-labeled and applied to the chip, where they bond with the previously attached DNA spots/genes. This bonding process is called *hybridisation*. The fluorescent labels of the sequences that bound produce a color signal, from which the expression of each spot/gene is calculated. Higher amount of the 'patient's color' means higher expression of the patient's gene (so-called *over-expression*), higher amount of the 'control's color' means lower expression of the patient's gene (so-called *under-expression*), same amount of both colors points to no difference between the patient's and control's gene expression.

This technology started progressing rapidly in the late 90s with increased knowledge about the DNA and with the development of new production methods (Bumgarner [2013]). Nevertheless, the technology is limited to the chosen DNA spots/-genes bound to the microarray chip and therefore cannot detect new unexpected transcripts. When there is a reason to investigate new DNA sequences, the chips need to be updated.

Sequencing is a method for determining the exact order of the nucleotides - adenine, guanine, cytosine, and thymine/uracil - within a DNA/RNA molecule, and hence aim for much more detail and precise description of the DNA/RNA, compared to microarrays. First whole nucleic acid sequence, isolated from yeast, was read already by Holley et al. [1965], but the major breakthrough came in 1977 with Sanger's *chain-termination* technique (Sanger et al. [1977]). This technique has been improved and new methods developed over the years, and, when commercialised around the year 2000, became known as the *next-generation sequencing* (NGS). Originally, these technologies were hugely expensive, and therefore unavailable for the majority of researchers. The costs of human genome sequencing dramatically dropped in the first years of the 21st century and since then the NGS technologies have been evolving dramatically (Goodwin et al. [2016]), lowering the costs and increasing the speed and precision. They have been gradually replacing microarray technologies, which are, nevertheless, still much cheaper and hence still widely used. A historical overview of the sequencing methods can be found in Heather and Chain [2016], and Goodwin et al. [2016].

NGS technologies can be generally divided into two groups: DNA- and RNA-

sequencing. DNA-sequencing methods determine the precise order of the nucleotides, hence they are used to detect single mutations, such as single nucleotide polymorphisms (SNPs)[1], novel sequences, splice variants[2], etc. RNA-sequencing methods analyse molecules of RNA, which reflect the rate of gene transcription. This means that, additionally to the sequence of nucleotides, also the gene expression can be detected. In more detail, millions of short RNA sequences are read and the number of reads, so-called *read count*, for each sequence is calculated. Higher read count for a particular gene corresponds to larger quantity of RNA, hence higher gene expression. Read counts for each gene are then compared between the experimental groups (e.g. patient versus control), and significant differences are reported as over- or under-expression.

Generally, sequencing technologies have higher resolution and much lower limit of detection than a standard microarray. Probe differences, which contribute to the high noise in microarray, are avoided in sequencing data. Nevertheless, combination of sequencing data is also non-trivial, because the preparation and the sequencing protocols (such as the methods for priming, fragmentation or amplification) are not standardised. Hence when combining sequencing experiments that used different protocols, related differences should be carefully checked and accounted for (Rung and Brazma [2013]). Sequencing methods have also additional biases, connected to gene length or gene GC content[3] (Zheng et al. [2011]). Therefore, careful normalisation should precede the analysis and consequent meta-analysis of sequencing data. The normalisation procedure should be chosen based on the data and experimental conditions, in order to remove existing biases while preserving real biological differences between samples (Aleksic et al. [2014]). Another important issue are the storage and computing requirements, which are substantially larger for sequencing data, compared to microarray.

After normalisation, for example by calculating 'Reads per Kilobase of gene length per Million reads' (RPKM), the sequencing data can be combined. Nevertheless, it has been shown that the normalised values can still be biased and should be additionally corrected (Zheng et al. [2011]).

[1]SNP - a variation in a single nucleotide that appears on the same position in at least a certain, yet small, fraction of population, e.g. $> 1\%$.

[2]Splice variant - a recombinant DNA molecule that origins from cutting and resealing DNA from different sources.

[3]Guanine-cytosine (GC) content - the percentage of guanine or cytosine bases in a specific fragment of DNA (e.g. human genome has 42% GC content).

Alternative technologies to microarray and sequencing are NanoString, with exceptionally high resolution; qPCR, with high sensitivity and specificity; or optical mapping (Goodwin et al. [2016]).

Databases Large collection of genomic data are published online in publicly available databases. Among the most used are ArrayExpress, Gene Expression Omnibus (GEO), DDBJ Omics Archive, or Sequence Read Archive (SRA). Additional databases with a specific topical interest are summarised in Rung and Brazma [2013]. Raw sequencing data represent a special ethical case, as the DNA donor can be potentially identified from them, and so there exist special genotype databases with restricted access, for example dbGaP (Genotypes and Phenotypes database), or European Genome-Phenome Archive (EGA).

2.2.2 Methods

The data used in meta-analysis can be provided as normalised raw data (e.g. color log ratios or counts) or already processed summary-level data (e.g. p-values, ranks or effect sizes from compared conditions). Combining raw data is less straightforward, as biases, arising for example from platform, laboratory or probe differences, have to be accounted for. Ramasamy et al. [2008] identify key issues and suggest a stepwise approach for microarray data meta-analysis. A review summary of reusing microarray, as well as sequencing data, can be found in Rung and Brazma [2013]. Tseng et al. [2012], in their comprehensive literature review, analysed the purposes and types of 620 published meta-analyses, and found that the most common purpose (62%) of microarray meta-analysis is finding consistently differentially expressed (DE), i.e. over- or under-expressed, genes; or detecting signaling pathways[1]. Because the next-generation sequencing data are relatively new, there is less literature concerning meta-analysis of such data. Nevertheless, most methods for summary-level data can be used for microarray, as well as sequencing data. Ultimately, microarray, sequencing and other data types can be also combined together.

[1]Signaling pathway - group of molecules that work together and control one or more cell functions

2.2.2.1 Combining raw data

Raw data can be combined after careful quality control, filtering and normalisation, which is of special importance. The biases in genomic experiments have strong effect on the data and increase the risk of heterogeneity, therefore combining raw data, especially microarray, is usually restricted to studies from the same platform. When using raw data from different platforms, additional adjustments for batch and probe effects have to be made (Chen et al. [2011a]). Nonetheless, even under the same platform, a special attention has to be paid to normalisation. Irizarry et al. [2003] designed a normalisation method specifically for data output by the one of the leading microarray producers, the Affymetrix company. Cheng et al. [2009] showed that gene-wise discrepancies across studies are often significant and proposed a ratio-adjusted gene-wise normalisation (rGN).

Most methods focus on searching for DE genes or classifying cancer types. For example, Wang et al. [2004] adopted a purely Bayesian approach to identify DE genes; Parmigiani et al. [2004] proposed the integrative correlation meta-analysis in order to classify several types of lung cancer; Qiao et al. [2010] suggested a weighted Distance Weighted Discrimination (wDWD), classification method based on Support Vector Machine (SVM), and used it to classify different lung cancer types from microarray data. Fishel et al. [2007] also used SVM techniques to obtain a ranked list of the most expressed genes. Meta-analysis of raw sequencing data was done for example by Xia et al. [2014] or Sudmant et al. [2015].

2.2.2.2 Combining summary-level data

An elegant solution to overcome various problems connected to the biologically complicated raw data, is to combine the results of individual studies, such as p-values, effect sizes or gene ranks, which we shall call *summary-level data*. The methods for combining summary-level data can be classified according to the type of data they combine:

Combining p-values Often, within an experiment, a p-value is calculated for each gene. The p-value represents the significance of difference between two experimental conditions (e.g. tumor vs. control). For example, when the p-value for gene A is smaller than a certain significance level, it suggests that gene A in the tumor sample expresses differently than the same gene in the control sample, and

hence might be connected to the formation of the tumor, and should be further investigated. Due to large number of genes involved in microarray studies, many false associations appear, and it is therefore important to make sure the p-values are adjusted for multiple testing. A major advantage of the p-value-data is that they are standardised to a common scale $[0, 1]$ and thus easily combinable across experiments. Some simple methods take the minimum (Tippett et al. [1931]) or the maximum (Wilkinson [1951]) p-value as a test statistic. The minimum method considers a gene DE if there is at least one study with a small p-value, while the maximum method is more conservative and needs all studies to have small p-values in order to find a gene DE. When comparing only a few studies (say 2-4), the overlaps in DE genes can be visualised by the naïve Venn diagram (Venn [1880]); when many studies are combined, simple counting methods can be used (for example counting how many studies have a p-value$\leq$ 0.05). Nevertheless, such simple methods are considered statistically inefficient (Friedman [2001]). Rhodes et al. [2002] were among the first to suggest more sophisticated statistical meta-analysis. They used the well-known Fisher's method (Fisher [1925]) to sum up minus log-transformed p-values: $F = -\sum_{j=1}^{n} \log(p_j)$, where n studies are combined and p_j is the p-value of study $j = \{1, \ldots n\}$ for a particular gene. A larger F score reflects stronger aggregated DE evidence. Some other methods use the Stouffer's score (Stouffer [1949]). Later, Li et al. [2011] proposed an adaptively weighted Fisher's method, adding a categorisation of the found DE genes in order to help further biological exploration. The Fisher's method was also used by Rau et al. [2014] in order to combine p-values from sequencing data.

Combining effect sizes When studies have similar design and ways to measure the outcome, methods that combine effect sizes, such as risk ratios, can be applied. These methods often use the classical fixed or random effect model (see section 2.1.2.1). One of the first such effect size methods for microarray data was suggested by Choi et al. [2003], and followed by other variations and extensions, for example by Hu et al. [2005], or Marot et al. [2009]. Also several Bayesian methods with MCMC simulations have been suggested (see a comparative study in Conlon et al. [2007]).

Combining ranks Of special interest for us are methods combining ranks of the values, instead of the values themselves. These methods range from simple mean

or sum of ranks to more sophisticated methods like top-scoring classifier (Xu et al. [2005]), which uses the ranks of the gene expression values; RankProd (Breitling et al. [2004]), which calculates the product of the rank of pairwise differences of samples between two groups across the studies; Rank-rank hypergeometric overlap (RRHO) by Plaisier et al. [2010], which calculates the significance of overlap between two ranked lists; and rank aggregation methods based on Markov chain Monte Carlo simulations (DeConde et al. [2006], Lin and Ding [2009]). Various methods using rank data are discussed in more detail in Chapter 3.

2.2.3 Tools and packages

The majority of methods for meta-analysis of genomic data are implemented in the R language. Some packages are more general, others are designed specifically for a certain data type. Below several packages are listed, with a short note on the software, aims, and implemented methods.

- `metaArray` - a Bioconductor package for DE genes detection, implements methods of Choi et al. [2003]; Parmigiani et al. [2002, 2004]; Wang et al. [2004].
- `MetaDE` - an R package for DE genes detection, combines p-values and effect sizes, implements 12 major methods (most of them mentioned in this chapter).
- `metaMA` - an R package for DE genes detection, combines p-values and effect sizes, implements the method of Marot et al. [2009].
- `GeneMeta` - a Bioconductor package for DE genes detection, combines effect sizes, based on Choi et al. [2003].
- `METRADISC` (METa-analysis of RAnked DISCovery) - a freely downloadable software[1] for DE genes detection, implements the method of Zintzaras and Ioannidis [2012], based on Monte Carlo simulations.
- `RankProd` - a Bioconductor package for DE genes detection, based on the rank product non-parametric method (Breitling et al. [2004]).
- `metaRNASeq` - an R package for DE genes detection, combines p-values, implements the method of (Rau et al. [2014]).

[1] `http://biomath.med.uth.gr/default.aspx?id=232164AC-9C6B-4A27-A595-2A22C35B6260`

- `RRHO` (Rank-Rank Hypergeometric Overlap) - an R package, as well as a web application[1], implements the method of Plaisier et al. [2010].
- `TopKSpace` - a submodule of the `TopKLists` R package, based on the method of Lin and Ding [2009].
- `MultiMeta` - an R package for meta-analysis of Genome Wide Association Studies (GWAS).
- `MetABEL` - an R package for meta-analysis of GWAS.
- `EasyStrata` - an R package for meta-analysis of GWAS.
- `MetaPath` - an R package for pathway analysis.

2.3 Discussion

Meta-analysis is a powerful tool to improve already existing results. This is achieved by combining the results and inferring stronger conclusions. The main focus of this chapter was on combining results from clinical and genomics studies.

Clinical trials provide valuable evaluations of medical treatments. Nevertheless, due to a large amount of possible biases skewing the results, combining them using meta-analysis can improve the findings without the need for any additional trials.

Genomic data face a similar problem. Experiments containing design and processing errors (the amount of which largely depends on the technology used) are expensive to duplicate or reproduce. The desired, more stable results can be achieved cheaply by combining the already existing studies, even across technologies, i.e. by meta-analysis.

As discussed in this chapter, meta-analysis must always be performed with careful data quality and compatibility control.

Although the methods and approaches for meta-analysis might differ depending on the field and data types, the aims always remain the same: to add more certainty to the results, to resolve possibly conflicting results, and to obtain new findings that a single study cannot provide.

[1] `http://systems.crump.ucla.edu/rankrank/rankranksimple.php`

Chapter 3

Ranking

Ranking from 'the best' to 'the worst' is part of our every day life. We rank our favourite meals, films, holiday destinations, politicians, schools, restaurants or football teams. Academic journals are ranked according to the Impact Factor (or other measures), search engines rank search results according to relevance or date. In medicine, treatments are ranked according to their efficacy. In molecular biology, genes are ranked according to their connection to a disease. There are countless examples of ranking data and their usage in day-to-day life, science or industry. It is therefore important to understand and extract useful information from such data.

This chapter introduces the concept of multiple ranking data and focuses on methods relevant to the multiple ranking model we suggest in Chapter 4. Section 3.1 introduces the concept of ranking data, section 3.2 describes the most common modelling approaches, and section 3.3 is dedicated to meta-analysis and inference from ranking data. The last two sections, 3.4 and 3.5, introduce some correlation and distance measures, and briefly describe special data structures. A more general overview of ranking data models can be found in the standard reference book for ranking data by Marden [1995], or a more recent book by Alvo and Yu [2014].

3.1 Ranking data

Let us define some terminology crucial for our model, using an example. Imagine having a set of objects $\{A, B, C, D\}$, which can be practically anything: films, countries, products, drugs, genes, and so on. These objects can be ranked by human or machine *assessors* from 'the best' (e.g. the tallest, the strongest, the most relevant)

to 'the worst' (e.g. the shortest, the weakest, the least relevant), depending on what measure of goodness is chosen (e.g. height, strength, relevance). For example, an assessor decides that in his opinion the object B is the best, C second best, A third best and D the worst, whatever 'best' and 'worst' might represent. One can provide his/her assessment as a sequence of object names, i.e. (B, C, A, D), which is called *the ordering*. Or, one can assign ranks to each object, for example 1 to the best one and 4 to the worst one. So in this example A gets the rank 3, B the rank 1, C the rank 2 and D the rank 4. This way we get a sequence of integers $(3, 1, 2, 4)$, which is called *the ranking*. The ranking determines the positions of the objects in the original set $\{A, B, C, D\}$. In some cases, where the objects can be judged by an objective measure, such as weight or color intensity, there exists what we call the *true ranking* (and analogically the *true ordering*). The true ranking is the ranking of objects by the objective measure, so for example from the heaviest to the lightest. Throughout the thesis, the term 'a rank' is used for one rank position, the term 'ranks' for several rank positions, the term 'ranking' for a full set of ranks, and the term 'rankings' for several full sets of ranks.

Typically, ranking data consist of rankings by several assessors, not just one. We call such data *multiple ranking data*. Consider an example where three people are asked to rank four drinks, each 100ml, according to their sugar content (Table 3.1). The true ordering is (Coca-cola (11g), Sprite (10g), Beer (2.7g), Water (0g)) and

	Person 1	Person 2	Person 3
Coca-cola	1	2	1
Sprite	2	1	2
Beer	3	3	3
Water	4	4	4

Table 3.1: An example of ranking data. Four drinks were ranked by three people according to their sugar content.

so the true ranking is $(1, 2, 3, 4)$. All three assessors agreed that the least sugar is in water and second least in beer. They ranked Coca-cola and Sprite as the top 2 sweetest drink, but they do not all agree on the order of the top 2.

Theoretically, there are $p!$ possible orderings/rankings of p objects. For our example this means that there are $4! = 24$ possible orderings of the presented drinks. Considering the number of possible rankings, our assessors are in high agreement. This suggest that the drinks, or more precisely their sugar contents, are quite distinct from each other and hence easy to be ranked correctly. Generally, the more the

assessors agree on a choice of ranks, the more distinct the objects probably are (e.g. water versus beer). Higher disagreement, on the other hand, points to unclear differences between the objects (e.g. Coca-cola versus Sprite).

The question is, how to extract and summarise more information from the multiple ranking data. This is the topic of the following section.

3.2 Modelling ranking data

Deeper understanding of the data can be obtained by modelling the ranking data and the processes that originated them. Enormous amount of ranking data models have been suggested since the beginning of the 20^{th} century. This section does not intend to summarise them all but rather to introduce the background concepts the models are based on.

According to Heiser and D'Ambrosio [2013], the modelling approaches can be split into three groups: (1) exploring the structure of ranking data, (2) probabilistic modelling of the ranking process, with the assumption of substantial agreement among the assessors (*homogeneity*), (3) probabilistic modelling of the population of assessors, with the assumption of substantial disagreement between some of them (*heterogeneity*). However, these groups are not mutually exclusive, as some approaches from one group can be used to develop models for another. The following text introduces all three groups, with focus on the second one, probabilistic modelling of the ranking process, as it will be adapted for our model in Chapter 4.

3.2.1 Exploring the structure of ranking data

Methods for exploring the data structure are the least ambitious, trying to simply describe the structure of the observed rankings. Exploratory multivariate methods can be used to provide insight into the data. For example, simple calculation of the mean rank and standard deviation for each object over all assessors, or plotting histograms of rank frequencies. More sophisticated methods include principal component analysis (Gabriel [1971]), multidimensional scaling (Cox and Cox [2008]; de Leeuw and Heiser [1982]), factor analysis (Alvo and Yu [2014], Chap.9), or cluster analysis (Jacques and Biernacki [2014]).

3.2.2 Probabilistic modelling of the ranking process

More complex type of models aim for mathematical description of the discriminal process (physical, mechanical, chemical, neurological) that an assessor (or 'a judge', as used in psychology) undergoes to produce a ranking of objects (or 'stimuli', as used in psychology).

Thurstonian models. The first one to consider such a model was an U.S. psychologist named Louis Leon Thurstone (1887–1955) with a series of 6 articles in 1927 (among the most cited are Thurstone [1927a,b,c]), having purely psychological applications in mind. He first considered comparing two objects only, so-called *pairwise comparisons*, but later generalised the model to multiple comparisons (Thurstone [1945]). Statistical considerations of Thurstone's model were pioneered by Mosteller [1951] and Daniels [1950], and many followers extended the model further (David [1988]; Ennis and Johnson [1993]; Glenn and David [1960]; Rosner and Kochanski [2015]; Stern [1990]).

The main idea behind Thurstonian models is the following: It is assumed that there exists a ground truth, also called *latent variable*, which determines the correct ranking of the concerned objects. For example, sugar content of each drink determines the correct ranking from the sweetest one to the least sweet one. This ground truth is not necessarily known. Each assessor evaluates the objects in mind and assigns them *mental scores*, which represent his/hers perception of the ground truth, e.g. how sweet he/she thinks a drink is. The assessor then ranks the objects based on the mental scores. These scores are considered to be drawn from a Gaussian distribution with mean being the value of the ground truth, and standard deviation representing the quality or consistency of the assessors.

The assumption of a Gaussian (normal) distribution later became a subject of debate. Thurstone himself claimed that "in most of the experiments, the distributions are close to normal" (Thurstone [1927c]). Luce [1994] describes this claim as being "a bit overly optimistic" and discusses the suitability of the Gaussian assumption. Common argument for the choice of a normal distribution is the central limit theorem. However, according to Luce, many psychologists find normal distribution in this situation psychologically implausible. Luce [1994] also discusses alternative distributions, namely exponential, double exponential, logistic and multivariate distributions, suggested in the previous literature. Despite some criticism,

the normal distribution assumption seems to be still used today as the gold standard in applications of the Thurstonian models.

Defined as such, Thurstonian models have been greatly used in behavioral research, psychometrics and sensory analysis, where the data come mainly from questionnaires, surveys and consumer ratings. Among the most common tasks today is determining the degree of difference among two or more choice options (e.g. consumer products), modelling the relationship between choice options and decision-makers (e.g. consumers), and finding ways to supplement choice data (Böckenholt [2006]). Another important goal for cognitive science is measuring the quality of expertise in ranking tasks (Lee et al. [2011, 2014]).

Despite its origin in psychology, the concept of Thurstonian models has been applied to other disciplines, for example in health science for evaluation of European Quality of Life (EuroQoL) questionnaires (Dolan [1997]; Ratcliffe et al. [2009]; Salomon [2003]), or in statistical learning classifications tasks (Geng and Luo [2014]).

In Chapter 4, we use this approach to build a model for estimating the underlying signal from multiple ranked lists, and apply it in network meta-analysis to estimate effect size, or in genomic meta-analysis to identify potential cancer therapeutics (Chapter 5).

Another class of models aiming for modelling of the ranking process are ***distance-based ranking models*** (Mallows [1957]) which are based on the Bradley-Terry-Luce model (Bradley and Terry [1952]; Luce [1959]); and ***multistage models*** (Fligner and Verducci [1988]) which decompose the ranking process into a series of independent stages, and in each stage assume the Bradley-Terry-Luce model.

3.2.3 Probabilistic modelling of the population of assessors

In situations where heterogeneity among the assessors is likely, methods modelling the population of assessors are used. For example, in voting and elections, where different age or social groups tend to prefer different parties, it would not make sense to assume that there is one objective true ranking. Hence the subgroups of a population have to be taken into account. There are three types of approaches modelling the population of assessors (Heiser and D'Ambrosio [2013]). One group of such models consist of so-called ideal-point probabilistic methods, which assume one model for the whole population (e.g. Kamakura and Srivastava [1986]; van

Blokland-Vogelesang [1989]; Zinnes and Griggs [1974]). In the second group are models that consider an (unknown) mixture of subpopulations, each having its specific model parameters, for example a mixture of Bradley-Terry-Luce type models (Croon [1989]), or a mixture of distance-based models (Murphy and Martin [2003]). The third group of models also assume subpopulations but this time they are known, i.e. they make use of covariates, such as age, gender, education, social status, and so on (see, for example, Böckenholt [2001]; Chapman and Staelin [1982]; Francis et al. [2014]; Gormley and Murphy [2008]; Skrondal and Rabe-Hesketh [2003].

3.3 Meta-analysis of ranking data

Models for ranking data, as introduced in Section 3.2, can be used with various aims in mind. This section focuses on two specific meta-analytical problems: (1) search for an overall summary ranking, so-called *consensus ranking*, and (2) search for the top most informative subset of objects in long lists, so-called *top-k objects*.

3.3.1 Obtaining a consensus ranking: rank aggregation

One of the common problems is obtaining a *consensus ranking*, or *central ranking*. As the names suggest, it is a ranking that is the 'closest' to most of the individual rankings. Essentially, a consensus ranking is an 'average' ranking vector calculated from multiple ranking vectors. And just like there are various methods for calculating the average of values (e.g. arithmetic mean, geometric mean, median, mode), there are various methods for calculating a consensus ranking. In fact, a consensus ranking is an estimation of the true ranking. By looking at our example ranking matrix of drinks (Table 3.1), we can say, without any calculations, that a reasonable consensus ranking is $(1, 2, 3, 4)$. Generally, the task of obtaining a consensus rank is not that easy. Ranking data typically contain a larger amount of objects and assessors than in our example. Together with higher dimensions comes more disagreement between the assessors, and the problem becomes impossible to be solved by simply looking at the ranking matrix.

An estimation of the consensus ranking is crucial in all the fields where rankings by assessors exist. Bookmakers may base their odds on a consensus quality ranking of sport teams. Yearly university rankings often originate as a consensus of rankings by students or by various metrics. Companies may distribute finances in advertising

based on a consensus of product rankings by customers. A journal ranking can be a consensus of rankings by various quality metrics. A consensus ranking of genes, obtained from rankings by several laboratories, can point to new biomarkers.

In all these situations a consensus ranking is not only useful as an average summary ranking, but it is often of better quality and more objective than any of the individual rankings. This phenomena was described in psychology and economics as *The Wisdom of Crowds* (Surowiecki [2004]).

The methods that aim for a consensus ranking estimation are called *rank aggregation* methods. These have been traditionally used in applied psychology (marketing, advertisement), and later became an important tool for combining information from search engines or biological studies. Rank aggregation methods can be divided into three classes: distributional, heuristic, and stochastic approaches. Basic principles are summarised below, more details on these methods can be found in Lin [2010a]. Our method (Chapter 4) could be classified as a combination of the distributional and stochastic approaches. Nevertheless, it cannot be strictly called a rank aggregation method, as it primarily aims for estimation of the underlying signal (which Thurstone called 'latent variable') and provides a consensus ranking only as a byproduct.

Distributional approaches are based on the Thurstonian model, explained in Section 3.2.2. These methods are ideal for many short lists, i.e. a few objects and many assessors, as they yield poor estimates for small number of assessors. Mostly, they are applied in psychometrics and marketing (see, e.g., Green and Tull [1978], Conklin and Lipovetsky [1999], Lee et al. [2014], Selker et al. [2017]), and many of these methods are implemented in the Bayesian software JAGS (see section 2.1.3).

Heuristic approaches are deterministic in nature, intuitive and computationally simple. They are suitable for a few long lists, i.e. many objects and a few assessors, and have been effectively used for combining search engine results. An example of such a method is the classical voting Borda count method (de Borda [1781]), a rather simple approach based on the Borda score. For the i^{th} object, the Borda score BS_i is defined as

$$BS_i = \sum_{j=1}^{n} R_{ij}, \tag{3.1}$$

where R_{ij} is the rank assigned to object i by assessor j. The consensus ranking is obtained by ordering their Borda scores in an increasing manner. Other modifications have been proposed and are today addressed as *Borda's methods*. They have been, for example, used for elections in some countries (Saari [2001]). Less intuitive but more elegant alternatives to Borda's methods are *Markov chain methods* (variations of Dwork et al. [2001] and DeConde et al. [2006]), which use only pairwise ranking comparisons.

Examples of the heuristic method, specifically designed for combining results from search engines, are Google's PageRank algorithm (Page et al. [1999]), Supervised Rank Aggregation (Liu et al. [2007]), or Hyperlink-Induced Topic Search (HITS, Kleinberg [1999]). According to Lin [2010a], the performance of heuristic methods for biological applications is limited.

Stochastic optimisation approaches are based on some optimisation criterion, which aims to minimise the amount of disagreement between the input lists and the aggregate rankings. The disagreement is measured using some rank distance measure (e.g. Kendall's τ or Spearman's footrule distance - see Section 3.4). These methods are ideal for a few long lists, i.e. many objects and only a few assessors. Hence they are often used for combining long lists in biological experiments. For example, the Order Explicit Algorithm (OEA), suggested by Lin and Ding [2009], is a stochastic optimisation approach using the Cross Entropy Monte Carlo (CEMC) method (Rubinstein [1999]). OEA algorithm is implemented in the R package `TopKLists` (Schimek et al. [2015]), submodule `TopKSpace`. Pihur et al. [2007] also used the CEMC approach to aggregate rankings of p-values in microarray experiments and to combine and assess various clustering algorithms (Pihur et al. [2008]). Their approach is implemented in the R package `RankAggreg` (Pihur et al. [2009]).

3.3.2 Obtaining a list of top-k objects

A common problem with long lists of objects is determining the point where the agreement between two or more rankings degenerates into noise. This concerns long lists of hundreds or thousands of objects, for example gene expression measurements, search engine results, or customer products surveys. Having long noisy lists becomes an issue when applying methods which assume a certain level of agreement

between the lists, for example rank aggregation methods (Section 3.3.1). When such a method is applied on (partly-)noisy ranked lists, the results have likely no informative value. Hence it is important to truncate the lists and only apply the methods on a subset of so-called ***top-k*** objects. Typically, in these lists the agreement is prevalent only for a small subset of, say, the first 100 objects (hence $k = 100$), and the rankings are extremely noisy for the remaining, say, $10,000$ of objects.

Yang et al. [2006] addressed this problem specifically for estimating similarity between gene ranked lists and detecting the top genes with the most prominent ranks. Their approach is implemented in the Bioconductor package `OrderedList` (Lottaz et al. [2006]). Plaisier et al. [2010] suggested a method for estimating significant overlap between two gene expression signatures, as an alternative to choosing a fixed threshold. Their method, called Rank-Rank Hypergeometric Overlap (RRHO), is implemented in a Web application [1], as well as the Bioconductor package `RRHO` (Rosenblatt and Stein [2013]). Hall and Schimek [2012] solved the problem of finding the top-k for a general pair of very long ranked list, using a moderate-deviation-based approach. This method is implemented in the `R` package `TopKLists` (Schimek et al. [2015]), submodule `TopKInference`. In a conference paper (Švendová and Schimek [2013]) we compared the RRHO method of Plaisier et al. to TopKInference of Hall and Schimek, when applied to 5 simulated ranked lists. We showed that TopKInference yields constantly good results, even for high correlation between the noisy part of the lists, while RRHO estimates the correct k only with the assumption of low correlation between the noisy part of the lists. Sampath and Verducci [2013] suggested an alternative approach to Hall and Schimek [2012], called moving average maximum likelihood estimator (MAMLE), based on the multistage model for rankings (Fligner and Verducci [1988]). All mentioned methods can only handle pairs of lists. When applied to multiple lists, top-k is calculated for all pairwise combinations and the maximum k is usually taken as the result.

3.4 Correlation and distance measures

There are several methods that measure the strength of relationship between two variables. This section defines those most used and discusses their advantages. For more distance and correlation measures, see Marden [1995].

[1] http://systems.crump.ucla.edu/rankrank/

Correlation measures are fairly easy to interpret, because the values of correlation always range between $[-1, 1]$, where 1 signifies perfect correlation and -1 perfect anti-correlation. Distance measures between two variables have always the minimum at 0, meaning the two variables are identical, but the maximum, if possible to determine, differs for each method. Hence it is not straightforward to assess how 'small' or 'large' the measured distance is. However, when possible, one can normalise the measured distance by dividing it by its maximum value and obtain values that range between $[0, 1]$.

Note that, when using more than one correlation measure, one cannot always directly compare the absolute correlation values. Despite the values always ranging between -1 and 1, the absolute values can point to a different amount of association between the variables. For example, Spearman's correlation between vectors $(1, 2, 3)$ and $(1, 3, 2)$ is 0.5, while Kendall's τ rank correlation between the same vectors is 0.3.

Let us consider a pair of variables $X = (x_1, \ldots, x_p)$ and $Y = (y_1, \ldots, y_p)$, $x_i, y_i \in \mathbb{R}$, and define their respective rankings as $X_R = (r_{x_1}, \ldots, r_{x_p})$ and $Y_R = (r_{y_1}, \ldots, r_{y_p})$, $r_{x_i}, r_{y_i} \in \mathbb{N}, 1 \leq r_{x_i}, r_{y_i} \leq p, i = 1, \ldots, p$. Because in our algorithm in Chapter 4 we need to compare real vectors as well as vectors of rankings, correlation and distance measures for both, X, Y and X_R, Y_R, are introduced here.

- **Pearson's correlation** coefficient between the variables X and Y measures the linear relationship between the variables and is defined as

$$r(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{\sum_{i=1}^{p}(x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{p}(x_i - \bar{X})^2 \sum_{i=1}^{p}(y_i - \bar{Y})^2}}, \tag{3.2}$$

  where $\text{cov}(X, Y)$ is the covariance, and σ_X, σ_Y are the standard deviations of the variables, therefore $\bar{X}$ and $\bar{Y}$ are the sample means of X and Y. Pearson's correlation measures the linear relationship between the variables. In other words, if X changes proportionally with Y, the variables are correlated.

- **Spearman's ρ correlation** between two variables X and Y is equal to Pearson's correlation between the ranks of the variables:

$$\rho(X, Y) = \frac{\text{cov}(X_R, Y_R)}{\sigma_{X_R} \sigma_{Y_R}}, \tag{3.3}$$

  where $\text{cov}(X_R, Y_R)$ is the covariance of the rankings, and $\sigma_{X_R}, \sigma_{Y_R}$ are the

standard deviations of the rankings.

Spearman's correlation assesses monotonic relationship between the two variables, rather than linear, as in the case of Pearson's correlation. So if X tends to increase when Y increases, the correlation is positive. If X tends to decrease when Y increases, the correlation is negative. The difference from Pearson's correlation is that the increase or decrease does not have to be proportional.

- **Euclidean distance** between variables X and Y is the 'direct line distance', given by the Pythagorean theorem:

$$d_E(X,Y) = \sqrt{\sum_{i=1}^{p}(x_i - y_i)^2}. \tag{3.4}$$

- **Kendall's τ distance** between two ordinal variables X_R and Y_R is the number of discordant pairs:

$$\tau_d(X_R, Y_R) = \sum\sum\nolimits_{1\leq i<j\leq p} I[(r_{x_i} - r_{x_j})(r_{y_i} - r_{y_j}) < 0], \tag{3.5}$$

where $I(x) \in \{0, 1\}$ is the indicator function. Kendall's distance defines the number of interchanges needed to achieve the order Y_R.

- **Kendall's τ rank correlation** between two ordinal variables X_R and Y_R is the normalised Kendall's distance:

$$\tau(X_R, Y_R) = 2\frac{\tau_c(X_R, Y_R) - \tau_d(X_R, Y_R)}{p(p-1)}, \tag{3.6}$$

where $\tau_d(X_R, Y_R)$ is the number of discordant pairs, i.e. Kendall's distance (3.5), and $\tau_c(X_R, Y_R) = \sum\sum_{1\leq i<j\leq p} I[(r_{x_i} - r_{x_j})(r_{y_i} - r_{y_j}) > 0]$ is the number of concordant pairs. Being a correlation measure, the values of τ range between -1 (all pairs discordant) and 1 (all pairs concordant) and are usually smaller than Spearman's ρ. Kendall's τ correlation is more intuitive than Spearman's correlation, it has a defined standard error and it is less sensitive to discrepancies. It is important to point out that Kendall's correlation assumes statistical independence, i.e. considers the probability of pairwise swaps between positions in two lists to be constant. For example, the probability that an objects is ranked 1^{st} in list X and 10^{th} in list Y is the same as if it was ranked 1^{st} in

X and 2nd in Y, even if the latter is more likely.

- **Spearman's footrule distance** between two ordinal variables X_R and Y_R is defined as:

$$S_{\text{foot}}(X_R, Y_R) = \sum_{i=1}^{p} |r_{x_i} - r_{y_i}|. \tag{3.7}$$

Spearman's footrule distance gives higher penalty to a larger distance between the ranks and therefore, unlike Kendall's distance/correlation, also captures the size of disagreement. We can see that the value depends on the length p of the variables X_R, Y_R. In order to compare quality of results for data with unequal p, we need to normalise the footrule distance by dividing it by the maximum value $p^2/2$ (Dwork et al. [2001]):

$$N_{\text{foot}}(X_R, Y_R) = \frac{S_{\text{foot}}(X_R, Y_R)}{p^2/2}, \tag{3.8}$$

The values of N_{foot} then conveniently range between 0 and 1. From the definitions we can see that Spearman's footrule works with the actual ranks, while Kendall's τ only considers the relative ranks.

In the evaluation of our results in Chapter 4, we used Pearson's and Kendall's correlation, as well as Euclidean distance and raw and normalised Spearman's footrule distance, in order to evaluate our results. We could see, depending on the simulation scenarios, that some measures were more suitable than others.

3.5 Dealing with special cases in data structure

In some practical scenarios, only a subset of objects is ranked. For example, in surveys, participants are often asked to choose only their top favourite films, songs, drinks, etc. out of a larger selection. Such rankings are called *partial rankings*. In other applications, some ranks might be missing. For instance, in array-based gene expression experiments, researches have to specify the genes they want to investigate, and this can easily result in genes present in one study but missing in another, and vice versa. Partial rankings and rankings with missing ranks create a group of *incomplete rankings*. In other situations, two or more objects can be ranked on the same position, resulting in so-called *tied rankings*. These special cases are here described only briefly, because our model (Chapter 4) assumes complete rankings.

For many of the models mentioned in this chapter, there exists an altered model allowing these special ranking cases, without need for modifying the data (see Marden [1995] and Lin [2010b]).

3.5.1 Incomplete rankings

Incomplete rankings pose a problem to the traditional non-parametric rank-based statistics. If a small percentage of ranks is missing, one can remove the concerning objects (e.g. genes, drugs) or assessors (e.g. samples) from the analysis. This solution sacrifices possibly valuable information and can also introduce a bias, when the missingness is not completely at random (Little and Rubin [2014]). There are other solutions depending on whether we have access to the data that produced the rankings (e.g. expression values).

If we only have access to the ranking, a Monte Carlo Expectation-Maximisation (MCEM) algorithm can be used to deal with incomplete ranks. Given the observed incomplete rankings, one calculates conditional expectation of the complete rankings log-likelihood function (E-step). In the next step (M-step), the likelihood function is maximised with respect to the parameters of interest. These steps are repeated until a certain precision is reached. More details on this approach can be found in Alvo and Yu [2014], Chapter 9.

If we have access to the actual values that produced the rankings, we can impute the missing values, i.e. artificially create the missing information using the information that is available. Then we can rank the imputed data and obtain complete rankings for further analysis. There are many data imputation models and large amount of literature on this topic (see, e.g., Molenberghs et al. [2014]). Here we only briefly describe the model approaches. Imputation methods can be generally divided into three types:

1. Likelihood and Bayesian methods, which assume a parametric model for the complete data, together with a parametric model for the missing data mechanism.

2. Weighting methods, which weight the observed data in an appropriate way, depending on the data and problem type.

3. Multiple imputations, which repeat a chosen imputation method multiple times and create multiple complete datasets. Each dataset is then analysed

separately, and the resulting estimates and their standard errors are combined, according to Rubin's rules (Rubin [1987]). In particular, say we impute the data m times and obtain m estimates $\hat{\theta}^{(1)}, \ldots, \hat{\theta}^{(m)}$ of a parameter of interest θ. Then the combined estimate of θ is defined by Rubin's rules as

$$\hat{\theta}^{\mathrm{MI}} = \frac{1}{m} \sum_{k=1}^{m} \hat{\theta}^{(k)}. \tag{3.9}$$

Let us denote the variance of estimates $1, \ldots, m$ as $V^{(1)}, \ldots, V^{(m)}$. Then the variance of $\hat{\theta}^{\mathrm{MI}}$ is defined by Rubin's rules as

$$\mathrm{V}^{\mathrm{MI}} = \bar{\mathrm{V}} + (1 + \frac{1}{m})B, \tag{3.10}$$

where $\bar{\mathrm{V}} = \sum_{k=1}^{m} \mathrm{V}^{(k)}/m$ is the within-imputation variability, and $B = \sum_{k=1}^{m} (\hat{\theta}^{(k)} - \hat{\theta}^{\mathrm{MI}})^2/(m-1)$ is a parameter correcting for the missing data uncertainty, i.e. between-imputation variability. In order to make the final results reproducible, it was suggested (White et al. [2011]) that the number of imputations should be at least as great at the percentage of incomplete cases.

