Evaluation of clustering results and novel cluster algorithms: A metascientific perspective

Theresa Ullmann

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

eingereicht am 28.09.2022

Erstgutachterin: Prof. Dr. Anne-Laure Boulesteix
Zweitgutachter: Prof. Dr. Mark Robinson
Drittgutachter: Prof. Dr. Iven van Mechelen
Tag der Disputation: 02.12.2022

Summary

Cluster analysis is frequently performed in many application fields to find groups in data. For example, in medicine, researchers have used gene expression data to cluster patients suffering from a particular disease (e.g., breast cancer), in order to detect new disease subtypes. Many cluster algorithms and methods for cluster validation, i.e., methods for evaluating the quality of cluster analysis results, have been proposed in the literature. However, open questions about the evaluation of both clustering results and novel cluster algorithms remain. It has rarely been discussed whether a) interesting clustering results or b) promising performance evaluations of newly presented cluster algorithms might be over-optimistic, in the sense that these good results cannot be replicated on new data or in other settings.

Such questions are relevant in light of the so-called "replication crisis"; in various research disciplines such as medicine, biology, psychology, and economics, many results have turned out to be non-replicable, casting doubt on the trustworthiness and reliability of scientific findings. This crisis has led to increasing popularity of "metascience". Metascientific studies analyze problems that have contributed to the replication crisis (e.g., questionable research practices), and propose and evaluate possible solutions. So far, metascientific studies have mainly focused on issues related to significance testing. In contrast, this dissertation addresses the reliability of a) clustering results in applied research and b) results concerning newly presented cluster algorithms in the methodological literature. Different aspects of this topic are discussed in three Contributions.

The first Contribution presents a framework for validating clustering results on validation data. Using validation data is vital to examine the replicability and generalizability of results. While applied researchers sometimes use validation data to check their clustering results, our article is the first to review the different approaches in the literature and to structure them in a systematic manner. We demonstrate that many classical cluster validation techniques, such as internal and external validation, can be combined with validation data. Our framework provides guidance to applied researchers who wish to evaluate their own clustering results or the results of other teams on new data.

The second Contribution applies the framework from Contribution 1 to quantify over-optimistic bias in the context of a specific application field, namely unsupervised microbiome research. We analyze over-optimism effects which result from the multiplicity of analysis strategies for cluster analysis and network learning. The plethora of possible analysis strategies poses a challenge for researchers who are often uncertain about which method to use. Researchers might be tempted to try different methods on their dataset and look for the method yielding the "best" result. If only the "best" result is selectively reported, this may cause "overfitting" of the method to the dataset and the result might not be replicable on validation data. We quantify such over-optimism effects for

four illustrative types of unsupervised research tasks (clustering of bacterial genera, hub detection in microbial association networks, differential network analysis, and clustering of samples).

Contributions 1 and 2 consider the evaluation of clustering results and thus adopt a metascientific perspective on *applied* research. In contrast, the third Contribution is a metascientific study about *methodological* research on the development of new cluster algorithms. This Contribution analyzes the over-optimistic evaluation and reporting of novel cluster algorithms. As an illustrative example, we consider the recently proposed cluster algorithm "Rock"; initially deemed promising, it later turned out to be not generally better than its competitors. We demonstrate how Rock can nevertheless appear to outperform competitors via optimization of the evaluation design, namely the used data types, data characteristics, the algorithm's parameters, and the choice of competing algorithms. The study is a cautionary tale that illustrates how easy it can be for researchers to claim apparent "superiority" of a new cluster algorithm. This, in turn, stresses the importance of strategies for avoiding the problems of over-optimism, such as neutral benchmark studies.

Zusammenfassung

Clusteranalyse wird in vielen Anwendungsbereichen durchgeführt, um Gruppen in Daten zu finden. Beispielsweise verwenden Forscher in der Medizin Genexpressionsdaten, um Patienten mit einer bestimmten Krankheit (z.B. Brustkrebs) zu clustern, mit dem Ziel, neue Untergruppen der Krankheit zu entdecken. Viele Clusteralgorithmen und Methoden für Clustervalidierung, d.h. Methoden zur Bewertung der Qualität von Clusteringergebnissen, wurden in der Literatur vorgeschlagen. Jedoch bleiben offene Fragen in Bezug auf die Bewertung von Clusteringergebnissen und neuer Clusteralgorithmen. Bis jetzt wurde selten diskutiert, ob a) interessante Clusteringergebnisse oder b) vielversprechende Bewertungen neu präsentierter Clusteralgorithmen überoptimistisch sein könnten, in dem Sinne, dass sich die guten Ergebnisse nicht auf neuen Datensätzen oder in anderen Szenarien replizieren lassen.

Solche Fragen sind im Angesicht der sogenannten Replikationskrise relevant: In verschiedenen Forschungsfeldern, z.B. Medizin, Biologie, Psychologie und Wirtschaftswissenschaften, stellte sich heraus, dass viele wissenschaftliche Befunde nicht replizierbar sind. Dies ließ die Vertrauenswürdigkeit und Zuverlässigkeit wissenschaftlicher Befunde als fraglich erscheinen. Diese Krise hat zu einer erhöhten Popularität von „Metawissenschaft“ geführt. Metawissenschaftliche Studien untersuchen Probleme, die zur Replikationskrise beigetragen haben (z.B. zweifelhafte Forschungspraktiken); zudem schlagen sie mögliche Lösungen vor und evaluieren diese. Bis jetzt haben sich metawissenschaftliche Studien vor allem auf Probleme im Zusammenhang mit Signifikanztesten konzentriert. Im Gegensatz dazu betrachtet diese Dissertation die Zuverlässigkeit von a) Clusteringergebnissen in angewandter Forschung und b) Bewertungen von Clusteralgorithmen, die in der methodologischen Literatur neu vorgestellt werden. Verschiedene Aspekte dieser Themen werden in drei Beiträgen diskutiert.

Der erste Beitrag präsentiert ein Framework für die Validierung von Clusteringergebnissen auf Validierungsdaten. Der Gebrauch von Validierungsdaten ist essenziell, um die Replizierbarkeit und Verallgemeinerbarkeit von Ergebnissen zu prüfen. Wissenschaftler in angewandter Forschung benutzen manchmal Validierungsdaten, um ihre Clusteringergebnisse zu überprüfen. Unser Artikel ist der erste, in dem die verschiedenen Ansätze in der Literatur auf systematische Weise strukturiert werden. Wir verdeutlichen, dass viele klassische Ansätze für Clustervalidierung, wie etwa interne und externe Validierung, mit Validierungsdaten kombiniert werden können. Unser Framework bietet Orientierung für Wissenschaftler in angewandter Forschung, die ihre eigenen Clusteringergebnisse oder die Ergebnisse anderer Forschungsteams auf neuen Datensätzen evaluieren wollen.

Der zweite Beitrag wendet das Framework aus dem ersten Beitrag an, um überoptimistischen Bias zu quantifizieren. Dies wird im Kontext eines spezifischen Anwendungsfeldes vorgenommen, nämlich unüberwachter („unsupervised“) Mikrobiomanalyse. Wir unter-

suchen überoptimistische Effekte, die aus der Vielfalt an Analysestrategien für Clusteranalyse und Netzwerkgenerierung resultieren. Die Vielzahl möglicher Analysestrategien stellt eine Herausforderung für Forscher dar, die oftmals unsicher sind, welche Methode sie verwenden sollten. Forscher können daher versucht sein, verschiedene Methoden auf ihrem Datensatz auszuprobieren und nach derjenigen Methode zu suchen, die das „beste“ Ergebnis liefert. Wenn jedoch nur das „beste“ Ergebnis selektiv berichtet wird, könnte dies „Overfitting“ der Methode an den Datensatz verursachen. Das Ergebnis ist dann möglicherweise nicht auf Validierungsdaten replizierbar. Wir quantifizieren solche überoptimistischen Effekte für vier beispielhafte Forschungsfragen (Clustering von Bakteriengattungen, Entdeckung zentraler Knoten in Netzwerken basierend auf Assoziationen zwischen Mikroben, Vergleiche derartiger Netzwerke zwischen zwei Gruppen, und Clustering von Proben).

Beiträge 1 und 2 behandeln die Bewertung von Clusteringergebnissen und werfen somit einen metawissenschaftlichen Blick auf *angewandte* Forschung. Im Gegensatz dazu ist der dritte Beitrag eine metawissenschaftliche Studie über *methodologische* Forschung zur Entwicklung neuer Clusteralgorithmen. Dieser Beitrag untersucht die überoptimistische Bewertung und Präsentation neuer Clusteralgorithmen. Als illustratives Beispiel betrachten wir den kürzlich vorgestellten Clusteralgorithmus „Rock“. Dieser wurde zunächst als vielversprechend angesehen; wie sich jedoch später herausstellte, ist der Algorithmus im Allgemeinen nicht besser als konkurriende Algorithmen. Wir demonstrieren, dass Rock dennoch so präsentiert werden kann, als würde er besser als konkurrierende Algorithmen abschneiden, nämlich durch Optimierung des Evaluationsdesigns, genauer gesagt der verwendeten Datentypen, Dateneigenschaften, der Parameter des Algorithmus und der Wahl der konkurrienden Algorithmen. Unsere Studie ist ein warnendes Beispiel und beleuchtet, wie einfach es für Forscher sein kann, die vermeintliche Überlegenheit eines neuen Clusteralgorithmus zu behaupten. Dies wiederum hebt die Bedeutung von Strategien zur Vermeidung von Überoptimismus hervor, beispielsweise die Wichtigkeit neutraler Benchmarkstudien.

Acknowledgments

First, I would like to thank Anne-Laure for her wonderful supervision. She has supported me throughout my PhD with encouragement, patient guidance, and inspiring discussions. I am grateful that she always been available for questions, while at the same time giving me space to explore my ideas. This combination has been ideal, and I could not have wished for better supervision.

Moreover, I would like to thank:

- Mark Robinson and Iven van Mechelen for kindly agreeing to be the reviewers of this thesis.
- My colleagues at the Institute for Medical Information Processing, Biometry, and Epidemiology (IBE) at the LMU Munich, especially (in alphabetical order) Christina, Max, Raphael, Roman, Sabine, and Simon. Their knowledge, kindness, and humor have made working with them tremendously enjoyable.
- The co-authors of the Contributions of this thesis (again in alphabetical order): Anna Beer, Philipp Finger, Christian Hennig, Maximilian Hünemörder, Christian L. Müller, Stefanie Peschel, and Thomas Seidl. I am grateful that I had the opportunity to work with these talented researchers from various subject backgrounds. A special thanks goes to Stefanie, for patiently offering her expertise on microbiome network analysis.
- Anna Jacob and Christina Nießl for reading and commenting on the thesis.
- The Munich Center for Machine Learning (MCML) for generous financial support, as well as for providing opportunities for networking with researchers from different departments of the LMU Munich.

Finally, I thank my family and friends for their moral and practical support during my PhD. In particular, I am deeply grateful to my parents for their unwavering love and support.

Contents

1 Background and motivation

Cluster analysis describes a range of data-analytic techniques for finding groups in data. Cluster algorithms are applied in different research fields, such as medicine, biology, psychology, and economics. For example, in the field of cancer research, researchers have frequently used gene expression data or other molecular data to cluster cancer patients in order to find new disease subtypes (see, e.g., for breast cancer, Burstein et al., 2015; Curtis et al., 2012; The Cancer Genome Atlas Network, 2012). Such subtypes can have clinical implications and may guide targeted treatment (Garrido-Castro et al., 2019; Prat et al., 2015).

Apart from clustering patients, further insights about cancer come from single-cell analysis. Recent technological advances have enabled the simultaneous measurement of gene expression in a multitude of individual cells (single-cell RNA-seq). Based on this data, the cells can be clustered to detect cell types or cell states (Duò et al., 2018). For example, Zheng et al. (2017) analyzed clusters of immune cells in liver cancer, and Pastushenko et al. (2018) applied cluster analysis to study tumor transition cell states.