The first two groups of methods impute the missing values only once, hence are sometimes called 'single imputation methods'. These are prone to problems, such as overly precise estimates or biases due to ignoring the uncertainty of the imputations. Therefore multiple imputation approaches are preferred in practice (Molenberghs et al. [2014]).

In section 5.2, we applied multiple imputation in order to replace missing values in clinical trials.

3.5.2 Tied rankings

Ties in a ranking occur when an assessor cannot distinguish between two objects and hence ranks them at the same position. With human assessors, this can happen, for example, when a person cannot decide which of the two films he/she likes better. With machine assessors, this may occur, for example, when the calculated gene expression for two genes is identical, hence they share a rank position.

Formally, a tied ranking is represented by a p vector of a integers, $a < p$, where each integer appears at least once. For example, imagine we have 5 genes A,B,C,D,E and the following expression values associated with them: $0.1, 0.3, 0.1, 0.5, 0.1$, re-

spectively. If we were to rank the genes from the most to the least expressed, D would be 1st, B 2nd, and A, C and E would share the last 3 positions, i.e. they are partially ordered as (D, B, {A,C,E}). There are 3! rankings of genes A,B,C,D and E compatible with the above partial ordering: $(3, 2, 4, 1, 5), (4, 2, 1, 3, 5), (5, 2, 3, 1, 4)$, $(5, 2, 4, 1, 3), (3, 2, 5, 1, 4), (4, 2, 5, 1, 3)$.

When combining several rankings containing tied ranks, whether it is to find a consensus ranking or a list of top-k objects, one has generally three options

1. Arbitrarily order the tied objects by assigning them a random ranking, i.e. in the above mentioned example we could randomly choose one of the 3! rankings, e.g. $(4, 2, 1, 3, 5)$. Another option is to assign them ranks depending on the input order of the tied objects. So because in the group {**A,C,E**} A comes before C and C comes before E, ranking from the first to the last tied would be $(\mathbf{3}, 2, \mathbf{4}, 1, \mathbf{5})$, and from the last to the first tied $(\mathbf{5}, 2, \mathbf{4}, 1, \mathbf{3})$.

2. Assign the average ranking to the tied objects, i.e. the middle rank position, or the maximum or the minimum ranking. In the above example the average ranking would be $(\mathbf{4}, 2, \mathbf{4}, 1, \mathbf{4})$, maximum ranking would be $(\mathbf{5}, 2, \mathbf{5}, 1, \mathbf{5})$, and minimum ranking $(\mathbf{3}, 2, \mathbf{3}, 1, \mathbf{3})$. Note, that the average ranking is not an integer when there is odd number of tied objects.

3. Use an algorithm that adopts a special distance metric for partial rankings. Versions of Kendall's τ distance as well as Spearman's footrule distance can be found in Kendall [1945], Baggerly [1995], Fagin et al. [2006], or Critchlow [2012].

Our model (Chapter 4) does not use any special distance metric in order to handle ties, hence the concerned ranks have to be defined according to 1. or 2. above. By definition, repeated and non-integer ranks can be used in our model. However, investigating the effect of various methods for replacing tied rankings is beyond the scope of this thesis.

3.6 Discussion

In this chapter, the topic of ranking data was introduced and various modelling approaches described. As shown, human or machine 'rankers' ordering various objects appear across fields and research areas. With multiple rankers, the problem

of inferring information from ranking data falls into the category of meta-analysis. Combining the rankings is of special importance in areas where no other informative measures exist. Such areas include preference surveys, voting, or internet search. In other situations, analysing rankings of values instead of the values themselves might be a way of overcoming the problems with measurements, such as incomparability, or the presence of outliers.

Depending on what type of knowledge is desired, various methods for combining and inferring from ranking data can be used. Some methods aim to simply describe the structure of the data, other, more ambitious methods, intend to derive information about the rankers, objects, or the ranking process itself.

The elegance of using ranking data has been recognised in practically every research field, and with increasing data collection in public databases, new methods for analysing such data continue to appear.

Chapter 4

Multiple ranking model

This chapter is based on the paper by Švendová and Schimek [2017]. Our novel model for estimating the underlying signal from multiple ranked lists is proposed and tested on simulated data.

4.1 Introduction

Meta-analysis serves as a tool for combining existing results of various numerical data types (Chapter 2). One of these types are ranking data (Chapter 3). Ranking data are scale independent and therefore various types of results can be combined together without any normalisation procedures. There are scenarios where no other numerical information is available, for example in surveys where customers are asked to rank certain products. In such situations, there is no dispute that ranking data must be used for such an analysis. In other scenarios, different types of numerical information might be available, for example p-values, effect sizes, or gene expressions. Here, one can use methods that are designed to combine a specific data type, e.g. p-values, but cannot combine data across different data types, e.g. p-values with gene expressions. In situations where combining across data types is desirable, one can rank the objects according to the values and obtain a general scale-free dataset of rankings. A number of methods for combining ranking data have been suggested, mainly in the context of data aggregation (Section 3.3.1) and search for top-k objects (Section 3.3.2).

Rank-based methods have the advantage of being invariant to transformation and normalisation as long as the relative orderings are preserved. In addition, they

are robust to outliers, although some information is inevitably lost compared to metric approaches. For studies that comprise different data types but have some commonality, rank-based methods offer the opportunity to integrate individual results in order to arrive at some consensus that is more conclusive than any of the individual studies.

Take for example a situation where several lists of genes are ranked by their expression. Most existing rank aggregation methods produce an aggregated ranking of these genes (or a top-k subset of them), from the most to the least expressed. However, the information about the extend to which a gene is expressed compared to that ranked below it, is missing. Is the gene which is ranked first almost as equally expressed as the gene ranked second? Or does their expression differ dramatically, but in this case there happened to be no other genes between them? These are questions that classical rank aggregation methods cannot answer. We addressed these questions and propose a method that aims to reconstruct the true structure of the data.

Our model falls into the category of Thurstonian models (see Section 3.2), as it aims for modelling of the ranking process by assuming there is an underlying true variable, such as the 'true gene expression'. We figuratively construct the model as follows: we consider a number of assessors, which can either represent actual human judges, machines, or independent studies. The assessors aim for the one objective measure, namely the underlying true variable. With this goal in mind, they construct what Thurstone called *mental scores*, which are a form of evaluation of quality of each object. Because we are not restricted to psychological applications like Thurstone, we call these scores by the general term *attributes* and define them as any kind of numerical evaluation of the objects, either scores that are only mentally conceived but never numerically expressed, or actual measurements (e.g. gene expression or efficacy). Unconsciously, while evaluating, each assessor produces an error, whether it is a measurement error, or an error caused by a lack of knowledge. These errors result in a certain level of disagreement between the assessors and can create problematic contradictions between studies. We resolve the contradictions by combining the studies in a meta analysis.

The method suggested here provides:

1. Estimates of the underlying true signals from multiple ranked lists, under the assumption that the involved objects are informative, in other words there is

a high concordance in their rank positions.

2. Standard errors of the signal estimates, using a non-parametric bootstrap.
3. A consensus ranking derived from the signal estimates, that is, one without applying any data aggregation technique.
4. Stability assessment of the derived consensus ranking.

This chapter is structured as follows. Section 4.2 introduces the statistical model for estimation of the underlying signal. Section 4.3 describes numerical methods for the evaluation of the results, including standard error estimation and rank stability assessment. Pseudocode and implementation details are provided in Section 4.4. Simulations and their results are described in Sections 4.5 and 4.6. Section 4.7 discusses the results and usage of the model. Possible improvements and generalisations are also suggested.

4.2 The Method

In this section we suggest a stochastic model, which describes the relationship between the underlying true signal and an observed ranking matrix. We propose two iterative procedures (one frequentist and one Bayesian) for the signal parameters estimation. Both procedures involve calculation of a joint multivariate distribution of the observed rankings and a specific objective function that guarantees convergence to the true signal parameters. This section is an extended version of the text that appears in Švendová and Schimek [2017].

4.2.1 The statistical model

Let us consider a group of n assessors (humans or machines) assessing p objects. We assume that the j^{th} assessor either implicitly or explicitly observed random variables $X_{1j}, \ldots, X_{pj}$, which we shall call the *attributes*. In some but not all situations, these attributes can be observed or measured, for example gene expression or response rate. In this work, attributes are understood as a theoretical construct. The variable X_{ij} denotes the value of the attribute for the i^{th} object as seen by the j^{th} assessor. The order of these variables, say

$$X_{\sigma_{(1)j}} > \ldots > X_{\sigma_{(p)j}}, \tag{4.1}$$

defines the rankings that the j^{th} assessor gave to the objects. Let $R_{1j}, \ldots, R_{pj}$, where $R_{ij} \in \mathbb{N}, R_{ij} \leq p$, denote the rank of $X_{1j}, \ldots, X_{pj}$ according to (4.1). Then we say that $\{R_{ij}\}$ is the column *rank matrix* of the attributes $\{X_{ij}\}$. Ties can be handled by assigning random or average ranks, for example.

Furthermore, we suppose, that the values $X_{1j}, \ldots, X_{pj}$ follow the model

$$X_{ij} = \theta_i + Z_{ij}, \qquad i = 1, \ldots, p; j = 1, \ldots, n, \tag{4.2}$$

where the θ_i are real-valued parameters and the Z_{ij} are random variables. Let us write $\mathbf{X} = \{X_{ij}\}$, $\theta = \{\theta_i\}$ and $\mathbf{Z} = \{Z_{ij}\}$. The parameters θ represent the 'true values' that all assessors try to rank, but due to the random errors $\mathbf{Z}$, their best approximation of the 'true values' are the attributes $\mathbf{X}$. A specific error distribution is not required for parameter estimation but normality is assumed for the simulations in this thesis. We assume that the θ's are shared by all assessors.

Because the scale of the values in $\mathbf{X}$ strongly depends on the data type (e.g. gene expression measurements are positive or negative real numbers, while count reads are positive integers), we resort to normalised values of θ. A normalisation is also required for the purpose of identifiability of these parameters in the process of estimation. We call the vector of the parameters the *underlying signal*, and denote it by $\tilde{\theta} = \tilde{\theta}_1, \ldots, \tilde{\theta}_p$,

$$\tilde{\theta} = \frac{\theta}{\|\theta\|}, \quad \text{where } \|\theta\| = \sqrt{\sum_{i=1}^{p} \theta_i^2}. \tag{4.3}$$

Let vector $r^\theta = (r_1, \ldots, r_p)$, where $r_i \in 1, \ldots, p$ represents the rank of θ_i (and hence of $\tilde{\theta}_i$). We call r^θ the *true ranking*.

The underlying signal $\tilde{\theta}$ can be understood as a measure of relative distance between the values of θ. This measure is interesting, because it is directly responsible for the stability of the observed rankings: small distance between $\tilde{\theta}_i$ and $\tilde{\theta}_{i+1}$ for an arbitrary object $i \in \{1, \ldots, p\}$ implies that the i^{th} and $(i+1)^{\text{th}}$ objects are very similar, and hence only a small perturbation is needed to swap their rank positions. On a scale between 0 and 1, for example, if $\tilde{\theta}_1 = 0.5$ and $\tilde{\theta}_2 = 0.6$, the objects 1 and 2 are much more similar than if $\tilde{\theta}_1 = 0.5$ and $\tilde{\theta}_2 = 1$. In both cases, object 2 will be ranked above object 1, but in the first case this rank order is much less stable than in the second case.

We do not make any assumptions about the distribution of θ or $\tilde{\theta}$. In practice,

exponential or normal distribution can often be expected, but this should not be assumed automatically (see e.g. Hardin and Wilson [2009] for gene expression data). Where we have prior knowledge about the distribution, it can be used for constructing an initial guess in our estimation procedure, however the statistical model itself does not pose any restrictions on θ.

Our main aim is to estimate the normalised vector of signals $\tilde{\theta} = (\tilde{\theta}_1, \ldots, \tilde{\theta}_p)$ (Equations 4.2, 4.3). In cases where the scale differs for each assessor, or where the attributes X_{ij} are non-existent, we have to refer ourselves to the rank representation of the attributes $\mathbf{X}$, i.e. its column rank matrix. We will use the notation $\mathbf{R}(\theta)$ for the column rank matrix of $\mathbf{X}$ from the model 4.2.

Let us stress that our algorithm only uses the ranks $\mathbf{R}(\theta)$ for the estimation of $\tilde{\theta}$. The values in $\mathbf{X}$ serve as a theoretical construct of the underlying signal, but they are not required for any calculations.

4.2.2 The distribution of ranks

In order to recover the underlying signal $\tilde{\theta}$ from the input rank matrix $\mathbf{R}(\theta)$, we first need to evaluate the distribution of ranks in $\mathbf{R}(\theta)$. Once having this information, we can compare the rank distribution in $\mathbf{R}(\theta)$ to the rank distribution in $\mathbf{R}(\hat{\tilde{\theta}})$, where $\hat{\tilde{\theta}}$ is an estimate of $\tilde{\theta}$. By comparing these two rank distributions, we indirectly compare the true $\tilde{\theta}$ to the estimate $\hat{\tilde{\theta}}$. In an iterative manner, we can then improve the estimate until a desired precision is reached.

Following this idea, we now construct a probability distribution of the rank positions in an arbitrary rank matrix $\mathbf{R}(\cdot)$, where the dot represents an arbitrary vector of p real values. Recall that the dimensionality of $\mathbf{R}(\cdot)$ is $p \times n$, where p stands for the number of objects, and n the number of assessors. Each column of $\mathbf{R}(\cdot) = \{R_{ij}\}$ contains rank positions of p objects, say objects $o_1, \ldots, o_p$. Let

$$\mathcal{P}_{\ell_{\max}} = \{(s^{(1)}, \ldots, s^{(\ell)}) \in \mathbb{N}^{\ell} | s^{(i)}_{i=1,\ldots,\ell} \leq p, \ell = 1, \ldots, \ell_{\max}\} \tag{4.4}$$

be a set of integer vectors of lengths 1 to $\ell_{\max}$, where $\ell_{\max}$ is an integer. The number of elements of $\mathcal{P}_{\ell_{\max}}$ is $\sum_{\ell=1}^{\ell_{\max}} p^{\ell}$. For example, $\mathcal{P}_1 = \{1, \ldots, p\}$ has p elements, $\mathcal{P}_2 = \mathcal{P}_1 \cup \{(s^{(1)}, s^{(2)}) \in \mathbb{N} \times \mathbb{N} | s^{(1)}, s^{(2)} \leq p\}$ has $p + p^2$ elements, $\mathcal{P}_3 = \mathcal{P}_1 \cup \mathcal{P}_2 \cup \{(s^{(1)}, s^{(2)}, s^{(2)}) \in \mathbb{N}^3 | s^{(1)}, s^{(2)}, s^{(3)} \leq p\}$ has $p + p^2 + p^3$ elements, and so on for higher values of $\ell_{\max}$. The set of all elements of $\mathcal{P}_{\ell_{\max}}$ represents all possible sub–rankings

of lengths $\ell = 1, \ldots, \ell_{\max}$, including tied rankings. We compare them to all sub–rankings of the same lengths from $\mathbf{R}(\cdot)$ and decide about their probabilities on each position (Equations 4.5 and 4.6). The coefficient ℓ defines the number of objects considered in each sub–ranking comparison. Increasing $\ell_{\max}$ means examining more sub-ranking possibilities, which leads to an increased precision but also substantially increased computational demand. Our simulations suggested that $\ell_{\max} = 2$ is a satisfactory compromise.

In order to compare each element of $\mathcal{P}_{\ell_{\max}}$, say $s_\ell, \ell = 1, \ldots, \ell_{\max}$, with each subset of ranks in $\mathbf{R}(\cdot)$, we have to systematically examine all ℓ-sub-rankings across all n assessors. Having this goal in mind, we can think of an imaginary window of width n and height ℓ that slides along the matrix $\mathbf{R}(\cdot)$ from the top to the bottom for each ℓ (see an example on Figure 4.1). At each location $k \in \{0, \ldots, p - \ell\}$, we calculate the probability of objects $o_{k+1}, \ldots, o_{k+\ell}$ having rank positions smaller or equal to $s_\ell = (s^{(1)}, \ldots, s^{(\ell)})$, for all $s_\ell \in \mathcal{P}_{\ell_{\max}}$, $\ell = 1, \ldots, \ell_{\max}$:

$$F(o_{k+1}, \ldots, o_{k+\ell}|s_\ell) = P(R_{k+1} \leq s^{(1)} \wedge \ldots \wedge R_{k+\ell} \leq s^{(\ell)}), \tag{4.5}$$

where $R_{k+1}, \ldots, R_{k+\ell}$ denote a generic sequence of ranks $R_{k+1,j}, \ldots, R_{k+\ell,j}$ of the j^{th} assessor.

$$R = \begin{array}{c} \\ o_1 \\ o_2 \\ o_3 \\ o_4 \end{array} \begin{array}{c} \begin{array}{ccc} A_1 & A_2 & A_3 \end{array} \\ \left(\begin{array}{ccc} 1 & 1 & 2 \\ 3 & 2 & 1 \\ 2 & 4 & 3 \\ 4 & 3 & 4 \end{array} \right) \end{array} \begin{array}{l} \\ \text{k=0} \\ \text{k=1} \\ \\ \\ \end{array}$$

Figure 4.1: The role of k for $\ell = 2$ in formulas 4.5 and 4.6. Increasing k slides the window of width $n = 3$ and height $\ell = 2$ from the top to the bottom of the rank matrix. Figure from Švendová and Schimek [2017].

We can obtain an estimate of $F(o_{k+1}, \ldots, o_{k+\ell}|s_\ell)$ as

$$\begin{aligned} &\hat{F}(o_{k+1}, \ldots, o_{k+\ell}|s_\ell) = \\ &\qquad \frac{1}{n} \sum_{j=1}^{n} I(R_{k+1,j} \leq s^{(1)} \wedge \ldots \wedge R_{k+\ell,j} \leq s^{(\ell)}), \end{aligned} \tag{4.6}$$

where I is the standard indicator function resulting in 1 or 0, and $s_\ell \in \mathcal{P}_{\ell_{\max}}$.

We calculate the probability 4.6 for all combinations of $s_\ell \in \mathcal{P}_{\ell_{\max}}$, $\ell = 1, \ldots, \ell_{\max}$ and $k = 0, \ldots, p - \ell$. For each $\ell = 1, \ldots, \ell_{\max}$, we save the calculated probabilities

in a $(p-\ell+1)\times p^{\ell}$ probability matrix $\mathcal{F}_{\ell}(\mathbf{R})$:

$$\mathcal{F}_{\ell}(\mathbf{R}) = \begin{pmatrix} \hat{F}(o_1,\ldots,o_{\ell}|s_{\ell 1}) & \ldots & \hat{F}(o_1,\ldots,o_{\ell}|s_{\ell p^{\ell}}) \\ \vdots & \ddots & \vdots \\ \hat{F}(o_{p-\ell+1},\ldots,o_p|s_{\ell 1}) & \ldots & \hat{F}(o_{p-\ell+1},\ldots,o_p|s_{\ell p^{\ell}}) \end{pmatrix}, \tag{4.7}$$

where $s_{\ell c} \in \mathcal{P}_{\ell_{\max}}, c = 1,\ldots,p^{\ell}$ is the c^{th} element in $\mathcal{P}_{\ell_{\max}}$ of length ℓ. So, for example for $p=3$, if $\ell=1$ then $s_{\ell 1},\ldots,s_{\ell p^{\ell}}$ are integers $s_{11}=1, s_{12}=2, s_{13}=3$; if $\ell=2$ then $s_{\ell 1},\ldots,s_{\ell p^{\ell}}$ become vectors of integers $s_{21}=(1,1), s_{22}=(1,2), s_{23}=(2,1), s_{24}=(2,2), s_{25}=(1,3), s_{26}=(3,1), s_{27}=(3,3), s_{28}=(3,2), s_{29}=(2,3)$ (the order does not matter as long as it is consistent for all elements of 4.7).

This means that, for example, for $\ell_{\max}=2$ we have to calculate two matrices, $\mathcal{F}_1(\mathbf{R})$ and $\mathcal{F}_2(\mathbf{R})$. We can write $\mathcal{P}_2 = \{s_{11},\ldots,s_{1p},s_{21},s_{22},\ldots,s_{2p^2}\}$, and then for $\ell=1$:

$$\mathcal{F}_1(\mathbf{R}) = \begin{pmatrix} \hat{F}(o_1|1) & \ldots & \hat{F}(o_1|p) \\ \vdots & \ddots & \vdots \\ \hat{F}(o_p|1) & \ldots & \hat{F}(o_p|p) \end{pmatrix}, \tag{4.8}$$

and for $\ell=2$:

$$\mathcal{F}_2(\mathbf{R}) = \begin{pmatrix} \hat{F}(o_1,o_2|s_{21}) & \ldots & \hat{F}(o_1,o_2|s_{2p^2}) \\ \vdots & \ddots & \vdots \\ \hat{F}(o_{p-1},o_p|s_{21}) & \ldots & \hat{F}(o_{p-1},o_p|s_{2p^2}) \end{pmatrix}. \tag{4.9}$$

To illustrate the probability estimation, let us consider a simple example input rank matrix $\mathbf{R}$ with 3 assessors A_1 to A_3 and 4 objects o_1 to o_4:

$$\mathbf{R} = \begin{array}{c} \\ o_1 \\ o_2 \\ o_3 \\ o_4 \end{array} \begin{array}{c} \begin{array}{ccc} A_1 & A_2 & A_3 \end{array} \\ \begin{pmatrix} 1 & 1 & 2 \\ 3 & 2 & 1 \\ 2 & 4 & 3 \\ 4 & 3 & 4 \end{pmatrix} \end{array}.$$

Then, for instance, the probability that object o_2 is ranked 3^{rd} or higher (i.e. $\ell =$

$1, s_{1\cdot} = 3)$ is

$$\hat{F}(o_2|3) = \frac{1}{3}[I(3 \leq 3) + I(2 \leq 3) + I(1 \leq 3)] = 1. \tag{4.10}$$

The probability that object o_1 is ranked 1^{st} or higher, and at the same time object o_2 is ranked 3^{rd} or higher (i.e. $\ell = 2, s_{2\cdot} = (1, 3)$) is

$$\begin{aligned} \hat{F}(o_1,o_2|(1,3)) = \frac{1}{3}[&I(1 \leq 1 \wedge 3 \leq 3)+ \\ &+ I(1 \leq 1 \wedge 2 \leq 3) + I(2 \leq 1 \wedge 1 \leq 3)] = \frac{2}{3}. \end{aligned} \tag{4.11}$$

Equivalently, such probabilities are calculated for all possible combinations of ranks and objects, as defined by formula 4.7.

In summary, the combination of the statistical model 4.2 and the rank distribution function 4.5, allowed us to adopt an indirect inference approach in the spirit of Gourieroux et al. [1993]. *Indirect inference* is understood as a simulation-based method for parameter estimation (or inferences about these parameters) under an auxiliary model which does not need to be an accurate description of the data generating process. As we do not know enough about the decision process that is responsible for the observed rank order, indirect inference is an adequate strategy. We resort to it because our estimation problem is too complex to apply standard methodology such as maximum likelihood. To be more precise, we cannot define a target function. Thus a simulation-based approach, at the expense of heavy computing, is the only feasible alternative. In our case, the parameters to be estimated are the signal values $\tilde{\theta}_i, i = 1, \ldots, p$, and the auxiliary model is a general model $\hat{\mathbf{X}} = \hat{\tilde{\theta}} + \hat{\mathbf{Z}}$ (analogous to 4.2). We iteratively improve the model by updating $\hat{\tilde{\theta}}$ and $\hat{\mathbf{Z}}$, until $\hat{\tilde{\theta}}$ is 'close enough' to $\tilde{\theta}$. The proximity of $\hat{\tilde{\theta}}$ and $\tilde{\theta}$ is measured by comparing $\mathcal{F}_\ell(\mathbf{R}(\hat{\tilde{\theta}}))$ and $\mathcal{F}_\ell(\mathbf{R}(\theta))$ (Equation 4.7), over $\ell = 1, \ldots \ell_{\max}$. The details on the iterative process are described in Section 4.2.3.

4.2.2.1 Behavior of the rank distribution

Let us now illustrate how the distribution of rank positions changes when we slide along the observed rank matrix (i.e. when we increase k), and how these changes depend on the level of agreement between the assessors. Imagine, for example, that we observe a rank matrix $\mathbf{R}_{40\times 20}$, i.e. the rankings of $p = 40$ objects $o_1, \ldots, o_{40}$ and

$n = 20$ assessors. Say we consider the maximum window height $\ell_{\max} = 2$, hence we have to construct a set $\mathcal{P}_2 = \{1, \ldots, 40\} \cup \{(s^{(1)}, s^{(2)}) \in \mathbb{N} \times \mathbb{N} | s^{(1)}, s^{(2)} \leq 40\}$. Then we can calculate the probabilities of rank distribution (Equation 4.7), i.e. for $\ell = 1$

$$\mathcal{F}_1(\mathbf{R}) = \begin{pmatrix} \hat{F}(o_1|s_{1_1}) & \ldots & \hat{F}(o_1|s_{1_{40}}) \\ \vdots & \ddots & \vdots \\ \hat{F}(o_{40}|s_{1_1}) & \ldots & \hat{F}(o_{40}|s_{1_{40}}) \end{pmatrix}_{40 \times 40}, \tag{4.12}$$

and for $\ell = 2$

$$\mathcal{F}_2(\mathbf{R}) = \begin{pmatrix} \hat{F}(o_1, o_2|s_{2_1}) & \ldots & \hat{F}(o_1, o_2|s_{2_{1600}}) \\ \vdots & \ddots & \vdots \\ \hat{F}(o_{39}, o_{40}|s_{2_1}) & \ldots & \hat{F}(o_{39}, o_{40}|s_{2_{1600}}) \end{pmatrix}_{39 \times 1600}. \tag{4.13}$$

Notice, that for $\ell = 1$, the above matrix has 40 columns, for $\ell = 2$ it is 1600 columns, and for $\ell = 3$ it would quickly climb to 64000 columns, which explains the computational burden with higher values of $\ell_{\max}$.

To demonstrate the behavior of the rank distribution, let us simulate the underlying signal $\tilde{\theta}$ so that the true ranking is $r^{\theta} = (1, \ldots, 40)$, i.e. object o_1 has rank 1, o_2 has rank 2, and so on.

Figure 4.2 shows how the probabilities in Equation 4.13 change when sliding an imaginary window of height $\ell = 2$ located at k. Each subplot represents the $(k+1)^{\text{th}}$ row of the matrix in 4.13. $s^{(1)}$ and $s^{(2)}$ are integers representing all possible rank positions (elements of $s_2 \in \mathcal{P}_2$).

Let `tau` be the median calculated from Kendall correlations between the columns of matrix $\mathbf{R}$. We can observe (Figure 4.2) that when highly agreeing assessors (`tau` $= 0.9$, first column) rank the objects, the probabilities are almost all equal or close to 1 for $k = 0$. When sliding down along the lists, i.e. increasing k, the number of high probabilities decreases and the large values concentrate in the top right corner, which represents the low rank positions close to 40. This behavior for high rank correlation is logical, because $\tilde{\theta}$ in this example is simulated so that the object o_1 has rank 1, object o_2 has rank 2, object o_3 has rank 3, and so on. Therefore, when computing the indicator function $I(R_{k+1,j} \leq s^{(1)} \wedge R_{k+2,j} \leq s^{(2)})$ in 4.6, we obtain the value 1 for most of the top windows of the ranked lists, whilst when sliding down to the bottom of the lists, zero values dominate.

For the extreme case of no correlation between the assessors (`tau` $= -0.01$, shown

in the third column of Figure 4.2), there is no concordance whatsoever, causing the indicator function to produce predominantly zeros, independently of the position of the sliding window. Values close to one are observed only at the top right corner, simply because there the algorithm compares assessors' ranks to the highest possible integers $s^{(1)}, s^{(2)}$, and the indicator function is more likely to produce ones. For a low rank correlation ($\texttt{tau} = 0.31$, shown in the second column in Figure 4.2), the output is between the two extremes. Note, that a low rank correlation is shown here merely to illustrate the behaviour of the distribution function. For multiple assessments in real world data, rank correlation above 0.5 can generally be expected.

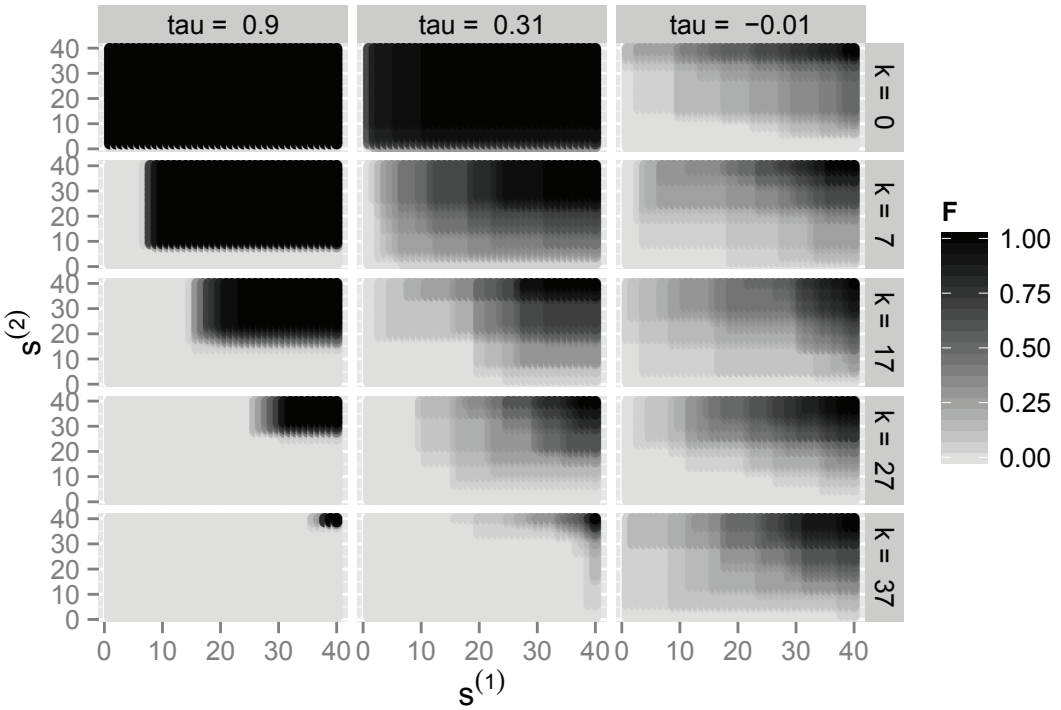


Figure 4.2: Visualisation of the rank probabilities for 20 assessors ranking 40 objects, with $\ell = 2$. Each of the 12 rectangles represents the probability values for all possible pairs of integers $s^{(1)}, s^{(2)}$ calculated in an imaginary window located at k. The values range between zero (light grey) and one (black). The 3 columns represent 3 different input rank matrices 40×20 having 3 different medians of Kendall's τ correlation between the assessments (1st column highly correlated, 2nd column lowly correlated, and 3rd column uncorrelated ranks).

4.2.3 Parameter optimisation

In order to estimate the underlying signal using the indirect inference procedure, one has to choose an optimisation technique. We applied a frequentist optimisation with a genetic algorithm, as well as a Bayesian random-walk Metropolis-Hastings Markov chain Monte Carlo (MCMC) procedure, and compared the results. The crucial part of the calculations is the choice of an objective function, which determines how close is an estimate to the true signal. Our goal is then naturally to minimise the objective function. We define an objective function for our algorithm in Section 4.2.3.1.

The pseudocode for both optimisation methods can be found in Section 4.4 (Algorithms 1 and 2). The R code for all involved functions in both algorithms is provided in the Appendix. The R package `GA` (Scrucca [2013]) was used for the genetic algorithm optimisation. The MCMC technique was programmed in R as part of this thesis. Here we shortly describe the two techniques and the involved algorithmic steps.

Genetic algorithms (GA) (Holland [1975], Goldberg and Holland [1988]) are optimisation methods inspired by the evolution principles. In each iteration, a *population* is composed of a number of *individuals* (vector estimates), which then undergo the process of *selection* (keeping the 'best' vectors only), *crossover* (swapping elements of the vectors) and *mutation* (replacing the elements of the vectors). The fitness of the individuals is evaluated in each iteration using a *fitness function*, which we call the objective function. The algorithm stops when either the desired maximum fitness, or the maximum number of iterations is reached. For an illustrative summary of the algorithm within our method, see the flow-chart on Figure 4.3.

Metropolis-Hastings MCMC (Metropolis et al. [1953], Hastings [1970], Liu [2008]) is a technique for random sampling from a probability distribution, especially useful for multi-dimensional distributions where direct sampling might be difficult. This is the case of our optimisation problem, where we try to find a minimum on a p-dimensional space. In each step, a sample candidate is suggested based on the previous sample, evaluated with an objective function, and either accepted as a new sample, or rejected. The algorithm stops when either a desired minimum of the objective function or the maximum number of steps is achieved. For an illustrative summary of the algorithm within our method, see the flow-chart on Figure 4.4.

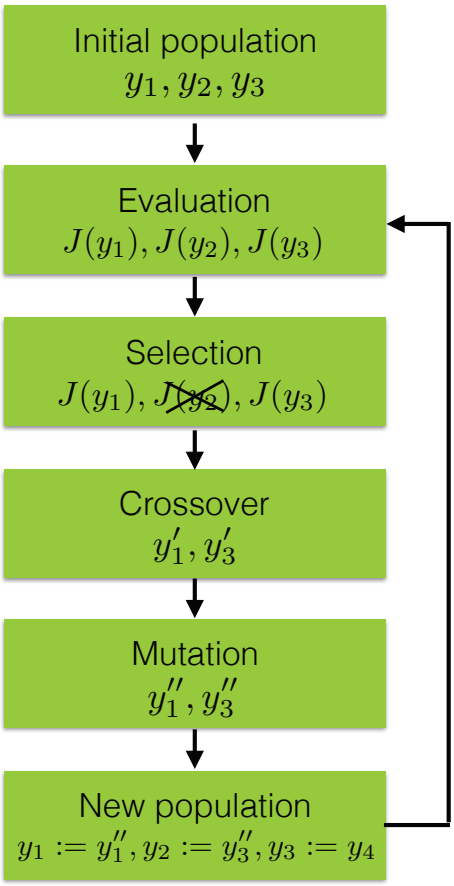


Figure 4.3: Genetic algorithm. y_i's denote the vector estimates (individuals in a population), and J denotes the fitness or objective function.

In order to check the convergence of the MCMC algorithm, we used the Gelman-Rubin's convergence diagnostic (Gelman and Rubin [1992]). Because our estimated signals are multivariate, we used the diagnostic on each object's point estimate separately, as recommended by the authors. In particular, we calculated the potential scale reduction factor (Gelman and Rubin [1992], Section 2.2), using the R package `coda`. For factors close to 1, we could be confident of converging to the target distribution. Note, that the Gelman-Rubin's diagnostic assumes normal target distribution. If there is a reason to think otherwise, one should apply a version that does not assume normality (e.g. Brooks and Gelman [1998], Sections 3 and 4).

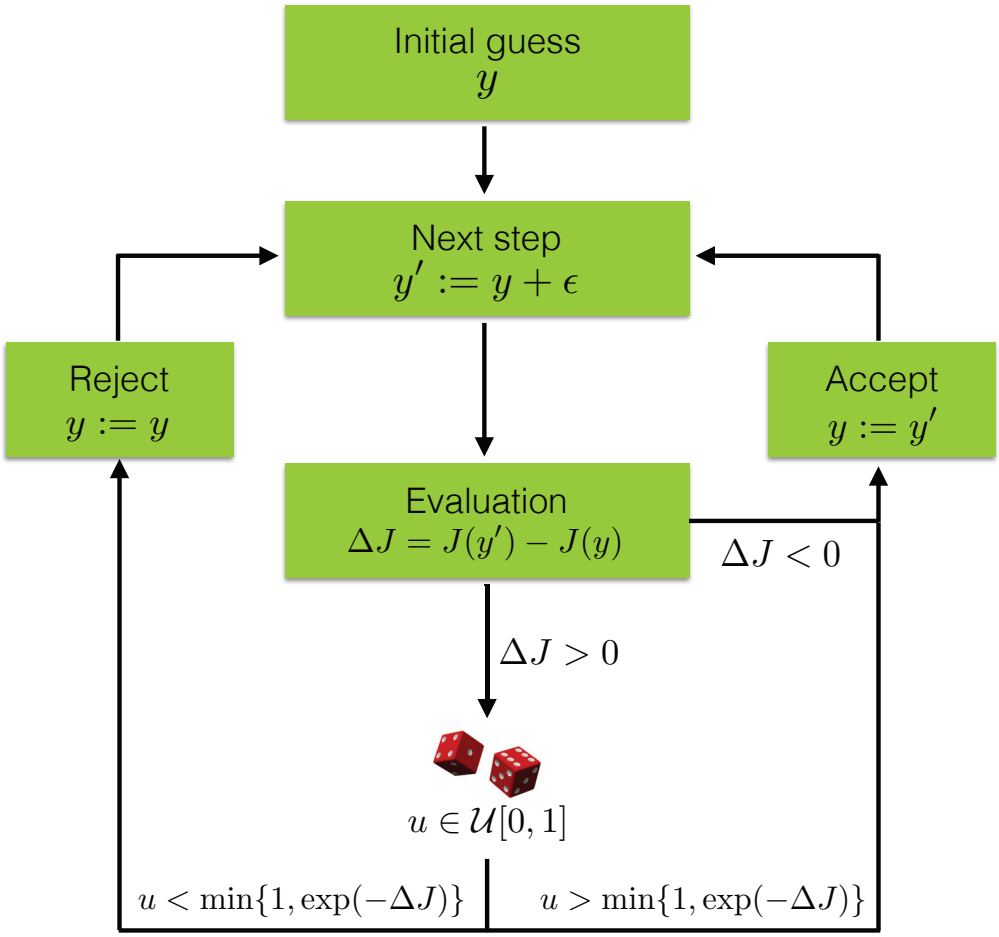


Figure 4.4: Metropolis-Hastings MCMC algorithm. y denotes a vector estimate, and J denotes the objective function.

4.2.3.1 Objective function

Both, the MCMC and the GA, optimisation procedures require an objective function $J(\cdot)$ that guarantees the convergence to the underlying signal $\tilde{\theta}$. In our situation, the objective function $J(\mathbf{y})$ should express the distance between $\tilde{\theta}$ and a vector $\mathbf{y}$. We assume that an arbitrary vector $\mathbf{y} = (y_1, \ldots, y_p)$ follows the model $\hat{X} = \mathbf{y} + \hat{Z}$, where $\hat{Z} = \{\hat{Z}_{ij} \sim \mathcal{N}(0, \sigma_j^2), i = 1, \ldots, p, j = 1, \ldots, n\}$ is a matrix of random variables. $\hat{X}$ is the matrix of attributes based on the underlying $\mathbf{y}$, and $\mathbf{R}(\mathbf{y})$ is the column rank matrix of $\hat{X}$. According to 4.7, we can calculate $\mathcal{F}_\ell(\mathbf{R}(\mathbf{y}))$, i.e. the probability matrix of $\mathbf{R}(\mathbf{y})$ for a window of height ℓ, for each $\ell = 1, \ldots, \ell_{\max}$.

Let us assume that $\mathbf{y}$ represents an estimate of $\tilde{\theta}$. We define the objective function

as $J : \mathbb{R}^p \to \mathbb{R}_0^+$

$$J(\mathbf{y}) = \frac{1}{\ell_{\max}} \sum_{\ell=1}^{\ell_{\max}} \sum_{c=1}^{p^\ell} \left[\mathcal{F}_{\ell c}(\mathbf{R}(\theta)) - \mathcal{F}_{\ell c}(\mathbf{R}(\mathbf{y}))\right]^2, \tag{4.14}$$

where $\ell_{\max}$ is the maximum height of the window, and $\mathcal{F}_{\ell c}(\mathbf{R}(\cdot))$ denotes the c^{th} column of the probability matrix $\mathcal{F}_\ell(\mathbf{R}(\cdot)), c = 1, \ldots, p^\ell$. The smaller $J(\mathbf{y})$ is, the more similar are the probability matrices, hence the more similar are the rank matrices $\mathbf{R}(\theta)$ and $\mathbf{R}(\mathbf{y})$, and hence the more similar are $\mathbf{y}$ and $\tilde{\theta}$.

To show that the objective function is indeed minimal when $\mathbf{R}(\theta) = \mathbf{R}(\mathbf{y})$ and $\mathbf{y} = \tilde{\theta}$, and that it increases with the degree to which the ranks in $\mathbf{R}(\mathbf{y})$ are disturbed, we conducted the following numerical experiment. We simulated an input rank matrix $\mathbf{R}(\theta)$ by generating the attributes as $X_{ij} = \theta_i + Z_{ij}, \theta_i \sim \mathcal{N}(0, 1), Z_{ij} \sim \mathcal{N}(0, 0.1^2)$ and ranking these attributes column-wise. Recall that the variance 0.1^2 reflects the size of the error introduced by the assessors. We defined $\mathbf{y} = (\theta_1 + \epsilon_1, \ldots, \theta_p + \epsilon_p)$, where $\epsilon_i \sim \mathcal{N}(0, \sigma_y^2)$, with variance $\sigma_y^2 = 0.01^2$. We then constructed the rank matrix $\mathbf{R}(\mathbf{y})$ and calculated $J(\mathbf{y})$. Consequently, we increased the variance by 0.05^2, so that $\sigma_y^2 = 0.015^2$, and repeated the same calculation. This was done consecutively until $\sigma_y^2 = 1$. By increasing the variance we kept 'contaminating' the underlying signal and we expected the objective function to be minimal where this contamination was equal to the real error introduced by the assessors, i.e. where $\sigma_y^2 = 0.1^2$. The calculated values are plotted in Figure 4.5 for $n = 10, p = 10, \ell_{\max} = 2$, where, for better visibility, the standard deviation σ_y (rather than the variance σ_y^2) is plotted against the values of $J(\mathbf{y})$. We can see that the minimal values of $J(\mathbf{y})$ were indeed around $\sigma_y^2 = 0.1^2$, and that the function increased in both directions away from this point. This behaviour can be observed for any combination of n, p, and ℓ (see Appendix Figure 6.1 for other settings); only the value of the minimum changes. We conclude, therefore, that it is reasonable to use the objective function as a distance measure between the underlying signal and the estimate. As a consequence, in terms of optimisation, our goal must be to minimise $J(\mathbf{y})$.

4.2.3.2 Choice of variances for an estimate y

In each optimisation step, we need to choose appropriate variance parameters $\sigma_j^2, j = 1, \ldots, n$. This is to obtain a rank matrix $\mathbf{R}(\mathbf{y})$ of $\hat{\mathbf{X}} = \mathbf{y} + \hat{Z}$, that can be compared with the observed $\mathbf{R}(\theta)$. We roughly choose a variance $\hat{\sigma}^2$, the same for all assessors,

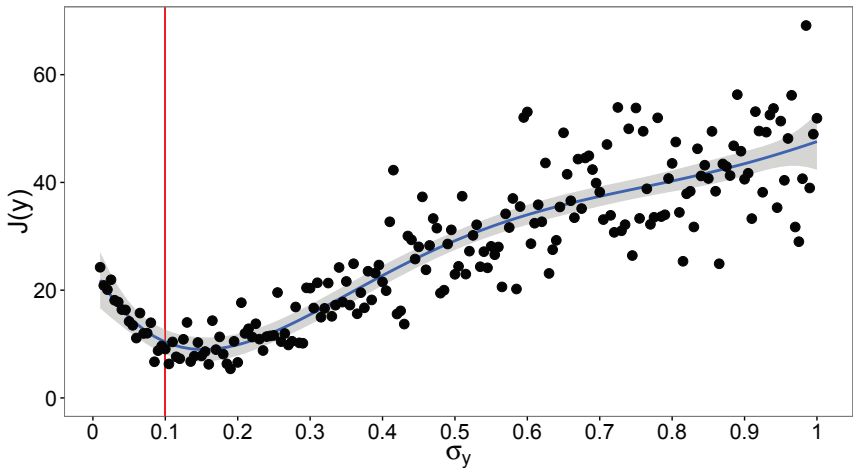


Figure 4.5: Dependence of the objective function $J(\mathbf{y})$ on the random error σ_y added to the underlying signal. The expected minimum is at 0.1. Figure from Švendová and Schimek [2017].

i.e. $\sigma_j^2 = \hat{\sigma}^2$ for all $j = 1, \ldots, n$, in such a way that the resulting $\mathbf{R}(\mathbf{y})$ resembles the observed $\mathbf{R}(\theta)$ as closely as possible. We have to use this approximate solution, because we do not have a priori knowledge about the assessing ability of the assessors. In fact, our main goal is to recover the signal and its uncertainty for each object, not the quality of the assessors.

The parameter $\hat{\sigma}^2$ is chosen in data driven manner via the following steps:

1. Calculate the median of Spearman's ρ correlations between the columns of the observed rank matrix $\mathbf{R}(\theta)$, say $\tilde{\rho}(\mathbf{R}(\theta))$.

2. Create 50 help rank matrices $\mathbf{R}(\mathbf{y})^1, \ldots, \mathbf{R}(\mathbf{y})^{50}$ using 50 different values of $\hat{\sigma}^2$ between 0.01^2 and 0.5^2, in increments of 0.01^2. For a small number of objects ($p \sim 10$), larger increments, i.e. less help matrices, are sufficient.

3. Find $\mathbf{R}(\mathbf{y})^{\text{best}}$, for which the median of Spearman's ρ correlations $\tilde{\rho}(\mathbf{R}(\mathbf{y})^{\text{best}})$ is closest to $\tilde{\rho}(\mathbf{R}(\theta))$. Spearman was chosen here, because it proved faster than other rank correlation coefficients.

4. Pick the smallest $\hat{\sigma}^2$ which was used to produce $\mathbf{R}(\mathbf{y})^{\text{best}}$, and define $\sigma_j^2 := \hat{\sigma}^2$ for all $j = 1, \ldots, n$.

4.3 Numerical evaluation

When performing simulations, we were able to compare our estimates to the simulated true underlying signal, using vector distance and correlation measures (Section 3.4). In real world scenarios, naturally, one does not know the underlying signal, hence the only means of quality evaluation are then calculation of the standard errors (4.3.2) and assessment of rank stability (4.3.4). This section is adapted from Švendová and Schimek [2017].

4.3.1 Quality of the signal estimate

In order to evaluate the quality of an estimated signal $\hat{\tilde{\theta}}$ in our simulations, we compared the estimate to the simulated true signal $\tilde{\theta}$ by calculating Pearson's correlation coefficient r and the Euclidean distance d_E (see Section 3.4 for definitions). The larger the correlation and the smaller the distance, the closer our estimate should be to the true signal.

4.3.2 Error estimate of the estimated signal

The uncertainty of our signal estimates was evaluated by calculating their standard errors in the following way. We bootstrap from the columns of the observed ranking matrix $\mathbf{R}(\theta)$, with replacement. This way we create B independent non-parametric bootstrap samples, say $\mathbf{R}^1, \ldots, \mathbf{R}^B$. We then estimate the corresponding parameter $\hat{\tilde{\theta}}(b)$ for each rank matrix $\mathbf{R}^b$, $b = 1, \ldots, B$. Finally, we calculate the standard errors of the estimated signal values for each object i (Efron and Tibshirani [1994], p.47):

$$\mathrm{SE}(\hat{\tilde{\theta}}_i) = \sqrt{\sum_{b=1}^{B} [\hat{\tilde{\theta}}_i(b) - \bar{\theta}_i]^2/(B-1)}, \tag{4.15}$$

where $\bar{\theta}_i = \sum_{b=1}^{B} \hat{\tilde{\theta}}_i(b)/B$.

We found that violin plots (Hintze and Nelson [1998]) adequately represent the distribution of the bootstrap estimates. We plot the true and estimated signal, together with the bootstrap violin plots and ±2SE estimates (for an example, see Figure 4.8).

4.3.3 Quality of the derived consensus ranking

We obtained a consensus ranking by ranking the values of the estimated signal. In our simulations, we compared the derived consensus ranking, say $\hat{r}$, to the simulated true ranking r^{θ} by calculating the normalised Spearman's footrule distance N_{foot}, and Kendall's τ rank correlation (see Section 3.4 for definitions).

Additionally, we compared the results obtained by our method with the results obtained by two popular rank aggregation methods:

- Borda's method (Marden [1995]), a rather simple approach based on the Borda score BS. For the i^{th} object

$$\text{BS}_i = \sum_{j=1}^{n} R_{ij},$$

  where R_{ij} is the rank assigned to object i by assessor j. The objects are then ordered by their Borda scores in an increasing manner.

- Order Explicit Algorithm (OEA), a modern stochastic optimisation approach using Cross Entropy Monte Carlo (CEMC) simulations. This method is less straightforward (for details see Lin and Ding [2009]). It is implemented in the R package `TopKLists` (Schimek et al. [2015]), function `CEMC`. We applied this function with Spearman's footrule distance as well as Kendall's τ distance. Because the OEA is a stochastic approach, and the aggregation results for the same input data can differ between runs, we determined its aggregated rank list 10 times. Finally, Kendall's τ correlation between the true signal and each of the 10 OEA result was calculated, and the median $\tilde{\tau}$ of the obtained correlation coefficients was taken for comparison with our method.

4.3.4 Stability of the derived consensus ranking

The stability of the estimated ranks between each pair of objects can be defined by the amount of overlap of their ±2SE intervals. The larger the overlap between two intervals, the easier it is to swap the two ranks, i.e. the less stable the ranks are. In order to evaluate the stability, we calculated the percentage of the ±2SE interval of object A which is overlapped by the ±2SE interval of object B, and denoted the overlap percentage as $o(A, B) \in \langle 0, 1 \rangle$. It is clear that this overlap measure is

not symmetric, i.e. $o(A, B) \neq o(B, A)$. Zero overlap, i.e. $o(A, B) = o(B, A) = 0$, points to the maximum rank stability between A and B. Full overlap, i.e. $o(A, B) = o(B, A) = 1$, points to the maximum rank instability, or, in other words, randomly interchangeable ranks between A and B. An *overlap matrix* $O_{p\times p}$ can be defined, where O_{ij} represents the percentage of the $\pm$2SE interval overlap of object i and object j, $i, j = 1, \ldots, p$ (for an example, see Table 4.6). From this matrix, one can identify groups of objects that are mutually rank-unstable.

4.4 Implementation

The algorithm was implemented in R. Its pseudocode can be found below (Algorithm 1 for the GA and Algorithm 2 for the MCMC optimisation) and the individual scripted functions in the Appendix. The iterative part of the algorithm (Figures 4.3, 4.4), was initialised 10 times and the results were averaged, in order to obtain a robust solution. For the standard error estimate, this had to be repeated B times, each time with a different input rank matrix $\mathbf{R}^b, b = 1, \ldots, B$ (see Section 4.3.2). The calculation loops are depicted on Figure 4.6. Each part of the calculation naturally adds to the computational time.

Algorithm 1: Signal estimation with **genetic algorithm (GA)**

Input :

1. Rank matrix $\mathbf{R} = \{R_{ij}\}, i = 1, \ldots, p, j = 1, \ldots, n$ for p objects and n assessors
2. Probability matrix $\mathcal{F}_\ell(\mathbf{R})$ for each $\ell = 1, \ldots, \ell_{\max}$ (see Equation 4.7)

Output: The estimate $\hat{\tilde{\theta}} = (\hat{\tilde{\theta}}_1, \ldots, \hat{\tilde{\theta}}_p)$ of the true signal $\tilde{\theta} = (\tilde{\theta}_1, \ldots, \tilde{\theta}_p)$

for *bootstrap* $b = 1$ **to** B **do**
  for *run* $r = 1$ **to** *maxrun* **do**
    generate pop_size initial population: $\mathbf{y}^{(01)}, \ldots, \mathbf{y}^{(0\text{pop\_size})}$, where $\mathbf{y}^{(0k)} = (y_1^{(0k)}, \ldots, y_p^{(0k)}), k = (1, \ldots, \text{pop\_size})$ and $y_i^{(0k)} \sim \mathcal{U}[-1, 1]$
    for *iteration* $t = 0$ **to** *maxit* **do**
      calculate $J(\mathbf{y}^{(tk)})$ for each k (see eq.4.14)
      $\mathbf{y}_{\text{sel}}^{(tk)}$:= pop_size individuals selected from $\mathbf{y}^{tk}$ with linear ranking selection
      $\mathbf{y}_{\text{cross}}^{(tk)}$:= pop_size individuals formed from $\mathbf{y}_{\text{sel}}^{(tk)}$ with local arithmetic crossover with probability 0.8
      $\mathbf{y}_{\text{mut}}^{(tk)}$:= pop_size individuals formed from $\mathbf{y}_{\text{cross}}^{(tk)}$ with uniform random mutation with probability 0.3
      $\mathbf{y}^{((t+1)\,k)} := \mathbf{y}_{\text{mut}}^{(tk)}$
    $\mathbf{y}^{(m)} = \underset{\mathbf{y} \in \{\mathbf{y}^{((\text{maxit}+1)\,1)}, \ldots, \mathbf{y}^{((\text{maxit}+1)\,\text{pop\_size})}\}}{\text{argmin}} J(\mathbf{y})$
    $\mathbf{y}^{(mr)} = (y_1^{(mr)}, \ldots, y_p^{(mr)}) := \mathbf{y}^{(m)}$ (the best estimate in run r)
  for *object* $i = 1$ **to** p **do**
    $\text{med}_i = \text{median}(y_i^{(m1)}, \ldots, y_i^{(m\,maxrun)})$ (median over all runs)
  $\mathbf{v} = (v_1, \ldots, v_p) := (\text{med}_1, \ldots, \text{med}_p)$
  $\hat{\tilde{\theta}}_b = \frac{\mathbf{v}}{\|\mathbf{v}\|}$, where $\|\mathbf{v}\| = \sum_{i=1}^p v_i^2$ (the estimate for bootstrap sample b)

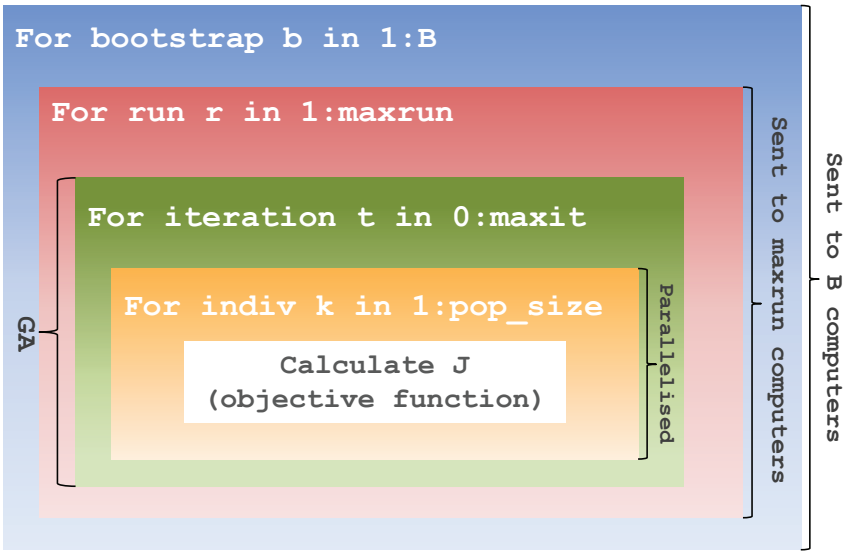
Algorithm 2: Signal estimation with **Metropolis-Hastings MCMC** (adapted from Švendová and Schimek [2017])

Input :

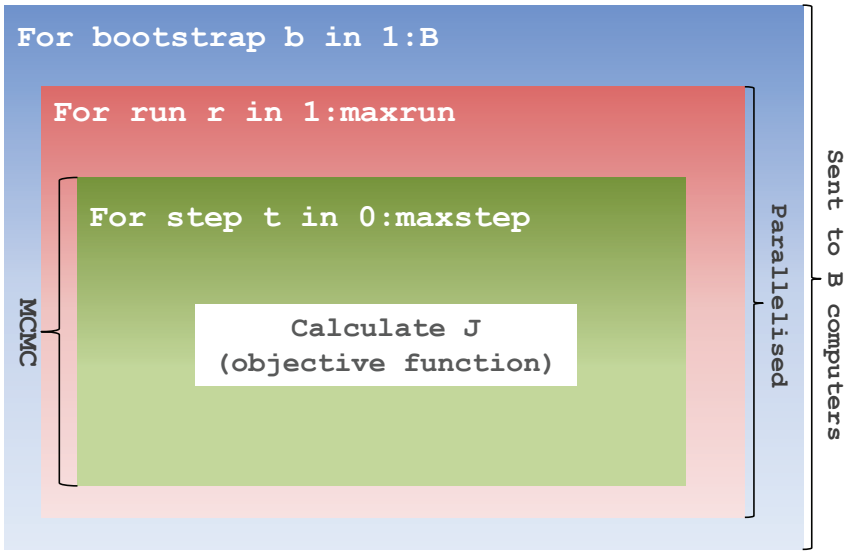
1. Rank matrix $\mathbf{R} = \{R_{ij}\}, i = 1, \ldots, p, j = 1, \ldots, n$ for p objects and n assessors
2. Probability matrix $\mathcal{F}_\ell(\mathbf{R})$ for each $\ell = 1, \ldots, \ell_{\max}$ (see Equation 4.7)

Output: The estimate $\hat{\tilde{\theta}} = (\hat{\tilde{\theta}}_1, \ldots, \hat{\tilde{\theta}}_p)$ of the true signal $\tilde{\theta} = (\tilde{\theta}_1, \ldots, \tilde{\theta}_p)$

for *bootstrap* $b = 1$ **to** B **do**
  for *run* $r = 1$ **to** *maxrun* **do**
    initial guess $\mathbf{y}^{(0)} = (y_1^{(0)}, \ldots, y_p^{(0)})$, where $y_i^{(0)} \sim \mathcal{U}[-1, 1]$
    for *step* $t = 0$ **to** *maxstep* **do**
      $u \sim \mathcal{U}[0, 1]$
      $\epsilon = (\epsilon_1, \ldots, \epsilon_p)$, where $\epsilon_i \sim \mathcal{N}[0, 0.1^2]$
      propose next step $\mathbf{y}' := (\mathbf{y}_1^{(t)} + \epsilon_1, \ldots, \mathbf{y}_p^{(t)} + \epsilon_p)$
      calculate $\Delta J = J(\mathbf{y}') - J(\mathbf{y}^{(t)})$ (see eq.4.14)
      if $u < min\{1, exp(-\Delta J)\}$ **then**
        $\mathbf{y}^{(t+1)} := \mathbf{y}'$
        else
          $\mathbf{y}^{(t+1)} := \mathbf{y}^{(t)}$
    $\mathbf{y}^{(m)} = \underset{\mathbf{y} \in \{\mathbf{y}^{(0)}, \ldots, \mathbf{y}^{(\mathbf{maxstep})}\}}{\operatorname{argmin}} J(\mathbf{y})$
    $\mathbf{y}^{(mr)} = (y_1^{(mr)}, \ldots, y_p^{(mr)}) := \mathbf{y}^{(m)}$ (the best estimate in run r)
  for *object* $i = 1$ **to** p **do**
    $\text{med}_i = \text{median}(y_i^{(m1)}, \ldots, y_i^{(m\,maxrun)})$ (median over all runs)
  $\mathbf{v} = (v_1, \ldots, v_p) := (\text{med}_1, \ldots, \text{med}_p)$
  $\hat{\tilde{\theta}}_b = \frac{\mathbf{v}}{\|\mathbf{v}\|}$, where $\|\mathbf{v}\| = \sum_{i=1}^{p} v_i^2$ (the estimate for bootstrap sample b)



(a) The algorithm with **GA.** For each bootstrap sample (blue field), several independent runs are calculated (red field). Each such run involves several iterations (green field). In each iteration, J is calculated (white field) separately for each individual of the current population (yellow field).



(b) The algorithm with **MCMC.** For each bootstrap sample (blue field), several independent runs are calculated (red field). Each such run involves calculation of a Markov chain with a sufficient number of steps (green field). J is calculated (white field) in each iteration of the chain.

Figure 4.6: Calculation diagrams with usage of Genetic Algorithm (GA) and Monte Carlo Markov chain (MCMC). The detail procedure within the green fields is schematically depicted on Figures 4.3 (GA) and 4.4 (MCMC). `B` is the number of bootstrap samples, `maxrun` the number of independent runs, `maxit` the number of iterations in GA, `pop_size` population size in GA, and `maxstep` the number of steps in an MCMC chain.

4.4.1 Computational considerations

First thing to consider from the computational point of view is the choice of the optimisation method. As described in the previous sections, we used two optimisation methods: a non-parametric GA, and a Bayesian MCMC. Both methods yield approximately the same results, but have different computational demands, depending on the implementation. Based on the code presented in the Appendix, MCMC requires multiple times more iterations (or steps) than GA.

Let us demonstrate the amount of calculations required on an example. Say we wanted to calculate 50 bootstrap samples (`B=50`), 10 runs (`r=10`), and 1000 steps (`maxstep=1000`) with the MCMC optimisation. We need to calculate the objective function $J(\cdot)$ $50 \times 10 \times 1000 = 500,000$ times. If the objective function took 0.01 second to calculate, the entire computation would take 1.4 hours. It is therefore extremely important to optimise the calculations.

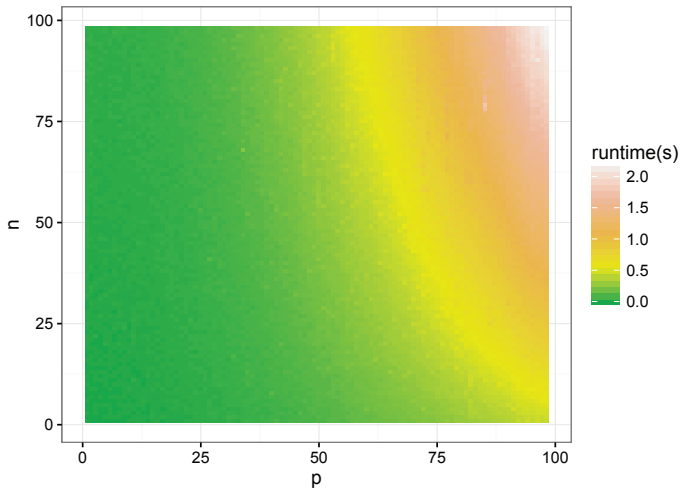


Figure 4.7: Runtime in seconds of the objective function, depending on the number of objects p and number of assessors n.

The computing time of the objective function depends heavily on the number of objects p vs. number of assessors n. Its runtime is shown in Figure 4.7. These times were obtained from a standard desktop machine Win7 64bit, Intel Xeon E5620@2.4GHz/12GB RAM. We can see that the runtime grows exponentially with increasing p and n, where p has a more dramatic impact on the runtime than n. In practical problems such as gene expression meta-analysis (Section 2.2) the number of objects is typically hundreds or thousands times larger than the number of assessors, i.e. $p >> n$. In other cases, such as clinical meta-analysis (Section 2.1),

there are only a few objects (e.g. treatments) and tens or hundreds of assessors (e.g. clinical trials), i.e. $n > p$. Different data types pose different computational demands on the algorithm, and therefore an efficient implementation is crucial.

Practically, there are two ways of speeding up the calculations: (i) optimise the objective function, and (ii) distribute independent tasks from the loops to multiple cores or, even better, multiple computers with multiple cores each.

Concerning (i), it is useful to find out which parts of the code are the most time consuming. This can be done with a code profiler. For example, I used the R profiler `profvis` (Chang and Luraschi [2016]) and found that the choice of variance (Section 4.2.3.2) contributes to most of the computational time. This part of the code was markedly sped up, nevertheless the runtime could be improved further by finding a better, faster, way of choosing the most appropriate variance. This should be an aim of future work.

Concerning (ii), the individual loop calculations of the core function are mainly independent and hence are perfect candidates for parallelisation and usage of computer cluster, as described in the following section.

4.4.2 Usage of parallel and cluster computing

Parallel computing is a way of calculating several tasks simultaneously, using multiple cores of a computer. This can speed up the calculations multiple times, depending on the number of cores. In order to make use of parallel computing in an algorithm, it must be possible to split the calculation into independent tasks. Each task is computed on one core and their results are brought back together in the end. If one has access to not only one, but several interconnected computers with multiple cores each, the whole procedure can be sped up even more, as each computer then calculates several simultaneous tasks. Such a set of interconnected computers working together as a single computing unit is called a *computer cluster*. The distribution of tasks within the cluster is usually done automatically by a queuing system, which evaluates the processing power needed and available. The user has to only specify the expected time, memory load and number of cores required.

There are several levels, where the calculations in our algorithm could be split into independent tasks. Diagrams 4.6a and 4.6b depict how cluster and parallel computing was used for each optimisation method. The bootstrap and run loops (blue and red fields) were completely independent of each other, hence they could

be parallelised with usage of the computer cluster. The GA iterations and MCMC random walk (green fields), on the other hand, were dependent, i.e. every iteration was directly using the result from the previous one, and therefore could not be parallelised or send to a cluster of computers. The calculation example from the previous section (4.4.1) then has a potential to reduce the computational time from 1.4 hours to 10 seconds.

During my three months internship at the Max Planck Institute in Berlin, I had access to a large computer cluster, consisting of 26 computers with about 1700 cores and 20TB of RAM, altogether. Only a fraction of this computing power was needed to get all the simulation results within 1 day, compared to several weeks of computing on a single core desktop computer.

4.5 Simulations

We investigated several simulated scenarios in order to test our model (Table 4.1). This section is an extended version of the text that appears in Švendová and Schimek [2017].
Our aim was to look at the estimates from several different perspectives:

Number of objects versus number of assessors. Does the proportion between objects and assessors influence the results? We presumed that more assessors would increase the strength of the signal.

Window size. Does larger maximum window height $\ell_{\max}$ improve the results? We presumed that larger maximum window height would improve the signal estimate, because more sub-ranking possibilities were taken into account, hence the rank probability should be more precise.

Levels of correlation between the input ranking. How does the level of overall agreement between the assessors influence the results? We expected that lower agreement between the assessors would disturb the signal strength, and hence produce a worse signal estimate.

Poor assessors. Let us define a *poor assessor* as such an assessor who is completely wrong, compared to the other assessors which are in good agreement. How do Poor assessments influence the results? The ranking of a poor assessor was generated as a random sequence of integers between 1 and p, for which Kendall's τ correlation with the true ranking was less than 0.5. We expected poor assessors to worsen the estimate but to a lesser degree than when overall correlation is lower.

GA vs. MCMC Are there differences in results depending on which optimisation method was used? We summarised the overall results from all settings. Based on the strong theory behind MCMC algorithms, it was expected to produce more stable results than GA. The GA procedures have no theoretical guarantee of convergence to a global optimum, hence we expected it to result in worse, or less reliable estimates, compared to MCMC.

Runtime. How long does it take to calculate an estimate depending on number of objects, assessors, window size and optimisation method? We expected the

runtime to increase with larger number of objects/assessors, as well as larger window height. Because MCMC optimisation requires more iteration steps, we expected longer times compared to GA.

The underlying (normalised) signal was simulated as

$$\tilde{\theta} = \frac{\theta}{\|\theta\|},$$

where $\theta = (\theta_1, \ldots, \theta_p), \theta_i \sim \mathcal{N}(0,1)$ and added random error $Z_{ij} \sim \mathcal{N}(0, \sigma^2), i = 1, \ldots, p$ for each assessor $j = 1, \ldots, n$. The size of p, n, σ varied depending on the required setting (see Table 4.1). For easier visual comparison, the underlying signal was always ordered in a decreasing manner, and therefore the true ranking was always $1, \ldots, p$. All settings with $p = 10$ shared the same underlying signal, and the results were therefore directly comparable. The same held for all settings with $p = 20$.

	p	n	$l_{\max}$	%poor	$\tilde{\rho}/\tilde{\tau}$	σ
Setting 1	10	10	2	0% (0)	0.84/0.68	0.125
Setting 2	10	10	2	10% (1)	0.82/0.64	0.125
Setting 3	10	10	2	20%(2)	0.75/0.60	0.125
Setting 4	10	20	2	0% (0)	0.84/0.69	0.135
Setting 5	10	20	2	10% (2)	0.82/0.64	0.135
Setting 6	10	20	2	20% (4)	0.77/0.62	0.135
Setting 7	20	10	2	0% (0)	0.84/0.68	0.09
Setting 8	20	10	2	10% (1)	0.82/0.65	0.09
Setting 9	20	10	2	20% (2)	0.79/0.61	0.09
Setting 11	10	10	2	0% (0)	0.56/0.38	0.29
Setting 12	10	10	2	0% (0)	0.18/0.13	0.61
Setting 13	10	20	2	0% (0)	0.57/0.42	0.29
Setting 14	10	20	2	0% (0)	0.18/0.15	0.68
Setting 15	20	10	2	0% (0)	0.56/0.42	0.19
Setting 16	20	10	2	0% (0)	0.18/0.13	0.55
Setting 17	10	10	3	0% (0)	0.84/0.68	0.125
Setting 18	10	20	3	0% (0)	0.84/0.69	0.135

Table 4.1: Simulation settings, where p is the number of objects, n is the number of assessors, $\ell_{\max}$ is the maximum height of the window, %poor is the percentage of poor assessments, and $\tilde{\rho}/\tilde{\tau}$ is the median of the Spearman correlation/Kendall's τ rank correlation between the assessors, σ the standard deviation used to simulate the input rank matrix $\mathbf{R}$. Table for Settings 1-9 adapted from Švendová and Schimek [2017].

4.6 Simulation results

This section is an extended version of the text that appears in Švendová and Schimek [2017].

4.6.1 Results summary

For each settings (Table 4.1), we evaluated the signal with both optimisation methods, GA and MCMC. We summarised the results using the quality assessment measures defined in Section 4.3. The calculated measures are listed in Table 4.2. In the following, the results for the first four issues from Section 4.5 are compared in terms of (i) signal estimation (Pearson's correlation r, Euclidean distance d_E), (ii) standard error estimates (Median of standard errors MedSE), and (iii) derived consensus ranking, i.e. ranking of the estimated signal (normalised Spearman's footrule distance N_{foot}, Kendall's τ correlation). Rank stability results are commented on separately. The optimisation methods GA versus MCMC are compared in terms of quality of estimate. The runtime is compared between the settings and between optimisation methods.

Number of objects versus number of assessors. Does the proportion between objects and assessors influence the results? We compared three different scenarios: $n = p = 10$ (Setting 1), $n = 20, p = 10$ (Setting 4), $n = 10, p = 20$ (Setting 7). The resulting measures are plotted in the Appendix, Figure 6.2a.

(i) Signal: r remained constantly high, above 0.96, d_E varied between 0.09 and 0.32. The highest correlation and lowest distance ($r = 0.99, d_E = 0.16$ for GA and $r = 0.99, d_E = 0.09$ for MCMC) was observed when $n > p$ (Setting 4), confirming our expectation that a higher number of assessors than objects increases the signal strength.

(ii) SE estimates varied between 0.025 and 0.050. The lowest SE estimates were observed for $n < p$, which we, nevertheless, linked to the differences in the true signal rather than differences between the settings. There were no substantial differences in SE between settings with $n = p$ and $n > p$.

(iii) Consensus ranking . The best ranking was, as expected, produced from the $n > p$ setting ($N_{\text{foot}} = 0.08, \tau = 0.91$ for GA and $N_{\text{foot}} = 0.04, \tau =$

0.96 for MCMC). The differences between $n = p$ and $n < p$ were minor and inconclusive.

More assessors than objects visibly improved the signal estimate, supporting the hypothesis that more assessors add to the signal strength. Because the setting with $p = 20$ had inevitably different simulated true signal than the settings with $p = 10$, we were not able to directly compare $n > p$ versus $n < p$.

Window height. Does larger maximum window height $\ell_{\max}$ improve the results? We compared $\ell_{\max} = 2$ vs. $\ell_{\max} = 3$ (Settings 1, 4, 17, 18). The resulting measures are plotted in the Appendix, Figure 6.2b.

(i) Signal: In our rather small dataset, the correlation between the signal and its estimate was high already with $\ell_{\max} = 2$ ($0.96 < r < 0.99$), so there was not much space for improvement. The signal estimate was therefore comparable for $\ell_{\max} = 2$ and $\ell_{\max} = 3$.

(ii) SE estimates, ranging between $0.037 < d_E < 0.054$, were slightly larger for cases with $\ell_{\max} = 3$. This fact suggests that larger maximum window height might require also larger number of iterations in order to preserve the SE precision.

(iii) Consensus ranking estimates were in good correlation with the true ranking for both $\ell_{\max} = 2$ and $\ell_{\max} = 3$ settings. No conclusive differences between the settings were observed.

In the calculated scenarios, $\ell_{\max} = 2$ was obviously satisfactory and increasing it by 1 did not bring much improvement to already very good estimates. We assume that only for cases with lower overall agreement, or cases with higher p and n, would increasing the maximum window hight make sense. Remember, that increasing the window height comes with exponentially increased computational time.

Levels of correlation between the input ranking. How does the level of overall agreement between the assessors influence the results? We compared high-, mid- and low- median correlation between the columns of the input rank matrix (Settings 1, 4, 7, 11-16). The resulting measures are plotted in the Appendix, Figure 6.3a for $n = p$, Figure 6.3b for $n > p$, and Figure 6.4a for $n < p$.

(i) Signal: The highest Pearson's correlation, as well as the lowest Euclidean distance was observed for the largest overall correlation (Settings 1,4,7, median(r) = 0.98, median(d_E) = 0.2). Expectedly, the lowest correlation and highest distance from the true signal had the estimates from the lowest overall correlation (Settings 12,14,16, median(r) = 0.77, median(d_E) = 0.7).

(ii) SE estimates reflected the level of disagreement between the assessors by having larger values for the lowest correlation (median(MedSE) = 0.10), and smaller values for the largest correlation (median(MedSE) = 0.04).

(iii) Consensus ranking estimates were also clearly better for overall large correlation (median(τ) = 0.91, median(N_{foot}) = 0.08) compared to low correlation (median(τ) = 0.73, median(N_{foot}) = 0.20).

All calculated measures confirmed our expectation, that a decrease in the overall correlation between the assessors causes a decrease in signal estimate precision.

Poor assessors. When all assessors are in good agreement, only a few are completely wrong (we call them *poor assessors*), how does that influence the results? We compared scenarios with 0%, 10% and 20% of poor assessments (Settings 1-9). The resulting measures are plotted in the Appendix, Figure 6.4b for $n = p$, Figure 6.5a for $n > p$, and Figure 6.5b for $n < p$.

(i) Signal Pearson's correlation stayed high, above 0.94, even for 20% poor assessors. The Euclidean distance did slightly increase with introducing poor assessors (median(d_E) = 0.2 for 0%, and median(d_E) = 0.3 for 20% poor assessors).

(ii) SE estimates showed a slight increase for the cases with more poor assessors (median(MedSE) = 0.04 for 0%, and median(MedSE) = 0.05 for 20% poor assessors), reflecting decrease in signal certainty.

(iii) Consensus ranking estimates were not conclusively worse or better when poor assessors were introduced. Curiously, the best estimates were achieved for Setting 5 (10% poor assessors) with both, GA and MCMC, methods ($\tau = 1, N_{\text{foot}} = 0$).

Up to 20% of poor assessments had either none or only very little influence on the resulting estimates. It is likely that the percentage of poor assessors would need to be over 20% in order to disturb the signal noticeably.

Rank stability. As described in Section 4.3.4, the location and width of the SE intervals can be used to asses the mutual rank stability of the objects. The easiest assessment is to visually compare the SEs in the bootstrap plots (see, e.g., Figure 4.8): the most overlapping SE intervals suggest the highest rank instability. The overlap matrix (see, e.g., Table 4.6) contains the exact percentage of overlap between each pair of objects, which can be interpreted as the chance of swopping their ranks. Such examination can point to groups, or 'clusters', of objects that are mutually highly rank-unstable and rank-interchangeable. The overlap matrix also helps to explain the differences in consensus rankings between various aggregation methods. The interpretation of the rank stability is described in more details for Setting 1 in Section 4.6.4.

As we showed above, the settings with lower correlation (whether caused by poor assessors or overall disagreement) generally result in larger SEs. This necessarily means that the overlap between the SEs is also larger and hence the ranks become less stable.

GA versus MCMC. Are there differences in results depending on which optimisation method was used?

There were no major differences in the single estimates by these two methods. Note, that GA is a non-parametric, while MCMC is a Bayesian method. The fact that both methods provided very similar estimates is therefore an excellent result.