Finally, another approach to study cancer comes from microbiome research which focuses on communities of microbes, for example, those living in the human gut. The human microbiome is assumed to play an important role in the health of an individual (Shreiner et al., 2015). While causal statements are difficult to make, changes in the microbiome are hypothesized to be associated with cancer. Dohlman et al. (2021) and Loftus et al. (2021) clustered bacteria to better understand the microbial community structure in colorectal cancer samples.

A multitude of cluster algorithms and methods for evaluating clustering results has been proposed in the literature. Yet, issues remain regarding the reliability and trustworthiness of a) clustering results in applied research, and b) results concerning newly presented cluster algorithms in the methodological literature. With regards to point a), scarce attention has been directed to the question of whether interesting clustering results are replicable or not (i.e., whether they can be confirmed in subsequent studies), and how replicability might be assessed. Moreover, one might wonder whether the plethora of existing clustering methods poses a challenge in itself; for example, might clustering results be non-replicable on validation data due to "method selection bias" (when researchers have picked the "best" method after trying many different ones on their dataset)? Finally, with regards to point b), one might suspect that promising performance evaluations of newly presented cluster algorithms are often over-optimistic, i.e., the good evaluation results do not hold up on other datasets or in alternative study designs.

So far, these issues have been rarely addressed in the cluster analysis literature, yet they appear relevant in light of the "replication crisis" that has plagued empirical scientific investigation in recent decades (Baker, 2016). Many scientific findings from diverse research

disciplines have turned out to be non-replicable, i.e., the results could not be confirmed in subsequent studies. Several large-scale replication projects reported a low percentage of successful replications in fields such as preclinical cancer research (Begley & Ellis, 2012; Errington et al., 2021), psychology (Open Science Collaboration, 2015), economics (Camerer et al., 2016), and social science (Camerer et al., 2018). An influential (and controversial) paper from Ioannidis (2005) even argued that "most published research findings are false".

The replication crisis has contributed to the rise of *metascience* (Schooler, 2014). Metascience (also called metaresearch) is broadly defined as "science on science", i.e., metascientific studies analyze the scientific process itself. Metascience is not necessarily a novel discipline; as Hardwicke et al. (2020) point out, the roots of metascience may be traced back to the scientific revolution in the 16th and 17th century, when philosophers such as Francis Bacon argued for rigorous scientific methods guided by skepticism (Bacon, 1620/1995). In the following centuries, metascientific considerations could be regarded as a branch of philosophy of science; *philosophical* methods were applied to study empirical science, and the term "metascience" was used mostly in this context until some decades ago (see, e.g., Pearce and Rantala, 1983). Metascience moved beyond the discipline of philosophy in the 20th century, when researchers started to apply *empirical* methods to study empirical science (Hardwicke et al., 2020). For example, Sterling (1959) noted that most findings published in major psychological journals were reported to be statistically significant. The phenomenon that "positive" results are much more likely to be published than "negative" results has since come to be known as "publication bias".

The popularity of metascience surged in the 21st century due to the aforementioned replication crisis. Metascientific studies now employ a broad spectrum of theoretical and empirical approaches to analyze problems that have contributed to the crisis (Hardwicke et al., 2020). These problems include questionable research practices such as p-hacking, cherry-picking, data dredging, or HARKing (for a brief overview, see Andrade, 2021). Such practices often exploit the *multiplicity of analysis strategies*; typically, there are many researcher degrees of freedom regarding the choice of analysis plan (Gelman & Loken, 2014; Simmons et al., 2011). These degrees of freedom might be problematic when coupled with *selective reporting* (reporting only the "best" or "most interesting" results). The multiplicity issue plays an important role in this thesis, and will be discussed in more detail below in Section 5.

Researchers do not necessarily engage in questionable research practices in an intentional or malicious manner, but might fall into the trap of self-deception (Nuzzo, 2015) when encountering ambiguity in methodology and results. Selective reporting is often strongly motivated by institutional incentive structures; journals, funders, and universities typically prefer "novel" and "interesting" results.

So far, metascience has focused mainly on issues related to significance testing (e.g., Head

et al., 2015; Simonsohn et al., 2014; Wasserstein et al., 2019), with several studies also considering explanatory and/or predictive modeling (Hoffmann et al., 2021; Patel et al., 2015; Steegen et al., 2016).[1] Moreover, some studies have discussed the replication crisis in the context of machine learning and artificial intelligence (Hutson, 2018), often with a focus on supervised learning (Hullman et al., 2022; Kapoor & Narayanan, 2022).

In contrast, metascientific studies on cluster analysis (unsupervised learning) are rare. An exception is the study of Beijers et al. (2022) where specification curve analysis (Simonsohn et al., 2020) was used to analyze the impact of the multiplicity of clustering methods on the resulting number of clusters in a psychiatric dataset. Moreover, there are some articles which used mostly formal and philosophical arguments to take a critical look at certain practices in cluster analysis. Such studies could be classified as metascientific in the broader sense. For example, Hennig (2015) cautioned that researchers should be more wary of using datasets that come with a "true" clustering for evaluating the performance of cluster algorithms. He argued that there is hardly ever a unique "true" clustering, and that a "good" clustering always depends on the context and the aim of the analysis (see also Von Luxburg et al., 2012 for similar considerations).

Another direction of research on cluster analysis related to metascience has recently emerged in computer science; for the ReScience initiative (Rougier et al., 2017), some research teams have attempted to reproduce and replicate the results of papers presenting new cluster algorithms (Eijkelboom et al., 2022; Teule et al., 2021), see Section 5 for more details.

Despite the examples listed, a lack of research into metascientific aspects of clustering, in particular topics such as (non)replicability, multiplicity of analysis strategies, and over-optimism still exists. This thesis aims at closing this gap by demonstrating that these issues are not limited to significance testing or supervised learning. Besides cluster analysis, the thesis will also–to a lesser extent–consider further unsupervised approaches, namely methods for network generation.

Metascience not only studies problems related to the replication crisis, but also aims to identify and evaluate potential solutions (Hardwicke et al., 2020), such as open science practices (Nosek et al., 2015), preregistration (Nosek et al., 2018), and neutral comparison studies (Boulesteix et al., 2013; Boulesteix et al., 2017). In this spirit, this thesis not only analyzes problems, but also discusses possible solutions.

The remainder of this thesis is structured as follows. In the first part of the thesis, I will introduce various concepts and methods relevant for this work, starting with Section 2 which will discuss some popular cluster algorithms as well as methods for cluster validation. While Contributions 1 and 3 mostly use datasets that have a simple structure,

[1] It is often not possible to strictly distinguish between metascientific studies that study significance testing and those that study exploratory/predictive modeling, given that model coefficients are often tested for significance. In cluster analysis, significance testing can also play a role when evaluating the clustering results (see Contribution 1), but this will not be the main focus of the thesis.

Contribution 2 considers microbiome data, which is more complex and will be explored in Section 3. Based on microbiome data, microbial networks can be generated, which in turn can be used as input for network-based clustering approaches. Such clustering methods will be explained in Section 4. In Section 5, I will discuss some terms from the introduction (in particular, replication, validation data, over-optimism, and multiplicity of analysis strategies) in more detail. The three Contributions of the thesis are summarized in Section 6. Section 7 provides an outlook on possible future directions of research.
In the second part of the thesis, the three Contributions are attached. These are Contribution 1 (Ullmann, Hennig, et al., 2022) which discusses validating clustering results on validation data, Contribution 2 (Ullmann et al., 2023) which covers over-optimism in unsupervised microbiome research, and Contribution 3 (Ullmann, Beer, et al., 2022) which is about the over-optimistic evaluation and reporting of novel cluster algorithms.

2 Classical cluster algorithms and cluster validation

This section explains some popular cluster algorithms such as k-means and hierarchical clustering, as well as methods for evaluating clustering results. Before considering any specific cluster algorithm, I will begin with some general remarks.

2.1 Basic concepts in cluster analysis

The goal of clustering is to assign a set of entities $\{x_1, \ldots, x_m\}$ to k groups (clusters) $C_1, \ldots, C_k$. Objects inside of a cluster should be "similar" to each other, and "dissimilar" from objects in other clusters. What exactly "(dis)similar" here means cannot be defined uniquely because it depends on the particular *cluster concept* of each cluster algorithm (Hennig, 2015). Therefore, different cluster algorithms will often yield different clusterings on the same dataset. For example, k-means aims to find spherical clusters, while DBSCAN looks for regions of higher density that are not necessarily spherical.
Cluster algorithms can be divided into *hard* (crisp) and *soft* (fuzzy) clustering methods (Hennig & Meila, 2015). In hard clustering, each entity can be assigned to only one cluster. In soft clustering, each entity can belong to multiple clusters, and a weight γ_{il} denotes the degree of membership of object x_i to cluster C_l. In the Contributions of this thesis, we regard hard clustering as the "default" case, and do not consider algorithms specifically designed for soft clustering. However, many presented arguments could also be applicable in soft clustering.
Both hard and soft clusterings can be considered as *flat* clusterings. In contrast, a *hierarchical* clustering consists of sequences of nested clusters, which can be visualized as a dendrogram (for more details, see the paragraph on hierarchical clustering below).
Cluster algorithms typically start from either object-by-variable data or (dis)similarity

data (Van Mechelen et al., 2018). This is depicted in Table 1. Part (a) shows the structure of an $n \times p$ object-by-variable dataset. We assume that the n objects represent samples drawn from a population (e.g., patients with breast cancer), while the p variables represent a fixed set (e.g., a set of genes for which gene expression is measured). In Contribution 1, we distinguish between the following two cases. It might be of interest to either cluster the n samples (*inferential* clustering, which aims to gain insights about the underlying population from which the samples were drawn) or to cluster the p variables (*descriptive* clustering, which aims to describe the fixed set of interest). This distinction is explained in more detail in Contribution 1. Instead of $n \times p$ object-by-variable data, the data might also be available in the form of a (dis)similarity matrix. Table 1(b) shows an $n \times n$ (dis)similarity matrix of the n samples for inferential clustering, Table 1(c) a $p \times p$ (dis)similarity matrix of the p variables for descriptive clustering.

Table 1: Data input for cluster algorithms

(a) $n \times p$ data

	Var 1	...	Var p
id 1	...	...	...
⋮	⋮	⋱	⋮
id n	...	...	...

(b) $n \times n$ data

	id 1	...	id n
id 1	...	...	...
⋮	⋮	⋱	⋮
id n	...	...	...

(c) $p \times p$ data

	Var 1	...	Var p
Var 1	...	...	...
⋮	⋮	⋱	⋮
Var p	...	...	...

An object-by-variable dataset of type $n \times p$ can be transformed into a (dis)similarity of type $n \times n$ or $p \times p$ by applying a suitable (dis)similarity measure (such as the Euclidean distance, Pearson/Spearman correlation, etc.). For the reverse route, multidimensional scaling techniques can be used (Borg & Groenen, 2005).
In the following section, each cluster algorithm will be explained in turn along with which data structure it takes as input (i.e., object-by-variable or (dis)similarity data). A special case of similarity data is *network data*. Algorithms specifically designed for network data will be considered in Section 4 below, after discussing methods for network generation in Section 3.

2.2 Popular clustering methods

This section introduces some classical cluster algorithms, namely k-means, hierarchical clustering, DBSCAN, and Mean Shift, which are used in the three Contributions to illustrate the raised issues. Note, however, that our results should not be considered as strictly specific to these particular algorithms. These clustering methods are merely used as examples, and we would expect that similar illustrations could also be performed with other algorithms (this remark also applies to the network-based cluster algorithms discussed further below). The algorithms are therefore sketched only briefly, with more detailed information available in the given references.