The main difference between the two methods was in the width of the SE intervals. When closely investigating the results on Appendix Figures 6.6 to 6.22, one can immediately notice that, even though both single estimates are very close to the true signal, the GA's SE intervals often fail to include the true signal. This is a major drawback, because in order to make our estimate reliable, the SE intervals need to contain the value of the true signal. Take, for example, the GA result from Setting 1 (Figure 6.6a). The GA's SEs do not include the true signals of objects o_1, o_6 and o_9. Additionally, the true signals of objects o_2 and o_{10} lie on the very edge of the SE intervals. When looking at the MCMC result (Figure 6.6b) the true signals for objects o_6 and o_9 lie on the edge of the SE interval, but none of the true

signals lie completely outside the intervals. Overall, it seems that MCMC adapts to lower correlation more reliably by widening the SEs (e.g., compare Setting 4 on Figure 6.9b and Setting 13 on Figure 6.17b). One possible explanation why GA produced less reliable results is that the optimisation procedure got stuck in local minimum instead of converging to a global minimum.

For higher number of objects (e.g. Setting 7, Figure 6.12), nevertheless, even MCMC fails to include some true signals in its SE intervals. Compared to GA, though, it is in fewer cases. This fact suggests, that with increasing number of objects, also the number of iterations should be increased in order to preserve informative SE intervals.

Runtime. How long does it take to calculate an estimate depending on number of objects, assessors and window size?

The runtimes are listed in Table 4.2. The parameter Runtime(m) denotes the runtime in minutes to calculate a signal estimate. The expected runtime for B bootstrap samples on a single machine is Runtime(m)*B. The calculations were performed on a standard 4-core desktop machine Win7 64bit, Intel Core i5-3470CPU@ 3.2GHz/8GB RAM. As expected, the runtime increased when increasing the number of objects, assessors and window height, where the window height had the biggest impact on runtime. The runtime for MCMC was about twice as long as for GA. Nevertheless this comparison hugely depends on the choice of iteration steps. The current implementation parallelises the individual chains within each bootstrap run, hence the maximum speed on a single machine can be achieved by using 10 cores. The bootstrap calculations are independent of each other and can be also run in parallel, reducing the runtime further.

Overall summary. Overall, the best results were obtained in situation with more assessors n than objects p and, at the same time, the highest correlation (Setting 4). The opposite extreme, the worst estimates, were achieved in situation with more objects than assessors and, at the same time, the lowest correlation (Setting 16). Our assumption that high correlation together with $n > p$ guarantees stronger signal, and hence better estimates, was therefore confirmed.

Both methods resulted in very similar estimate, although MCMC proved to be more reliable in terms of SE estimation. Nevertheless, our simulated scenarios are not exhaustive and there are also many additional variables that can influence the

comparison of GA versus MCMC, as well as overall quality of the estimates. For example, each method involves several tuning parameters that have high impact on precision and should be tuned for each problem individually. If computationally possible, one should use both methods for comparison when applied on real data where the true signal is unknown.

4.6.2 The estimate: single estimate versus bootstrap mean estimate

The figures resulting from the simulations (Appendix Figures 6.6 to 6.22) revealed an interesting fact. Our single estimate (blue lines), i.e. the estimate calculated from the input rank matrix, is in most settings very close to what we call the *bootstrap mean estimate* (black dots inside the violin plots), i.e. the mean over B estimates, where B is the number of the bootstrap samples (see Equation 4.16 below). In some cases, however, the bootstrap mean estimate seems to be closer to the true signal, than the single estimate. This is true especially when the correlation between the assessors is rather mild or low, i.e. Settings 11-16. From this we conclude that a more stable solution is achieved from the bootstrap sampling, rather than from the single estimate. Hence a general recommendation is to consider the bootstrap mean estimate to be the best estimate. Let us denote the bootstrap mean estimate $\hat{\Theta} = (\hat{\Theta}_1, \ldots, \hat{\Theta}_p)$, where

$$\hat{\Theta}_i = \frac{\sum_{b=1}^{B} \hat{\hat{\theta}}_i(b)}{B}, \tag{4.16}$$

where $i = 1, \ldots, p$, B is the number of bootstrap samples and $\hat{\hat{\theta}}_i(b)$ is a single estimate for object i from bootstrap sample $b = 1, \ldots, B$. This notation means that the single estimate we considered in our simulated scenarios $\hat{\hat{\theta}}$ is equivalent to $\hat{\hat{\theta}}(1)$.

In the applications of our algorithm (Chapter 5), the signal estimate is always the bootstrap mean estimate $\hat{\Theta}$, rather than a single estimate $\hat{\hat{\theta}}$.

4.6.3 Comparison to rank aggregation methods

By ranking our estimated signals, we obtain a consensus ranking. Such ranking can be also calculated with rank aggregation methods, as described in Section 3.3.1. We chose the simple Borda method, as well as stochastic Order Explicit Algorithm (OEA) with Spearman's distance (here denoted OEA.s) and Kendall's τ distance

(here denoted OEA.k), in order to verify and compare our method to these well established techniques. We calculated consensus ranking estimates for all considered settings with both methods and compared to our estimates, using Kendall's τ rank correlation with the true ranking (Table 4.3).

Our method, independent of the optimisation procedure, produced consensus ranking estimates, whose correlation to the true ranking was similar to the standard rank aggregation methods. In some low agreement scenarios (Settings 3, 6, 9, 13, 14), our estimates were even better than of the other methods.

It is important to note, that rank aggregation techniques lack any statistical error assumption. Therefore, quality measures such as rank stability, crucial for many practical applications, cannot be derived from their results.

4.6.4 Setting 1 in detail with MCMC optimisation

In the example in setting 1, we show the input rank matrix and the obtained results. Let us repeat that setting 1 comprises $n = 10$ assessors $A_1, \ldots, A_{10}$ and $p = 10$ objects $o_1, \ldots, o_{10}$. The input ranking matrix $\mathbf{R}(\theta)$ is depicted in Table 4.4. From the input matrix, we can immediately see some patterns of the signal $\tilde{\theta} = (\tilde{\theta}_{o_1}, \ldots, \tilde{\theta}_{o_{10}})$. For example, objects o_1 and o_2 are ranked exclusively first or second. This should be reflected by close values of $\tilde{\theta}_{o_1}$ and $\tilde{\theta}_{o_2}$, and a considerable gap between these two values and the remaining $\tilde{\theta}_{o_3}, \ldots, \tilde{\theta}_{o_{10}}$. Similarly, object o_{10} is ranked 10^{th} by seven assessors, and hence we can expect $\tilde{\theta}_{o_{10}}$ to be substantially smaller than the remaining values.

After running our algorithm with MCMC optimisation for the input rank matrix $\mathbf{R}(\theta)$ and 50 bootstrap samples $\mathbf{R}^1, \ldots, \mathbf{R}^{50}$, we obtained a signal estimate $\hat{\tilde{\theta}}$, its standard errors SE and derived consensus ranking $\hat{r}$ (Table 4.5).

The results from Table 4.5 are visualised in Figure 4.8. The plot shows the true signals (red line), the signal estimates (blue line), the densities (in violin display format) of the bootstrap signal estimates, and the $\pm$2SE intervals (green horizontal lines) around the mean of the bootstrap estimates (dots). Additionally to the correlation and distance measures (Table 4.2), based on this plot we can visually compare the true and estimated signals.

From the violin plots we can derive three facts for each object: (i) How well the normal error assumption is reflected in the obtained estimates, (ii) how much variation is associated with the signal (signal error), and (iii) how much variation

is associated with the ranks (rank stability). Regarding (i), we can see that the violin plots show unimodal, approximately symmetric densities throughout, hence the normal error assumption for the evaluated model holds. Regarding (ii), the width of the ±2SE intervals corresponds to the signal uncertainty. The narrower the interval, the more confident the estimate. For example, the ±2SE interval for object o_5 is about 40% narrower than that for o_{10}, resulting in a 40% more confident estimate. Regarding (iii), the location and width of the ±2SE intervals define mutual rank stability between pairs of objects. The larger the overlap, the less stable the ranks. We calculated the percentages of overlaps for each pair of objects (Table 4.6). One can see that, for example, the ±2SE intervals of objects o_1 and o_2 are overlapping by 99%. This means that their mutual ranks are extremely unstable. In such cases, one needs to be careful with conclusions about the correct rank order, as the rank positions are highly interchangeable. The opposite is true in the case of objects o_2 and o_3. Their ±2SE intervals do not overlap and the derived rank order can be trusted.

The true signal, as well as the estimated signal, is expected to lie within the ±2SE interval. This is the case for all objects, with the exception of o_6, where the true signal is just outside the ±2SE interval. Such situations, where the true signal does not lie inside, can be explained from the input rank matrix (Table 4.4). The true rank of object o_6 is 6, nevertheless eight assessors out of ten ranked it higher than 6 and consequently the estimates were shifted upwards. If most of the rank positions comprise the same type of error, the signal estimates will reflect it too. On the other hand, if each assessment comprises a different error, the signal estimate can be recovered correctly. For instance, object o_5 was ranked in various positions between 3 and 9, yet the signal estimate was recovered with high precision.

4.7 Discussion

In this chapter we have proposed a novel approach for estimating the signals in multiple ranked lists, as well as their standard errors. Such lists, where several assessors (humans or machines) judge a fixed set of objects by sorting these objects, are common in many fields, such as marketing, opinion research, evaluation research, Web analytics, and biosciences. By combining these lists, we can obtain estimates of the underlying signals responsible for the observed rankings. Such estimates provide

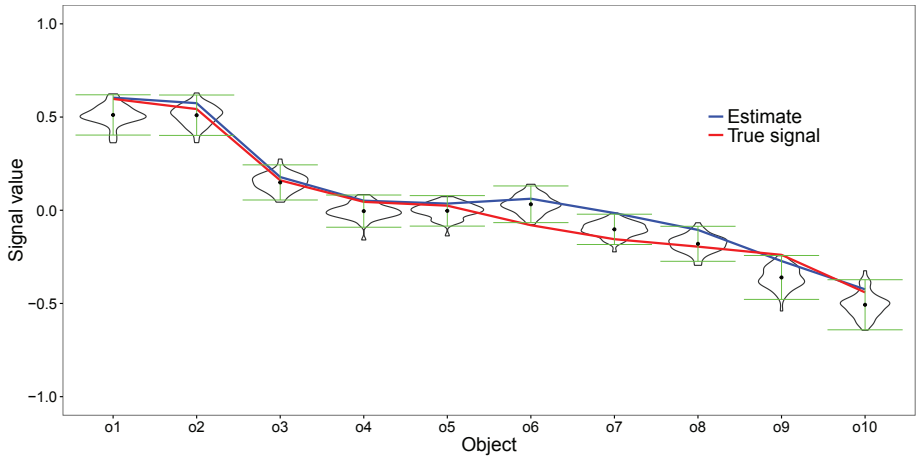


Figure 4.8: Results of Setting 1 with MCMC optimisation. The red line represents the underlying true signals $\tilde{\theta}$, the blue line the estimates $\hat{\tilde{\theta}}$ calculated from the input rank matrix, the dots the bootstrap means, and the horizontal lines the $\pm$ 2SE intervals. Figure from Švendová and Schimek [2017].

not only a consensus ranking (which current rank aggregation methods can already produce), but also reveal rank stability of the rank positions of the objects.

Our method is an iterative indirect inference procedure, which aims to minimise an objective function. The objective function we defined represents the difference between the true signals and their estimates, and is based on calculating the multivariate distribution of ranks in several lists of rankings. We offered two optimisation methods in order to achieve the minimum: a frequentist genetic algorithm (GA), and a Bayesian Metropolis Markov chain Monte Carlo (MCMC) sampling. We calculated standard errors of the estimates using a non-parametric bootstrap.

On a number of simulated scenarios, using both optimisation methods, we were able to demonstrate the potential of the suggested approach at reconstructing the signal from multiple ranked lists. In addition, the obtained signal estimates correctly reflected the patterns of rank instability caused by the objects with almost identical true signal values.

Our results showed that:

1. There was almost perfect correlation between the true and the estimated signals.

2. Our proposed method was robust to a reasonable amount of poor assessments.

3. Weaker signals, caused by high levels of disagreements between the assessors,

were reflected by wider SE intervals.

4. GA and MCMC produced very similar estimates, although MCMC approach was more reliable in terms of SE estimation.

5. Runtime on a standard desktop machine was rather high and increased with larger datasets.

6. The mean over bootstrap estimates was more robust than a single estimate, especially in cases with lower agreement between the assessors.

7. The true rankings correlated firmly with our estimated consensus rankings, obtained by ordering the estimated signal values.

8. Our consensus rankings were comparable to the results from some popular aggregation techniques (Borda, Order Explicit Algorithm).

A limitation of our approach is that it does not allow for missing values in an assessor's ranking. However, this could be solved by imputing ranks based on other assessors' decisions, which is basically a generalisation of our distribution function rank representation. The computations are also rather slow for larger datasets. Future work would include the implementation of more efficient MCMC procedures, such as simulated annealing. Experimenting with cooling schedules could improve the results further, as well as increase the efficiency of the algorithm.

GA	r	d_E	N_{foot}	τ	its.	MedSE ± MAD	Runtime(m)
Setting 1	0.96	0.32	0.12	0.82	250	0.039± 0.0046	7.2
Setting 2	0.94	0.34	0.12	0.82	250	0.040± 0.0040	7.2
Setting 3	0.95	0.36	0.08	0.91	250	0.049± 0.0026	7.2
Setting 4	0.99	0.16	0.08	0.91	250	0.036± 0.0042	8.8
Setting 5	0.98	0.27	0.00	1.00	250	0.037± 0.0030	8.8
Setting 6	0.96	0.28	0.12	0.82	250	0.038± 0.0030	8.8
Setting 7	0.98	0.29	0.10	0.88	250	0.025± 0.0044	9
Setting 8	0.97	0.30	0.10	0.88	250	0.028± 0.0068	9
Setting 9	0.97	0.31	0.10	0.87	250	0.031± 0.0066	9
Setting 11	0.94	0.34	0.20	0.73	250	0.047± 0.0030	7.2
Setting 12	0.76	0.71	0.32	0.56	250	0.091± 0.0087	7.2
Setting 13	0.99	0.18	0.04	0.96	250	0.044± 0.0033	8.8
Setting 14	0.95	0.31	0.16	0.82	250	0.118± 0.0304	8.8
Setting 15	0.94	0.43	0.16	0.78	250	0.058± 0.0543	9
Setting 16	0.78	0.66	0.29	0.60	250	0.109± 0.0290	9
Setting 17	0.97	0.26	0.12	0.82	250	0.052± 0.0083	12.7
Setting 18	0.98	0.32	0.08	0.91	250	0.049± 0.0096	14.5
MCMC	r	d_E	N_{foot}	τ	steps	MedSE ± MAD	Runtime(m)
Setting 1	0.98	0.22	0.08	0.91	10000	0.048± 0.0088	13.6
Setting 2	0.96	0.31	0.12	0.87	10000	0.053± 0.0036	13.6
Setting 3	0.98	0.26	0.08	0.91	10000	0.063± 0.0054	13.6
Setting 4	0.99	0.09	0.04	0.96	10000	0.050± 0.0056	20.7
Setting 5	0.99	0.11	0.00	1.00	10000	0.050± 0.0060	20.7
Setting 6	0.98	0.23	0.00	1.00	10000	0.050± 0.0025	20.7
Setting 7	0.98	0.23	0.11	0.87	10000	0.027± 0.0035	21.8
Setting 8	0.94	0.36	0.12	0.84	10000	0.037± 0.0064	21.8
Setting 9	0.95	0.33	0.12	0.86	10000	0.081± 0.0243	21.8
Setting 11	0.92	0.50	0.20	0.73	10000	0.108± 0.0064	13.6
Setting 12	0.72	0.83	0.40	0.47	10000	0.165± 0.0134	13.6
Setting 13	0.95	0.33	0.08	0.91	10000	0.108± 0.0072	20.7
Setting 14	0.90	0.54	0.20	0.73	10000	0.180± 0.0112	20.7
Setting 15	0.92	0.42	0.18	0.80	10000	0.049± 0.0110	21.8
Setting 16	0.60	0.90	0.40	0.40	10000	0.091± 0.0127	21.7
Setting 17	0.97	0.27	0.12	0.87	10000	0.054± 0.0054	35.8
Setting 18	0.99	0.15	0.04	0.96	10000	0.047± 0.0044	43.6

Table 4.2: Correlations and distances between the signal values $\tilde{\theta}$ and the estimates $\hat{\tilde{\theta}}$ from GA ('its.' denotes number of iterations) as well as MCMC ('steps' denotes number of steps), where r is Pearson correlation (ideally $r = 1$), d_E is Euclidean distance (ideally $d_E = 0$), N_{foot} is normalised Spearman's footrule distance (ideally $N_{\text{foot}} = 0$), τ is Kendall's τ correlation (ideally $\tau = 1$), MedSE is the median of the bootstrap standard errors ± median absolute deviation (MAD), and Runtime(m) is the runtime in minutes. The best overall results are highlighted in green, the worst in red. Table for MCMC Settings 1-9 adapted from Švendová and Schimek [2017].

Method	GA	MCMC	Borda	OEA.s	OEA.k
Setting 1	0.82	0.91	0.91	0.86	0.86
Setting 2	0.82	0.87	0.87	0.87	0.87
Setting 3	0.91	0.91	0.86	0.87	0.82
Setting 4	0.91	0.96	0.95	0.95	0.95
Setting 5	1.00	1.00	0.95	0.95	0.95
Setting 6	0.82	1.00	0.91	0.95	0.95
Setting 7	0.88	0.87	0.95	0.92	0.89
Setting 8	0.88	0.84	0.89	0.88	0.86
Setting 9	0.87	0.86	0.85	0.86	0.85
Setting 11	0.73	0.73	0.82	0.82	0.82
Setting 12	0.56	0.47	0.60	0.24	0.38
Setting 13	0.96	0.91	0.91	0.80	0.78
Setting 14	0.82	0.73	0.78	0.73	0.78
Setting 15	0.78	0.80	0.86	0.77	0.84
Setting 16	0.60	0.40	0.55	0.29	0.42
Setting 17	0.82	0.87	0.91	0.86	0.86
Setting 18	0.91	0.96	0.95	0.95	0.95

Table 4.3: Kendall's τ correlation between the true and the estimated ranks from our method with GA and MCMC optimisation, and from popular rank aggregation methods; OEA.s - Order Explicit Algorithm with Spearman's footrule, OEA.k - Order Explicit Algorithm with Kendall's τ distance. The values for GA and MCMC are identical to those in Table 4.2. Table for Settings 1-9 adapted from Švendová and Schimek [2017].

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
o_1	2	2	1	1	1	2	2	1	1	2
o_2	1	1	2	2	2	1	1	2	2	1
o_3	3	5	3	7	3	5	3	3	3	3
o_4	4	6	6	3	6	4	4	6	6	7
o_5	9	3	4	6	5	3	6	7	8	4
o_6	8	4	5	4	4	6	5	4	5	5
o_7	5	8	7	5	8	7	7	8	4	6
o_8	7	7	8	8	7	9	9	5	7	8
o_9	6	10	9	9	9	8	8	9	10	10
o_{10}	10	9	10	10	10	10	10	10	9	9

Table 4.4: Setting 1: Input rank matrix $\mathbf{R}(\theta)$. Each row represents objects o_1 to o_{10}, and each column represents assessors A_1 to A_{10}. Table from Švendová and Schimek [2017].

	Truth		Estimate		
	$\tilde{\theta}$	r^{θ}	$\hat{\tilde{\theta}}$	$\hat{r}$	SE
o_1	0.60	1	0.60	1	0.054
o_2	0.54	2	0.57	2	0.054
o_3	0.16	3	0.18	3	0.047
o_4	0.05	4	0.05	5	0.043
o_5	0.02	5	0.04	6	0.041
o_6	-0.08	6	0.06	4	0.049
o_7	-0.16	7	-0.02	7	0.041
o_8	-0.20	8	-0.11	8	0.047
o_9	-0.24	9	-0.27	9	0.059
o_{10}	-0.44	10	-0.42	10	0.067

Table 4.5: Setting 1: The true signals $\tilde{\theta}$, the true ranking r^{θ}, the signal estimates $\hat{\tilde{\theta}}$, the derived ranking $\hat{r}$, and the standard errors SE. Table from Švendová and Schimek [2017].

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}
o_1	1	**0.99**	-	-	-	-	-	-	-	-
o_2	**0.99**	1	-	-	-	-	-	-	-	-
o_3	-	-	1	0.14	0.13	0.40	-	-	-	-
o_4	-	-	0.15	1	**0.96**	**0.86**	0.40	0.03	-	-
o_5	-	-	0.14	1.00	1	**0.89**	0.39	-	-	-
o_6	-	-	0.38	0.75	0.74	1	0.23	-	-	-
o_7	-	-	-	0.43	0.39	0.28	1	0.60	-	-
o_8	-	-	-	0.03	-	-	0.52	1	0.17	-
o_9	-	-	-	-	-	-	-	0.13	1	0.45
o_{10}	-	-	-	-	-	-	-	-	0.39	1

Table 4.6: Overlap matrix $O_{p\times p}$ for assessing rank stability in Setting 1. O_{ij} represents the percentage of $\pm$2SE of object i that is overlapped by $\pm$2SE of object j. Overlaps larger than 80% are highlighted in bold. Table from Švendová and Schimek [2017].

Chapter 5

Applications of the multiple ranking model

Our method can be used as a frequentist meta-analytic tool that can combine arbitrary types of objects ranked by any number of assessors. Such situations can be found in many disciplines, as mentioned throughout the thesis. In this chapter, a number of examples are introduced with an emphasis on medical applications.

The method is demonstrated on a simple toy example (Section 5.1) where the underlying signal is known. Section 5.2 describes an application in clinical network meta analysis, comparing several antidepressants. Section 5.3 presents an application from the field of genomics - comparing gene modifications caused by drugs versus those caused by lung cancer, and deriving possible lung cancer therapeutics.

As discussed in Section 4.6.2, the best estimate of $\tilde{\theta}$ is the mean over all bootstrap results $\hat{\Theta}$ (Equation 4.16), and is calculated in this way for all three applications.

5.1 Toy example: The Bottle Experiment

Before presenting some real data applications, let us explain the usage of the model on a toy example, where the true signal is known. An experiment was prepared, where 11 differently sized bottles ($p = 11$) were filled with various amounts of liquid and covered completely in aluminium foil, so that it was not visible how much liquid is inside (Figure 5.1). Each bottle was assigned a random letter from A to K. The bottles were purposely prepared so that they have exponentially distributed weights, i.e. the most pronounced differences for the heaviest bottles and only small

differences for the lightest bottles. The true weights of the bottles in grams, as well as the normalised true weights, are listed in Table 5.1.

Figure 5.1: The Bottle experiment. Bottles of different sizes filled with invisible amount of liquid. The task for each participant was to order the bottles from the heaviest to the lightest (lifting the bottles was allowed).

Twenty four of our colleagues ($n = 24$) volunteered to order these bottles according to their weight, from the heaviest to the lightest. They could lift the bottles and had no time limit. Each participant assigned ranks from 1 to 11, rank 1 to the bottle he/she thought was the heaviest and rank 11 to the bottle he/she thought was the lightest. The purpose of this experiment was to show that we can estimate the relative weights of the bottles solely from the rankings provided by our participants.

5.1.1 Method

This toy example is a typical scenario for the classical Thurstonian model, as introduced in Section 3.2.2. Human assessors evaluated the bottles according to their weight. Because they did not know the true weight, they had to assign what Thurstone called *mental scores* to each bottle and rank them accordingly. Mental score of a bottle is an approximation of its weight, thought of by an assessor. Our model is built the same way, only we call the normalised true weights *underlying signals*, and the mental scores *attributes*.

True order	True weights (grams) θ	Normalised true weights $\tilde{\theta}$	Estimated weights $\hat{\Theta}$	SE
F (Romenquelle 1.5l)	1312	0.67	0.60	0.021
J (Vöslauer 1l)	847	0.43	0.43	0.020
K (Innocent juice 0.9l)	782	0.40	0.39	0.016
I (Arizona green tea 0.5l)	556	0.29	0.29	0.016
A (Ganic water 0.5l)	425	0.22	0.23	0.013
E (Rauch juice 0.3l)	355	0.18	0.24	0.012
D (Coca Cola 0.5l)	249	0.13	0.17	0.010
G (Innocent juice 250ml)	187	0.10	0.17	0.009
B (Innocent juice 250ml)	132	0.07	0.09	0.013
H (Buttermilk 0.5l)	82	0.04	0.12	0.012
C (Yakult 80ml)	74	0.04	0.11	0.011

Table 5.1: True and estimated values of The Bottle Experiment. True weights are the actual weights in grams, normalised true weights are the underlying signals we try to estimate, and estimated weights are our estimates of the underlying signals. SE are the standard errors of the estimates.

Following the model defined in Section 4.2.1, the final rankings by the participants (Table 5.2) were used as the input rank matrix $\mathbf{R}(\theta)$. We assumed that the matrix $\mathbf{R}(\theta)$ is the column rank matrix of

$$X_{ij} = \theta_i + Z_{ij}, \qquad i = 1, \ldots, 11; j = 1, \ldots, 24, \tag{5.1}$$

where θ_i is the true weight (in grams) of bottle i, and each variable Z_{ij} expresses the error that participant j made when evaluating the weight of bottle i. We aimed to estimate the normalised true weights $\tilde{\theta} = \theta/\|\theta\|$, using only the participants' rankings $\mathbf{R}(\theta)$.

The normalised true weights were estimated as the mean ($\hat{\Theta}$, Equation 4.16) of the bootstrap sample estimates ($\hat{\tilde{\theta}}$), using GA optimisation with 200 iterations.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
F	1	1	1	1	1	1	2	1	1	1	1	1
J	2	2	2	2	2	2	1	3	2	2	4	2
K	3	3	3	3	3	3	3	2	3	3	3	3
I	4	4	4	4	4	4	6	4	4	4	2	4
A	6	6	6	5	6	6	4	5	5	6	6	5
E	5	5	5	6	5	5	5	6	6	5	5	6
D	8	7	7	7	8	8	7	7	8	7	8	7
G	7	8	8	8	7	7	8	8	7	8	7	8
B	11	11	9	11	11	9	11	11	9	9	11	9
H	9	9	11	9	9	11	9	9	11	11	9	10
C	10	10	10	10	10	10	10	10	10	10	10	11
	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24
F	1	1	1	1	1	1	1	1	1	1	1	1
J	2	2	2	2	3	2	2	2	3	3	2	2
K	3	3	3	3	2	3	3	3	2	2	3	4
I	4	4	4	4	6	4	4	4	4	4	6	3
A	6	5	6	5	4	5	5	5	6	6	4	6
E	5	6	5	6	5	6	6	6	5	5	5	5
D	8	8	8	8	8	8	7	8	11	7	7	8
G	7	7	7	7	7	11	8	7	7	8	8	7
B	11	11	11	9	11	7	11	11	8	11	11	11
H	9	9	9	11	9	9	9	9	9	9	9	9
C	10	10	10	10	10	10	10	10	10	10	10	10

Table 5.2: Input ranking matrix. Ranking of bottles from the heaviest to the lightest by 24 participants.

We also applied Borda and OEA aggregation techniques (Section 3.3.1) in order to obtain a single consensus ranking based on the input matrix of ranked lists.

5.1.2 Results

The resulting weight estimates are shown in Table 5.1, fourth column, where one can compare them to the normalised true values. The true and estimated values are plotted in Figure 5.2. The red line connects the normalised true weights, the blue line connects our bootstrap mean estimate $\hat{\Theta}$. The violin plots show the distribution of our estimates based on 100 bootstrap samples. The estimates of the true ranking by our method, as well as by other rank aggregation methods, are presented in Table 5.3. The correlation and distance measures between the estimates and the

true values are listed in Table 5.4.

Our signal estimate has remarkably high Pearson's correlation with the normalised true weights ($r = 0.99$). Also Kendall's τ rank correlation is high ($\tau = 0.85$), and normalised Spearman's footrule distance low ($N_{\text{foot}} = 0.13$). We can conclude that our signal estimate is a very good approximation of the true signal.

Regarding the consensus ranking estimation, the best estimate of the true ranking was achieved by the Borda algorithm ($\tau = 0.89$). Our estimate was identical to OEA.k and OEA.s ($\tau = 0.85$). Our method, as well as OEA.k and OEA.s resulted in swopped positions of the bottles A and E, which caused the slightly worse result compared to Borda. All four methods made the same 'mistake' when ordering the last three bottles B, H, and C. All methods being so consistently wrong may point to poor quality of the input data for these three bottles.

Bottle	True	**Our method**	Borda	OEA.k	OEA.s
F	1	1	1	1	1
J	2	2	2	2	2
K	3	3	3	3	3
I	4	4	4	4	4
A	5	6	5	6	6
E	6	5	6	5	5
D	7	8	8	8	8
G	8	7	7	7	7
B	9	11	11	11	11
H	10	9	9	9	9
C	11	10	10	10	10

Table 5.3: True ranking (True) and estimated ranking by our method (Our method), Borda's method (Borda), and the Order Explicit Algorithm with Kendall's τ distance (OEA.k), as well as Spearman's footrule distance (OEA.s).

Let us discuss the quality of our signal estimate and suggest an explanation for the mistakes in the estimates. We can see (Figure 5.2) that our algorithm estimated the true weight with good precision for the top 5 bottles, but then overestimated the weights slightly for the rest of them. This result reflects the quality of the rankings by our participants. Clearly, bigger differences in weight are easier to be recognised by a human, rather than more subtle ones. And hence, because the weights were distributed exponentially, the participants were less certain about the order of the lighter bottles, which necessarily effected our estimate. Let us have a closer look at the input ranks by our participants (Table 5.2).

	$\tilde{\theta}$ vs. $\hat{\Theta}$		r^θ vs. $\hat{r}$	
	r	d_E	N_{foot}	τ
Our method	**0.99**	**0.17**	**0.13**	**0.85**
Borda	-	-	0.10	0.89
OEA.s	-	-	0.13	0.85
OEA.k	-	-	0.13	0.85

Table 5.4: Correlation and distance measures between the estimates and the true values. $\tilde{\theta}$ is the normalised true weight, $\hat{\Theta}$ is the estimated weight, r^θ the true and $\hat{r}$ is the estimated ranking. OEA.s/OEA.k denotes the Order Explicit Algorithm with Spearman's footrule/Kendall's τ distance measure. r is Pearson's correlation (ideally $r = 1$), d_E is Euclidean distance (ideally $d_E = 0$), N_{foot} is the normalised Spearman's footrule distance (ideally $N_{\text{foot}} = 0$), τ is Kendall's τ correlation (ideally $\tau = 1$).

We can see that bottle F, being by far the heaviest (1312g), was correctly ranked first by 23 out of 24 assessors. This was picked up by our algorithm and the weight for F correctly estimated as much higher than the weights of the rest of the bottles. Bottle J (847g) was ranked second by 19 assessors, third by 4 assessors and fourth by 1 assessor, hence its weight was estimated as the second highest, but not too far from the third value for the bottle K (782g), which was ranked second by 4 assessors, third by 19 and fourth by 1 assessor. This way we can explain the connection between the input ranking and our estimate for each bottle. Interesting results can be seen for bottles A (425g) and E (355g), where the assessors had trouble distinguishing which one is heavier - approximately half of them ranked A as the heavier one and the other half ranked E as the heavier one - even though the difference in weight was quite substantial (425-355=70g). This uncertainty can be explained by optical illusion known as *size-weight illusion* and described already by Charpentier [1891] (see Murray et al. [1999] for an English summary), which causes that smaller objects feel heavier compared to the bigger objects of the same weight. From Table 5.1 we can see that bottle A, weighing 425g, was 500ml in volume, and bottle E, weighing 355g, was 300ml in volume. The size-weight illusion likely caused bottle A feeling a little lighter and bottle E feeling a little heavier, which was enough to compensate for the 70 grams difference, and their weights became practically indistinguishable for a human. The same pattern can be observed with bottles D and G, also ranked equally by the participants, yet there was a 62 gram difference between them. Ranking of bottles G and B confirms this illusion theory in another way - these bottles have the same volume (250ml) and the difference between them is only 55 grams, yet the

assessors were very confident (and correct) about their mutual order, because the identical size of the bottles prevented from the illusion we saw in the cases above.

Whenever the assessors were unsure about the order of two bottles, their SE ranges in Figure 5.2 highly overlapped (e.g. A and E), and we could hence interpret their ranks as interchangeable.

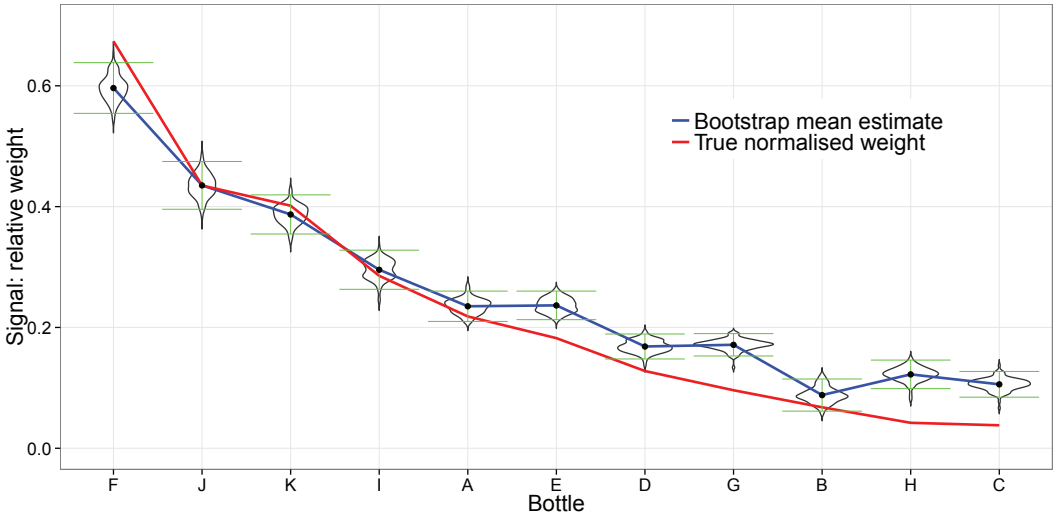


Figure 5.2: Relative weight estimates of the bottles based on participants' ranking. The red line connects the normalised true weights $\tilde{\theta}$, by which the bottles are ordered. The blue line connects our bootstrap mean estimated weights $\hat{\Theta}$ (black dots). Green horizontal lines denote $\pm 2SE$ intervals based on the bootstrap estimates. Settings: GA optimisation, 200 iterations in each run, 10 runs for each bootstrap sample, 100 bootstrap samples.

5.1.3 Discussion

This section described an experiment, in which 24 of our colleagues ranked 11 bottles by their supposed weight. Our algorithm was used to estimate the relative weights of these bottles, based only on the rankings from our colleagues. Our estimates very well agreed with the true weights (with exceptionally high Pearson's correlation of 0.99). The individual estimates got slightly worse when the differences between the weights became less pronounced. This decline in precision was explained by the decrease of quality of the input rankings. When most of the assessors ranked a bottle wrongly higher (or lower) than its true rank, our estimate assigned it a weight that was higher (or lower) than its true weight. When the assessors were indecisive about the order of two bottles, our estimate reflected it by assigning them

very similar weights.

From this toy example we can see that our algorithm was able to provide a very precise estimate of the underlying signal, i.e. the normalised weights, using solely the rankings made by human judges. In the following two sections, this ability of estimating the underlying signal is challenged by two real world datasets.

5.2 Multiple treatment meta-analysis of clinical studies comparing 12 antidepressants

To illustrate how our multiple ranking model can be used in network meta-analysis (see Section 2.1.2.2), we analyzed a dataset provided by Cipriani et al. [2009]. They selected 117 randomised controlled trials (RCTs), where any of the following 12 new-generation antidepressants were compared: bupropion, citalopram, duloxetine, escitalopram, fluoxetine, fluvoxamine, milnacipran, mirtazapine, paroxetine, reboxetine, sertraline, and venlafaxine. These antidepressants were used as monotherapy in the acute-phase treatment (8 weeks) of adults with unipolar major depression. Cipriani et al. assessed the quality of the data as part of the Meta-Analyses of New Generation Antidepressants (MANGA) project[1].

The objectives of Cipriani et al. study were to compare the individual new-generation antidepressants in terms of efficacy and acceptability. Here we focus on efficacy only. Efficacy was assessed by the response rate to treatment. Response rate was hence the main outcome measure and was defined as the proportion of patients who had a reduction of at least 50% in depression severity at 8 weeks, i.e. who responded to the treatment:

$$\text{response rate} = \frac{\text{number of responded}}{\text{number of treated patients}}.$$

The drugs can be ordered from the highest to the lowest response rate and hence are suitable for our algorithm. We used our method to assess the efficacy of the 12 drugs, and compared our results to the results of Cipriani et al.

5.2.1 Input data

The response rates were available from 111 RCTs. The objects in our model were the antidepressant drugs, i.e. $p = 12$, and the assessors were the RCTs, i.e. $n = 111$. We merged all response rates in one table (Table 5.5). Each of the RCTs compared only two or three of the complete collection of 12 treatments, and therefore the table had many missing values. In order to use our algorithm, an imputation of the missing values was necessary.

[1] `http://www.psychiatry.univr.it/docs/ResearchActivities/MTM_Protocol.pdf`

	T1	T2	T3	T4	T5	T6	...	T111
escitalopram	0.45	-	-	0.65	-	-	...	-
fluoxetine	0.37	-	-	-	0.42	0.12	...	-
sertraline	-	0.71	-	-	-	-	...	-
venlafaxine	-	0.67	-	-	-	-	...	-
fluvoxamine	-	-	0.57	-	-	-	...	-
paroxetine	-	-	0.53	-	-	-	...	0.51
reboxetine	-	-	-	-	-	-	...	-
bupropion	-	-	-	0.58	-	-	...	-
mirtazapine	-	-	-	-	0.59	-	...	-
citalopram	-	-	-	-	-	0.09	...	-
duloxetine	-	-	-	-	-	-	...	-
milnacipran	-	-	-	-	-	-	...	0.50

Table 5.5: An illustration of the observed response rate input data for 12 antidepressants and 111 trials (T1–T111).

5.2.2 Missing data imputation

Probably the simplest imputation technique would be to randomly copy the response rates from other trials. But this naïve approach ignores the differences in sample sizes between the trials and may lead to biased estimates. In order to produce more precise estimates, we employed the multiple imputation approach described in Section 3.5.1. Multiple imputation requires to impute the data multiple times and pool the results over the imputations. According to White et al. [2011], the number of imputations should be at least as large as the percentage of incomplete cases. This can be a computational challenge for our algorithm, if the amount of missing data is large. Note that a larger amount of missing data causes larger differences in the completed datasets and hence larger between-imputation variance.