When describing the algorithms, I assume without loss of generality that the n objects (samples) are to be clustered. Otherwise, if the p variables are to be clustered, the roles of objects and variables can simply be switched. Note that Contribution 1 also generally uses the terms "objects" to refer to the entities to be clustered.

k-means In its classical form, k-means (Lloyd, 1982) is based on real-valued object-by-variable data. Let k be the desired number of clusters. The aim is to find a clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ such that each cluster C_l is represented well by its centroid c_l (thus k-means tries to find spherical and equally sized clusters). This can be formulated in mathematical terms as follows. Let $x_i \in \mathbb{R}^p, i = 1, \ldots, n$ denote the objects to be clustered. Then the centroid c_l of Cluster C_l is calculated as $c_l = \frac{1}{|C_l|} \sum_{x_i \in C_l} x_i$, where $|C_l|$ denotes the number of objects in cluster C_l. The goal is to minimize the sum of squares error:

$$SSE(\mathcal{C}) = \sum_{l=1}^{k} \sum_{x_i \in C_l} \|x_i - c_l\|^2 \tag{1}$$

The classical k-means method of Lloyd (1982) aims to (approximately) minimize (1) with an iterative algorithm consisting of the following steps. 1) Initialization: k random "centroids" are chosen from the set of objects $\{x_1, \ldots, x_n\}$. 2) Each object x_i is assigned to the nearest centroid (in terms of Euclidean distance), resulting in a preliminary clustering $\tilde{\mathcal{C}}$. 3) The centroids are updated, i.e., the centroid of each cluster $\tilde{C}_l$ in $\tilde{\mathcal{C}}$ is calculated. Steps 2) and 3) are then repeated until the algorithm converges, i.e., until the cluster memberships do not change anymore. Note that this algorithm does not necessarily find the global minimum of (1). In particular, the performance of the algorithm depends on the initialization of the centroids, and some random initializations may lead to suboptimal performances. A popular strategy is to repeat the random initialization several times and choose the best end result according to criterion (1).

Hierarchical clustering Hierarchical clustering refers to a class of cluster algorithms which return hierarchies as output, i.e., sequences of nested clusters. This can be visualized with dendrograms (Figure 1). A flat clustering with a specified number of clusters k can be derived from a hierarchical clustering by horizontally cutting through the dendrogram at a suitable height.

Methods for hierarchical clustering can be divided into two categories, namely *agglomerative* vs. *divisive* approaches. Agglomerative clustering starts with n clusters, each consisting of a single object, and successively merges these clusters into larger ones. Divisive clustering starts with a single cluster containing all objects, and successively partitions this cluster into smaller clusters. In the following, I focus on agglomerative clustering, which is the more popular approach.

Agglomerative clustering can be performed with *linkage* functions. This requires data in

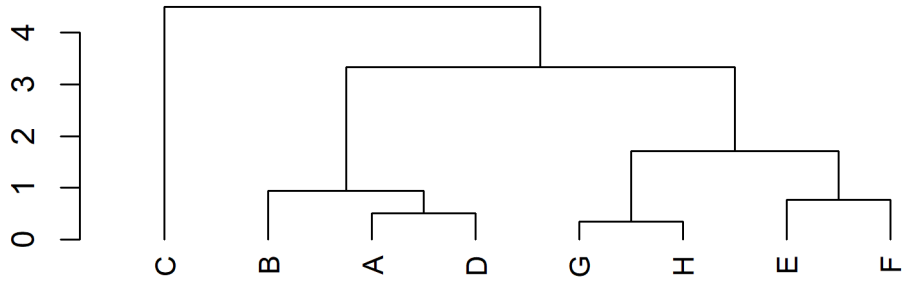


Figure 1: Dendrogram showing an exemplary hierarchical clustering of objects named $\{A, B, \ldots, H\}$. The dendrogram was generated with the dendextend R package (Galili, 2015).

the form of a dissimilarity matrix as input. Let $d(x_i, x_j)$ denote the dissimilarities of the objects x_i, x_j. In each step of the cluster algorithm, the most similar (least dissimilar) clusters are merged, where the dissimilarity $d(C, \tilde{C})$ of two clusters is calculated based on the dissimilarities of the objects in the clusters. The concrete calculation depends on the linkage function:

- Complete linkage (Sorensen, 1948): $d(C, \tilde{C}) = \max\{d(x, y) : x \in C, y \in \tilde{C}\}$
- Single linkage (Sneath, 1957): $d(C, \tilde{C}) = \min\{d(x, y) : x \in C, y \in \tilde{C}\}$
- Average linkage (Sokal & Michener, 1958): $d(C, \tilde{C}) = \frac{1}{|C||\tilde{C}|} \sum_{x \in C, y \in \tilde{C}} d(x, y)$

The different linkage functions can produce clustering results with different properties (Everitt et al., 2011). Single linkage is often not recommended in practice, as it can lead to so-called chaining effects: two clusters which are intuitively "separated" from each other, but connected by a chain of intermediate points, may be joined together early in the agglomerative process, which is frequently undesired. See Everitt et al. (2011) for an illustration of this issue.

DBSCAN DBSCAN (Ester et al., 1996) is a density-based algorithm, i.e., clusters are conceptualized as regions of higher density which might not be necessarily spherical, but could also be elongated, drawn-out, etc. The algorithm accepts a dissimilarity matrix as input, based on a distance d. Moreover, two input parameters $\epsilon > 0$ and $minPts \in \mathbb{N}$ are required. In the following, the objects to be clustered are called "points". DBSCAN is based on placing "connected" points in the same cluster. More precisely, the following concepts are used (Ester et al., 1996):

- A *core point* p is a point with at least $minPts$ points (including p itself) in its ϵ-neighborhood, where the latter is defined as the set $\{q : d(p, q) \leq \epsilon\}$.

- A point p is *directly density-reachable* from a point q if $d(p,q) \leq \epsilon$ (i.e., p is in the ϵ-neighborhood of q) and q is a core point.
- A point p is *density-reachable* from a point q if there is a chain of points $p_1 \ldots, p_m$, $p_1 = q, p_m = p$ such that p_{i+1} is directly density-reachable from p_i.
- A point p is *density-connected* to a point q if there is a point o such that both p and q are density-reachable from o.

A cluster C is then defined as a set of points such that a) if point p is in C and q is density-reachable from p, then q is also in C, and b) if points p, q are in C, then p, q must be density-connected.
The DBSCAN algorithm finds such clusters by starting with an arbitrary point p and evaluating its ϵ-neighborhood. If p is a core point, then the density-connected cluster including p is determined. Once this cluster is computed, the algorithm visits a new point. This procedure does not necessarily assign every point to a cluster. Points that remain unclustered after the algorithm stops are called *noise points* and can be considered as outliers.
For fixed values of ϵ and $minPts$, the above definition of clusters does not necessarily imply a unique clustering. More precisely, the dataset may contain border points (points which are neither core points nor noise points) that are density-reachable from more than one cluster, and thus could be assigned to different clusters. DBSCAN assigns such border points to the first cluster that they are reachable from (Schubert et al., 2017). This means that the clustering found by DBSCAN does not necessarily remain the same if the order of the points in the dataset is permuted. However, as Schubert et al. (2017) noted, this is a rare issue in practice.

Mean Shift The Mean Shift algorithm (Fukunaga & Hostetler, 1975) takes real-valued object-by-variable data as input. The objects are thus interpreted as points in the standard Euclidean space. Mean Shift proceeds as follows: For each point p, its local area is considered. This local area can be defined as an ϵ-neighborhood. In Mean Shift clustering, ϵ is called the *bandwidth* (which is the only input parameter of the algorithm). The mean of all points in the local area is computed, and the point p is "shifted" towards this mean. Starting from the new position of p, the procedure is then repeated until p arrives at its final position. After this shifting process is concluded for all points, different points with a similar final position are assigned to the same cluster.
The idea of this procedure is that each point will be gradually shifted towards positions in increasingly denser regions, i.e., towards "density modes". In fact, the algorithm seeks to estimate the local maxima (modes) of the probability density function from which the data was sampled (Cheng, 1995; Comaniciu & Meer, 2002). These modes correspond intuitively to cluster centers.

Modifications of Mean Shift include the Rock algorithm (Beer et al., 2019) that is used in Contribution 3 to demonstrate over-optimistic presentation of novel cluster algorithms. Rock replaces the bandwidth-defined local area with a K-nearest neighbor approach, i.e., in each step, the points are shifted towards the mean of their K nearest neighbors.

2.3 Evaluating the results of cluster algorithms

Once a clustering is obtained, how can its quality be evaluated? Different *cluster validation* procedures have been proposed to answer this question. These are described in more detail in Contribution 1 and are only briefly summarized here. *Internal validation* is based only on the data which was used for clustering. Typically, so-called internal validation indices are calculated, for example, indices measuring the homogeneity and/or separation of the clusters. In contrast, *external validation* uses external information that was not used for clustering. For example, the agreement of a clustering with a previously known categorization is evaluated. For *visual validation*, plots are generated to visualize the clustering (e.g., principal component plots, heatmaps,...). Researchers can inspect these plots to judge the clustering quality. *Stability evaluation* is based on the following idea: Let $\mathcal{C}$ be a clustering of the dataset D resulting from the clustering method M. The data D was sampled from a distribution F. Suppose that further datasets $D^{(1)}, \ldots, D^{(H)}$ are sampled from F, and the method M is applied to these datasets to obtain clusterings $\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(H)}$. If these clusterings are similar to each other and/or similar to the original clustering $\mathcal{C}$, this indicates a high degree of stability. Different approaches exist for measuring the similarity of the clusterings. As the distribution F is not known in practice, F is approximated by resampling techniques (e.g., subsampling or bootstrapping).
In applied research, cluster validation techniques can be used for evaluating a single clustering, but also for comparing multiple clusterings. Moreover, the validation procedures can be used in methodological research to evaluate the performance of cluster algorithms (see Contribution 3). A particular focus of this thesis lies on combining classical validation techniques with *validation on validation data*, a concept extensively discussed in Contribution 1 (see also Sections 5.2 and 6).

3 Microbiome data and microbial networks

In all three Contributions of this thesis, various datasets are used for demonstration or illustration of the raised issues. Contributions 1 and 3 mostly use simple data examples, in the form of real-valued $n \times p$ object-by-variable data that can be interpreted in standard Euclidean space. On the other hand, Contribution 2 (as well as an example in the supplement of Contribution 1) focuses on *microbiome data* which has a particular structure, and thus also requires specific data analysis techniques (Gloor et al., 2017). This section

explains basic properties of microbiome data. Moreover, methods for generating microbial association networks are discussed. Network-based analyses are considered extensively in Contribution 2.

3.1 Microbiome count data

The term *microbiome* refers to the community of microbes (including bacteria, archaea, fungi, and viruses) that live in a particular environment, e.g., the human gut. Microbiome data is typically obtained by collecting n samples (e.g., stool samples from n persons to study the gut microbiome). Possible goals are to find out which – and how many – microbes are contained in each of these samples, and how these microbes are associated with each other.

For this purpose, so-called *marker genes* of microbes can be sequenced. For bacteria, a very popular marker gene is the 16S ribosomal RNA (rRNA) gene. This gene is suitable because it contains highly variable regions that can serve to distinguish between different types of bacteria (Li, 2015). Such a variable region can be targeted with high-throughput amplicon sequencing. The resulting sequences are then clustered into operational taxonomic units (OTUs). That is, sequences that differ by less than a specific threshold (e.g., 3%) are clustered together (Callahan et al., 2016). (Note that this thesis does not specifically consider this clustering task, rather, we focus on cluster analysis performed at a later step in the network analysis, see below.) Each OTU can be seen as a proxy for a bacterial species (Li, 2015). The clustering into OTUs accounts for sequencing errors; small differences between sequences are often caused by technical error, not by genuine biological differences. The downside is a certain loss of information, i.e., the fine-scale variation that is, in principle, accessible by modern sequencing techniques, cannot be detected. Newer computational methods (Callahan et al., 2016) can correct sequencing errors without OTU clustering. The resulting sequences are called amplicon sequence variants (ASVs). In Contribution 2, we consider OTU data. However, the study design of Contribution 2 would also apply to ASV data.

The result of the sequencing process can be represented as an $n \times p$ count matrix consisting of non-negative integers, see Table 2(a). Each entry w_{ij} denotes how often taxon j was observed in sample i ("taxon" is a general term for a taxonomic group and stands here for an OTU or an ASV). Additionally, the count table comes with metadata about the taxonomic categorization of the OTUs or ASVs; by consulting a bacterial database, each taxon can be assigned a taxonomic lineage, i.e., information about which higher-level taxonomic ranks the taxon belongs to, such as genus, family, order, etc. (Li, 2015). As an example, the taxonomic lineage of the famous bacterial species Escherichia coli (E. coli) is displayed in Table 3 (Schoch et al., 2020).

The count data is sometimes agglomerated to a higher taxonomic level (for example, the

Table 2: Microbiome data

(a) $n \times p$ count matrix

	taxon 1	...	taxon p
id 1	w_{11}	...	w_{1p}
$\vdots$	$\vdots$	$\ddots$	$\vdots$
id n	w_{n1}	...	w_{np}

(b) $p \times p$ adjacency matrix

	taxon 1	taxon 2	...	taxon p
taxon 1	0	a_{12}	...	a_{1p}
taxon 2	a_{21}	0	...	a_{2p}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
taxon p	a_{p1}	a_{p2}	...	0

Table 3: Taxonomic lineage of the bacterial species Escherichia coli (E. coli). Together with other species, E. coli belongs to the genus Escherichia, and so on.

Taxonomic classification	
Domain:	Bacteria
Phylum:	Pseudomonadota
Class:	Gammaproteobacteria
Order:	Enterobacterales
Family:	Enterobacteriaceae
Genus:	Escherichia
Species:	Escherichia coli

genus level, which is the case in Contribution 2). The structure of this agglomerated data would then again look like Table 2(a), with a "taxon" now standing, e.g., for a genus. Consequently, the number p of taxa is reduced. Agglomeration is sensible if one is interested in specific functions that different bacterial species share with their higher-level taxonomic group (Röttjers & Faust, 2018). Moreover, agglomeration may facilitate the interpretation of microbial association networks (see below). On the other hand, agglomeration can lead to information loss.

3.2 Microbial association networks

Microbial count data can be used for different types of analyses. This section focuses on inferring a microbial association network from the data, i.e., a network in which each node represents a microbial taxon, and each edge represents an association between two taxa. Such networks will be used in Contribution 2 to demonstrate over-optimism effects. See Figure 2 for the visualization of an exemplary network. Microbial networks are important tools to better understand the complex interactions between microbes that live in a specific habitat (Faust et al., 2012). They may help to generate new hypotheses, e.g., about key players in the microbiome (Röttjers & Faust, 2018).

Networks with many nodes and edges can be difficult to interpret. This issue can be partially addressed by agglomeration to a higher taxonomic level, as this reduces the

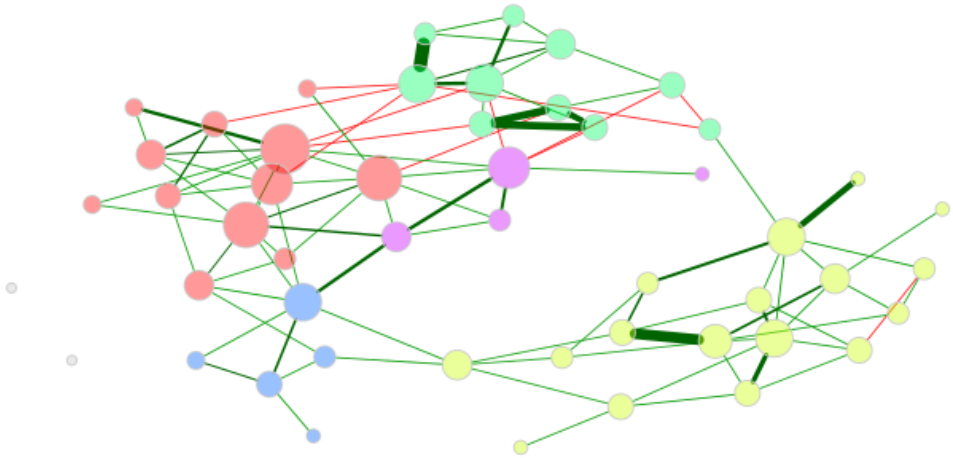


Figure 2: Exemplary network generated with the NetCoMi R package (Peschel et al., 2021). This network shows associations between a subset of microbes living in the human gut, based on OTU count data from the American Gut Project (McDonald et al., 2018). Edges are colored according to the direction of the association (green: positive, red: negative). The edge widths represent the association strengths. Nodes are colored according to a clustering of the network with fast greedy modularity optimization (Clauset et al., 2004). The size of a node represents its degree centrality, which is defined as the number of adjacent nodes (Freeman, 1978). For better interpretability, labels could be added to the nodes (e.g., with information about the taxonomic rank), but to keep the visualization clear, this was not done here.

number of taxa and thus the number of nodes in the networks (Röttjers & Faust, 2018).

A microbial association network can be represented as a $p \times p$ adjacency matrix as in Table 2(b). Each entry a_{ij} represents the edge weight, i.e., the strength of the association between taxa i and j, with $a_{ij} = 0$ signifying that there is no association (and thus no edge) between i and j. The entries on the diagonal are zero to indicate that there are no self-loops in the network. We assume that the network is undirected, and thus the adjacency matrix is symmetric.

How can such a network be generated, i.e., how can an adjacency matrix be obtained from a count matrix? This requires calculating the associations between the taxa. For this purpose, one cannot directly apply classical correlation measures, such as Pearson correlation, to the count matrix. This is not valid due to the *compositional* nature of the count data (Gloor et al., 2017). For each sample i, the counts sum up to a fixed number $m^{(i)} = \sum_{j=1}^{p} w_{ij}$, which is called the *sequencing depth* or *library size*. The sequencing depth does not correspond to the true total bacterial abundance in the sample. Instead, the depth is determined by technical factors, and constitutes an upper limit of the sequencing instruments. Put differently, if the instrument has delivered a read of a taxon, that read is then not available anymore for further counts. Moreover, the sequencing depth varies between samples due to technical reasons (Gloor et al., 2017).

Consequently, the counts must be interpreted as relative abundances: only the proportions $w_{ij}/m^{(i)}$ carry information, not the absolute abundances w_{ij}. Evidently, for each sample, the proportions sum up to one. This constraint implies that, in particular, the count data cannot be interpreted in standard Euclidean space. Instead, the proportions are elements of the so-called simplex space (Aitchison, 1982).[2]

As a consequence of compositionality, classical correlation measures such as the Pearson or Spearman correlation cannot be applied to the count matrix directly. As Lovell et al. (2015) illustrate, completely different absolute abundances can give rise to the same relative abundances. Uncorrelated absolute abundances can yield correlated relative abundances, or vice versa. In particular, not accounting for compositionality can lead to spurious negative correlations (Friedman & Alm, 2012): if the proportion of a taxon increases, the proportions of the other taxa must necessarily decrease, simply because the proportions must add up to one.

To calculate associations between taxa, the count data is thus first normalized. Suitable normalization methods are discussed in the next paragraph.[3]

Normalization and zero handling. Aitchison (1982) proposed the centered log-ratio (clr) transformation, which maps proportions from the simplex space into the Euclidean space. For each sample i, the count vector $\boldsymbol{w}^{(i)} = (w_{i1}, \ldots, w_{ip})$ is transformed as follows:

$$clr\left(\boldsymbol{w}^{(i)}\right) = \left(\log\left(\frac{w_{i1}}{g(\boldsymbol{w}^{(i)})}\right), \ldots, \log\left(\frac{w_{ip}}{g(\boldsymbol{w}^{(i)})}\right)\right),$$

where $g\left(\boldsymbol{w}^{(i)}\right)$ is the geometric mean of $\boldsymbol{w}^{(i)}$. Note that $clr\left(\boldsymbol{w}^{(i)}\right) = clr\left(\boldsymbol{w}^{(i)}/m^{(i)}\right)$, i.e., the clr transformation yields the same results for the absolute counts as for the relative abundances.

Both the logarithm and the division by the geometric mean require non-zero counts. This poses a challenge as microbiome count data is typically *sparse*, i.e., contains many zero counts. Some taxa may indeed have low abundance, and thus be present in only few samples. However, zero counts can also stem from technical issues related to the sequencing process (Tsilimigras & Fodor, 2016).

To make the clr applicable to sparse count data, a *pseudo count* of one can be added to the data. However, as Yoon et al. (2019) note, this addition is somewhat arbitrary, and may have undesired effects on subsequent analyses. The authors thus proposed the

[2]Note that the property of compositionality is not restricted to microbiome data, but extends more generally to data obtained with high-throughput sequencing, including RNA-Seq gene expression data (Quinn et al., 2018), due to the same technical constraints of sequencing instruments as described above.

[3]For simplicity, I will use the term "normalization" both for log-ratio transformations, such as the centered log-ratio (clr) transformation, as well as for normalizations to effective library size, such as the variance-stabilizing (VST) procedure (both clr and VST are described below). Some authors emphasize the distinction between "transformation" and "normalization" (Quinn et al., 2018), but a more detailed discussion of this issue goes beyond the scope of this thesis.

modified clr transformation (mclr), which applies the clr to non-zero counts only, and leaves the zero counts unmodified. Alternative approaches to deal with zero counts are summarized in Tsilimigras and Fodor (2016) and Peschel et al. (2021).

An alternative to normalization based on log-ratios (such as the clr and mclr) is the variance-stabilizing transformation (VST; Anders and Huber, 2010), which is based on fitting a negative binomial distribution to the count data, and aims to (approximately) eliminate the dependence of the variance of the counts on the mean. Like the clr, the VST requires prior handling of zeros.

Association estimation. Normalization methods such as the clr and VST alleviate some issues related to compositionality, by correcting for varying sequencing depths and mapping the proportions into standard Euclidean space. Classical correlation measures can then be applied to the transformed data. There is some evidence that normalization with the clr or VST combined with Pearson correlation yields consistent correlation estimates, provided the sample size is not too small (Badri et al., 2020). Moreover, classical correlation measures are well-known and easy to understand. We thus include normalization with clr/mclr/VST coupled with Pearson or Spearman correlation as part of the methods for generating microbial networks in Contribution 2.

However, some issues related to compositionality and/or association estimation may remain when applying Pearson or Spearman correlation, even after normalizing the data (Tsilimigras & Fodor, 2016). Therefore, alternative methods for association estimation have been proposed. These include 1) latent correlation estimation, 2) partial correlation estimation, and 3) proportionality measures.

Latent correlation estimation (Yoon et al., 2020; Yoon et al., 2021) refers to estimating the latent correlation ("latentcor") matrix of a truncated Gaussian copula model. More precisely, the microbial counts are modeled as realizations of a Gaussian copula variable which is *truncated* to reflect that the counts are either zero or positive. Compared to the Pearson correlation, the latentcor approach is better suited to deal with excess zeros in the microbial count matrix (Yoon et al., 2019). The latentcor estimation is applied to mclr-transformed data (as mentioned above, the mclr transformation leaves the zeros in the count data intact).

Methods for estimating partial correlations include the SPRING approach (Yoon et al., 2019), which starts by calculating the latentcor matrix and then applies the neighborhood selection technique (Meinshausen & Bühlmann, 2006) to infer conditional dependencies. That is, for each pair of taxa, the associations between the two taxa *conditioned* on all other taxa are estimated. In contrast to classical correlation methods, SPRING can thus distinguish between direct and indirect associations. An alternative to SPRING is the SPIEC-EASI method (Kurtz et al., 2015), which also estimates partial correlations, but starts from the Pearson correlation matrix of the clr-transformed data instead of the

latentcor matrix. For Contribution 2, we use the newer SPRING method.
An alternative approach for association estimation is the concept of proportionality (Lovell et al., 2015; Quinn et al., 2017), which we also consider in Contribution 2. To describe the underlying idea, let $\boldsymbol{w}_{\text{rel}}^{(\bullet j)} = \left(\frac{w_{ij}}{m^{(i)}}\right)_{i=1,\ldots,n}$ denote the j'th column of the relative abundance matrix. If the relative abundances $\boldsymbol{w}_{\text{rel}}^{(\bullet j)}, \boldsymbol{w}_{\text{rel}}^{(\bullet k)}$ of two taxa j, k are proportional to each other, then the underlying absolute abundances are also proportional. Put differently, proportionality is a property of the underlying absolute abundances which can be inferred from the observed relative abundances. Thus, proportionality is a suitable measure for association strength. To calculate the "extent" of proportionality, a factorization of the log-ratio variance $var(\log(\boldsymbol{w}_{\text{rel}}^{(\bullet j)}/\boldsymbol{w}_{\text{rel}}^{(\bullet k)}))$ can be used (see Lovell et al., 2015 and Quinn et al., 2017 for details). While proportionality is a compositionally aware method (i.e., it takes compositionality directly into account), the data should still be clr-transformed before applying the proportionality measure, such that associations between different pairs of taxa are on the same scale and thus comparable (Lovell et al., 2015).