The overall missing rate in Table 5.5 is as high as 80% and hence we should ideally impute 80 times. To reduce the computational burden, nevertheless, we decided to impute the response rates only $m = 50$ times, which proved satisfactory for our purpose. Hence we created 50 complete input matrices. Each matrix was analysed separately and the resulting signal estimates and their standard errors were pooled according to Rubin's rules (Equations 5.5 and 5.6 below).

Let us describe the process of imputation. In each of the m imputations, the missing response rates were replaced by response rates from the other trials using the following weighting method. Assuming that the trials with more patients carry more information, we assigned the probability to be chosen as a replacement

proportionally to the number of patients involved in each trial.

Let us denote t_i the number of trials investigating drug i, and d_{ij} the number of patients that received drug i in trial $j = 1, \ldots, t_i$. Then the probability of imputing the response rate of drug i from trial j is

$$\frac{d_{ij}}{\sum_{j=1}^{t_i} d_{ij}}. \tag{5.2}$$

Imputing the data m times according to the probabilities in Formula 5.2 resulted in m complete tables of response rates (see an illustrative Table 5.6).

	T1	T2	T3	T4	T5	T6	...	T111
escitalopram	**0.45**	0.45	0.65	**0.65**	0.40	0.76	...	0.55
fluoxetine	**0.37**	0.74	0.82	0.57	**0.42**	**0.12**	...	0.53
sertraline	0.58	**0.71**	0.41	0.66	0.67	0.57	...	0.71
venlafaxine	0.39	**0.67**	0.64	0.74	0.64	0.78	...	0.65
fluvoxamine	0.59	0.57	**0.57**	0.59	0.61	0.59	...	0.57
paroxetine	0.39	0.48	**0.53**	0.67	0.67	0.63	...	**0.51**
reboxetine	0.47	0.47	0.39	0.39	0.39	0.64	...	0.39
bupropion	0.56	0.47	0.56	**0.58**	0.56	0.47	...	0.56
mirtazapine	0.72	0.85	0.56	0.56	**0.59**	0.38	...	0.50
citalopram	0.88	0.68	0.45	0.61	0.62	**0.09**	...	0.68
duloxetine	0.30	0.41	0.67	0.46	0.30	0.67	...	0.41
milnacipran	0.50	0.70	0.54	0.54	0.38	0.50	...	**0.50**

Table 5.6: An illustration of the imputed response rates for 12 antidepressants and 111 trials (T1–T111). The observed, i.e. not imputed, values marked bold.

5.2.3 Signal estimate versus odds ratio (OR)

The output of Cipriani et al. were odds ratios (ORs) for each treatment, obtained from a meta-regression NMA (see Section 2.1.2.2). OR is the ratio between the odds of response given drug A, compared to the odds of response given drug B. The odds of response for a drug is calculated as the number of patients with response, divided by the number of patients without response. For example, let us consider the following outcome for drug A and drug B:

	A	B
response	20	50
no response	80	120

Then the OR for drug A versus drug B will be:

$$\text{OR} = \frac{20/80}{50/120} = 0.6,$$

i.e. the odds for drug A to cause a response are 0.6 times lower than for drug B. In other words, drug B is likely to be more efficacious than drug A.

In NMA, a *reference drug* is chosen and hence its OR = 1 (because the odds of the reference are divided by the odds of the reference). The ORs of the other drugs are then calculated with the odds of the reference drug in the nominator. Therefore when the resulting OR is above 1, the reference drug is more efficacious, while when it is below 1, the reference drug is less efficacious, compared to the other drug. Based on previous literature and usage of the investigated antidepressants, Cipriani et al. used fluoxetine as the reference drug.

Our method does not work with the response rates directly, but takes their ranks instead, as described below in Section 5.2.4. The underlying signals calculated from these ranks represent the relative responses to the drugs. We stress the word 'relative', as one value on its own is uninformative and the values only make sense when mutually compared. Larger the value, higher the response compared to the other drugs.

Both, ORs from NMA as well as our signal estimates, report on the relative drug responses. Nevertheless, the two results are not directly comparable for two reasons: (i) NMA uses a reference drug which by definition has OR value of 1, and the other ORs can theoretically vary between $[0, \infty)$, while our signal estimates do not consider any reference drug and can theoretically vary between $(-\infty, \infty)$; (ii) the two results have opposite directions, i.e. higher response rate is represented by higher signal values, but smaller ORs. In order to make our results comparable to Cipriani's ORs, we have to (i) make fluoxetine the 'reference drug' with the value of 1 and shift our estimates for all the other drugs proportionally, and (ii) change the direction of our results, so that smaller values mean higher response. The solution of (i) is straightforward - one has to divide the signal values of all drugs by the value of fluoxetine. The solution of (ii) depends on the maximum and minimum value of the estimates. Because our estimated values lie between $(0, 1)$, we could change their direction by simply substracting the values from 1.

Formally, the adjustments (i) and (ii) can be described as follows. Let us denote our signal estimates for the 12 drugs as $\hat{\Theta} = (\hat{\Theta}_1, \ldots, \hat{\Theta}_{12})$, where $\hat{\Theta}_i \in [0, 1], i =$

$1, \ldots, 12$. Then the shifted estimate, centered around fluoxetine at 1, will be

$$\hat{\Theta}_{\text{shift}} = \frac{1 - \hat{\Theta}}{1 - \hat{\Theta}_f}, \tag{5.3}$$

where $\hat{\Theta}_f$ is the signal value estimated for fluoxetine. The values of $\hat{\Theta}_{\text{shift}}$ are then directly comparable to the values of OR.

5.2.4 Method

Our aim was to compare all the drugs, using the imputed response rates. If trying to do that without any sophisticated method, one could, for example, simply average the response rates for each drug over all trials. However, relying so strongly on the imputed rates seems questionable, as the amount of missing values in Table 5.5 is very large (over 80%) and hence lots of uncertainty is involved. Instead, we ranked the drugs within each trial and replaced the untrustworthy accuracy, i.e. imputed rates, by a less specific generality, i.e. ranks. Specifically, we ranked the drugs within each trial from those with the highest response rate (rank 1) to those with the lowest response rate (rank 12), as illustrated in Table 5.7. We did so for each of the m imputed matrices. By ranking the drugs we reduced the unknown within- and between-study heterogeneity, as well as other sources of uncertainty, such as an assumptions about a distribution. Such rankings were then directly analysed by our method.

We denoted the ranked lists in the same manner as in Chapter 4. That means we had m input rank matrices $\mathbf{R}(\theta)^{(1)}, \ldots, \mathbf{R}(\theta)^{(m)}$, where $m = 50$ was the number of imputations. Using our model, we assumed that each matrix $\mathbf{R}(\theta)^{(k)}, k = 1, \ldots, m$ was the column rank matrix of

$$X_{ij}^{(k)} = \theta_i + Z_{ij}^{(k)}, \qquad i = 1, \ldots, 12; j = 1, \ldots, 111, \tag{5.4}$$

where $X_{ij}^{(k)}$ is the k-th imputed matrix of response rates (like in Table 5.6), θ_i is the unknown true response rate of drug i, and each variable $Z_{ij}^{(k)}$ represents the error produced in trial j when evaluating the response of drug i, for the k-th imputed matrix. We aimed to estimate the normalised true response $\tilde{\theta}_i = \theta_i / \|\theta\|$ for each drug i, using only the drug rankings $\mathbf{R}(\theta)^{(1)}, \ldots, \mathbf{R}(\theta)^{(m)}$.

For each imputation $k = 1, \ldots, m$ and each antidepressant $i = 1, \ldots, 12$, we cal-

	T 1	T 2	T 3	T 4	T 5	T 6	...	T 111
escitalopram	8	11	3	4	9	2	...	6
fluoxetine	11	2	1	8	8	11	...	7
sertraline	4	3	11	3	1	7	...	1
venlafaxine	9	6	4	1	3	1	...	3
fluvoxamine	3	7	5	6	5	6	...	4
paroxetine	10	8	9	2	2	5	...	8
reboxetine	7	9	12	12	10	4	...	12
bupropion	5	10	6	7	7	9	...	5
mirtazapine	2	1	7	9	6	10	...	9
citalopram	1	5	10	5	4	12	...	2
duloxetine	12	12	2	11	12	3	...	11
milnacipran	6	4	8	10	11	8	...	10

Table 5.7: An illustration of the ranked drugs based on the imputed response rates (Table 5.6).

culated our bootstrap mean signal estimate $\hat{\Theta}_i^{(k)}$ (Equation 4.16), using the MCMC optimisation and 30 bootstrap samples.

Then, we combined $\hat{\Theta}^{(1)}, \ldots, \hat{\Theta}^{(m)}$ and their variances $\mathrm{V}(\hat{\Theta})^{(1)}, \ldots, \mathrm{V}(\hat{\Theta})^{(m)}$, where $\mathrm{V}(\hat{\Theta})^{(k)}$ is the variance of the bootstrap estimates from all imputations $1, \ldots, m$, according to the Rubin's rules as follows. For each drug i we calculated (Molenberghs et al. [2014]) the combined estimate

$$\hat{\Theta}_i^{\mathrm{MI}} = \frac{1}{m} \sum_{k=1}^{m} \hat{\Theta}_i^{(k)}, \tag{5.5}$$

and combined variance

$$\mathrm{V}_i^{\mathrm{MI}} = \bar{\mathrm{V}}_i + (1 + \frac{1}{m}) B_i, \tag{5.6}$$

where $\bar{\mathrm{V}}_i = \sum_{k=1}^{m} \mathrm{V}(\hat{\Theta}_i)^{(k)}/m$ is the within-imputation variability, and $B_i = \sum_{k=1}^{m} (\hat{\Theta}_i^{(k)} - \hat{\Theta}_i^{\mathrm{MI}})^2/(m-1)$ is the between-imputation variability that should correct for the missing data uncertainty. Standard errors were then calculated as $\mathrm{SE}_i = \sqrt{\mathrm{V}_i^{\mathrm{MI}}}$ and confidence intervals as

$$\mathrm{CI}(\hat{\Theta}_i^{\mathrm{MI}}) = \hat{\Theta}_i^{\mathrm{MI}} \pm 1.96 \ \mathrm{SE}_i,$$

for 0.95 confidence probability.

Finally, we transformed our estimate according to the Formula 5.3 and obtained $\hat{\Theta}_{\mathrm{shift}}$, an estimate comparable with OR from Cipriani et al. [2009].

5.2.5 Results

To be consistent with the results of Cipriani et al. [2009], instead of referring to a drug's response rate, we will refer to its efficacy. The connection is straightforward: the higher the response rate, the higher the efficacy. ORs and our estimates $\hat{\Theta}_{\text{shift}}$ take then the smallest values for the drugs with the highest efficacy.

Cipriani et al. [2009] admit that there were issues that can undermine the validity of their findings. For example financial biases in the studies (four of the drugs were assessed by pharmaceutical companies), some limitations of the primary trials (discrepancies between direct and indirect evidence, incomplete randomisation information), and potential confounders (e.g. dose issues). Acknowledging these biases, we compared the Cipriani and our results.

Figure 5.3 shows the comparison of results from our method and the meta-regression NMA used in Cipriani et al. [2009]. Table 5.8 lists the values of the point estimates and CIs, together with the correlation and distance measures between the results of these two methods. We can see that the point estimates are very close, especially for the most efficacious drugs. The CIs are also similar and highly overlapping between the two methods. Our CIs are wider than Cipriani's, which is probably due to the large missingness in the data that causes large between-imputation variance. The group of the first four drugs, i.e. sertraline, mirtazapine, escitalopram, and venlafaxine, were estimated by both methods as significantly more efficacious than the reference fluoxetine.

Main differences between the two methods were observed for the last three drugs: duloxetine, milnacipran and reboxetine. Table 5.9 shows the number of trials each drug was investigated in, as well as the number of drugs each drug was compared to. We can see that the above mentioned three drugs were investigated in the least number of trials and compared to the least number of other drugs. This means that the amount of direct evidence was low and the analysis heavily depended on the indirect evidence.

Consistency, i.e. the agreement between the direct and indirect evidence, is a condition necessary for reliable results with NMA. If we look closely at the ORs reported in Cipriani et al. [2009] (Table 3 in the paper) we can see that in several cases (concerning duloxetine, milnacipran and reboxetine) indirect evidence contradicted direct evidence. For example, from direct evidence milnacipran and reboxetine had lower, and duloxetine higher response rate than fluoxetine, while according to indi-

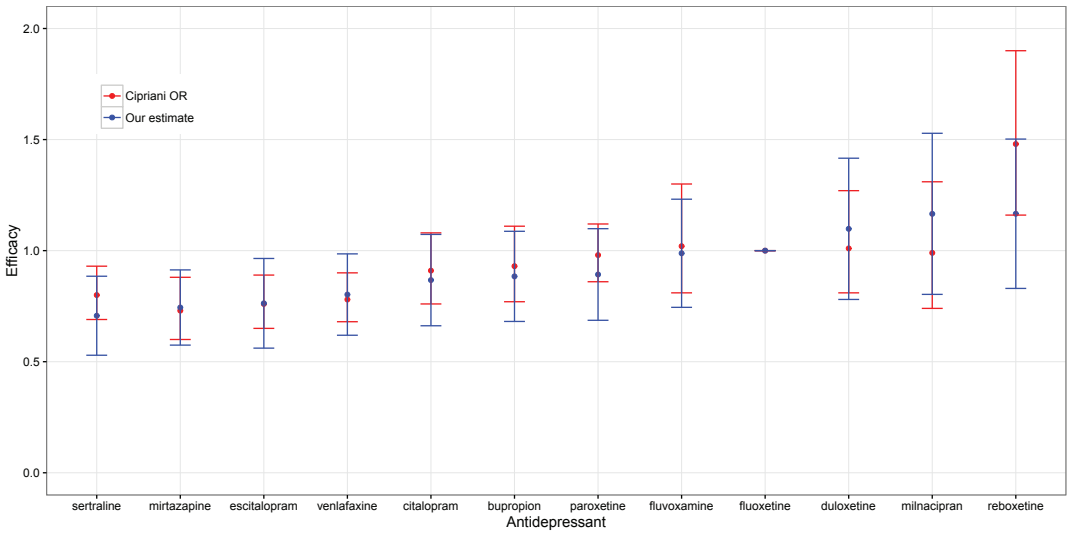


Figure 5.3: Efficacy estimates for the 12 antidepressants with fluoxetine being the reference drug, our results (blue) versus Cipriani (red). Lower value represents more efficacious drug. Red dots are ORs with 95%CI intervals from Cipriani et al. [2009], Table 4. Blue dots are our signal estimates $\hat{\Theta}_{\text{shift}}$ with 95%CI intervals (horizontal lines). The drugs are ordered by our estimate from the most to the least efficacious. Settings: MCMC optimisation, 5000 steps in each run, 10 runs for each bootstrap, 30 bootstrap samples.

rect evidence, milnacipran and reboxetine should have lower, and duloxetine higher response rate than fluoxetine. These inconsistencies together with a low number of trials and comparisons are likely the reason for larger CIs and more pronounced differences (between our method and meta-regression) in the estimates of efficacy of duloxetine, milnacipran and reboxetine.

	$\hat{\Theta}_{\text{shift}}$ (95%CI)	r($\hat{\Theta}_{\text{shift}}$)	OR (95%CI)	r(OR)
sertraline	0.71 (0.53–0.88)	1	0.80 (0.69–0.93)	4
mirtazapine	0.74 (0.57–0.91)	2	0.73 (0.60–0.88)	1
escitalopram	0.76 (0.56–0.96)	3	0.76 (0.65–0.89)	2
venlafaxine	0.80 (0.62–0.98)	4	0.78 (0.68–0.90)	3
citalopram	0.87 (0.66–1.07)	5	0.91 (0.76–1.08)	5
bupropion	0.88 (0.68–1.09)	6	0.93 (0.77–1.11)	6
paroxetine	0.89 (0.68–0.10)	7	0.98 (0.86–1.12)	7
fluvoxamine	0.99 (0.74–1.23)	8	1.02 (0.81–1.30)	11
fluoxetine	1	9	1	9
duloxetine	1.09 (0.78–1.41)	10	1.01 (0.81–1.27)	10
milnacipran	1.16 (0.80–1.53)	11	0.99 (0.74–1.31)	8
reboxetine	1.16 (0.82–1.50)	12	1.48 (1.16–1.90)	12
Euclidean distance		0.39 (ideally 0)		
Pearson correlation		0.81 (ideally 1)		
Kendall's τ correlation		0.76 (ideally 1)		
Normalised footrule distance		0.17 (ideally 0)		

Table 5.8: Odds ratios (OR) with 95% CI by Cipriani et al. [2009] versus our estimates $\hat{\Theta}_{\text{shift}}$ with 95% CI and their ranks, and distance and correlation measures. r($\hat{\Theta}_{\text{shift}}$) is the ranking according to our estimate $\hat{\Theta}_{\text{shift}}$, r(OR) is the ranking according to the OR. The drugs are ordered by our estimate from the most to the least efficacious.

	No.trials	No.drugs
sertraline	27	10/11
mirtazapine	13	6/11
escitalopram	19	9/11
venlafaxine	28	8/11
citalopram	16	8/11
bupropion	14	5/11
paroxetine	32	10/11
fluvoxamine	11	7/11
fluoxetine	54	11/11
duloxetine	8	3/11
milnacipran	6	4/11
reboxetine	8	4/11

Table 5.9: The number of trials (No.trials) for each drug and the number of drugs (No.drugs) each drug was compared to, out of 11 possible comparisons. Values copied from the supplementary tables in Cipriani et al. [2009].

5.2.6 Discussion

In this section, we used our novel method as a tool for network meta-analysis (NMA) of 111 RCTs assessing 12 antidepressants. We calculated an estimate of the true response of drugs, and compared it to odds ratios (OR) calculated by Cipriani et al. [2009], who used meta-regression NMA.

Our estimates were in good agreement with the Cipriani estimates. The top-4 most efficacious drugs identified by meta-regression NMA, were also found the most efficacious by our method. Both methods resulted in wider CIs for drugs investigated in only a small number of trials, reflecting the expected lower certainty of the signal. The main differences between the two methods were linked to the lack of direct evidence, as well as contradictions between direct and indirect evidence.

Overall, we saw that our method can be used as an NMA tool with results comparable to meta-regression NMA, providing the assumptions of NMA are satisfied. Our approach uses the ranks of the observed response rates and this way reduces various sources of errors connected to them. Meta-regression NMA accounts for errors by estimating within- and between-study variances. Unlike the classical NMA, our method does not need any reference drug to assess the efficacy, and the point estimates can be compared straightforwardly between any subset of studied drugs. Computational requirements, nevertheless, remain a challenge for practical usage of the method.

5.3 Meta-analysis of drug gene signatures in order to identify new therapeutic candidates

As a genomics meta-analysis application of our method, we analysed a subset of the data provided in a study by Fortney et al. [2015]. In this study changes in gene expression of 21 lung cancer signatures[1] (tumour vs. normal) were compared with gene expression responses to drug treatment in cultured human cells for 1309 drugs, using the Connectivity Map database (Lamb et al. [2006]). Each drug was assigned a correlation score, called *connectivity score*, ranging between -1 and 1. A large negative score indicates that the drug reverses gene modifications connected to lung cancer, and hence is a potential lung cancer therapeutic. Fortney et al. converted the connectivity scores into ranks and calculated a consensus ranked list, using the Rank Product method (Breitling et al. [2004]). The drugs with the highest ranks, i.e. ranks close or equal to 1309, were believed to be lung cancer therapeutic candidates. They found 247 such drugs. The procedure of obtaining the therapeutics candidates is depicted in Figure 5.4.

We aimed to estimate the true connectivity score. Having such estimate, we can: (i) rank the drugs according to their potential to reverse lung cancer-related gene modifications, and (ii) identify groups of rank-unstable drugs, for which the true connectivity score with lung cancer is very similar, but definitely larger or smaller compared to other drugs or groups of drugs. Drugs within a group can be then considered to have a very similar effect on lung cancer.

5.3.1 Input data

Input data for our method were lists of drugs ranked by their connectivity scores (green lists in Figure 5.4). For illustrative purposes, we chose 50 drugs, out of which 10 were ranked at the top of the 247 significant drugs that resulted from the Fortney et al. study as those consistently reversing lung cancer gene modifications. The remaining 40 drugs were chosen randomly from the remaining 1309-10=1299 drugs. The input rank matrix was obtained by ranking the connectivity scores of the chosen 50 drugs.

[1]Gene signature - group of genes in a cell whose combined expression pattern is uniquely characteristic of a biological phenotype

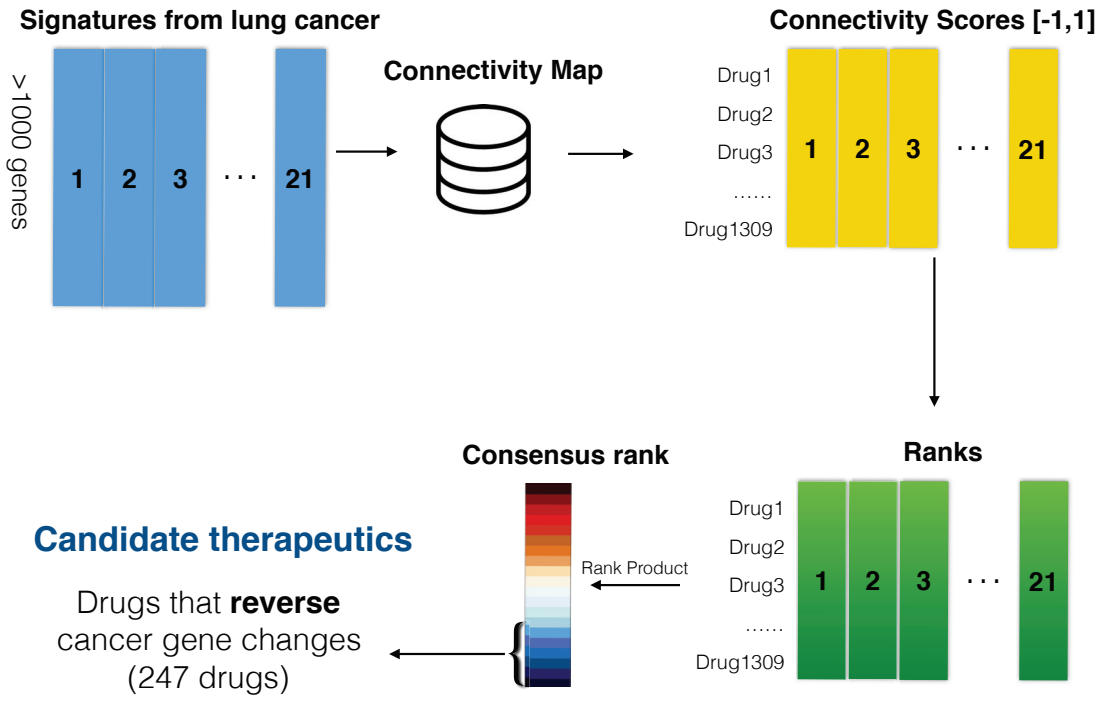


Figure 5.4: Diagram depicting the procedure of obtaining ranked lists of drugs using the Connectivity map database, as used in Fortney et al. [2015]. Lung cancer gene signatures were compared to the drug gene signatures from the database and each drug was assigned a connectivity score. The drugs were subsequently ranked by these scores and aggregated into a single list, where those drugs ranked at the bottom were those of interest, as they supposedly reverse lung cancer gene changes.

5.3.2 Method

The connectivity scores form the matrix of attributes $\{X_{ij}\}$ with $p = 50$ objects (drugs) and $n = 21$ assessors (lung cancer samples), and, according to the assumptions in Section 4.2.1, we assumed that they follow the model

$$X_{ij} = \theta_i + Z_{ij}, \qquad i = 1, \ldots, 50; j = 1, \ldots, 21. \tag{5.7}$$

Each parameter θ_i can be understood as the 'true relationship score' of drug i with lung cancer. Each random error Z_{ij} expresses the noise added to the true score of drug i by lung cancer sample j. The aim was to estimate the normalised true scores $\tilde{\theta}_i$ for each drug i solely from the rankings, say $\{R_{ij}\}$, of the measured connectivity

scores $\{X_{ij}\}$.

The input data comprised a few visibly poor assessments, i.e. lung cancer samples whose ranks have low correlation with the rest of the drugs. As we have seen in the simulation scenarios with bad experts or low correlation, a single estimate can be 'contaminated' by poor assessments (here bad samples). Hence we assumed that the mean over the bootstrap estimates is a more robust approximation of the true signal than a single estimate. We ran our method with GA optimisation for 30 bootstrap samples, each with 1000 iterations, and, like in the previous examples, calculated the final signal estimate for each drug as the mean over the bootstrap single estimates $\hat{\Theta}$ (Equation 4.16).

Typically, one does not know the values of the underlying signal, and hence it is not easy to verify how close an estimate is to its unknown signal. However, here we have all the attributes (connectivity scores) that produced the rankings. In addition we know that they are defined on the same scale $[-1, 1]$. Therefore we can assume that the normalised medians over the connectivity scores for each drug are a representation of the underlying signal, i.e. a "ground truth". Hence we could evaluate the quality of the bootstrap signal estimates by calculating the Spearman's correlation and Euclidean distance between the ground truth and our signal estimate.

By ordering our estimate we obtained a consensus ranking estimate. We also calculated consensus ranking estimates by Borda's method and OEA methods (3.3.1), and compared all results to the ranking of the ground truth by calculating Kendall's correlation and Spearman's footrule distance.

5.3.3 Results

Our estimates are plotted in Figure 5.5. We were primarily interested in those drugs with signal values below zero, as they should be anti-correlated with lung cancer and might reverse associated genetic modifications.

The drug that stands out the most is dexverapamil and thus should be proposed as the best therapeutic candidate. The next 10 drugs have highly overlapping standard error intervals, which are also larger than those of the subsequent drugs. This pattern suggests that the 10 drugs in this group behave in a similar fashion with respect to lung cancer gene modifications. Because we purposely chose ten drugs from the top of the Fortney's consensus ranking, we expected these ten drugs to

be also at the top of our estimated consensus ranking. Indeed, 10 out of the top-ranked 11 drugs obtained using our method are those preselected from Fortney et al. This finding confirms that our approach performs well in identifying the most distinguished therapeutic candidates. Moreover, the top-10 rank positions we have obtained are in good agreement with the Fortney results (the Fortney top-10 ranks are written beside the drug names in Figure 5.5).

The ground truth is plotted in Figure 5.5 in red. The correlation measures and additional distance measures are summarised in Table 5.10. Based on the calculated measures (e.g. Pearson's correlation $r = 0.97$ and Kendall's τ correlation $\tau = 0.93$), our estimate was very close to the ground truth. We can see that our point estimates are generally above the ground truth (Euclidean distance $d_E = 0.7$), nevertheless, they follow the increase of the ground truth (hence the very high Pearson's correlation). Comparison of the increasing trend is more important than comparison of the values themselves, because the chosen ground truth (median of the connectivity scores) is not the true underlying signal, but only its approximation.

The following results were achieved when comparing the estimated consensus rankings and the ground truth ranking. The Borda's method achieved Kendall's correlation $\tau = 0.91$, OEA with Spearman's footrule distance $\tau = 0.83$, and for OEA with Kendall's τ distance $\tau = 0.89$. Our method performed the best ($\tau = 0.93$), estimating a consensus ranking closest to the ground truth. Borda's method performed surprisingly well yielding a result similar to our approach. The Monte Carlo-based techniques were both outperformed by our method.

	Signal vs. ground truth		Est.rank vs. ground truth rank	
	r	d_E	N_{foot}	τ
Our method	**0.97**	**0.7**	**0.056**	**0.93**
Borda	-	-	0.072	0.91
OEA.s	-	-	0.125	0.83
OEA.k	-	-	0.096	0.89

Table 5.10: Resulting correlation and distance measures with the ground truth. OEA.s/OEA.k - Order Explicit Algorithm with Spearman's footrule/Kendall's τ distance measure. r is Pearson correlation (ideally $r = 1$), d_E is Euclidean distance (ideally $d_E = 0$), N_{foot} is the normalised Spearman's footrule distance (ideally $N_{\text{foot}} = 0$), τ is Kendall's τ correlation (ideally $\tau = 1$).

Let us have a closer look at the 8 drugs surrounded by black bounding box in Figure 5.5: resveratrol, luteolin, daunorubicin, vorinostat, bromocriptine, rolite-

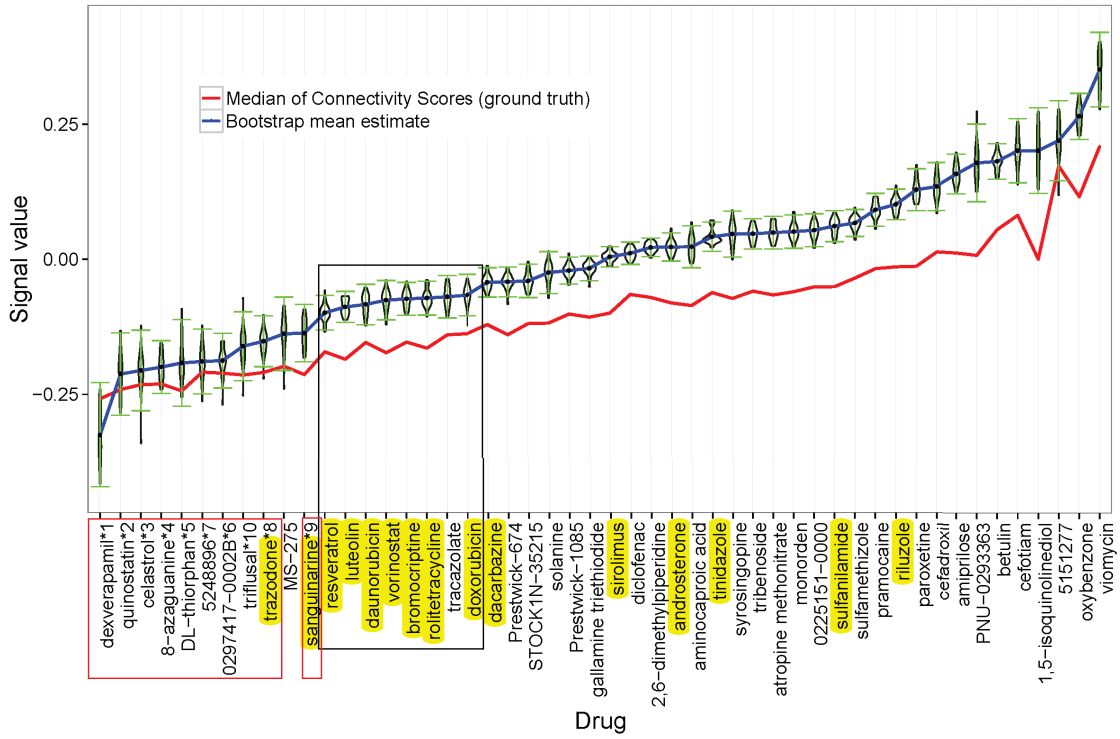


Figure 5.5: Resulting signal from 21 gene signatures. Drugs are ordered by our bootstrap mean estimate. Drugs with largest negative value supposedly reverse lung cancer gene changes. Top-10 drugs preselected by the Fortney et al. study are highlighted by red bounding box and their rank by the number beside their name. Black bounding box surrounds a group of drugs with very similar estimated signal, described in the text. Drugs that target enzymes connected to lung cancer are highlighted in yellow. The short horizontal lines denote $\pm 2SE$ intervals. Settings: GA optimisation 1000 iterations in each run, 10 runs for each bootstrap, 30 bootstrap samples.

tracycline, tracazolate and doxorubicin. Our algorithm grouped them together as drugs with very similar signal estimate (around -0.07). This is an interesting result, because these drugs were randomly chosen from the entire set of 1309 drugs, hence are not predetermined to be ranked beside each other. Yet our algorithm suggested that they are homogeneous in terms of gene modifications on lung cancer cells. It may be because they have similar chemical structure or similar biological function, which caused them being ranked similarly and hence our algorithm assigned them similar signal values. In the following paragraph, we try to support this hypothesis by looking at the enzymes that these drugs target and inhibit.

According to the DrugBank database (Knox et al. [2011]), 7 out of the 8 drugs above target a group of enzymes called Cytochrome P450 (CYP). These enzymes

have been associated with cancer formation, and activation and inactivation of anticancer drugs (Rodriguez-Antona and Ingelman-Sundberg [2006], Timofeeva et al. [2009]). In particular, most interestingly for our application, some specific CYPs have been shown to be over-expressed in lung cancer cells and are promising drug target candidates: CYP1A1, CYP2A6, CYP2E1 (Oyama et al. [2007]), CYP2A6, CYP2B6, CYP3A4/5 (Rodriguez-Antona and Ingelman-Sundberg [2006]), or CYP1B1 (Su et al. [2009]). DrugBank search revealed that the above mentioned drugs (excluding tracazolate), function as inhibitors of these particular CYPs connected to lung cancer. Specifically, resveratrol inhibits CYP3A4, luteolin inhibits CYP1A1 and CYP1B1, daunorubicin inhibits CYP1A1, CYP3A4, CYP3A5, vorinostat inhibits CYP3A4, bromocriptine inhibits CYP3A4, rolitetracycline inhibits CYP3A4, and doxorubicin inhibits CYP2B6, CYP3A4 and CYP1B1. All the drugs that target the lung cancer related CYPs are highlighted in yellow in Figure 5.5. We can see that they are mainly concentrated in or around the black box of similar drugs. These arguments support the hypothesis that our algorithm detected the similarity between the lung cancer related CYP inhibitors.

5.3.4 Discussion

Using an example from genomics, we showed how our method can be applied to identify possible new therapeutics for gene related diseases, such as cancer. We analysed a subset of a previously analysed dataset of lung cancer gene signatures, in order to verify the performance of our method. The calculations yielded very satisfying results highly correlated with the ground truth. The limitation of our method is in its calculation time requirements, due to the many rank permutations involved in the procedure. Hence, very long lists typical for genomics data (> 1000) cannot, at this stage, be handled in a reasonable time frame. Nevertheless, the calculation time could be sped up by applying more efficient optimisation procedures, further parallelisation, as well as implementing the method in a more efficient programming language, such as C++.

Chapter 6

Discussion

This chapter summarises our novel method for generalised meta-analysis with its main strengths and limitations. Future research is also suggested.

6.1 Summary of the problem

Combining scientific evidence with meta-analysis is an efficient way of strengthening results, dealing with study-related biases, resolving contradicting conclusions, increasing statistical power, and, ultimately, answering additional research questions. Generally, a study consists of a number of objects or individuals, where each object is assigned a certain measured value. Ideally, without the presence of errors, when an experiment is repeated or replicated, one would obtain the same values. The reality, however, shows that this is not the case. The observed values are very likely to differ due to various errors, and sometimes even result in contradicting conclusions. Meta-analysis aims to reveal the underlying signal, which is assumed to lie behind the studies, in other words the signal which would be observed in error-free experiments. The existing methods for meta-analysis can be usually applied only on data of the same type and nature. Methods that can combine various data types are limited to rank aggregation. Rank-based methods are invariant to transformations and normalisation, and robust to outliers. Nevertheless, when using rank-based methods, the magnitude of the observed values is lost and, as a result, only the order of the values of the underlying signal can be recovered.

6.2 Key findings

Our aim was to benefit from the flexibility of the rank-based approaches, while achieving the same goals as classical meta-analysis, i.e. recovering not only the order, but also the values of the underlying signals themselves. To this end we have developed a method for estimating the normalised signal underlying multiple ranked lists. These lists can be rankings of any values where it makes sense to order them from 'best' to 'worst', whether the values are gene expression or response rates. It is of crucial importance that the studies or assessments are comparable in the sense that the definition of 'best' and 'worst' is the same in all studies. Naturally, our method can be also applied on ranked lists which are not based on known values, for example preference rankings.

Our method assumes a fixed-effect-type model of the underlying vector of signals θ

$$X_{ij} = \theta_i + Z_{ij}, \quad i = 1, \ldots, p, j = 1, \ldots, n,$$

where p is the number of objects, n the number of studies (or, generally, assessors), and Z is a matrix of normal random variables. Because the signals θ can be values on any scale between minus and plus infinity, our aim was to estimate the normalised values of θ. The signals θ were not restricted with any distributional assumptions. The rank positions in the observed ranked lists were presumed to be rank positions of the values in X. Using this model, we iteratively converged to an estimate of normalised θ, thereby reconstructing the observed ranked lists. In each iteration, a $\hat{\theta}$ and $\hat{Z}$ was chosen, and $\hat{X}$ was calculated and ranked. Then the rank distribution of its rank positions was evaluated and compared to the rank distribution of the originally observed ranks. This way we could quantify the proximity of our estimate $\hat{\theta}$ to the unknown true θ and either stop the algorithm, if close enough, or proceed with further iterations. The iterative optimisation part of our algorithm was done with Bayesian Monte Carlo sampling, and also with a non-parametric genetic algorithm.

We obtained a consensus ranking simply by ranking the estimated values. We estimated the standard errors connected to each signal value by applying a non-parametric bootstrap. Using these error estimates, we assessed the stability of the rank estimates.

The algorithm was programmed in R with two choices of optimisation algorithm: genetic algorithm (applying the package `GA`) and Metropolis Markov chain

Monte Carlo (MCMC). The implementation of the algorithm was described, and the pseudocode and implementation details were provided. We found that the most computationally time consuming part of the algorithm was the choice of the variance of the random variables $\hat{Z}$ in each iteration (Section 4.2.3.2).

We examined the method on simulated data, addressing various issues, such as the number of objects/studies, the amount of agreement between the studies, and the choice of optimisation technique. The results confirmed our expectations that the signal estimates improve with an increased number of studies, and with increased agreement between them. Both chosen optimisation methods provided good estimates, nevertheless the standard errors were more reliably estimated when using the MCMC optimisation.

On a toy example we explained how the method can be used in practical problems. Additionally, we analysed a clinical dataset consisting of 111 randomised controlled trials that evaluated the patients' responses to 12 antidepressants. After imputing missing values with multiple imputation, our method was able to estimate the relative efficacy of the drugs and yielded results similar to odds ratios calculated with the network meta-analysis meta-regression method. We also analysed a genomic dataset of 50 drugs and their gene-altering lung cancer similarity scores. The results correctly (in agreement with the original paper) identified 10 drugs that caused the most antagonistic gene changes in comparison to lung cancer gene changes.

6.3 Strengths and limitations

The main advantage of the method is that it is not data-specific, unlike standard meta-analytical methods. It does not matter which device or method was used to produce the values, nor what scale they were defined on. It also does not matter whether the values describe a biological, chemical, economic, or behavioral process. The method is general and can be applied in any field where the ranking of objects is a logical way of comparing them.

The main disadvantage of the method is its high computational demand, originating in the complexity of the algorithm, as well as the rather large amount of iterations required. The computational time can be improved, however, by (i) optimising the algorithmic steps of the core function, (ii) finding a more efficient optimisation method, and (iii) distributing the calculation to a computing cluster. Our

method cannot handle missing values, nevertheless data imputation methods work to complete the dataset, as we have shown on a real-data example.

6.4 Outlook

Future research should be focused on computational optimisation, in order to allow for excessive testing and possible algorithmic improvements. In particular, one could experiment with other, more efficient, stochastic optimisation procedures, such as simulated annealing (Kirkpatrick et al. [1983]) and its variations (Szu and Hartley [1987], Ingber [1989], Ingber et al. [2012]), parallel tempering (Swendsen and Wang [1986]), the cross-entropy method (Rubinstein [1999]), or stochastic tunneling (Wenzel and Hamacher [1999]). The core function for evaluating estimates could be also sped up, especially by finding a faster way of choosing the variance of the random variables $\hat{Z}$ in each iteration.

The method could be potentially generalised to the case of not only one, but several (up to n) underlying signals. Specifically, the model could be generalised for situations where not all studies share the same underlying signal, i.e. the above model would be a random-effect-type model

$$X_{ij} = \Theta_{ij} + Z_{ij}, \quad i = 1, \ldots, p, j = 1, \ldots, n,$$

where Θ_{ij} is an underlying signal for object i and study j. Such a generalisation would permit differentiation between groups of studies that were initiated by the same process, in other words groups of studies that share the same underlying signal, i.e. for which $\Theta_{\cdot\cdot}$ is identical. One could estimate the groups iteratively using, for example, an expectation-maximisation algorithm (Dempster et al. [1977]). The method could then be used as a clustering algorithm.

6.5 Closing remarks

Combining studies with meta-analysis can yield large improvements compared to existing results, as well as answer additional research questions. The main contribution of this thesis is a novel rank-based method for combining studies or assessments, independent of data types, and applicable across fields. Simulations have shown that the method reflects the signal present in the data very well. Despite the disadvantage of high computational demands, real-world examples demonstrated that the

method can be successfully used as a meta-analytical tool in biomedical research.

R code

In this chapter, the R code is given for all newly written function used in this thesis (script MultiRankS.R). The genetic algorithm optimisation was ran using the R package `GA`. The usage of the functions is shown on examples.

6.6 Functions' description

- `norm_vec`: normalises a vector in order to have unit norm
- `st.error.estim`: calculates standard error
- `mean_terr`: calculates standard error and the mean of the values
- `F_perm`: calculates rank probability matrix for window height of specified maximum
- `J_theta`: evaluates the objective function for a specified probability matrix
- `FRgeneral`: given a vector and input matrix, calculates a rank matrix most similar to the input one
- `what.is.suitable.sigma`: estimates a standard deviation needed to create the most similar rank matrix (see `FRgeneral`)
- `MCMC.metropolis`: runs `onerun` function as many times as there are specified initial guesses (`theta.inis`). Distributes the run into a specified number of cores.
- `onerun`: runs the function `run_metropolis_MCMC` and summarises its results
- `run_metropolis_MCMC`: runs classical Metropolis-Hastings algorithm with a specified fixed step size

- `proposalfunction`: calculates a next step proposal in Metropolis-Hastings algorithm

6.7 The code - MultiRankS.R

```
library(gtools)
library(reshape)
library(grid)
library(parallel)
%library(adaptMCMC)

norm_vec = function(x) sqrt(sum(x^2)) # vector normalisation

st.error.estim = function(x){sqrt(sum((x-mean(x))^2/(length(x)-1)))} # standard error estimation

mean_terr <- function(x) { # estimation of mean and 2SE from a numeric vector x
  data.frame("y" = mean(x), "ymin" = mean(x) - 2*st.error.estim(x), "ymax" = mean(x) + 2*st.error.estim(x))
}

F_perm <- function(R, l_max){
# Calculates a probability matrix for each window size up to l_max. Outputs a list of all the probability matrices.
# R - matrix of rankings, objects in rows, assessors in columns
# l_max - maximum window height

  p=nrow(R) # number of objects
  n=ncol(R) # number of assessors
  F.l = list() # list of matrices F, each for different l
  for (ell in 1:l_max)
  {
    s_mat <- unname(as.matrix(do.call(expand.grid,rep(list(seq_len(p)),ell))))
    F_perm <- matrix(NA_real_,p-ell+1,nrow(s_mat))
  for (sri in seq_len(nrow(s_mat))) {
    s <- s_mat[sri,];
    F_perm[,sri] <- rowSums(Reduce('&',Map(function(e,i) R[i:(p-length(s)+i),]<=e,s,seq_along(s))))/n
```

```
    }
    F.l[[ell]] = F_perm
    names(F.l)[[ell]] = paste("l=",ell,sep="")
  }
  return(F.l = F.l)
}

J_theta = function(Fi, F.temp, l_max)
{
# Calculates the value of the objective function J.
# Fi - the input list of probability matrices (calculated
   from the observed rank matrix)
# F.temp - another list of probability matrices (calculated
   from a current estimate)
  SumSl = 0
  Suml = 0
  for (ell in 1:l_max)
  {
    for (sl in 1:ncol(Fi[[ell]]))
    {
      F1 = Fi[[ell]][,sl]
      F2 = F.temp[[ell]][,sl]
      SumSl = SumSl + (F1 - F2)%*%(F1 - F2)
    }
    Suml = Suml + SumSl
  }
  Suml = Suml / l_max
  return(Suml)
}

FRgeneral <- function(theta, l_max, R.input, F.input,
   increments=NULL)
{
# Calculates a rank matrix R.temp, based on a numerical
   vector theta. Matrix R.temp should resemble the input rank
    matrix R.input. The resemblance is achieved by generating
    a 'suitable' st.dev. with function what.is.suitable.sigma
   (). Outputs the rank matrix R.temp, its probability
   matrices F.temp and the value of J.theta.
# theta - a numerical vector (current estimate)
# l_max - maximum window size
# R.input - input rank matrix
```

```
64 # F.input - input list of probability matrices of R.input
65 # increments - see what.is.suitable.sigma() function
66
67   if (is.null(increments)) increments=0.01
68   p = nrow(R.input); n = ncol(R.input)
69   sigma = what.is.suitable.sigma(theta,R.input,increments)
70   X = matrix(nrow=p, ncol=n)
71   X = apply(X, 2, function(x) theta + rnorm(p, mean=0, sigma)
      )
72   R.temp = apply(X, 2, function(x) rank(-x, ties.method='
      random'))
73   F.temp = F_perm(R.temp,l_max)
74   J.theta = J_theta(F.input, F.temp, l_max)
75   return(list(J = J.theta, F = F.temp, R = R.temp))
76 }
77
78 what.is.suitable.sigma <- function(theta, R.input, increments
      =NULL)
79 {
80 # Estimates a st.dev. needed to create a rank matrix as
      similar to R.input as possible, using the vector theta.
      Outputs the most suitable sigma for theta.
81 # theta - a numerical vector (current estimate)
82 # R.input - input rank matrix
83 # increments - the size of the increments of sigmas tested
      for suitability. Larger increments makes the algorithm
      faster, but tests less sigma-possibilities. Should be
      adjusted depending on the number of objects p (for p~10
      increments = 0.05 are satisfactory)
84
85   p = nrow(R.input); n = ncol(R.input)
86   if (is.null(increments)) increments = 0.01
87   corInput = cor(R.input, method='spearman')
88   rho = median(corInput[lower.tri(corInput)]) # average
      correlation
89   sigma_range = seq(0.01,1, by = increments) # testing what
      correlation will be caused by each of the sigmas
90   rho.temp=numeric()
91   Xs = list() # list of attribute matrices, as produces by
      using the individual sigmas from sigma_range
92   for (sig in 1:length(sigma_range))
93   {
```

```
94      Xs[[sig]] = apply(matrix(nrow=p, ncol=n),2,function(x) x=
        theta + rnorm(p, mean=0, sd=sigma_range[sig]))
95    }
96    corm=Map(function(x) cor(x,method='spearman'), Xs)
97    rho.temp = sapply(corm, function(x) median(x[lower.tri(x)])
       )
98    sigma = sigma_range[which(abs(rho.temp-rho)==min(abs(rho.
       temp-rho)))][1]
99    return(sigma)
100 }
101
102 objective.fun <- function(theta, n, p, l_max, R.input, F.
       input, increments=NULL)
103 {
104 # Calculates the value of objective function
105 # theta - numeric vector (an estimate)
106 # n - number of assessors/studies
107 # p - number of objects
108 # l_max - height of the window
109 # R.input - the observed rank matrix
110 # F.input - the distribution of the observed rank matrix R.
       input
111 # increments - see what.is.suitable.sigma()
112   if (is.null(increments)) increments=0.01
113   sigma = what.is.suitable.sigma(theta,R.input,increments)
114   Z_ij = matrix(nrow=p, ncol=n)
115   Z_ij = apply(Z_ij, 2, function(x) rnorm(p, mean=0, sigma))
116   X = apply(Z_ij, 2, function(x) theta + x)
117   R = apply(X, 2, function(x) rank(-x, ties.method='random'))
118   F.temp = F_perm(R,l_max)
119   J.theta = J_theta(F.input, F.temp, l_max)
120   return(-J.theta)
121 }
122 #-----------------------------------------------------------
123 #
124 #        Classical Metropolis-Hastings MCMC algorithm
125 #
126 #-----------------------------------------------------------
127
128 MCMC.metropolis <- function(theta.inis, in.data, dev, chain.
       len, cores = detectCores(), increments=NULL, l_max){
```

```
129 # Runs several independent Metropolis MCMC chains (function
      onerun()). Number of chains depends on the length of
      initial guesses theta.inis. Outputs list of the results (
      chain + best estimate + minimum found) and runtime
130 # theta.inis - a list of vectors of initial guesses, the
      length of the list determines the number of chains
131 # in.data - list of 2: first rank matrix R, second list of
      matrices F (result of F_perm)
132 # dev - the standard deviation defining every next proposal
      step of the random walk
133 # chain.len - length of each chain (number of steps of the
      random walk)
134 # cores - number of cores to be used, if not specified, all
      cores available are used
135 # increments - see what.is.suitable.sigma() function
136
137   if(is.null(increments)) increments = 0.05
138   if(is.null(l_max)) l_max = 2
139   # clusterApply() for Windows
140   if (Sys.info()[1] == "Windows"){
141   cl <- makeCluster(cores)
142   clusterExport(cl, list("onerun", "run_metropolis_MCMC", "
      FRgeneral", "what.is.suitable.sigma", "F_perm", "J_theta",
       "proposalfunction"))
143   runtime <- system.time({
144      res = clusterApplyLB(cl=cl, x=theta.inis, fun=onerun,
      input=in.data, dev=dev, its=chain.len, l_max=l_max,
      increments)
145   })[3]
146   stopCluster(cl)
147   # mclapply() for everybody else
148   } else {
149     runtime <- system.time({
150       res = mclapply(X=theta.inis, FUN=function(x) onerun(
      theta.ini=x, input=in.data, dev=dev, its=chain.len, l_max=
      l_max, increments=increments), mc.cores=cores)
151     })[3]
152   }
153   return(list(avg=res, runtime=runtime))
154 }
155
```

```r
onerun <- function(theta.ini, input, dev=NULL, its, l_max, increments=NULL){
# Runs one chain of the Metropolis MCMC (function run_metropolis_MCMC()) and finds the minima. Outputs all found vectors where the minimum was found (x.in.min), all points of the chain(res.chain), and the value of the minimum (minJ).
# theta.ini - the initial guess (numeric vector)
# input - list of 2: first rank matrix R, second list of matrices F (result of F_perm)
# dev - the standard deviation defining every next proposal step of the random walk
# its - length of the chain (number of steps)
# increments - see what.is.suitable.sigma() function
  if(is.null(dev)) dev=0.1
  if(is.null(increments)) increments=0.05
  if(is.null(l_max)) l_max=2
  res.chain = list()
  mins = list()
  x.at.mins = list()
  acceptance = list()
  p = nrow(input[[1]])
  Ri = input[[1]]
  Fi = input[[2]]
  res.chain = run_metropolis_MCMC(theta.ini, iterations=its, dev=dev, Ri=Ri, Fi=Fi, l_0=l_max, increments)
  acceptance = 1-mean(duplicated(t(res.chain$chain)))
  if (acceptance < 0.15) warning(paste0('The acceptance rate is below 15% (',acceptance,'). The step size probably needs to be smaller.'))
  if (acceptance > 0.5) warning(paste0('The acceptance rate is above 50% (',acceptance,'). The step size probably needs to be larger.'))
  mins = which(res.chain$Js == min(res.chain$Js))
  x.at.mins = res.chain$chain[,mins]
  if (is.matrix(x.at.mins))
  { all.x.in.mins = as.data.frame(x.at.mins)
  dupl = apply(all.x.in.mins,1,duplicated) # find duplicated columns
  all.x.in.mins = all.x.in.mins[,which(apply(dupl,1,function(x) sum(x)) !=p)] # keep only unique columns
  }
```

```
184   else all.x.in.mins = x.at.mins
185   return(list(x.in.min=all.x.in.mins, chain=res.chain, minJ=
       min(res.chain$Js), acceptance=acceptance))
186 }
187
188 run_metropolis_MCMC <- function(theta.ini, iterations, dev,
       Ri, Fi,l_0,increments=NULL){
189 # Calculates Metropolis MCMC chain. Outputs the chain (all
       visited points) and the values of J in each point.
190 # theta.ini - initial point (numerical vector)
191 # iterations - number of mcmc steps (length of the chain)
192 # dev - the standard deviation defining every next proposal
       step of the random walk
193 # Ri - input rank matrix
194 # Fi - input list of probability matrices of Ri
195 # l_0 - maximum height of the window
196 # increments - see what.is.suitable.sigma() function
197   if (is.null(dev)) dev=0.1
198   if (is.null(increments)) increments=0.05
199   if (is.null(l_0)) l_0=2
200   p=nrow(Ri); n=ncol(Ri)
201   chain = array(dim = c(length(theta.ini),iterations+1)) #
       array of points I visit
202   F.for.chain = list() # saving the F values (might be needed
       for some reason)
203   Js = numeric() # J for each point I visit
204   chain[,1] = theta.ini
205   FR_ini = FRgeneral(theta.ini, l_0, Ri, Fi, increments) #
       calculates the F and J for the initial point
206   F.for.chain[[1]] = FR_ini$F
207   Js[1] = FR_ini$J
208
209   for (i in 1:iterations){
210     proposal = proposalfunction(param=chain[,i], stdev=dev) #
       create next point
211     FR_proposal = FRgeneral(theta=proposal, l_max=l_0, R.
       input=Ri, F.input=Fi,increments) # calculates the F and J
       for a current point
212     J_proposal = FR_proposal$J
213     probab = min(1,exp(-(J_proposal - Js[i]))) # the
       probability of acceptance
```

```
    if (runif(1) < probab){ # accept the new point with
   probability probab
      chain[,i+1] = proposal
      Js[i+1] = J_proposal
      F.for.chain[[i+1]] =  FR_proposal$F
    }else{ # otherwise stay in the current point
      chain[,i+1] = chain[,i]
      Js[i+1] = Js[i]
      F.for.chain[[i+1]] =  F.for.chain[[i]]
    }
  }
  return(list(chain=chain, Js=Js))
}

proposalfunction <- function(param, stdev){
# Calculates a proposal of the next step in MCMC algorithm
# param - a numerical vector (current point)
# stdev - st.dev. of the random values added the previous
   point to create the next point

  e = rnorm(length(param),mean = 0, sd = stdev)
  return(param+e)
}
```

6.8 Example of usage

Assuming we have an input matrix of ranks `R.input`, where objects are in rows and assessors in columns, we can calculate the signal estimate, using either classical Metropolis-Hastings MCMC (see pseudocode 2) or genetic algorithm (see pseudocode 1), as follows.

Prepare data:

```
source('MultiRankS.R')
F.input = F_perm(R=R.input, l_max=2)
num.chains = 10 # number of chains
theta.inis = vector('list',num.chains)# initial guesses - one
    for each chain
theta.inis = lapply(theta.inis, function(x) x= runif(p, -1,1)
   )
```

Metropolis-Hastings MCMC:

```
MCMC.result = MCMC.metropolis(theta.inis=theta.inis, in.data=
   list(R.input, F.input), l_max=2, dev=0.1, chain.len=5000,
   cores = detectCores())
```

Genetic algorithm:

```
library(GA)
GA.result = ga(type="real-valued", fitness = function(x)
   objective.fun(x, n=ncol(R.input), p=nrow(R.input),l_max =
   2,F.input = F.input,R.input = R.input), popSize = 10, min
   = rep(-1,nrow(R.input)), max = rep(1,nrow(R.input)),
   pmutation = 0.3, maxiter = 250, parallel = TRUE, keepBest
   = TRUE, suggestions = t(matrix(unlist(theta.inis),ncol(R.
   input),nrow(R.input))), monitor=plot)
```

List of abbreviations

BUGS	Bayesian inference using Gibbs sampler
CEMC	Cross entropy Monte Carlo
CI	Confidence interval
CYP	Cytochrome P450
dbGaP	Genotypes and phenotypes database
DE	Differential expression / differentially expressed
DNA	Deoxyribonucleic acid
EGA	European genome-phenome archive
FEM	Fixed–effect model
GA	Genetic algorithm
GC	Guanine-cytosine
GEO	Gene expression omnibus
GWAS	Genome-wide association study
JAGS	Just another Gibbs samples
MAD	Median absolute deviation
MANGA	Meta-analysis of new generation antidepressants
MAMLE	Moving average maximum likelihood estimator
MCEM	Monte Carlo expectation-maximisation
MCMC	Markov Chain Monte Carlo
MedSE	Median of standard errors
NGS	Next-generation sequencing
NMA	Network meta-analysis
OEA	Order Explicit Algorithm
OEA.k	OEA with Kendall's distance
OEA.s	OEA with Spearman's distance
OR	Odds ratio
RCT	Randomised controlled trial
REM	Random–effect model

RNA	Ribonucleic acid
RPKM	Reads per kilobase of gene length per million reads
RRHO	Rank-rank hypergeometric overlap
SE	Standard error
SNP	Single nucleotide polymorphism
SRA	Sequence read archive
SVM	Support vector machine

List of contributions

Publications

1. Švendová, V., and Schimek, M. G. (2017). A novel method for estimating the common signals for consensus across multiple ranked lists. *Computational Statistics and Data Analysis*, Volume 115, November 2017, pp. 122–135. `https://doi.org/10.1016/j.csda.2017.05.010`.

2. Lal, R., Lind, K., Heitzer, E., Ulz, P., Aubell, K., Kashofer, K., Middeke, J.M., Thiede, C., Schulz, E., Rosenberger, A., Hofer, S., Feilhauer, B., Rinner, B., Svendova, V., Schimek, M.G., Rücker, F.G., Hoefler, G., Döhner, K., Zebisch, A., Wölfler, A. and Sill, H. (2017). Somatic TP53 mutations characterize preleukemic stem cells in acute myeloid leukemia. *Blood*.

3. Bettermann, K., Kuldeep Mehta, A., Hofer, E.M., Wohlrab, C., Golob-Schwarzl, N., Svendova, V., Schimek, M.G., Stumptner, C., Thüringer, A., Speicher, M.R., Lackner, C., Zatloukal, K., Denk, H., Haybaeck, J. (2016). Keratin 18-deficiency results in steatohepatitis and liver tumors in old mice: A model of steatohepatitis-associated liver carcinogenesis. *Oncotarget*.

4. Schimek, M., Budinská, E. , Kugler, K.G., Švendová, V., Ding, J. and Lin, S. (2015). TopKLists: a comprehensive R package for statistical inference, stochastic aggregation, and visualization of multiple omics ranked lists. *Statistical Applications in Genetics and Molecular Biology*, 14(3) 311–316.

5. Schimek, M.G., Svendova, V. and Bloice, M.D. (eds.) (2015) Book of Abstracts ISNPS Meeting "Biosciences, Medicine, and novel Non-Parametric Methods", Graz 2015. ISBN 978-3-200-04319-0 (e-publication).

Posters, presentations and Proceedings

1. Schimek, MG., Svendova, V. (2016). Presentation: A distribution function approach for signal reconstruction from ranking data. *3rd Conference of the International Society for Non-Parametric Statistics*, Avignon, Italy.

2. Svendova, V., Schimek, M.G., Hall, P. (2015). Presentation: Estimating the underlying authority in multiple ranked lists. *8th International Workshop on Simulation*, Vienna, Austria.

3. Schimek, M.G., Svendova, V. (2015). Presentation and Proceedings paper: Novel methods for the statistical analysis of multiple and repeated rankings. *60th ISI World Statistics Congress*, Rio de Janeiro, Brazil.

4. Svendova, V., Schimek, M.G. (2014). Poster: The performance of recent statistical methods for inference and integration of ranked omics data, *27th International Biometric Conference*, Florence, Italy.

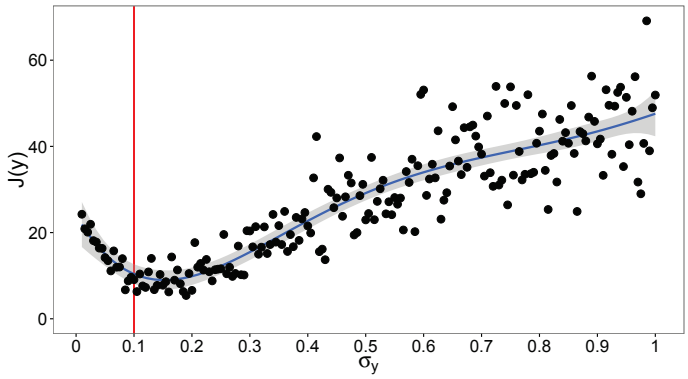
5. Schimek, M.G., Hall, P., Švendová, V. (2014). Presentation: Inference and modeling aspects of multiple ranked lists. *Joint Statistical Meeting*, Boston.

6. Schimek, M.G., Budinska, E., Švendová, V. (2013). Presentation: How to find consolidated top elements in omics ranked lists? *6th International Conference of the ERCIM WG on Computational and Methodological Statistics*, London.

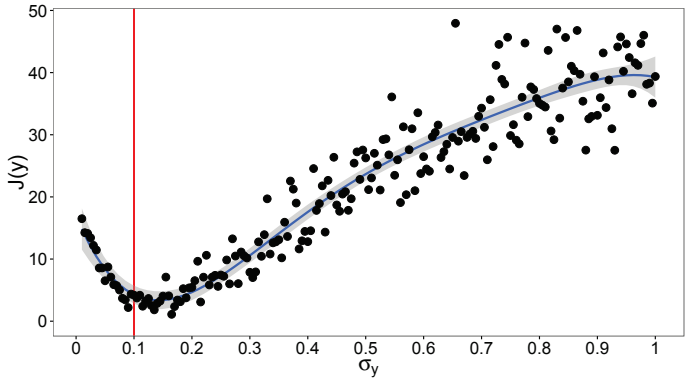
7. Schimek, M.G., Bloice, M., Svendova, V. (2013). Presentation: The analysis of time course ranking data by nonparametric inference. *7th International Workshop on Simulation*, Rimini, Italy.

8. Švendová, V. and Schimek, M.G. (2013). Poster and Proceedings paper: Search for top-k consensus objects in multiple ranked lists: TopKInference versus several recent procedures. *In Proceedings of the 59th ISI World Statistics Congress*, Hong Kong.

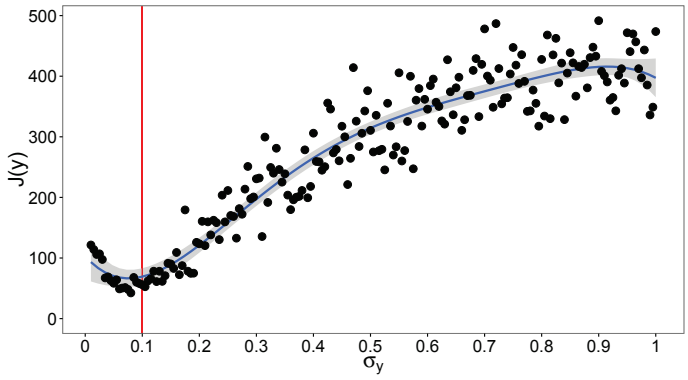
Figures



(a) Settings 1,2,3,11,12 and 17

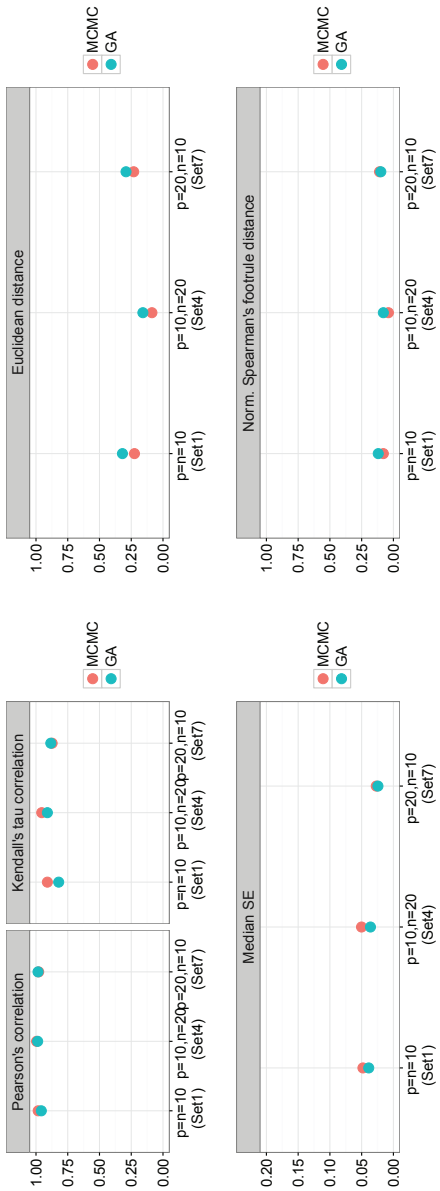


(b) Settings 4,5,6,13,14 and 18



(c) Settings 7,8,9,15 and 16

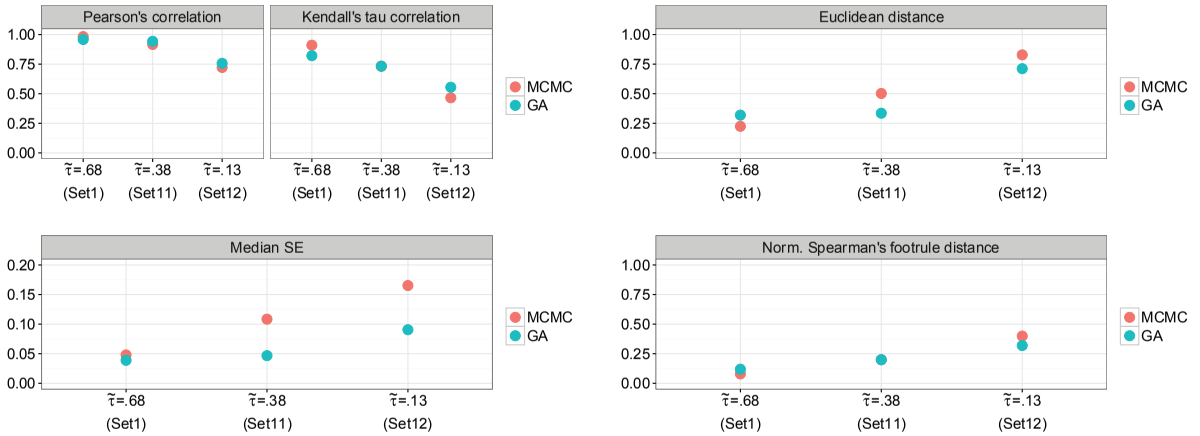
Figure 6.1: Dependence of the objective function on random error added to the true signal. The expected minimum is at 0.1. Figures adapted from Švendová and Schimek [2017].



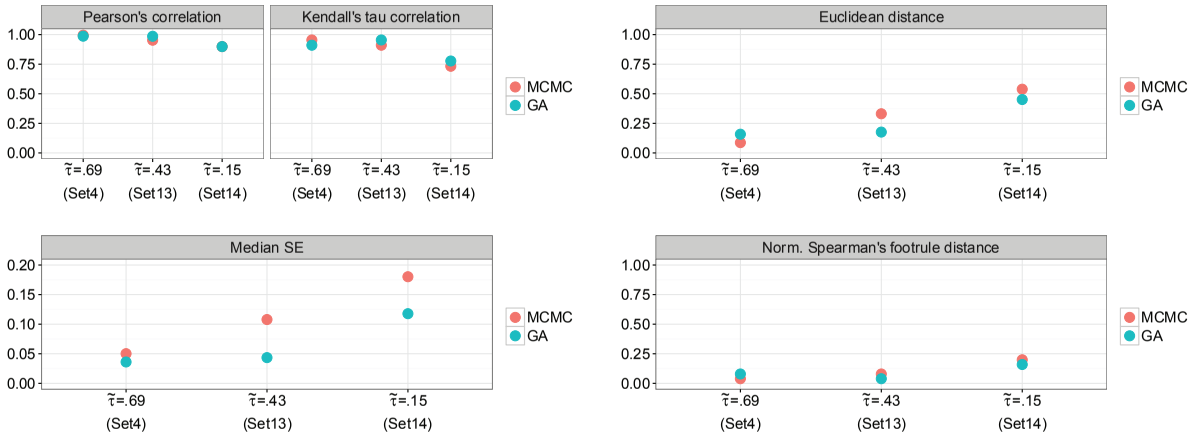
(a) Comparison: **number of objects** (p) **vs. number of assessors** (n).

(b) Comparison: **window size** Comparison between settings with $l_{\max} = 2$ vs. $l_{\max} = 3$.

Figure 6.2: Measures comparing the estimated and true signal I.

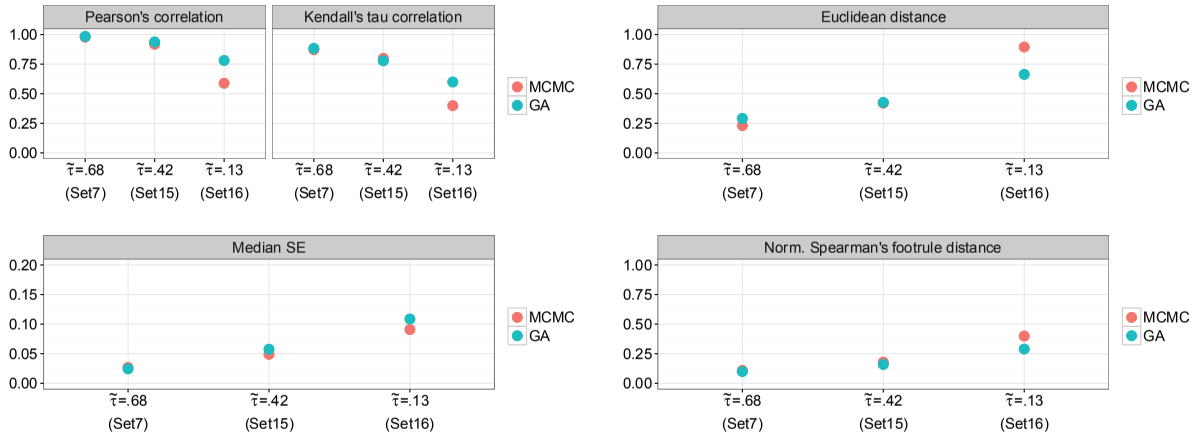


(a) Comparison: **levels of correlation between the input ranking,** $n = p$ Comparison between high-, mid- and low- correlation between the assessors, when the number of objects equal to the number of assessors.

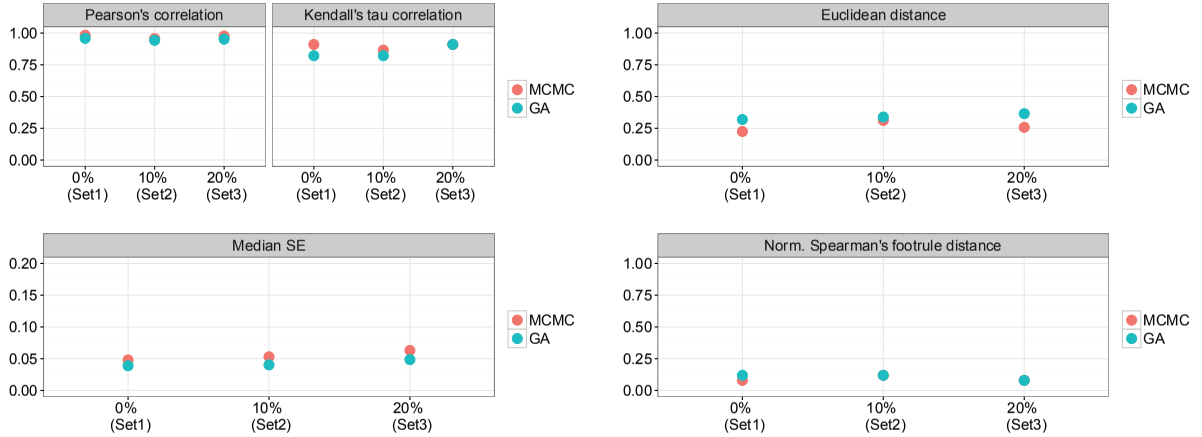


(b) Comparison: **levels of correlation between the input ranking,** $n > p$ Comparison between high-, mid- and low- correlation between the assessors, when the number of objects is smaller than the number of assessors.

Figure 6.3: Measures comparing the estimated and true signal II.

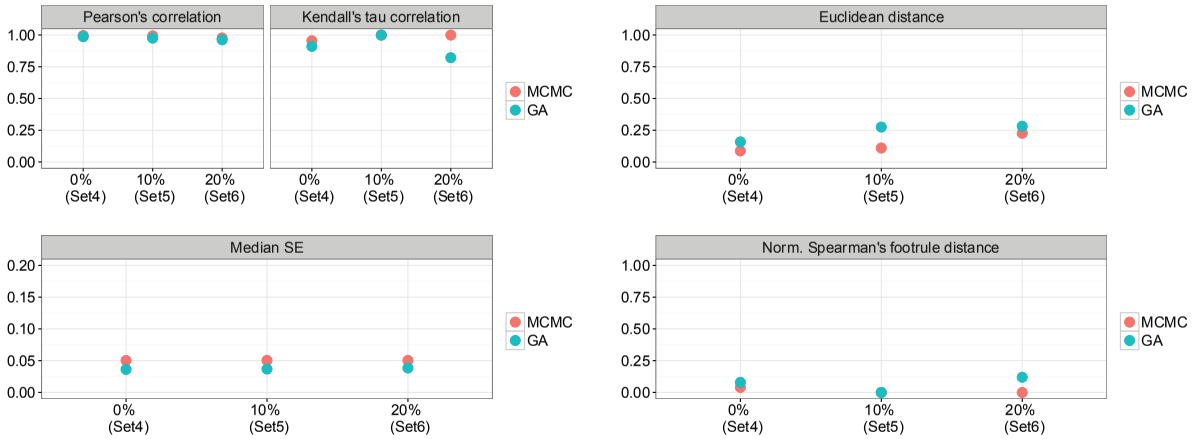


(a) Comparison: **levels of correlation between the input ranking,** $p > n$ Comparison between high-, mid- and low- correlation between the assessors, when the number of objects is higher than the number of assessors.

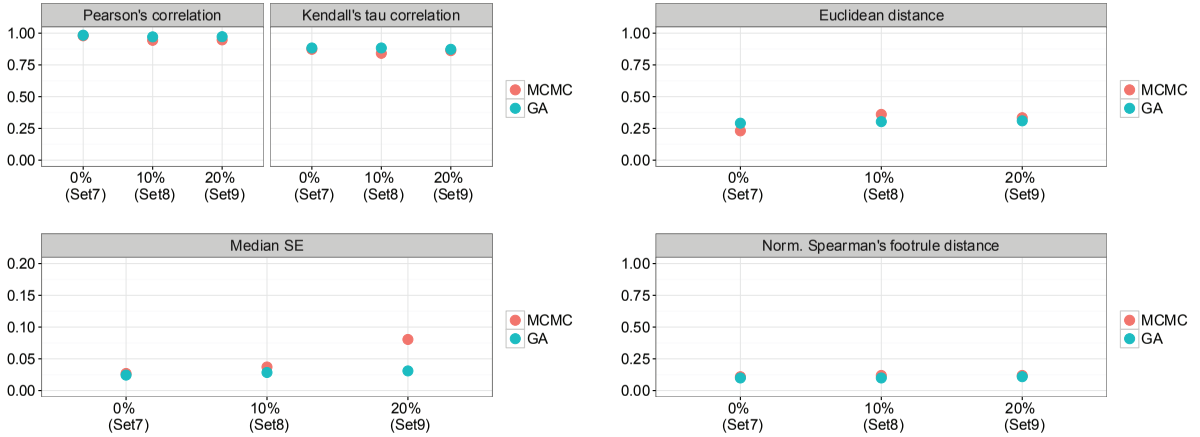


(b) Comparison: **bad experts,** $p = n$ Comparison between 0%, 10% and 20% of bad experts, when the number of objects is equal to the number of assessors.

Figure 6.4: Measures comparing the estimated and true signal III.

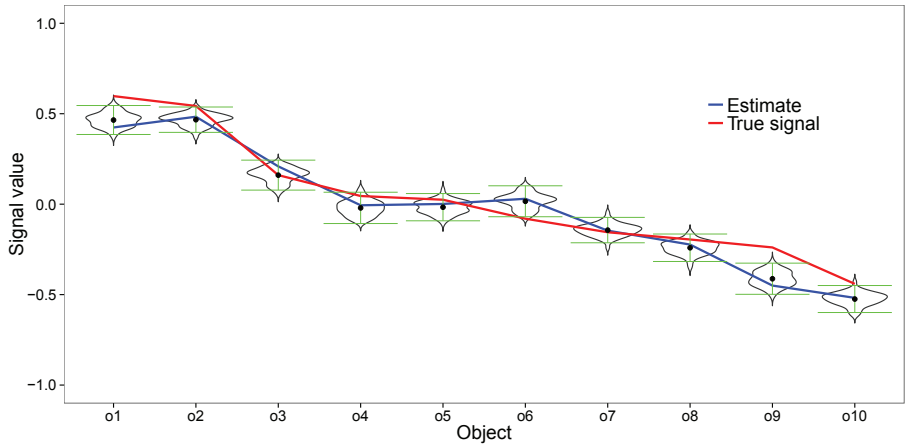


(a) Comparison: **bad experts,** $n > p$ Comparison between 0%, 10% and 20% of bad experts, when the number of objects is smaller than the number of assessors.