Sparsification of associations. After calculating associations between the taxa, the association matrix can be transformed into a $p \times p$ adjacency matrix as in Table 2(b), which represents a network as described above. However, this matrix may not be sparse, in which case the resulting network is *dense* (i.e., all or almost all nodes are connected), which hampers interpretability. Therefore, *sparsification* is frequently applied to the associations, i.e., small associations are set to zero (meaning that there is then no edge between these taxa in the network). Sparsification can be performed, for example, with a cut-off value: associations with an absolute value below a specified threshold are set to zero (Friedman & Alm, 2012). For the Pearson and Spearman correlation estimates, one can also apply a suitable significance test (e.g., a t-test or a bootstrap test) and only keep the associations that are significantly different from zero (Peschel et al., 2021).

Summary of the network generation steps. The workflow for generating microbial association networks based on count data is summarized in Figure 3. Note that this is a somewhat simplified depiction. Not every normalization method requires prior zero handling (e.g., the mclr transformation). Moreover, some methods for association estimation come with inbuilt sparsification (e.g., the SPRING method which estimates partial correlations in a sparse manner). More details about correctly combining methods for each of the four steps can be found in Peschel et al. (2021) and in Contribution 2.
Once the network is generated, further analyses can be applied, including clustering to identify groups of nodes. In the terminology of Section 2.1 and Contribution 1, *descriptive* clustering of the microbes is performed. Network-based clustering approaches are described in Section 4 below. Besides these approaches, we also apply hierarchical clustering to the *unsparsified* association matrices (details are given in Contribution 2). While

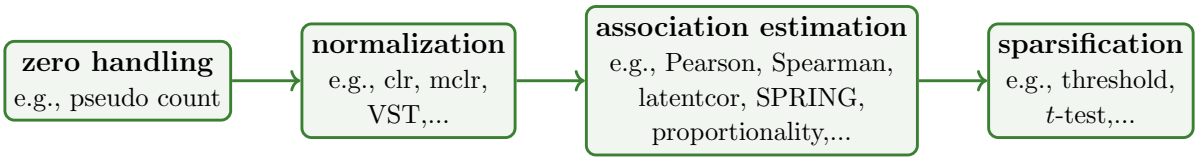


Figure 3: Simplified workflow for generating microbial association networks from microbiome count data.

the main focus of this thesis is on clustering, we also consider other analyses based on microbial association networks in Contribution 2. These include *hub detection* (identifying "influential" nodes of the network) or *differential network analysis* (comparing networks between different groups or conditions), as described in detail in Contribution 2. Finally, Contribution 2 not only considers the clustering of microbes, but also the clustering of *samples* (which constitutes *inferential* clustering in the terminology of Section 2.1 and Contribution 1). For this task, the focus is not on associations between microbes, but on similarities between samples. This is explained in detail in Contribution 2.

4 Network-based cluster algorithms

This section explores methods for clustering the nodes of a network, namely modularity optimization (Blondel et al., 2008; Clauset et al., 2004), spectral clustering (Ng et al., 2001; Weiss, 1999), and the manta algorithm (Röttjers & Faust, 2020). As in the previous section, a network is represented by a $p \times p$ adjacency matrix $A = (a_{ij})$, with a_{ij} denoting the weight of the edge between nodes i and j.

Modularity optimization The general aim of network-based cluster algorithms is to identify clusters of nodes such that nodes within a cluster are strongly connected (i.e., edges within a cluster have a high weight), while there are few respectively only weak connections between different clusters (i.e., edges between the clusters have a low weight). For a given network clustering, this property can be quantified by the *modularity measure*. In turn, network clustering can be performed by optimizing this measure.
Let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a clustering (partition) of the network nodes. Let $\delta_{\mathcal{C}}(i,j)$ be equal to 1 if nodes i and j are in the same cluster, and 0 otherwise. Moreover, let $W_{sum} = \sum_{i,j} a_{ij}$ be the sum over all edge weights in the network, and $d_i = \sum_{j=1}^{p} a_{ij}$ the (weighted) degree of node i. Then the modularity $q(\mathcal{C})$ is defined as follows (Newman, 2004; Newman & Girvan, 2004):

$$q(\mathcal{C}) = \frac{1}{W_{sum}} \sum_{i,j} \left(a_{ij} - \frac{d_i d_j}{W_{sum}} \right) \delta_{\mathcal{C}}(i,j) \tag{2}$$

The idea behind the modularity measure can be best understood for the special case of

an unweighted network, i.e., $a_{ij} = 1$ if there is an edge between i and j, and $a_{ij} = 0$ otherwise. Then the number of edges in the network is $W_{sum}/2$, and the fraction of within-cluster edges is $\left(\sum_{i,j} a_{ij}\delta_{\mathcal{C}}(i,j)\right)/W_{sum}$. Suppose that the edges between nodes are assigned randomly, given the fixed node degrees d_i. Then the probability of an edge existing between nodes i and j is given by $d_i d_j / W_{sum}$. Therefore, if the fraction of within-cluster edges is equal to what is expected for the randomized network, the modularity $q(\mathcal{C})$ is zero. On the other hand, higher modularity values indicate fractions "better than chance", and thus a better quality of the clustering.
Contribution 2 uses two approaches for modularity optimization, namely the fast greedy algorithm of Clauset et al. (2004) and the Louvain method of Blondel et al. (2008). Both methods have similarities to agglomerative hierarchical clustering (Section 2.2), as they start by putting each node in its own cluster, and then proceed to successively merge clusters to achieve increases in modularity.

Spectral clustering The name "spectral clustering" (Ng et al., 2001; Weiss, 1999) is derived from the term *spectrum*, i.e., the set of the eigenvalues of a matrix. For spectral clustering, the matrix of interest is the graph Laplacian L of the adjacency matrix A. The unnormalized graph Laplacian is defined as $L = D - W$, with $D = \text{diag}(d_1, \ldots, d_p)$ the diagonal matrix of the weighted degrees. Normalized versions of the graph Laplacian can also be used, see Von Luxburg (2007) for an overview. The graph Laplacian and its eigenvalues and eigenvectors are connected to properties of the network (Mohar, 1991). Spectral clustering consists of the following steps (Von Luxburg, 2007), with the adjacency matrix A and a fixed number of clusters k as input:

1. Compute the graph Laplacian L of A.
2. Compute the k first eigenvectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ of L.
3. Let $U \in \mathbb{R}^{p \times k}$ be the matrix with the eigenvectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ as its columns. For $j = 1, \ldots, p$, let $\boldsymbol{y}_j \in \mathbb{R}^k$ be the vector corresponding to the j'th row of U.
4. Apply the k-means algorithm to group the vectors $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_p$ into clusters $\tilde{C}_1, \ldots, \tilde{C}_k$.
5. The clustering of the network nodes is given by clusters $C_1, \ldots, C_k$ with $C_l := \{j : \boldsymbol{y}_j \in \tilde{C}_l\}$.

While this procedure appears rather abstract at first, it can be shown that the algorithm fulfills the general aim of network-based clustering as stated above (i.e., edges within a group should have a high weight, and edges between the groups a low weight). Spectral clustering does not optimize modularity, but instead approximates a "graph cut" procedure (Von Luxburg, 2007) which involves the minimization of the sums of edge weights between different clusters.

In contrast to the other network-based algorithms described in this section, which are typically applied only to networks, spectral clustering is also frequently applied to (real-valued) $n \times p$ object-by-variable data. In this case, the generation of the network adjacency matrix is often considered as a part of the spectral clustering pipeline; the object-by-variable data is first transformed into a similarity matrix, and then sparsified to obtain an adjacency matrix. For Contribution 2, we apply spectral clustering to the similarity matrix obtained by calculating microbial associations as described in Section 3 above. While sparsification could be performed with the approaches described in that section (e.g., the threshold method), we use another approach that is popular for spectral clustering, namely obtaining the sparsified adjacency matrix by computing the K-nearest neighbor graph based on the similarity matrix (Von Luxburg, 2007). This is why we classify spectral clustering as a similarity-based approach in Contribution 2, instead of grouping it together with modularity optimization and manta.

manta In contrast to modularity optimization and spectral clustering, manta (Röttjers & Faust, 2020) was specifically designed for microbial association networks and takes biological principles into account, such as "the enemy of my enemy is my friend". That is, even if there is no direct association between two microbial species in the network, they might still be clustered together if they have a shared negative association with another species. To incorporate such effects, manta accepts adjacency matrices with negative edge weights as input, which is not the case for modularity optimization or spectral clustering. Based on the adjacency matrix, manta generates a scoring matrix by iterating expansion and inflation of the edge weights, a process inspired by the Markov cluster algorithm (Van Dongen, 2000) which aims to find clusters based on random walks on the adjacency matrix. The scoring matrix is then clustered via agglomerative hierarchical clustering.

5 Concepts related to the replication crisis

This section explains in more detail some terms that were briefly mentioned in the introduction, including replication, validation data, multiplicity of analysis strategies, and over-optimism. These concepts are related to each other (for example, I define over-optimistic results as findings that cannot be successfully replicated or validated), and are relevant for all Contributions of this thesis. As a more general understanding of the terms is helpful before defining how they might apply to cluster analysis, the present section discusses the concepts in a broader context, mostly without specific reference to cluster analysis. Section 6 as well as the three Contributions will then explore the concepts specifically for clustering. In particular, Contribution 1 extensively discusses what replication and validation mean for evaluating clustering results.

Throughout this section, I distinguish between the terms in the context of *applied* stud-

ies vs. *methodological* studies. For example, an *applied* study might be a study which clusters cancer patients to find new disease subtypes, or a study in which a network is learned from a microbiome dataset to elucidate the structure of the gut microbiome in individuals with a certain illness. A typical example of a *methodological* study is a study introducing a new cluster algorithm. This thesis considers both aspects; Contributions 1 and 2 adopt a metascientific perspective on *applied* studies, while Contribution 3 focuses on *methodological* science. The terms discussed in this section can differ between the two cases. For example, replicating the results of an applied study does not necessarily mean the same as replicating the results of a methodological study.

5.1 Reproducibility and replicability

When discussing "replicability", it is important to distinguish this term from "reproducibility". The latter concept is also briefly explained in this section, although the focus remains on replicability.

The terms "reproducibility" and "replicability" are not used uniformly between (or even within) different research disciplines (Goodman et al., 2016; Nosek & Errington, 2020; Plesser, 2018). Given the lack of a broad consensus, this section takes a pragmatic approach; some (but not all) varying definitions of the terms are discussed, with a focus on how the terms are ultimately used in this thesis.

Reproducibility and replicability in applied research. A good starting point is the 2×2 matrix of Whitaker (2016) in Table 4. According to this scheme, *reproducibility* is defined as obtaining the exact same results when applying the same methods to the same data. Being able to reproduce one's own results or the results of other teams can be considered as a minimum standard in science (Hofner et al., 2016), but is often not fulfilled in practice, e.g., due to insufficient documentation and issues with data and/or code sharing (Gabelica et al., 2022; Gundersen & Kjensmo, 2018; Hardwicke et al., 2018).[4] *Replicability* is defined according to Table 4 as applying the same methods to *different* data and obtaining similar results. Due to sample variation, the results from new data will not typically be the exact same as from the original dataset. Therefore, it may be difficult to judge whether the results of a replication study indeed "successfully" replicate the original results. In the context of significance testing, different approaches have been proposed for defining replication success (e.g., Hedges, 2019; Held, 2020).

While the 2×2 scheme in Table 4 is rather clear-cut, one might argue that the definition of replication is too narrow, in particular the focus on applying the "same methods". In

[4]While reproducibility is discussed less in the Contributions of this thesis compared to replicability, we have made efforts to make our results reproducible, by using openly available datasets as well as publishing the full codes on Github with instructions for reproduction (for Contributions 2 and 3) or including the code in a supplement (for Contribution 1).