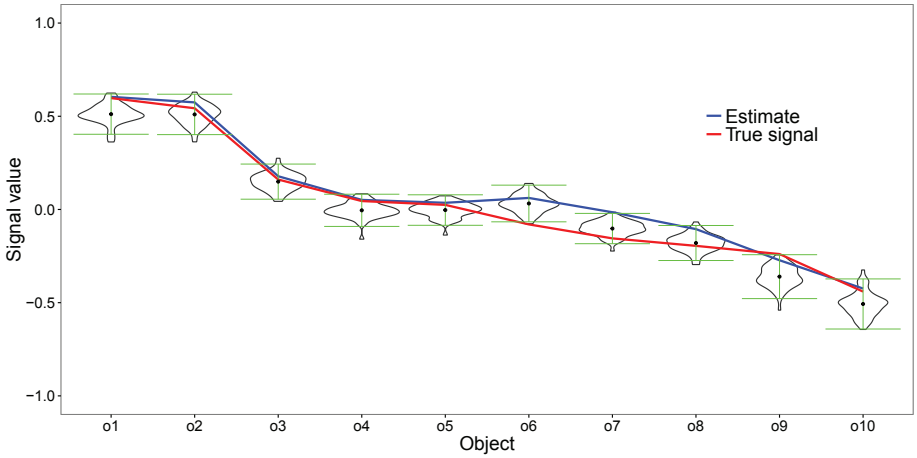


(b) Comparison: **bad experts,** $p > n$ Comparison between 0%, 10% and 20% of bad experts, when the number of objects is higher than the number of assessors.

Figure 6.5: Measures comparing the estimated and true signal IV.

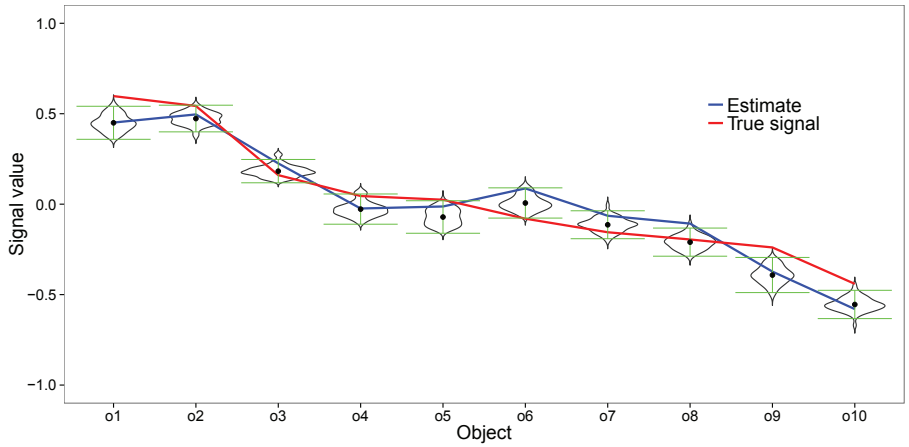


(a) GA: Settings 1

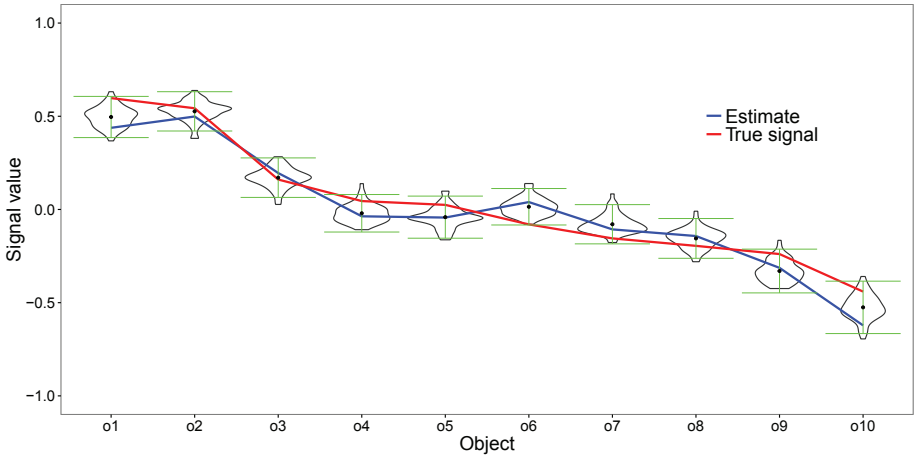


(b) MCMC: Setting 1. Figure from Švendová and Schimek [2017].

Figure 6.6: Setting 1: $n = p = 10, l_{\max} = 2$, bad experts=0, $\tilde{\rho}/\tilde{\tau} = 0.84/0.68$

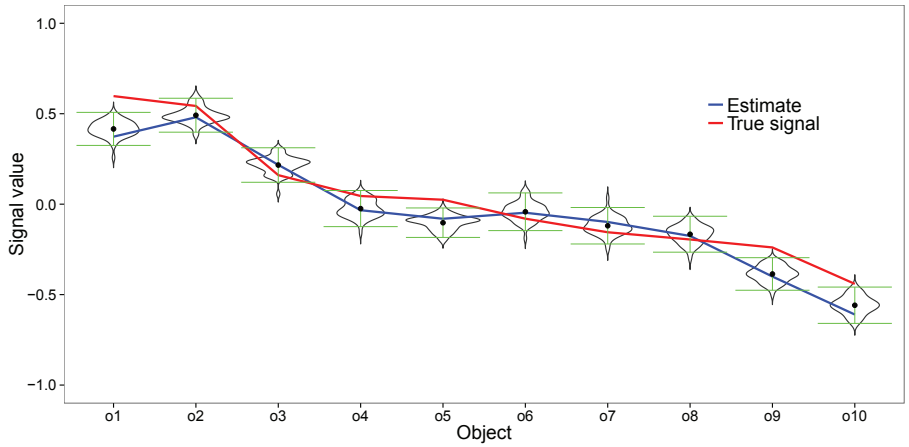


(a) GA: Settings 2

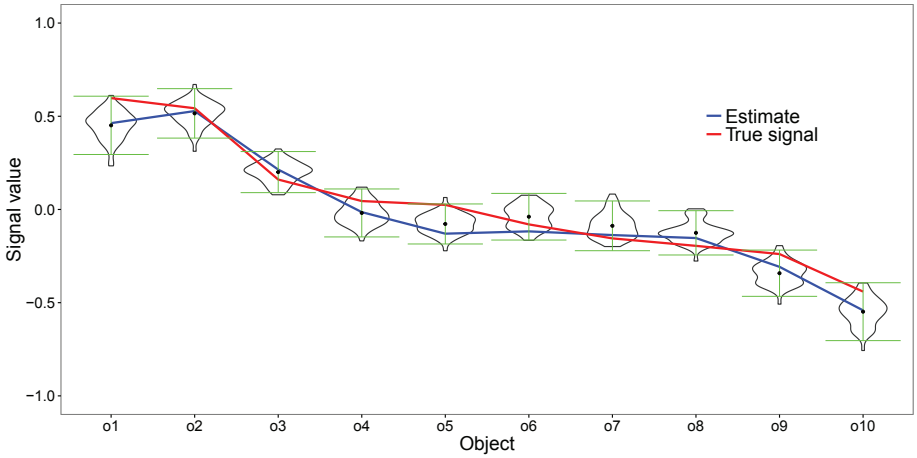


(b) MCMC: Setting 2. Figure from Švendová and Schimek [2017].

Figure 6.7: Setting 2: $n = p = 10, l_{\max} = 2$, bad experts=1, $\tilde{\rho}/\tilde{\tau} = 0.82/0.64$

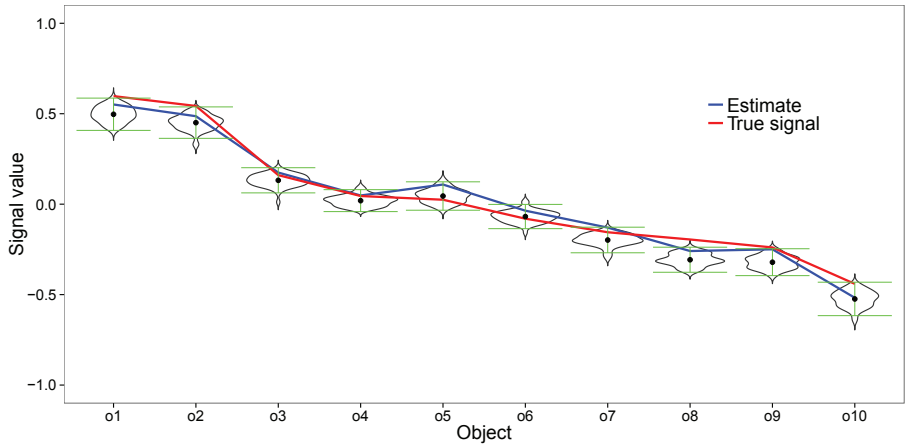


(a) GA: Settings 3

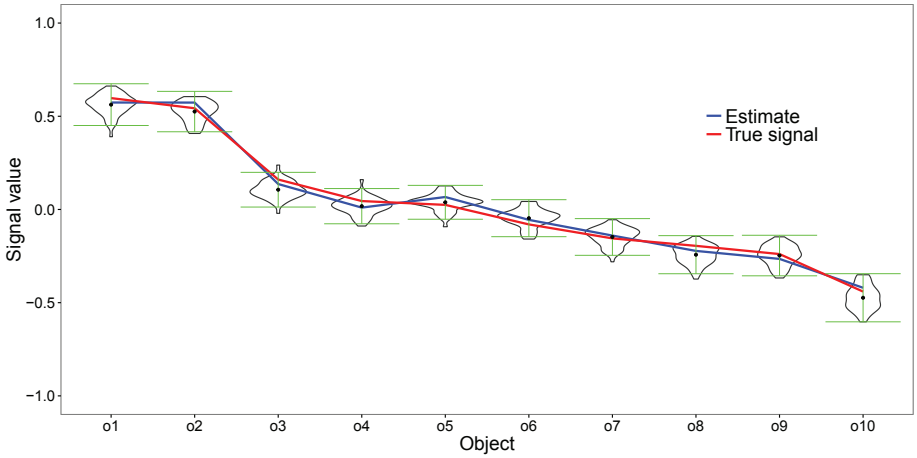


(b) MCMC: Setting 3. Figure from Švendová and Schimek [2017].

Figure 6.8: Setting 3: $n = p = 10, l_{\max} = 2$, bad experts=2, $\tilde{\rho}/\tilde{\tau} = 0.75/0.6$

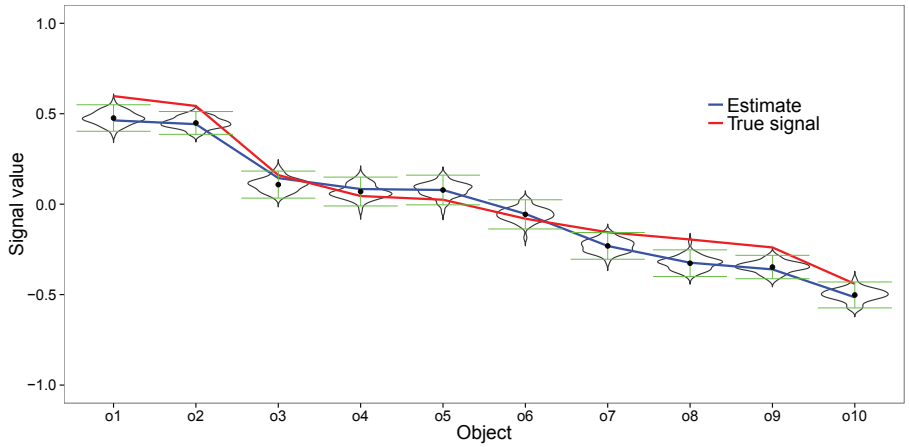


(a) GA: Settings 4

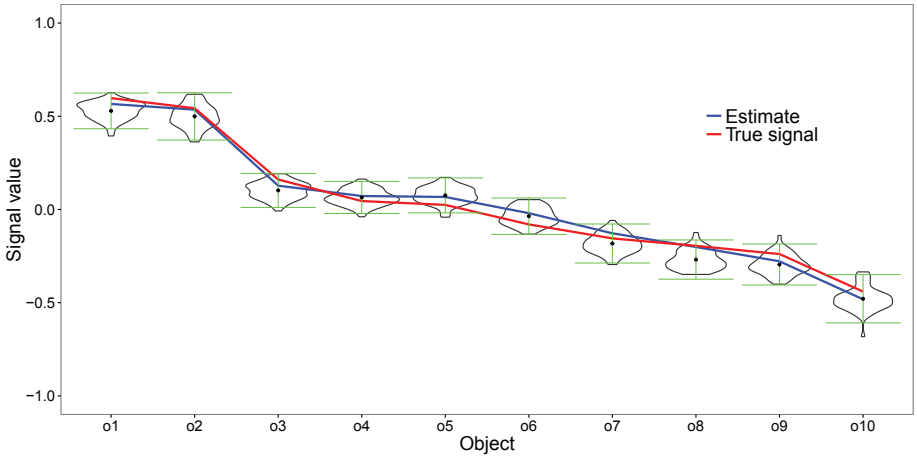


(b) MCMC: Setting 4. Figure from Švendová and Schimek [2017].

Figure 6.9: Setting 4: $n = 20, p = 10, l_{\max} = 2$, bad experts=0, $\tilde{\rho}/\tilde{\tau} = 0.84/0.69$

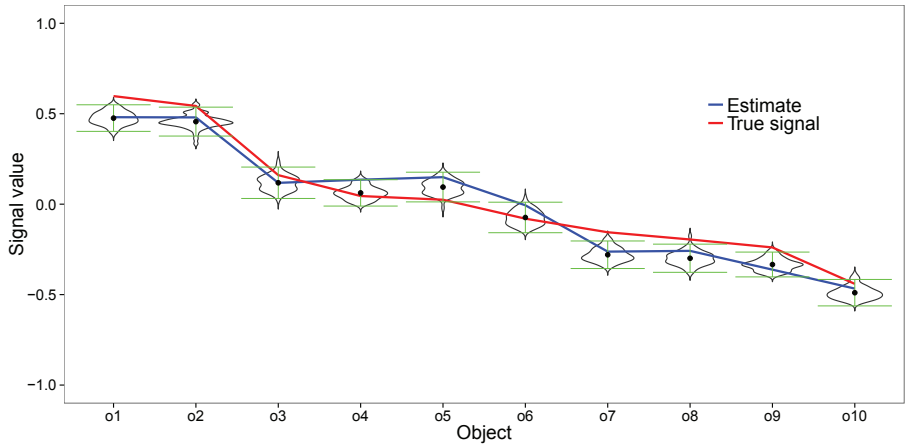


(a) GA: Settings 5

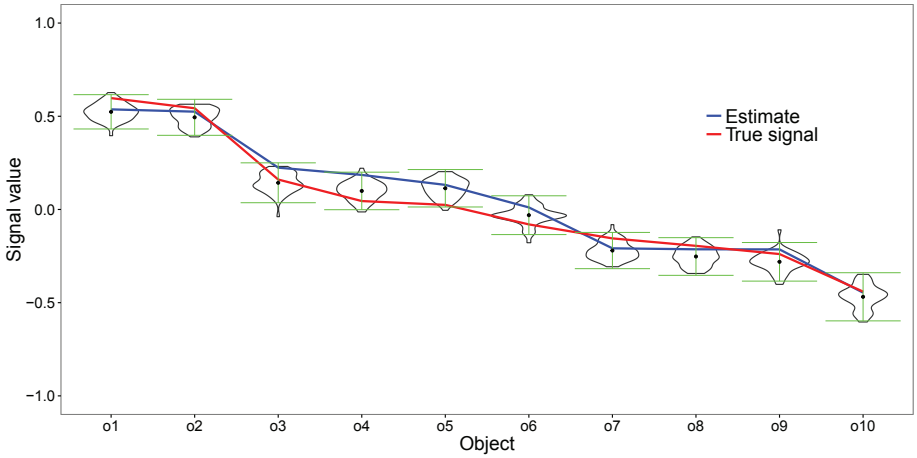


(b) MCMC: Setting 5. Figure from Švendová and Schimek [2017].

Figure 6.10: Setting 5: $n = 20, p = 10, l_{\max} = 2$, bad experts=2, $\tilde{\rho}/\tilde{\tau} = 0.82/0.64$

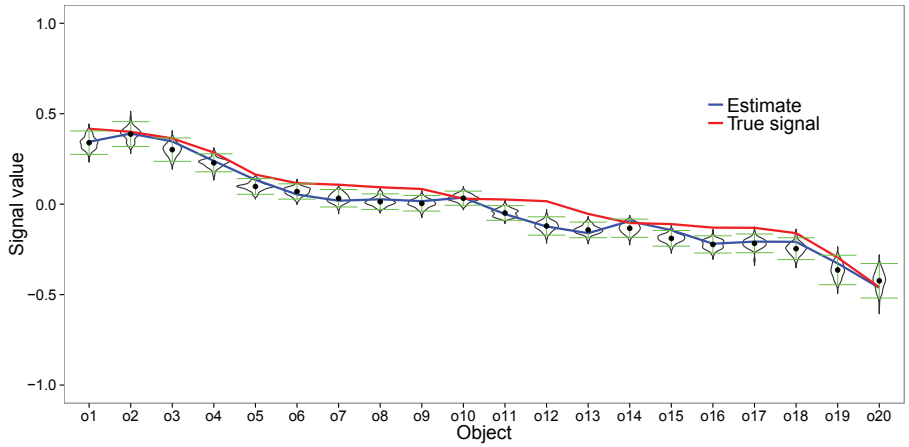


(a) GA: Settings 6

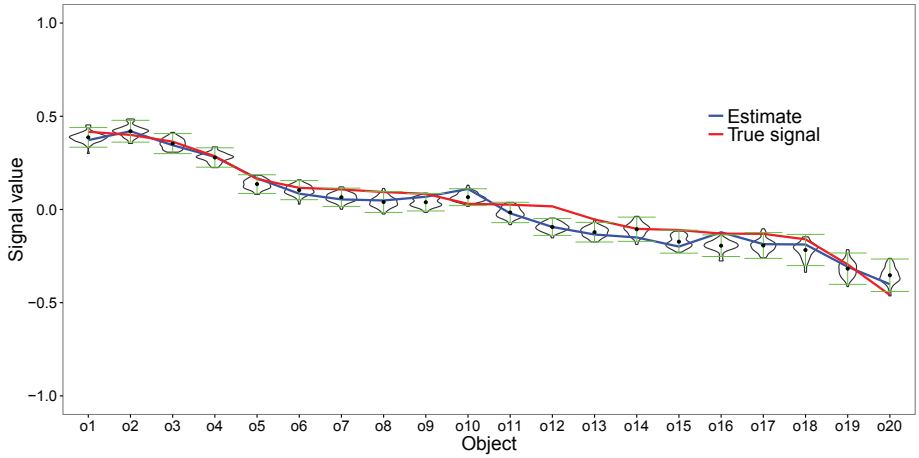


(b) MCMC: Setting 6. Figure from Švendová and Schimek [2017].

Figure 6.11: Setting 6: $n = 20, p = 10, l_{\max} = 2$, bad experts=4, $\tilde{\rho}/\tilde{\tau} = 0.77/0.62$

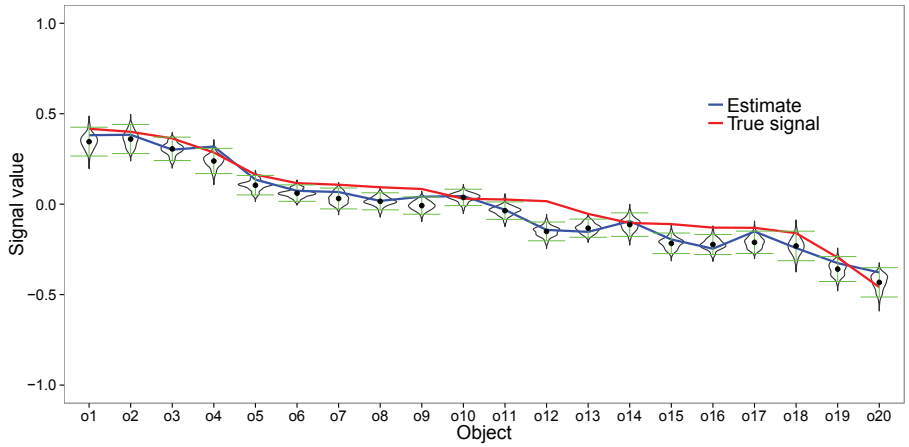


(a) GA: Settings 7

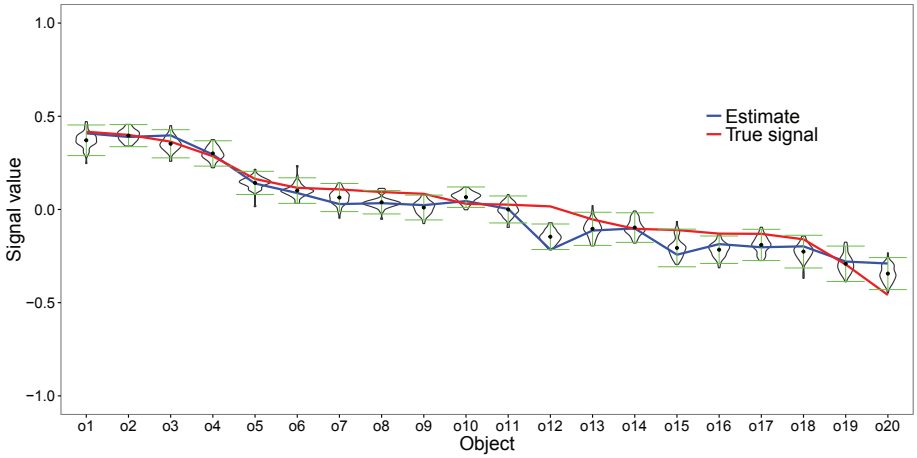


(b) MCMC: Setting 7. Figure from Švendová and Schimek [2017].

Figure 6.12: Setting 7: $n = 10, p = 20, l_{\max} = 2$, bad experts=0, $\tilde{\rho}/\tilde{\tau} = 0.84/0.68$

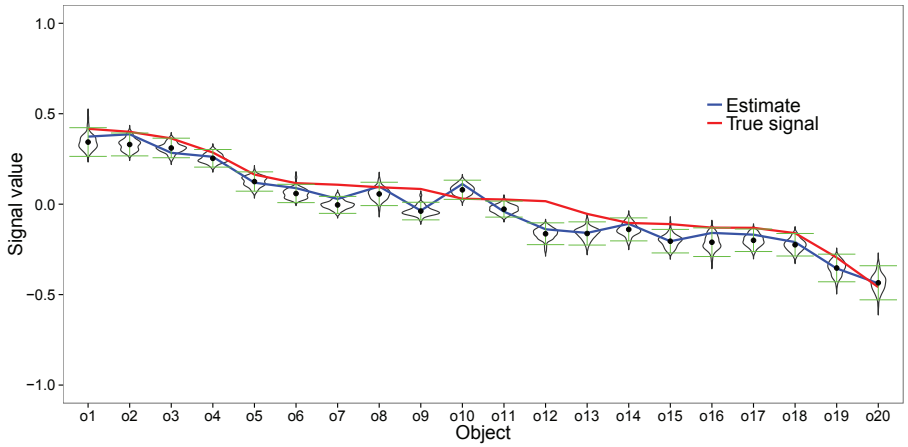


(a) GA: Settings 8

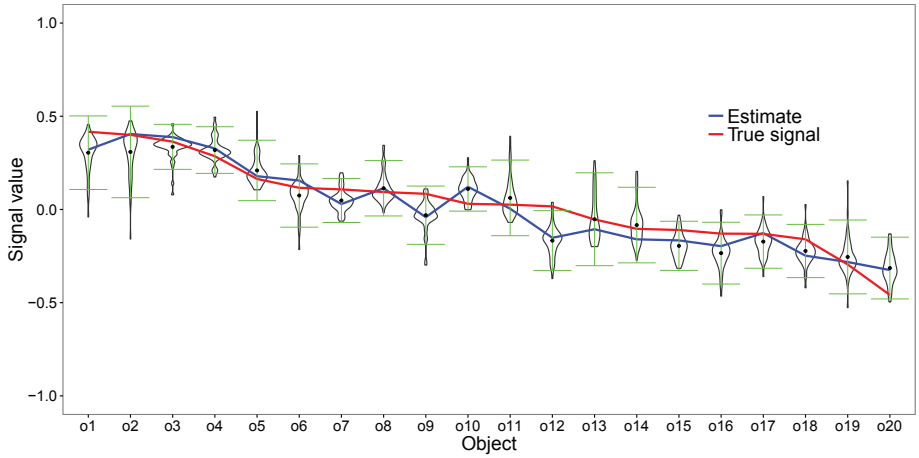


(b) MCMC: Setting 8. Figure from Švendová and Schimek [2017].

Figure 6.13: Setting 8: $n = 10, p = 20, l_{\max} = 2$, bad experts=1, $\tilde{\rho}/\tilde{\tau} = 0.82/0.65$

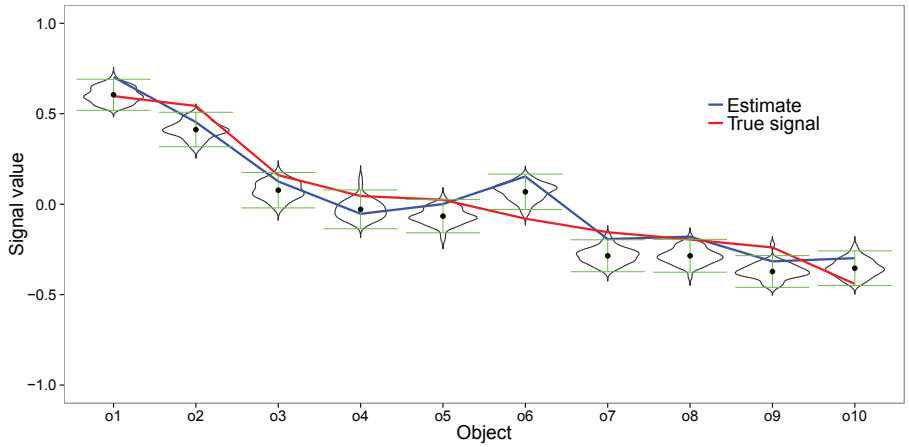


(a) GA: Settings 9

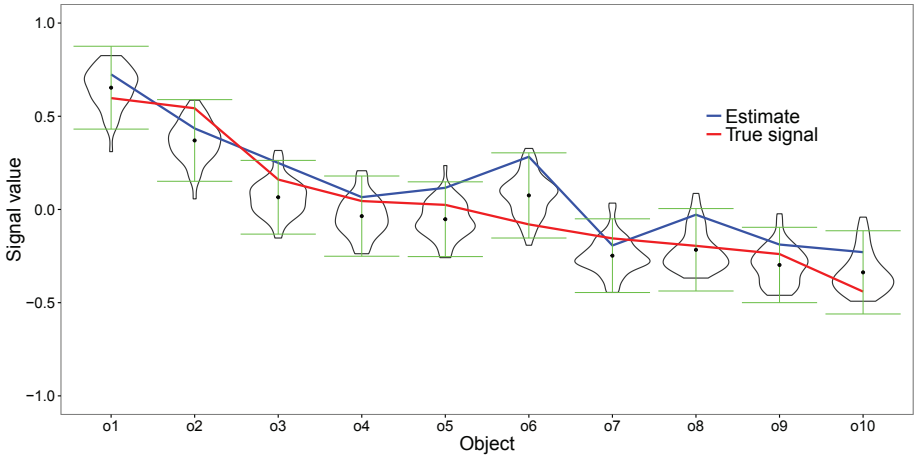


(b) MCMC: Setting 9. Figure from Švendová and Schimek [2017].

Figure 6.14: Setting 9: $n = 10, p = 20, l_{\max} = 2$, bad experts=2, $\tilde{\rho}/\tilde{\tau} = 0.79/0.61$

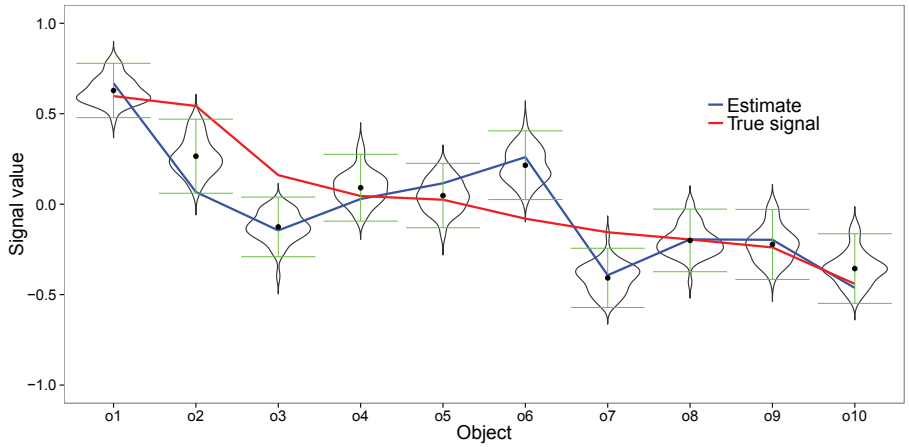


(a) GA: Settings 11

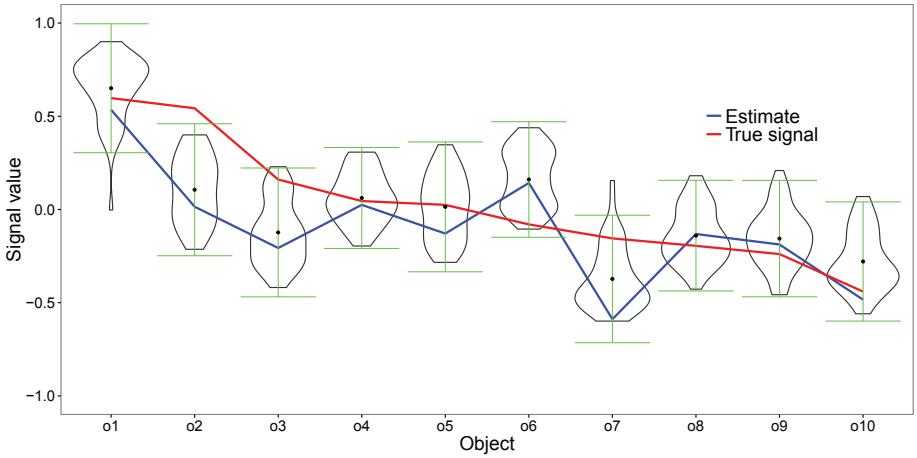


(b) MCMC: Setting 11

Figure 6.15: Setting 11: $n = p = 10, l_{\max} = 2$, bad experts=0, $\tilde{\rho}/\tilde{\tau} = 0.56/0.38$

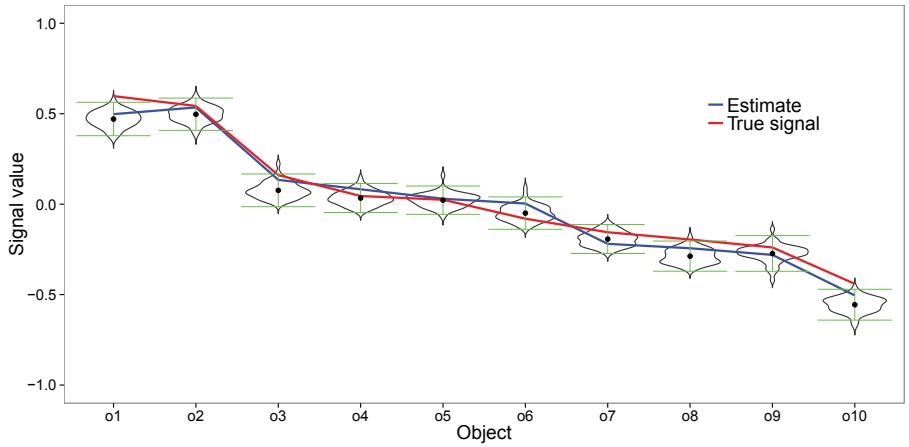


(a) GA: Settings 12

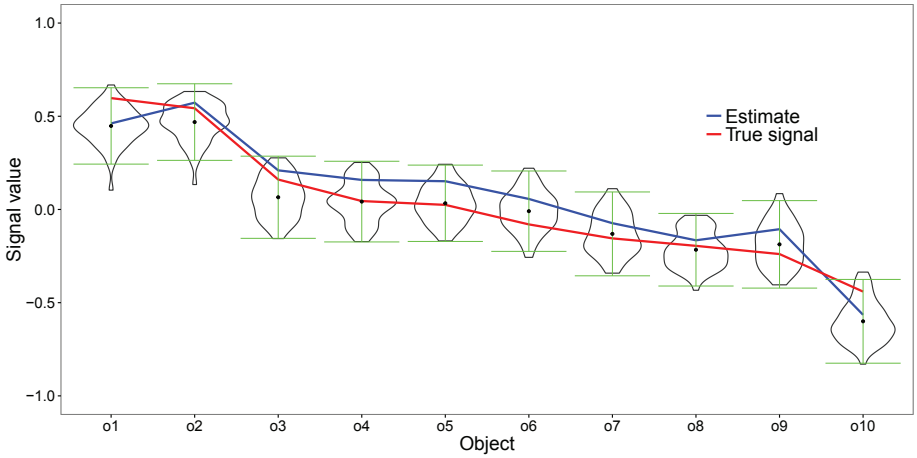


(b) MCMC: Setting 12

Figure 6.16: Setting 12: $n = p = 10, l_{\max} = 2$, bad experts=0, $\tilde{\rho}/\tilde{\tau} = 0.18/0.13$

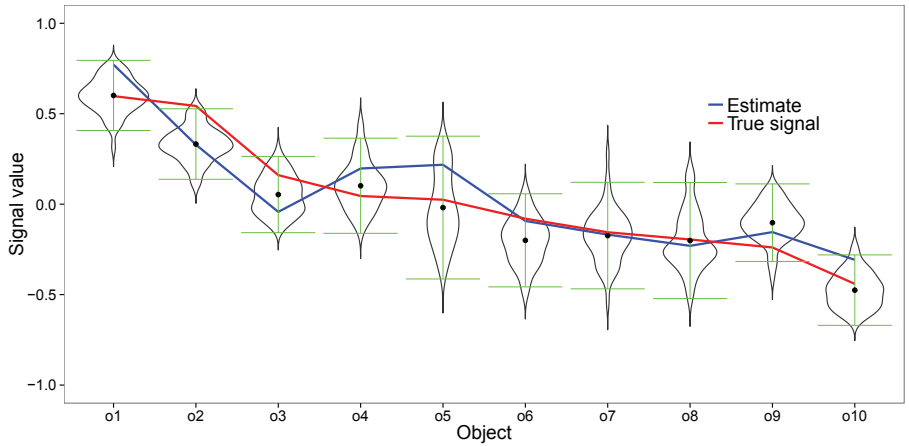


(a) GA: Settings 13

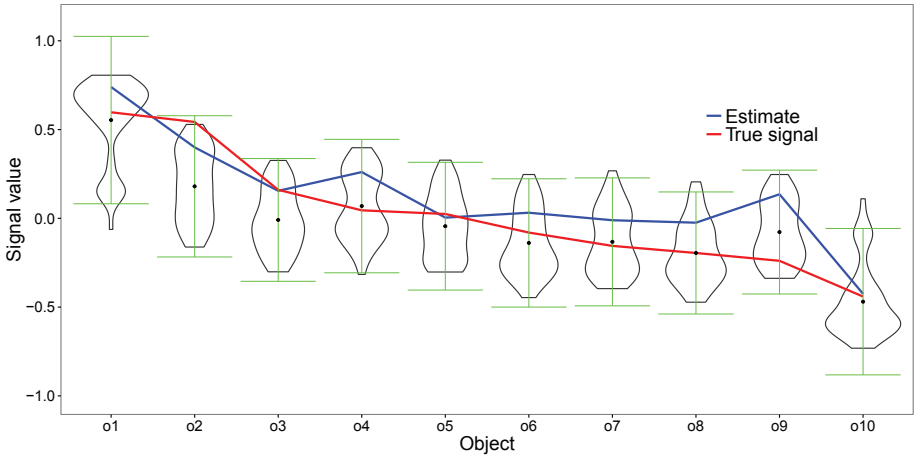


(b) MCMC: Setting 13

Figure 6.17: Setting 13: $n = 20, p = 10, l_{\max} = 2$, bad experts=0, $\tilde{\rho}/\tilde{\tau} = 0.57/0.42$

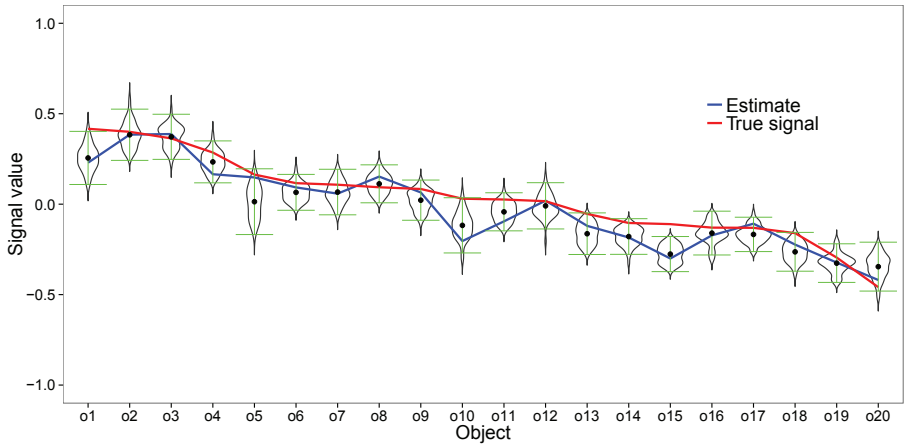


(a) GA: Settings 14

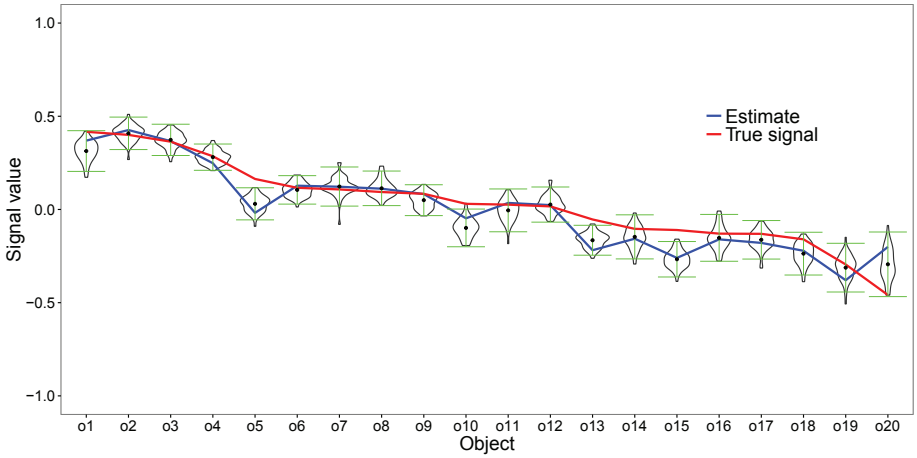


(b) MCMC: Setting 14

Figure 6.18: Setting 14: $n = 20, p = 10, l_{\max} = 2$, bad experts=0, $\tilde{\rho}/\tilde{\tau} = 0.18/0.15$