Table 4: The 2×2 scheme of Whitaker (2016) for defining reproducibility and replicability in applied research. The scheme is used in this section as a starting point for discussing the definitions. Overall, the present thesis uses the term "replicability" in a slightly broader sense.

		data	
		same	different
methods	same	reproducibility	replicability
	different	robustness	generalizability

their discussion about replication, Nosek and Errington (2020) argued for less focus on the "repetition of the technical methods" because of the difficulty, non-sensibility, or even impossibility of repeating the exact same procedures and measurements in a new study (for example, when conducting a new survey study in a different country, the original survey must be translated and perhaps also adapted to the cultural context). Instead, Nosek and Errington (2020) focused on the interplay between theory/hypotheses and evidence that is central to empirical science. In research studies, empirical evidence provides support (or not) for claims deduced from a theory. Nosek and Errington (2020) then defined replication as "a study for which any outcome would be considered diagnostic evidence about a claim from prior research".[5]

The authors acknowledged that under this somewhat broader definition, not everyone will agree whether a particular study is indeed a replication study (let alone when replication can be considered "successful"). Moreover, Nosek and Errington (2020) mentioned that in practice, using the same or very similar methods as in the original study can be a reasonable starting point for replication, particularly in fields where theoretical understanding is not yet profound. Nevertheless, the flexibility of the broader definition of replication has its advantages, also in the context of this thesis. For example, in Contribution 1, so-called result-based validation is discussed, which is not based on applying the same methods again, but might still be considered relevant for replicability (see Section 6 for more details). I therefore broadly define replication as re-assessing a claim on new data respectively in a new study, which may often (but not necessarily always) include applying the same methods as in the original study. This flexible definition also fits better with our understanding of the term "replication" in the context of methodological research, as will be discussed next.

[5] In comparison with the definition of replication according to the 2×2 scheme in Table 4, Nosek and Errington (2020) kept the focus on different data, but the applied methods may not be the exact same. Their definition of replication thus verges into the territory of "generalizability" according to the 2×2 scheme. However, Nosek and Errington (2020) still discerned replicability from generalizability: "[T]o be a replication, 2 things must be true: outcomes consistent with a prior claim would increase confidence in the claim, and outcomes inconsistent with a prior claim would decrease confidence in the claim." While replications must fulfill both criteria, generalizability tests do not necessarily fulfill the second criterion.

Reproducibility and replicability in methodological research. The concept of reproducibility can be transferred with relative ease to methodological research, namely by defining reproducibility as obtaining the same results when using the same methods, data, and code as the original authors. Defining replication, however, is to some extent even less clear-cut for methodological research than for applied research. There is not much literature on this topic, with exceptions mostly from the field of computer science (Plesser, 2018; Rougier et al., 2017), and there is again a lack of consensus among authors. Some researchers focus on the aspect of implementation. For example, Rougier et al. (2017) defined replication as "writing and then running new software based on the description of a computational model or method provided in the original publication". In the same article, the authors introduced the ReScience initiative which aims at encouraging reproduction and replication in computer science. For this initiative, several studies have since attempted to reproduce and replicate papers about new cluster algorithms or clustering frameworks (Eijkelboom et al., 2022; Teule et al., 2021). These papers did not always strictly follow the replication definition of Rougier et al. (2017), and instead also considered other modifications to the original study design, such as alternative datasets.

This fits with the broader definition of Boulesteix et al. (2020) who stated that "the goal of [replication] studies would be to confirm the results of previous methodological papers, using, say, alternative simulation designs, other real data sets and a different implementation". While this definition encompasses many cases, it still remains challenging to define replication exactly. To illustrate this, take a study which evaluates a new method, say method A, with a particular study design including several (simulated or real) datasets, competing methods, and evaluation criteria. Is a replication study then constrained to evaluating method A on new datasets or with different simulation designs, but with the same competing methods and evaluation criteria as in the original study? Could a study which uses new datasets, but additionally more competing methods or other evaluation criteria, also be counted as a replication study, or is this a generalizability test (recall the 2×2 scheme of Whitaker, 2016 in Table 4 above, which defines using different "methods" as generalizability)? What about a study which uses the *same* datasets to evaluate method A, but different competing methods or evaluation criteria? Is this (only) a robustness test?

As mentioned above, Nosek and Errington (2020) defined replication as "a study for which any outcome would be considered diagnostic evidence about a claim from prior research". For this definition, it is essential what the "claim from prior research" is. Typically, authors of a study introducing a new method claim that their method is superior over previous approaches in some sense. Usually, the method is not claimed to be better in *every* setting, but it is frequently (explicitly or implicitly) implied that the method's superiority extends beyond the particular datasets that the authors considered. However, the authors often do not clearly define this range of assumed superiority (see Nießl, Hoff-

mann, et al., 2022 for a detailed discussion of this issue), which makes it difficult to define what a replication should look like.

Overall, further research is required on this topic. The lack of clarity regarding the definition of replicability in methodological research is related to the fact that there is much less metascientific literature on methodological research than on applied research (for some exceptions, see the aforementioned articles of Boulesteix et al., 2020 and Nießl, Hoffmann, et al., 2022; as well as Boulesteix, Stierle, et al., 2015; Lohmann et al., 2021; Nießl, Herrmann, et al., 2022; Pawel et al., 2022; and further references in Contribution 3).

For the purposes of this thesis, I use the definition of Boulesteix et al. (2020), that is, replication may encompass using "alternative simulation designs, other real data sets and a different implementation". To this list, I tentatively add using different competing methods and/or alternative evaluation criteria (while acknowledging that other researchers might prefer to define this as "generalizability"). For the replication of studies which introduce new methods, re-evaluating the method in a neutral comparison study is of particular importance. A neutral comparison study is a study whose authors do not have a vested interest in one of the competing methods, and are (as a group) approximately equally familiar with all considered methods (Boulesteix et al., 2013; Boulesteix et al., 2017).

Like for replication in applied research, it is difficult to judge what constitutes "successful replication" of a methodological study. While the thesis does not discuss this issue in detail, notably worse results in replication studies will be considered as an indicator of over-optimism (see Section 5.3 below).

Besides the difficulty of defining (successful) replication, there are also many practical challenges when attempting to replicate a method's performance result with another study design. Nießl, Hoffmann, et al. (2022) performed an illustrative experiment (in which I was involved as a co-author) considering different exemplary methods, including clustering methods, that were recently proposed in the literature. More precisely, Nießl, Hoffmann, et al. (2022) considered pairs of methods (say, method A and method B) that were proposed by different authors for the same data analysis task (e.g., clustering cancer patients based on multi-omics data), and were evaluated by the original authors with study design A and B, respectively. Nießl, Hoffmann, et al. (2022) then evaluated method A with study design B, and method B with study design A (an approach that Nießl, Hoffmann, et al., 2022 call "cross-design validation"), to check whether the original performance results could be replicated with alternative study designs. In accordance with my tentative definition of replication as given above, the term "alternative study design" here refers not only to additional or different datasets, but also to different competing methods and evaluation criteria. The experiment not only demonstrated researcher degrees of freedom in the assessment of novel methods (see Sections 5.3 and 6 below), but also illustrated

many practical challenges encountered when conducting replications of methodological studies, e.g., the choice of a method's parameters when applying it to new datasets.

5.2 Validation on validation data

The term "validation", particularly "validation on validation data", refers to re-assessing a certain result or model on a dataset other than the original one. The term thus has some similarities with "replication". However, we use validation as a somewhat broader term. Replication, particularly in the context of applied research, has a strong focus on "new" data (e.g., independent samples from a different study center). In contrast, validation data might be genuinely new data, but could also be obtained by splitting a single dataset into training/discovery and test/validation data. This broader definition is why we mostly use the term "validation" in Contributions 1 and 2 instead of "replication". In our understanding, validation procedures (e.g., the procedures we discuss in Contribution 1 for clustering) might be used for replication in a new study with new data, but may also be used by authors to check their own results, using data that was obtained by splitting the original dataset before the start of the analysis.

This usage of the term "validation data" is inspired by supervised learning, where the concept is much more established compared to cluster analysis. Here, validation data might also refer to either split-apart or new data. For example, a prediction model assessing cardiovascular risk can be evaluated via *internal* validation; the model is fitted on one part of the data (the training data) and evaluated on the other part (the test or validation data). Often, the split into training and validation data is repeated multiple times, leading to resampling procedures such as cross-validation. Additionally, the prediction model may be evaluated (either by the same authors or by others) with *external* validation, i.e., by assessing the model's predictions on genuinely "new" data, i.e., independent samples (Steyerberg & Harrell, 2016). Note that these terms should not be confused with internal and external validation of clustering results (Section 2.3).

The above paragraph refers to supervised modeling in applied research, i.e., evaluating a concrete model fitted on a particular dataset. On the other hand, in methodological research, the performance of a model-fitting *procedure* is typically of more interest.[6] Validation data is used for both purposes, but for resampling schemes such as cross-validation, careful attention must be paid to which of the two purposes the scheme serves (Bates et al., 2021).

What does this mean for validation in the context of cluster analysis? Supervised learning is different from unsupervised learning. In particular, clustering a dataset does not yield

[6]For example, in an applied study, researchers might want to estimate how well a particular model for predicting cardiovascular risk, fitted via the Random Forest method on a specific dataset, will perform for new samples. In contrast, in methodological research, researchers might be interested in the mean performance of the Random Forest method over multiple samplings from a certain distribution. See Boulesteix, Hable, et al. (2015) for a mathematical formulation of this distinction.

a fitted "model" that can then be used to classify new samples from a validation dataset. Still, some analogies regarding the use of validation data can be drawn. For the context of *applied* research, I will discuss this in Section 6, where Contribution 1 is described in more detail. This thesis does not contain a similarly extensive discussion of validation data in the context of *methodological* research on clustering, although the topic is touched upon in Contribution 3. It would be interesting to explore this topic in more detail, as briefly discussed in Section 7.

Table 5 contains an overview of the terms "reproducibility", "replication", and "validation data" as used in this thesis. As stressed before, this should be considered as a pragmatic outline instead of a definite account.

Table 5: Usage of the terms reproducibility, replication, and validation on validation data in this thesis.

	applied studies	methodological studies
reproducibility	obtaining the exact same results when using the same methods, data, and code as the original authors	
replication	re-assessing a claim from prior research on new data, often using the same or very similar methods	re-assessing a claim from prior research on new data, with a different simulation design, or with a different implementation, potentially also with alternative competing methods and evaluation criteria
validation on validation data	re-assessing results from a present study or from prior research on validation data (either new data or data obtained by splitting the original dataset)	re-assessing results from a present study or from prior research on validation data (either new data or data obtained by splitting the original dataset), potentially also combined with varying other aspects of the study design (see the points in "replication")

5.3 Over-optimism and the multiplicity of analysis strategies

For the purposes of this thesis, *over-optimistic results* are defined as findings that cannot be successfully replicated or validated. While slightly worse results on replication or validation data might be due to chance (sample variation), I use the term over-optimism for *systematic* biases. In particular, Contributions 2 and 3 consider the bias that arises

from the *multiplicity of analysis strategies* (Gelman & Loken, 2014; Hoffmann et al., 2021; Steegen et al., 2016) combined with *selective reporting*.

The term "multiplicity of analysis strategies" refers to the following issue: for a given research question, there is often a plethora of acceptable analysis options and it is frequently unclear which analysis strategy is the "best one", leading to *method uncertainty* (Klau et al., 2020). Consequently, studies contain many *researcher degrees of freedom* (Simmons et al., 2011). In applied studies, these degrees of freedom might, e.g., refer to the choice of network generation method or cluster algorithm for analyzing the dataset at hand (see Contribution 2). In methodological studies, there are also many degrees of freedom regarding the study design, i.e., the choice of the datasets, simulation design, competing methods, evaluation criteria, etc. (see Nießl, Herrmann, et al., 2022; Nießl, Hoffmann, et al., 2022; and Contribution 3).

The multiplicity of analysis strategies may lead to over-optimistic results if multiple analysis options are tried and only the best result is reported (*selective reporting*, sometimes also called *cherry picking*), while less desirable results are left in the figurative "file drawer" (Rosenthal, 1979). In this case, there is risk of having "overfitted" the analysis choice to the data. The term *overfitting* is typically used in the context of supervised learning, often in relation to the (hyper)parameters of a model. When these (hyper)parameters are overly fitted to the noise and "irrelevant" characteristics of the training data, the model will perform worse on validation data. While overfitting is often considered in the context of fitting a model with a single algorithm, several studies have demonstrated that overfitting can also occur through trying different algorithms (e.g., different classification methods) and only reporting the best model (Bernau et al., 2013; Boulesteix, 2010; Boulesteix & Strobl, 2009; Westphal & Brannath, 2020).

Apart from the context of supervised learning, the negative effects of selective reporting have mostly been studied in the context of significance testing, where the multiplicity of analysis options can be exploited with p-hacking (Stefan & Schönbrodt, 2022). For example, researchers who want to perform a t-test for testing the difference between two groups might try different options for outlier handling and imputation of missing values, and report only the strategy that yields the lowest p-value of the t-test (which is obviously a questionable research practice).

To the best of my knowledge, the only study which explicitly analyzes the multiplicity of *clustering* strategies was performed by Beijers et al. (2022). Using a psychiatric dataset, the authors studied the impact of the multiplicity of clustering methods on the resulting number of clusters. This multiplicity was visualized with specification curve analysis (Simonsohn et al., 2020). However, Beijers et al. (2022) did not consider selective reporting, i.e., the effects of only reporting the "best" result. In contrast, Contributions 2 and 3 not only illustrate that the multiplicity of analysis strategies and degrees of freedom in a study design can lead to varied results, but also demonstrate over-optimism arising from

selective reporting.

In applied research, the multiplicity issue has been illustrated by a growing number of *multi-analysts studies*, for example in psychology (Schweinsberg et al., 2021; Silberzahn et al., 2018) and neuroscience (Botvinik-Nezer et al., 2020). The organizers of these studies asked multiple research teams to analyze the same dataset to test the same hypothesis (also called "crowdsourcing research"). There was notable variation in the chosen analysis strategies, as well as in the resulting conclusions. Our approach in Contributions 2 and 3 is different; we do not let multiple researchers analyze a dataset, but instead "model" the behavior of a single research team who tries multiple methods for network generation and clustering on a dataset (Contribution 2), or varies multiple aspects of the study design for demonstrating the "superiority" of a novel clustering method (Contribution 3). This hypothetical research team then selects only the "best" result or study design. In principle, Contributions 2 and 3 could also be interpreted as "modeling" the behavior of *multiple* teams, with each team trying a different analysis strategy, and only the team with the "best" result being able to publish their findings (e.g., due to publication bias).

In this section, I have so far considered over-optimism as unsuccessful replication or validation, stemming from the multiplicity of analysis strategies. Still, other notions of over-optimism are conceivable. For example, over-optimism can be caused by the misuse of statistical tests, e.g., by *selective inference*, also called "double-dipping". In applied studies, the following "cluster evaluation" procedure is still frequently performed. To demonstrate that the clusters are dissimilar from each other, the differences between the cluster means are tested. But as the clustering of the data is used to define the null hypothesis (double-dipping), the Type I error rate is inflated (Gao et al., 2022). In this sense, statements such as "there is a statistically significant difference between the clusters" are likely over-optimistic.

Results generated via double-dipping may be successfully replicable (if the authors of the replication study again use the faulty test procedure), but this replicability obviously does not imply that the results are not over-optimistic. Several studies have recently proposed valid post-inference procedures for testing differences between clusters (Chen & Witten, 2022; Gao et al., 2022; Grabski et al., 2022; Zhang et al., 2019). While this is an important issue, this thesis will not consider this aspect further, and instead focuses on over-optimism caused by the multiplicity of analysis strategies coupled with selective reporting.

6 Summary of the Contributions

The three Contributions of this cumulative thesis consider different aspects of replication, validation, and over-optimism in the context of cluster analysis. This section summarizes each Contribution.

Contribution 1 This article (Ullmann, Hennig, et al., 2022) discusses the role of validation data for the evaluation of clustering results in *applied* research. We address how a clustering obtained on a "discovery dataset" can subsequently be validated on validation data. As discussed in Section 5.2, validation data can result from splitting a single dataset into discovery and validation sets, but could also consist of new independent data. The article was motivated by the observation that applied researchers who perform cluster analysis sometimes use validation data, but that systematic overviews of such procedures were lacking. We thus reviewed the literature to identify various existing approaches, and then structured these approaches in a systematic framework.

We distinguish between two main approaches for using validation data, namely result-based and method-based validation. *Result-based* validation has certain analogies to the evaluation of a supervised model on validation data as discussed in Section 5.2 (although the analogy should not be overstretched). A clustering obtained on the discovery data is used to "predict" clusters of the entities in the validation data, thus yielding a clustering on the validation data. The quality of that clustering is then evaluated, e.g., with classical cluster validation techniques. Indeed, using validation data does not conflict with classical cluster validation techniques such as internal, external, and visual validation (Section 2.3). Rather, these classical procedures can be combined with validation data.

Method-based validation refers to re-applying the same method that yielded the clustering on the discovery data to the validation data. The two clusterings can then be compared, e.g., again via classical cluster validation. This validation approach is evocative of the definition of replication according to the 2×2 scheme of Whitaker (2016) that was explained in Section 5.1 above, where replication was defined as applying the same methods to new data. Indeed, the method-based validation approach could be used for the replication of a clustering study, with the validation data being a new dataset. However, as also mentioned in Section 5.1, there are more flexible definitions of replication such as the one given by Nosek and Errington (2020). Following this definition, it might also be possible to use result-based validation in replication studies.

As mentioned in Section 2.1, we consider both inferential clustering (where the entities to be clustered form a sample drawn from an underlying population) and descriptive clustering (where the entities to be clustered form a fixed set of specific interest). This distinction is relevant for various aspects of the validation framework. For example, in descriptive clustering, the objects to be clustered are the same for both discovery and validation data. For result-based validation, "predicting" the clusters on the validation data is thus trivial; the cluster memberships on the validation data are simply the same as on the discovery data. On the other hand, in inferential clustering, the objects to be clustered are different between discovery and validation data. Therefore, "predicting" the clusters of the objects in the validation data is not trivial, and requires a proper classification step.

Our framework offers guidance to applied researchers who wish to validate or replicate a clustering result. However, we stress that specific recommendations are difficult to make, because a suitable validation approach always depends on the context and aim of a concrete applied study. To help researchers become better acquainted with the concepts discussed in the article, we have illustrated different validation approaches in the supplement. This is presented in the style of a tutorial with R code and openly available datasets.

Contribution 2 This Contribution (Ullmann et al., 2023) is again positioned within the context of *applied* research, more specifically, microbiome research. The validation framework from Contribution 1 is used to demonstrate over-optimistic effects in unsupervised microbiome analysis. Using an exemplary microbiome dataset from the American Gut Project (McDonald et al., 2018), we quantify over-optimistic bias stemming from the multiplicity of methods for network generation and clustering.

For this purpose, we model the approach of a hypothetical researcher who has four unsupervised microbiome research tasks in mind: 1) clustering bacterial genera, 2) detecting "hubs" (influential nodes) in microbial association networks, 3) differential network analysis (comparing networks between two sample groups), and 4) clustering samples. For each task, the hypothetical researcher tries multiple methods and chooses the method yielding the "best" result according to a specific evaluation criterion (e.g., for the first research task, the evaluation criterion is the agreement of the clustering with a previously known taxonomic categorization). Note that we do not assume that the researcher does this with malicious intentions–their behavior might simply be caused by uncertainty regarding which of the many available methods should be used.

This behavior is modeled as follows. For each research task in turn, the microbiome dataset is repeatedly split into discovery and validation sets.[7] Multiple method combinations are applied on the discovery data. For the first three research tasks, these method combinations include different options for steps involved in generating microbial association networks, as displayed in Figure 3. For the fourth research task, multiple method combinations are obtained by varying methods for calculating (dis)similarities between samples. For the first and fourth research task, clustering methods are also part of the method combinations. These include, for example, the network-based cluster algorithms described in Section 4.

After applying multiple method combinations on the discovery data, the combination yielding the "best" result is chosen according to the respective evaluation criterion. The hypothetical researcher would stop here, and report only the best result. To estimate

[7]We split a single dataset into two parts instead of using an independent dataset as validation data. For the latter approach, we could not have determined whether worse performance on the validation data indeed stemmed from the multiplicity of analysis strategies combined with selective reporting, or was simply due to substantial differences between discovery and validation data.

the over-optimistic bias induced by this selective reporting, we re-apply the best method combination to the validation data. (In the terminology of Contribution 1, we use method-based validation.) The results are then compared between discovery and validation data. Worse results on the validation data imply over-optimistic bias. Indeed, for all four research tasks, we detect notable over-optimism effects.

These results illuminate the importance of strategies for avoiding over-optimism. For example, guidance from neutral comparison studies could help to reduce the multiplicity of possible analysis strategies *before* the start of the analysis. Preregistration of the analysis plan, as well as reporting the results of *all* attempted methods, might also help prevent over-optimistic effects.

Contribution 3 In contrast to Contributions 1 and 2, Contribution 3 (Ullmann, Beer, et al., 2022) adopts a metascientific perspective on *methodological* research. The article analyzes the over-optimistic presentation of novel cluster algorithms. In methodological research in general, authors who present a new method typically claim that this method is superior to existing approaches. However, such claims cannot always be taken at face value, because publication bias constitutes an incentive for authors to present their new method as favorably as possible (Boulesteix, Stierle, et al., 2015). Therefore, studies introducing new methods are likely to be over-optimistic, in the sense that the good performance results cannot be replicated in comparison studies later performed by other authors.

What mechanisms lead to the over-optimistic presentation of a new method? For supervised learning, this question was addressed by Jelizarow et al. (2010) and Pawel et al. (2022). Each study considered a "promising" novel method (for classification or regression), which in reality was not superior to other methods. Yet, the authors were able to demonstrate that a favorable presentation of the new method's performance could still be achieved, namely by exploiting researcher degrees of freedom in the evaluation design. This issue was also discussed by Dehghani et al. (2021), who noted that in many subfields of supervised machine learning, there is no consensus on which study design should be used for evaluating a novel method. This, in turn, allows researchers to find an experimental setup that best fits their new method. Another related work is the study of Nießl, Hoffmann, et al. (2022) (see Section 5.1), who demonstrated researcher degrees of freedom regarding the performance assessment of novel methods by conducting a cross-design validation experiment as described above.

We apply an approach similar to the studies of Jelizarow et al. (2010) and Pawel et al. (2022), to analyze over-optimization mechanisms specifically for the case of new clustering methods. Indeed, we argue that over-optimistic evaluation concerns novel cluster algorithms just as much as novel methods for supervised learning. To illustrate the issue, we use the recently proposed cluster algorithm Rock (Beer et al., 2019) as an example. Rock

was initially deemed to be a promising approach, but was later revealed to generally not perform better than alternative clustering methods. We demonstrate that Rock can still appear superior to competing methods by "optimizing" the study design and selectively reporting only the "optimal" design settings. More precisely, this concerns 1) optimizing the datasets on which Rock is evaluated, 2) optimizing Rock's hyperparameters without using validation data in a suitable way (or neglecting to properly tune the hyperparameters of the competing methods), and 3) optimizing the choice of the competing methods. Recall that Section 5.3 defined over-optimistic results as findings that cannot be successfully replicated or validated. We show that Rock's performance result, as obtained with the above optimizations, is indeed over-optimistic by demonstrating that the "superiority" of the algorithm disappears when we use a study design different from the "optimal" settings (e.g., when using datasets different from the "optimal" ones).
Our illustration provokes the discussion of possible solutions to the problem of over-optimism in methodological clustering research. For example, we recommend that after developing a novel clustering method, researchers should evaluate this method on fresh validation data that has been kept apart during the development phase and initial assessment of the algorithm (e.g., to detect possible overfitting of the algorithm's hyperparameters to the datasets used in the initial phase). We also note the importance of *neutral* benchmark studies, whose results will often be more reliable than the results of studies which introduce new methods. As publication bias is an institutional issue, we also stress the role of journals, funders, universities, etc. in tackling the over-optimism problem.