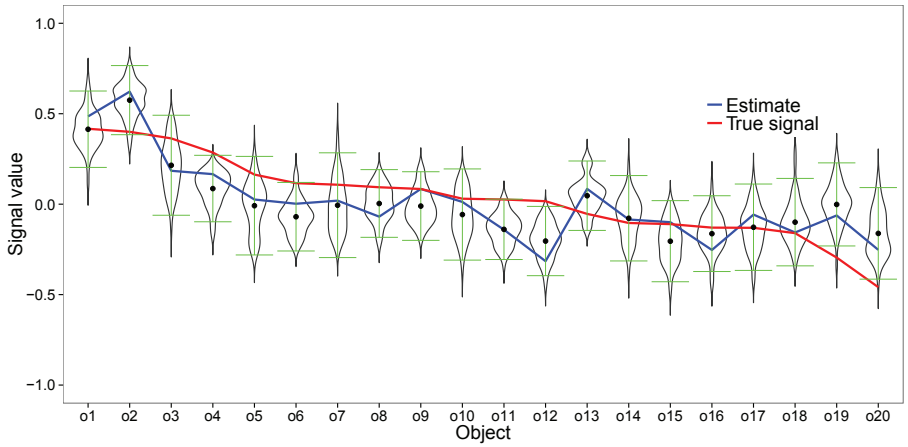


(a) GA: Settings 15

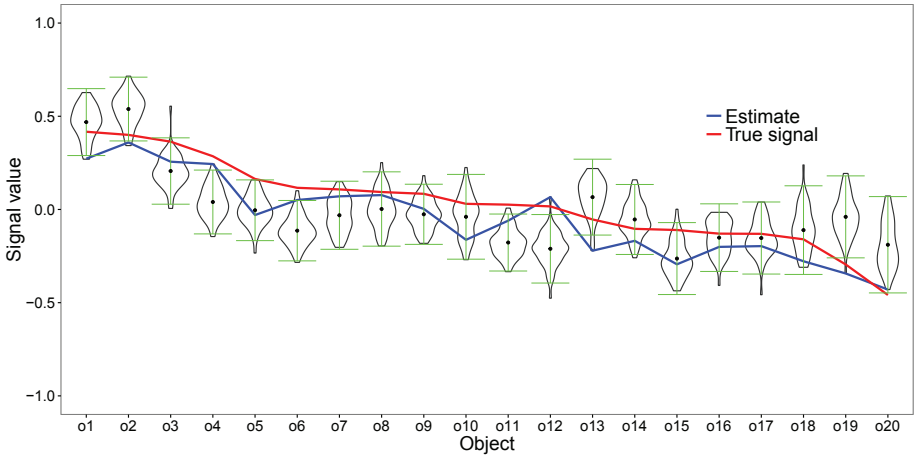


(b) MCMC: Setting 15

Figure 6.19: Setting 15: $n = 10, p = 20, l_{\max} = 2$, bad experts=0, $\tilde{\rho}/\tilde{\tau} = 0.56/0.42$

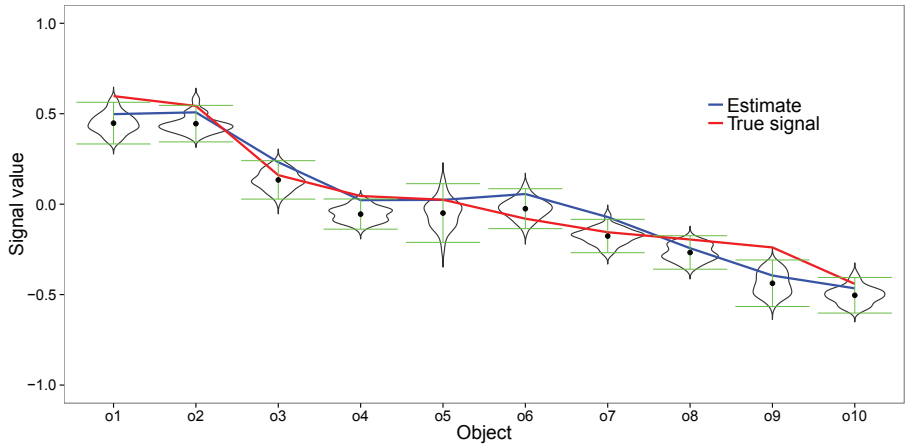


(a) GA: Settings 16

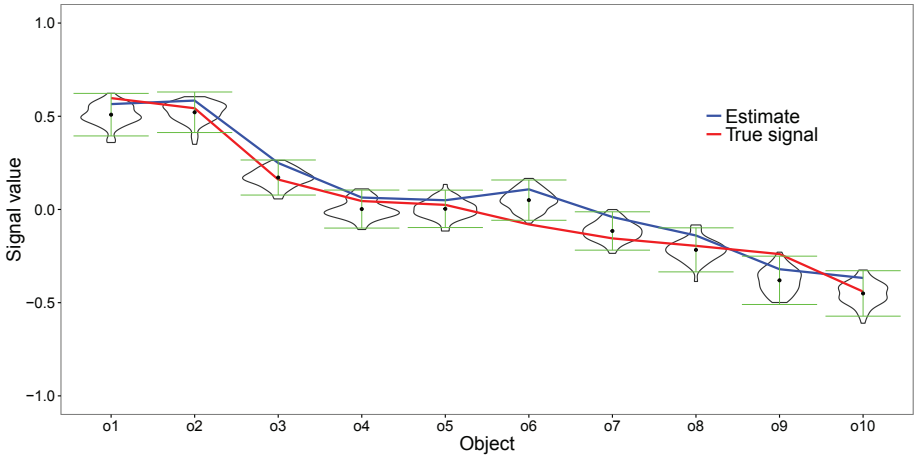


(b) MCMC: Setting 16

Figure 6.20: Setting 16: $n = 10, p = 20, l_{\max} = 2$, bad experts=0, $\tilde{\rho}/\tilde{\tau} = 0.18/0.13$

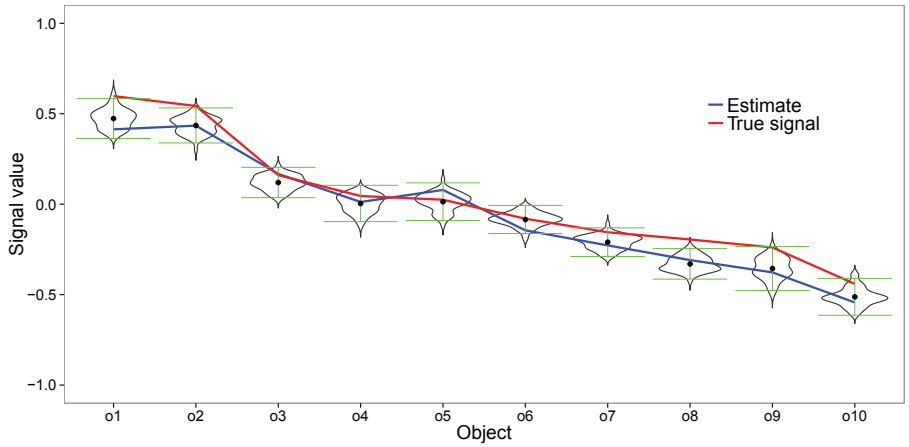


(a) GA: Settings 17

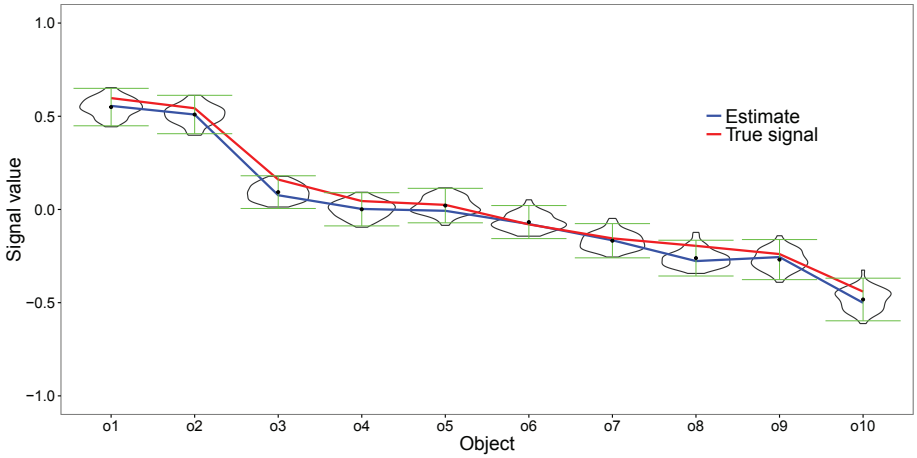


(b) MCMC: Setting 17

Figure 6.21: Setting 17: $n = p = 10, l_{\max} = 3$, bad experts=0, $\tilde{\rho}/\tilde{\tau} = 0.84/0.68$



(a) GA: Settings 18



(b) MCMC: Setting 18

Figure 6.22: Setting 18: $n = 20, p = 10, l_{\max} = 3$, bad experts=0, $\tilde{\rho}/\tilde{\tau} = 0.84/0.69$

References

Airy, G. B. (1861). *On the Algebraical and Numerical Theory of Errors of Observations and the Combination of Observations*. Macmillan&Company, London.

Aleksic, J., Carl, S. H., and Frye, M. (2014). Beyond library size: a field guide to NGS normalization. *bioRxiv*, page 006403.

Alvo, M. and Yu, P. L. (2014). *Statistical Methods for Ranking Data*. Springer, New York.

Bafeta, A., Trinquart, L., Seror, R., and Ravaud, P. (2014). Reporting of results from network meta-analyses: methodological systematic review. *BMJ*.

Baggerly, K. A. (1995). *Visual Estimation of Structure in Ranked Data*. PhD thesis, University of Texas at Dallas.

Berkey, C., Hoaglin, D., Antczak-Bouckoms, A., Mosteller, F., and Colditz, G. (1998). Meta-analysis of multiple outcomes by regression with random effects. *Statistics in Medicine*, 17(22):2537–2550.

Böckenholt, U. (2001). Mixed-effects analyses of rank-ordered data. *Psychometrika*, 66(1):45–62.

Böckenholt, U. (2006). Thurstonian-based analyses: Past, present, and future utilities. *Psychometrika*, 71(4):615–629.

Bradley, R. A. and Terry, M. E. (1952). Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Breitling, R., Armengaud, P., Amtmann, A., and Herzyk, P. (2004). Rank products: a simple, yet powerful, new method to detect differentially regulated genes in replicated microarray experiments. *FEBS Letters*, 573(1-3):83–92.

Brooks, S. P. and Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455.

Bucher, H. C., Guyatt, G. H., Griffith, L. E., and Walter, S. D. (1997). The results of direct and indirect treatment comparisons in meta-analysis of randomized controlled trials. *Journal of Clinical Epidemiology*, 50(6):683–691.

Bumgarner, R. (2013). Overview of DNA microarrays: types, applications, and their future. *Current Protocols in Molecular Biology*, pages 22–1.

Carlin, B. P., Hong, H., Shamliyan, T. A., Sainfort, F., and Kane, R. L. (2013). Case study comparing bayesian and frequentist approaches for multiple treatment comparisons. *Methods Research Reports*.

Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M. A., Guo, J., Li, P., and Riddell, A. (2016). Stan: A probabilistic programming language. *Journal of Statistical Software*.

Chalmers, T. (1988). Meta-analysis in clinical medicine. *Transactions of the American Clinical and Climatological Association*, 99:144.

Chang, W. and Luraschi, J. (2016). profvis: Interactive visualizations for profiling R code. *R package version 0.3.2*.

Chapman, R. G. and Staelin, R. (1982). Exploiting rank ordered choice set data within the stochastic utility model. *Journal of Marketing Research*, pages 288–301.

Charpentier, A. (1891). Analyse experimentale de quelques elements de la sensation de poids (*Experimental analysis: On some of the elements of sensations of weight*). *Archives de Physiologie Normale et Pathologique*, 3:122–135.

Chen, C., Grennan, K., Badner, J., Zhang, D., Gershon, E., Jin, L., and Liu, C. (2011a). Removing batch effects in analysis of expression microarray data: an evaluation of six batch adjustment methods. *PloS One*, 6(2):e17238.

Chen, M., Wang, K., Zhang, L., Li, C., and Yang, Y. (2011b). The discovery of putative urine markers for the specific detection of prostate tumor by integrative mining of public genomic profiles. *PLoS One*, 6(12):e28552.

Cheng, C., Shen, K., Song, C., Luo, J., and Tseng, G. C. (2009). Ratio adjustment and calibration scheme for gene-wise normalization to enhance microarray inter-study prediction. *Bioinformatics*, 25(13):1655–1661.

Choi, J. K., Yu, U., Kim, S., and Yoo, O. J. (2003). Combining multiple microarray studies and modeling interstudy variation. *Bioinformatics*, 19(suppl 1):i84–i90.

Cipriani, A., Furukawa, T. A., Salanti, G., Geddes, J. R., Higgins, J. P., Churchill, R., Watanabe, N., Nakagawa, A., Omori, I. M., McGuire, H., et al. (2009). Comparative efficacy and acceptability of 12 new-generation antidepressants: a multiple-treatments meta-analysis. *The Lancet*, 373(9665):746–758.

Cochrane, A. (1972). *Effectiveness and Efficiency. Random Reflections on Health Services.* London: Nuffield Provincial Hospitals Trust.

Conklin, M. and Lipovetsky, S. (1999). Efficient assessment of self-explicated importance using latent class Thurstone scaling. In *The 10th Annual Advanced Research Techniques Forum, American Marketing Association*, Santa Fe, New Mexico.

Conlon, E. M., Song, J. J., and Liu, A. (2007). Bayesian meta-analysis models for microarray data: a comparative study. *BMC Bioinformatics*, 8(1):80.

Cox, T. F. and Cox, M. A. (2008). Multidimensional scaling. In *Handbook of Data Visualization*, pages 315–347. Springer Berlin Heidelberg.

Critchlow, D. E. (2012). *Metric Methods for Analyzing Partially Ranked Data*, volume 34. Springer Science & Business Media.

Croon, M. A. (1989). Latent class models for the analysis of rankings. *Advances in Psychology*, 60:99–121.

Daniels, H. (1950). Rank correlation and population models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 12(2):171–191.

David, H. A. (1988). *The Method of Paired Comparisons, second ed.*, volume 12. Charles Griffin & Company Ltd., London.

de Borda, J. C. (1781). *Mémoire sur les Élections au Scrutin.* Histoire de l'Academie Royale des Sciences.

de Leeuw, J. and Heiser, W. (1982). Theory of multidimensional scaling. *Handbook of Statistics*, 2:285–316.

DeConde, R. P., Hawley, S., Falcon, S., Clegg, N., Knudsen, B., and Etzioni, R. (2006). Combining results of microarray experiments: a rank aggregation approach. *Statistical Applications in Genetics and Molecular Biology*, 5(1).

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (methodological)*, pages 1–38.

Dolan, P. (1997). Modeling valuations for EuroQol health states. *Medical Care*, 35(11):1095–1108.

Donegan, S., Williamson, P., D'Alessandro, U., and Tudur Smith, C. (2013). Assessing key assumptions of network meta-analysis: a review of methods. *Research Synthesis Methods*, 4(4):291–323.

Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. (2001). Rank aggregation methods for the web. In *Proceedings of the 10th International Conference on World Wide Web*, pages 613–622. ACM.

Efron, B. and Tibshirani, R. J. (1994). *An Introduction to the Bootstrap*. Chapman & Hall, London.

Efthimiou, O., Debray, T., Valkenhoef, G., Trelle, S., Panayidou, K., Moons, K. G., Reitsma, J. B., Shang, A., and Salanti, G. (2016). GetReal in network meta-analysis: a review of the methodology. *Research Synthesis Methods*.

Ennis, D. M. and Johnson, N. L. (1993). Thurstone-Shepard similarity models as special cases of moment generating functions. *Journal of Mathematical Psychology*, 37(1):104–110.

Fagin, R., Kumar, R., Mahdian, M., Sivakumar, D., and Vee, E. (2006). Comparing partial rankings. *SIAM Journal on Discrete Mathematics*, 20(3):628–648.

Fishel, I., Kaufman, A., and Ruppin, E. (2007). Meta-analysis of gene expression data: a predictor-based approach. *Bioinformatics*, 23(13):1599–1606.

Fisher, R. A. (1925). *Statistical Methods for Research Workers*, volume 5. Oliver and Boyd, Edinburgh and London.

Fligner, M. A. and Verducci, J. S. (1988). Multistage ranking models. *Journal of the American Statistical Association*, 83(403):892–901.

Fortney, K., Griesman, J., Kotlyar, M., Pastrello, C., Angeli, M., Sound-Tsao, M., and Jurisica, I. (2015). Prioritizing therapeutics for lung cancer: an integrative meta-analysis of cancer gene signatures and chemogenomic data. *PLoS Computational Biology*, 11(3):e1004068.

Franchini, A., Dias, S., Ades, A., Jansen, J., and Welton, N. (2012). Accounting for correlation in network meta-analysis with multi-arm trials. *Research Synthesis Methods*, 3(2):142–160.

Francis, B., Dittrich, R., Hatzinger, R., and Humphreys, L. (2014). A mixture model for longitudinal partially ranked data. *Communications in Statistics-Theory and Methods*, 43(4):722–734.

Freiman, J. A., Chalmers, T. C., Smith Jr, H., and Kuebler, R. R. (1978). The importance of beta, the type II error and sample size in the design and interpretation of the randomized control trial: Survey of 71 negative trials. *New England Journal of Medicine*, 299(13):690–694.

Friedman, L. (2001). Why vote-count reviews don't count. *Biological Psychiatry*, 49(2):161–162.

Gabriel, K. R. (1971). The biplot graphic display of matrices with application to principal component analysis. *Biometrika*, 58(3):453–467.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–472.

Geng, X. and Luo, L. (2014). Multilabel ranking with inconsistent rankers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3742–3747.

Glass, G. V. (1976). Primary, secondary, and meta-analysis of research. *Educational Researcher*, 5(10):3–8.

Glenn, W. and David, H. (1960). Ties in paired-comparison experiments using a modified Thurstone-Mosteller model. *Biometrics*, 16(1):86–109.

Goldberg, D. E. and Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine Learning*, 3(2):95–99.

Goodwin, S., McPherson, J. D., and McCombie, W. R. (2016). Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6):333–351.

Gormley, I. C. and Murphy, T. B. (2008). A mixture of experts model for rank data with applications in election studies. *The Annals of Applied Statistics*, pages 1452–1477.

Gourieroux, C., Monfort, A., and Renault, E. (1993). Indirect inference. *Journal of Applied Econometrics*, 8:S85–S118.

Green, P. E. and Tull, D. S. (1978). *Research for Marketing Decisions*. Prentice-Hall, New Jersey.

Grunstein, M. and Hogness, D. S. (1975). Colony hybridization: a method for the isolation of cloned DNAs that contain a specific gene. *Proceedings of the National Academy of Sciences*, 72(10):3961–3965.

Haidich, A. (2011). Meta-analysis in medical research. *Hippokratia*, 14(1):29–37.

Hall, P. and Schimek, M. G. (2012). Moderate-deviation-based inference for random degeneration in paired rank lists. *Journal of the American Statistical Association*, 107(498):661–672.

Hardin, J. and Wilson, J. (2009). A note on oligonucleotide expression values not being normally distributed. *Biostatistics*, 10:446–450.

Hasselblad, V. (1998). Meta-analysis of multitreatment studies. *Medical Decision Making*, 18(1):37–43.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Heather, J. M. and Chain, B. (2016). The sequence of sequencers: The history of sequencing DNA. *Genomics*, 107(1):1–8.

Heiser, W. J. and D'Ambrosio, A. (2013). Clustering and prediction of rankings within a Kemeny distance framework. In *Algorithms from and for Nature and Life*, pages 19–31. Springer.

Higgins, J., Jackson, D., Barrett, J., Lu, G., Ades, A., and White, I. (2012). Consistency and inconsistency in network meta-analysis: concepts and models for multi-arm studies. *Research Synthesis Methods*, 3(2):98–110.

Higgins, J. and Whitehead, A. (1996). Borrowing strength from external trials in a meta-analysis. *Statistics in Medicine*, 15(24):2733–2749.

Hintze, J. L. and Nelson, R. D. (1998). Violin plots: a box plot-density trace synergism. *The American Statistician*, 52(2):181–184.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* U Michigan Press.

Holley, R. W., Apgar, J., Everett, G. A., Madison, J. T., Marquisee, M., Merrill, S. H., Penswick, J. R., and Zamir, A. (1965). Structure of a ribonucleic acid. *Science*, 147(3664):1462–1465.

Hu, P., Greenwood, C. M., and Beyene, J. (2005). Integrative analysis of multiple gene expression profiles with quality-adjusted effect size models. *BMC Bioinformatics*, 6(1):1.

Ingber, L. (1989). Very fast simulated re-annealing. *Mathematical and Computer Modelling*, 12(8):967–973.

Ingber, L., Petraglia, A., Petraglia, M. R., Machado, M. A. S., et al. (2012). Adaptive simulated annealing. In *Stochastic Global Optimization and its Applications with Fuzzy Adaptive Simulated Annealing*, pages 33–62. Springer.

Irizarry, R. A., Bolstad, B. M., Collin, F., Cope, L. M., Hobbs, B., and Speed, T. P. (2003). Summaries of affymetrix genechip probe level data. *Nucleic Acids Research*, 31(4):e15–e15.

Jackson, D., White, I. R., and Thompson, S. G. (2010). Extending DerSimonian and Laird's methodology to perform multivariate random effects meta-analyses. *Statistics in Medicine*, 29(12):1282–1297.

Jacques, J. and Biernacki, C. (2014). Model-based clustering for multivariate partial ranking data. *Journal of Statistical Planning and Inference*, 149:201–217.

Kamakura, W. A. and Srivastava, R. K. (1986). An ideal-point probabilistic choice model for heterogeneous preferences. *Marketing Science*, 5(3):199–218.

Kendall, M. G. (1945). The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251.

Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., et al. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.

Knox, C., Law, V., Jewison, T., Liu, P., Ly, S., Frolkis, A., Pon, A., Banco, K., Mak, C., Neveu, V., et al. (2011). DrugBank 3.0: a comprehensive resource for 'omics' research on drugs. *Nucleic Acids Research*, 39(suppl 1):1035–1041.

Lamb, J., Crawford, E. D., Peck, D., Modell, J. W., Blat, I. C., Wrobel, M. J., Lerner, J., Brunet, J.-P., Subramanian, A., Ross, K. N., et al. (2006). The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease. *Science*, 313(5795):1929–1935.

Lee, M. D., Steyvers, M., De Young, M., and Miller, B. J. (2011). A model-based approach to measuring expertise in ranking tasks. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*.

Lee, M. D., Steyvers, M., and Miller, B. (2014). A cognitive model for aggregating people's rankings. *PloS One*, 9(5):e96431.

Li, J., Tseng, G. C., et al. (2011). An adaptively weighted statistic for detecting differential gene expression when combining multiple transcriptomic studies. *The Annals of Applied Statistics*, 5(2A):994–1019.

Lin, S. (2010a). Rank aggregation methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(5):555–570.

Lin, S. (2010b). Space oriented rank-based data integration. *Statistical Applications in Genetics and Molecular Biology*, 9(1).

Lin, S. and Ding, J. (2009). Integration of ranked lists via Cross Entropy Monte Carlo with applications to mRNA and microRNA studies. *Biometrics*, 65(1):9–18.

Little, R. J. and Rubin, D. B. (2014). *Statistical Analysis with Missing Data*. John Wiley & Sons, New Jersey.

Liu, J. S. (2008). *Monte Carlo Strategies in Scientific Computing*. Springer Science & Business Media, New York.

Liu, Y.-T., Liu, T.-Y., Qin, T., Ma, Z.-M., and Li, H. (2007). Supervised rank aggregation. In *Proceedings of the 16th International Conference on World Wide Web*, pages 481–490. ACM.

Lottaz, C., Yang, X., Scheid, S., and Spang, R. (2006). OrderedList - a bioconductor package for detecting similarity in ordered gene lists. *Bioinformatics*, 22(18):2315–2316.

Lu, G. and Ades, A. (2004). Combination of direct and indirect evidence in mixed treatment comparisons. *Statistics in Medicine*, 23(20):3105–3124.

Lu, G., Welton, N. J., Higgins, J., White, I. R., and Ades, A. E. (2011). Linear inference for mixed treatment comparison meta-analysis: A two-stage approach. *Research Synthesis Methods*, 2(1):43–60.

Luce, R. D. (1959). *Individual Choice Behavior: A Theoretical Analysis*. John Wiley and sons.

Luce, R. D. (1994). Thurstone and sensory scaling: Then and now. *Psychological Review*, 101(2):271–277.

Lumley, T. (2002). Network meta-analysis for indirect treatment comparisons. *Statistics in Medicine*, 21(16):2313–2324.

Lunn, D., Spiegelhalter, D., Thomas, A., and Best, N. (2009). The BUGS project: Evolution, critique and future directions. *Statistics in Medicine*, 28(25):3049–3067.

Lunn, D. J., Thomas, A., Best, N., and Spiegelhalter, D. (2000). WinBUGS-a Bayesian modelling framework: concepts, structure, and extensibility. *Statistics and Computing*, 10(4):325–337.

Madden, L. V., Piepho, H.-P., and Paul, P. A. (2016). Statistical models and methods for network meta-analysis. *Phytopathology*.

Mallows, C. L. (1957). Non-null ranking models. I. *Biometrika*, 44(1/2):114–130.

Mantione, K. J., Kream, R. M., Kuzelova, H., Ptacek, R., Raboch, J., Samuel, J. M., and Stefano, G. B. (2014). Comparing bioinformatic gene expression profiling methods: microarray and RNA-Seq. *Medical Science Monitor Basic Research*, 20:138–141.

Marden, J. I. (1995). *Analyzing and Modeling Rank Data*. Chapman & Hall, London.

Marot, G., Foulley, J.-L., Mayer, C.-D., and Jaffrézic, F. (2009). Moderated effect size and p-value combinations for microarray meta-analyses. *Bioinformatics*, 25(20):2692–2699.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

Molenberghs, G., Fitzmaurice, G., Kenward, M. G., Tsiatis, A., and Verbeke, G. (2014). *Handbook of Missing Data Methodology*. CRC Press.

Mosteller, F. (1951). Remarks on the method of paired comparisons. *Psychometrika*, 16:203–218.

Murphy, T. B. and Martin, D. (2003). Mixtures of distance-based models for ranking data. *Computational Statistics & Data Analysis*, 41(3):645–655.

Murray, D. J., Ellis, R. R., Bandomir, C. A., and Ross, H. E. (1999). Charpentier (1891) on the size - weight illusion. *Perception & Psychophysics*, 61(8):1681–1685.

Neupane, B., Richer, D., Bonner, A. J., Kibret, T., and Beyene, J. (2014). Network meta-analysis using R: a review of currently available automated packages. *PloS One*, 9(12):e115065.

O'Rourke, K. (2007). An historical perspective on meta-analysis: dealing quantitatively with varying study results. *Journal of the Royal Society of Medicine*, 100(12):579–582.

Oyama, T., Sugio, K., Uramoto, H., Kawamoto, T., Kagawa, N., Nadaf, S., Carbone, D., and Yasumoto, K. (2007). Cytochrome P450 expression (CYP) in non-small cell lung cancer. *Frontiers in Bioscience*, 12:2299–2308.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The PageRank citation ranking: bringing order to the web. *Technical Report*.

Parmigiani, G., Garrett, E. S., Anbazhagan, R., and Gabrielson, E. (2002). A statistical framework for expression-based molecular classification in cancer. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4):717–736.

Parmigiani, G., Garrett-Mayer, E. S., Anbazhagan, R., and Gabrielson, E. (2004). A cross-study comparison of gene expression studies for the molecular classification of lung cancer. *Clinical Cancer Research*, 10(9):2922–2927.

Pihur, V., Datta, S., and Datta, S. (2007). Weighted rank aggregation of cluster validation measures: a Monte Carlo cross-entropy approach. *Bioinformatics*, 23(13):1607–1615.

Pihur, V., Datta, S., and Datta, S. (2008). Finding common genes in multiple cancer types through meta–analysis of microarray experiments: A rank aggregation approach. *Genomics*, 92(6):400–403.

Pihur, V., Datta, S., and Datta, S. (2009). RankAggreg, an R package for weighted rank aggregation. *BMC Bioinformatics*, 10(1):1.

Plaisier, S. B., Taschereau, R., Wong, J. A., and Graeber, T. G. (2010). Rank–rank hypergeometric overlap: identification of statistically significant overlap between gene-expression signatures. *Nucleic Acids Research*, 38(17):e169–e169.

Plummer, M. et al. (2003). JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*, volume 124, page 125. Technische Universität Wien.

Qiao, X., Zhang, H. H., Liu, Y., Todd, M. J., and Marron, J. S. (2010). Weighted distance weighted discrimination and its asymptotic properties. *Journal of the American Statistical Association*, 105(489):401–414.

Ramasamy, A., Mondry, A., Holmes, C. C., and Altman, D. G. (2008). Key issues in conducting a meta-analysis of gene expression microarray datasets. *PLoS Medicine*, 5(9):e184.

Ratcliffe, J., Brazier, J., Tsuchiya, A., Symonds, T., and Brown, M. (2009). Using DCE and ranking data to estimate cardinal values for health states for deriving a preference-based single index from the sexual quality of life questionnaire. *Health Economics*, 18(11):1261–1276.

Rau, A., Marot, G., and Jaffrézic, F. (2014). Differential meta-analysis of RNA-seq data from multiple studies. *BMC Bioinformatics*, 15(1):1.

Rhodes, D. R., Barrette, T. R., Rubin, M. A., Ghosh, D., and Chinnaiyan, A. M. (2002). Meta-analysis of microarrays interstudy validation of gene expression profiles reveals pathway dysregulation in prostate cancer. *Cancer Research*, 62(15):4427–4433.

Rodriguez-Antona, C. and Ingelman-Sundberg, M. (2006). Cytochrome P450 pharmacogenetics and cancer. *Oncogene*, 25(11):1679–1691.

Rosenblatt, J. D. and Stein, J. L. (2013). Rrho: Test overlap using the rank-rank hypergeometric test. *R package version 1.0.0*.

Rosner, B. and Kochanski, G. (2015). Categorical judgment with a variable decision rule. *arXiv preprint arXiv:1601.08244*.

Rubin, D. B. (1987). *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Sons, New Jersey.

Rubinstein, R. (1999). The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1(2):127–190.

Rücker, G., Schwarzer, G., Krahn, U., and König, J. (2015). netmeta: Network meta-analysis using frequentist methods. *R package version 0.7-0*.

Rung, J. and Brazma, A. (2013). Reuse of public genome-wide gene expression data. *Nature Reviews Genetics*, 14(2):89–99.

Saari, D. (2001). *Chaotic Elections!: A Mathematician Looks at Voting*. American Mathematical Society.

Salanti, G., Higgins, J. P., Ades, A., and Ioannidis, J. P. (2008). Evaluation of networks of randomized trials. *Statistical Methods in Medical Research*, 17(3):279–301.

Salomon, J. A. (2003). Reconsidering the use of rankings in the valuation of health states: a model for estimating cardinal values from ordinal data. *Population Health Metrics*, 1(1):1.

Sampath, S. and Verducci, J. S. (2013). Detecting the end of agreement between two long ranked lists. *Statistical Analysis and Data Mining*, 6(6):458–471.

Sanger, F., Nicklen, S., and Coulson, A. R. (1977). DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, 74(12):5463–5467.

Schimek, M. G., Budinská, E., Kugler, K. G., Švendová, V., Ding, J., and Lin, S. (2015). TopKLists: a comprehensive R package for statistical inference, stochastic aggregation, and visualization of multiple omics ranked lists. *Statistical Applications in Genetics and Molecular Biology*, 14(3):311–316.

Scrucca, L. (2013). GA: a package for genetic algorithms in R. *Journal of Statistical Software*, 53(4):1–37.

Selker, R., Lee, M. D., and Iyer, R. (2017). Thurstonian cognitive models for aggregating top-n lists. *Decision*, 4(2).

Simpson, R. and Pearson, K. (1904). Report on certain enteric fever inoculation statistics. *The British Medical Journal*, pages 1243–1246.

Skrondal, A. and Rabe-Hesketh, S. (2003). Multilevel logistic regression for polytomous data and rankings. *Psychometrika*, 68(2):267–287.

Stern, H. (1990). Models for distributions on permutations. *Journal of the American Statistical Association*, 85(410):558–564.

Stouffer, S. A. (1949). A study of attitudes. *Scientific American*, 180(5):11.

Su, J.-M., Lin, P., Wang, C.-K., and Chang, H. (2009). Overexpression of cytochrome P450 1B1 in advanced non-small cell lung cancer: a potential therapeutic target. *Anticancer Research*, 29(2):509–515.

Sudmant, P. H., Alexis, M. S., and Burge, C. B. (2015). Meta-analysis of RNA-seq expression data across species, tissues and studies. *Genome Biology*, 16(1):1.

Surowiecki, J. (2004). *The Wisdom of Crowds: Why the Many are Smarter than the Few and How Collective Wisdom Shapes Business, Economics, Society and Nations*. Anchor Books, New York.

Sutton, A. J., Abrams, K. R., Jones, D. R., Jones, D. R., Sheldon, T. A., and Song, F. (2000). *Methods for Meta-Analysis in Medical Research*, volume 348. Wiley, Chichester.

Swendsen, R. H. and Wang, J.-S. (1986). Replica Monte Carlo simulation of spin-glasses. *Physical Review Letters*, 57(21):2607.

Szu, H. and Hartley, R. (1987). Fast simulated annealing. *Physics Letters A*, 122(3-4):157–162.

Thurstone, L. L. (1927a). A law of comparative judgment. *Psychological Review*, 34(4):273.

Thurstone, L. L. (1927b). The method of paired comparisons for social values. *The Journal of Abnormal and Social Psychology*, 21(4):384.

Thurstone, L. L. (1927c). Psychophysical analysis. *The American Journal of Psychology*, 38(3):368–389.

Thurstone, L. L. (1945). The prediction of choice. *Psychometrika*, 10(4):237–253.

Timofeeva, M., Kropp, S., Sauter, W., Beckmann, L., Rosenberger, A., Illig, T., Jäger, B., Mittelstrass, K., Dienemann, H., Bartsch, H., et al. (2009). CYP 450 polymorphisms as risk factors for early onset lung cancer: gender specific differences. *Carcinogenesis*, page 102.

Tippett, L. H. C. et al. (1931). *The Methods of Statistics. An Introduction Mainly for Workers in the Biological Sciences*. Williams and Norgate, Ltd., London.

Tseng, G. C., Ghosh, D., and Feingold, E. (2012). Comprehensive literature review and statistical considerations for microarray meta-analysis. *Nucleic Acids Research*, page 1265.

van Blokland-Vogelesang, R. A. (1989). Unfolding and consensus ranking: A prestige ladder for technical occupations. *Advances in Psychology*, 60:237–258.

van Houwelingen, H. C., Arends, L. R., and Stijnen, T. (2002). Advanced methods in meta-analysis: multivariate approach and meta-regression. *Statistics in Medicine*, 21(4):589–624.

van Valkenhoef, G. and Kuiper, J. (2014). gemtc: Network meta-analysis using Bayesian methods. *R package version 0.6-1*.

Venn, J. (1880). On the diagrammatic and mechanical representation of propositions and reasonings. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 10(59):1–18.

Švendová, V. and Schimek, M. G. (2013). Search for top-k consensus objects in multiple ranked lists: TopKInference versus several recent procedures. In proceedings: WSC Hong Kong 2013.

Švendová, V. and Schimek, M. G. (2017). A novel method for estimating the common signals for consensus across multiple ranked lists. *Computational Statistics and Data Analysis*, 115:122–135.

Wang, J., Coombes, K. R., Highsmith, W. E., Keating, M., and Abruzzo, L. V. (2004). Differences in gene expression between B-cell chronic lymphocytic leukemia and normal B cells: a meta-analysis of three microarray studies. *Bioinformatics*, 20(17):3166–3178.

Wenzel, W. and Hamacher, K. (1999). Stochastic tunneling approach for global minimization of complex potential energy landscapes. *Physical Review Letters*, 82(15):3003.

White, I. R., Barrett, J. K., Jackson, D., and Higgins, J. (2012). Consistency and inconsistency in network meta-analysis: model estimation using multivariate meta-regression. *Research Synthesis Methods*, 3(2):111–125.

White, I. R., Royston, P., and Wood, A. M. (2011). Multiple imputation using chained equations: issues and guidance for practice. *Statistics in Medicine*, 30(4):377–399.

Wilkinson, B. (1951). A statistical consideration in psychological research. *Psychological Bulletin*, 48(2):156.

Xia, J., Jia, P., Hutchinson, K. E., Dahlman, K. B., Johnson, D., Sosman, J., Pao, W., and Zhao, Z. (2014). A meta-analysis of somatic mutations from next generation sequencing of 241 melanomas: a road map for the study of genes with potential clinical relevance. *Molecular Cancer Therapeutics*, 13(7):1918–1928.

Xu, L., Tan, A. C., Naiman, D. Q., Geman, D., and Winslow, R. L. (2005). Robust prostate cancer marker genes emerge from direct integration of inter-study microarray data. *Bioinformatics*, 21(20):3905–3911.

Yang, G., Liu, D., Liu, R. Y., Xie, M., and Hoaglin, D. C. (2014). Efficient network meta-analysis: A confidence distribution approach. *Statistical Methodology*, 20:105–125.

Yang, X., Bentink, S., Scheid, S., and Spang, R. (2006). Similarities of ordered gene lists. *Journal of Bioinformatics and Computational Biology*, 4(03):693–708.

Zhang, J., Carlin, B. P., Neaton, J. D., Soon, G. G., Nie, L., Kane, R., Virnig, B. A., and Chu, H. (2014). Network meta-analysis of randomized clinical trials: Reporting the proper summaries. *Clinical Trials*, 11(2):246–262.

Zheng, W., Chung, L. M., and Zhao, H. (2011). Bias detection and correction in RNA-Sequencing data. *BMC Bioinformatics*, 12(1):1.

Zinnes, J. L. and Griggs, R. A. (1974). Probabilistic, multidimensional unfolding analysis. *Psychometrika*, 39(3):327–350.

Zintzaras, E. and Ioannidis, J. P. (2012). METRADISC-XL: A program for meta-analysis of multidimensional ranked discovery oriented datasets including microarrays. *Computer Methods and Programs in Biomedicine*, 108(3):1243–1246.