7 Outlook

Based on the ideas presented in this thesis, this section discusses possible directions for future research.

Illustrating over-optimism in different applied research fields Contribution 2 demonstrates over-optimism effects stemming from the multiplicity of clustering strategies in the context of microbiome analysis. The issue of multiplicity coupled with selective reporting is not constrained to microbiome research; in numerous other application contexts, there is also uncertainty about which method(s) to use for clustering, and over-optimistic bias is to be expected. It would therefore be of interest to quantify the extent of over-optimistic effects in further research fields. An approach analogous to the study design of Contribution 2 could be used for that purpose.

Dealing with the multiplicity of clustering strategies Contribution 2 discusses some strategies for alleviating the problem of over-optimistic effects in applied clustering research. For example, we advise against selectively reporting a single clustering result af-

ter multiple methods were tried. Instead, reporting the results of all analyses is preferable. But how can researchers report and visualize multiple clusterings in a clear and accessible manner? As previously mentioned, Beijers et al. (2022) visualized the multiplicity of clustering methods for analyzing a psychiatric dataset with specification curve analysis (Simonsohn et al., 2020). Alternatives to specification curve analysis were recently summarized by Hoffmann et al. (2021), including, e.g., "multiverse analysis" (Steegen et al., 2016) and the "vibration of effects" framework (Patel et al., 2015). These approaches were designed for significance testing and/or explanatory and predictive modeling. It might be interesting to explore whether the frameworks could be transferred to the case of cluster analysis, and whether they might help applied researchers in reporting the results of different clustering methods in a systematic manner.

Additionally or alternatively to reporting the multiplicity of methods and results, researchers could try to *integrate* method uncertainty (Hoffmann et al., 2021). More precisely, the results of different cluster algorithms could be combined into a single clustering via cluster ensemble methods (Fred & Jain, 2005; Strehl & Ghosh, 2002), an approach inspired by ensemble learning in supervised classification.

However, cluster ensembles combining the results of very different algorithms should not be applied blindly. Recall from Section 2.1 that each cluster algorithm is based on a certain cluster concept. Before the start of the analysis, researchers should carefully think about which type of clusters they are looking for, and which concept best suits the context and aim of their analysis (Hennig, 2015). Still, even if researchers can decide on a single cluster concept, there are often several algorithms which address the same or a similar concept (e.g., network-based cluster algorithms as explained in Section 4). In this case, combining the results of these algorithms via a cluster ensemble approach might reduce over-optimistic bias, compared to trying multiple algorithms and selectively reporting a single result.

Neutral comparison studies Both Contributions 2 and 3 stress the importance of neutral benchmark studies to compare different clustering methods. Van Mechelen et al. (2018) provided guidance for performing such studies. Several neutral benchmark studies have already been published. For example, Hennig (2022) compared popular clustering methods on several real datasets in a rather general context (i.e., without focusing on a particular application field). Benchmark studies comparing clustering methods in more specific contexts have also been performed, e.g., regarding clustering methods for single-cell RNA-seq data (Duò et al., 2018), or methods for cancer subtyping using multi-omics data (Duan et al., 2021). In addition to these existing studies, it would be desirable if further benchmark studies were published. For example, there is a lack of such studies in the context of microbiome data.

As new clustering methods are continually introduced each year, systematic benchmark

studies risk being "outdated" soon after publication. To alleviate this issue, benchmark studies should be published together with a public repository containing the used code and datasets, ideally in such a way that other researchers could easily add novel methods to the existing comparisons (Weber et al., 2019); for an example of this practice, see the study of Duò et al. (2018) mentioned above. In the same context (single-cell RNA-seq data), Germain et al. (2020) developed an R framework that can be used for the benchmarking of clustering pipelines. The flexibility of the framework allows for extensible benchmarks. Ultimately, such efforts reflect a move from "static" to "dynamic" benchmarking (Mangul et al., 2019; Robinson & Vitek, 2019). Note, however, that software infrastructures which allow the continual updating of benchmark studies can be time-consuming to develop and maintain. It would be informative to explore best practices for this process.

Using validation data in methodological clustering research As mentioned in Section 5.2, this thesis discusses using validation data for clustering evaluation mostly in the applied context. Regarding the methodological context, Contribution 3 mentions the importance of using fresh datasets after developing a novel clustering method (see also Section 6 above), and briefly touches upon using validation data obtained by splitting a single dataset. It would be interesting to analyze how validation data could be more routinely included in methodological clustering research, both in studies introducing new methods as well as in studies comparing existing methods. In supervised learning, using validation data for the evaluation of methods is routine, and suitable resampling schemes have been extensively discussed in the literature. For example, it is well known that for evaluating a supervised classifier in combination with hyperparameter optimization, *nested* resampling schemes are required in order to avoid over-optimistic performance evaluation (Bischl et al., 2021). It might be insightful to study whether such schemes would also be sensible for the evaluation of clustering methods, where hyperparameters (such as the number k of clusters) are frequently optimized, e.g., via resampling-based stability methods.

Clarifying the distinction between exploratory and confirmatory research in cluster analysis Cluster analysis is often considered to be exploratory; the clustering is performed to gain a first impression of the data, without fixed hypotheses in mind. This raises the question whether careful validation of the results, e.g., with validation data, is as important as it would be for confirmatory research. Indeed, if the cluster analysis is performed purely for exploratory purposes, then following a strict procedure including validation data may not always be required (as long as it is clearly reported that the analysis was exploratory in nature). Yet cluster analysis may not always be constrained to exploratory research. Consider the first example in Section 1, namely the clustering of cancer patients (Burstein et al., 2015; Curtis et al., 2012; The Cancer Genome Atlas

Network, 2012). The authors of these studies did not just perform cluster analysis for visualization purposes or to gain a first impression of the data; rather, the clear aim was to detect novel cancer subtypes. While there were no pre-specified hypotheses in the sense of confirmatory hypothesis testing, the authors aimed to find clusters related to clinical outcomes, e.g., clusters associated with survival. In Contribution 2, we also assume that the hypothetical researcher has specific evaluation criteria in mind when clustering the microbiome data, even if the cluster analysis is a priori unsupervised.
If researchers have specific aims when performing the clustering, this might go beyond the scope of exploratory research, and careful validation of the results is of particular importance, even more so if the clustering is eventually intended to be used in practice (e.g., using cancer subtypes to develop tailored treatments). Cluster analysis may also verge into confirmatory research if researchers try to replicate the clustering result of another research team. To the best of my knowledge, it has not been systematically discussed, so far, in what sense cluster analysis may sometimes be considered as "confirmatory". A more detailed discussion of this issue might be of interest, particularly with regards to what this means for suitable validation procedures.

In summary, this thesis has discussed topics such as validation, (non-)replicability, and over-optimism in the context of cluster analysis. Going forward, I hope that this thesis motivates the use of good research practices in cluster analysis, which subsequently would help increase reliability and replicability for both clustering results in applied research and performance evaluations of novel clustering methods.

References

Aitchison, J. (1982). The statistical analysis of compositional data. *Journal of the Royal Statistical Society: Series B (Methodological)*, *44*(2), 139–160.

Anders, S., & Huber, W. (2010). Differential expression analysis for sequence count data. *Nature Precedings*. https://doi.org/10.1038/npre.2010.4282.1

Andrade, C. (2021). HARKing, cherry-picking, p-hacking, fishing expeditions, and data dredging and mining as questionable research practices. *The Journal of Clinical Psychiatry*, *82*(1), 20f13804.

Bacon, F. (1995). *Novum organum* (P. Urbach & J. Gibson, Eds.). Open Books. (Original work published 1620)

Badri, M., Kurtz, Z. D., Bonneau, R., & Müller, C. L. (2020). Shrinkage improves estimation of microbial associations under different normalization methods. *NAR Genomics and Bioinformatics*, *2*(4), lqaa100.

Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. *Nature*, *533*, 452–454.

Bates, S., Hastie, T., & Tibshirani, R. (2021). Cross-validation: What does it estimate and how well does it do it? *arXiv preprint*. https://doi.org/10.48550/arXiv.2104.00673

Beer, A., Kazempour, D., & Seidl, T. (2019). Rock-let the points roam to their clusters themselves. *Proceedings of the 22nd International Conference on Extending Database Technology (EDBT)*, 630–633.

Begley, C. G., & Ellis, L. M. (2012). Raise standards for preclinical cancer research. *Nature*, *483*, 531–533.

Beijers, L., van Loo, H. M., Romeijn, J.-W., Lamers, F., Schoevers, R. A., & Wardenaar, K. J. (2022). Investigating data-driven biological subtypes of psychiatric disorders using specification-curve analysis. *Psychological Medicine*, *52*(6), 1089–1100.

Bernau, C., Augustin, T., & Boulesteix, A.-L. (2013). Correcting the optimal resampling-based error rate by estimating the error rate of wrapper algorithms. *Biometrics*, *69*(3), 693–702.

Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A.-L., Deng, D., & Lindauer, M. (2021). Hyperparameter optimization: Foundations, algorithms, best practices and open challenges. *arXiv preprint*. https://doi.org/10.48550/arXiv.2107.05847

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2008*(10), P10008.

Borg, I., & Groenen, P. J. F. (2005). *Modern multidimensional scaling: Theory and applications* (2nd ed.). Springer.

Botvinik-Nezer, R., Holzmeister, F., Camerer, C. F., Dreber, A., Huber, J., Johannesson, M., Kirchler, M., Iwanir, R., Mumford, J. A., Adcock, R. A., Avesani, P., Baczkowski, B. M., Bajracharya, A., Bakst, L., Ball, S., Barilari, M., Bault, N., Beaton, D., Beitner, J., ... Schonberg, T. (2020). Variability in the analysis of a single neuroimaging dataset by many teams. *Nature*, *582*(7810), 84–88.

Boulesteix, A.-L. (2010). Over-optimism in bioinformatics research. *Bioinformatics*, *26*(3), 437–439.

Boulesteix, A.-L., Hable, R., Lauer, S., & Eugster, M. J. (2015). A statistical framework for hypothesis testing in real data comparison studies. *The American Statistician*, *69*(3), 201–212.

Boulesteix, A.-L., Hoffmann, S., Charlton, A., & Seibold, H. (2020). A replication crisis in methodological research? *Significance*, *17*(5), 18–21.

Boulesteix, A.-L., Lauer, S., & Eugster, M. J. (2013). A plea for neutral comparison studies in computational sciences. *PloS One*, *8*(4), e61562.

Boulesteix, A.-L., Stierle, V., & Hapfelmeier, A. (2015). Publication bias in methodological computational research. *Cancer Informatics*, *14s5*, 11–19.

Boulesteix, A.-L., & Strobl, C. (2009). Optimal classifier selection and negative bias in error rate estimation: An empirical study on high-dimensional prediction. *BMC Medical Research Methodology*, *9*, 85.

Boulesteix, A.-L., Wilson, R., & Hapfelmeier, A. (2017). Towards evidence-based computational statistics: Lessons from clinical research on the role and design of real-data benchmark studies. *BMC Medical Research Methodology*, *17*, 138.

Burstein, M. D., Tsimelzon, A., Poage, G. M., Covington, K. R., Contreras, A., Fuqua, S. A., Savage, M. I., Osborne, C. K., Hilsenbeck, S. G., Chang, J. C., Mills, G. B., Lau, C. C., & Brown, P. H. (2015). Comprehensive genomic analysis identifies novel subtypes and targets of triple-negative breast cancer. *Clinical Cancer Research*, *21*(7), 1688–1698.

Callahan, B. J., McMurdie, P. J., Rosen, M. J., Han, A. W., Johnson, A. J. A., & Holmes, S. P. (2016). DADA2: High-resolution sample inference from Illumina amplicon data. *Nature Methods*, *13*(7), 581–583.

Camerer, C. F., Dreber, A., Forsell, E., Ho, T.-H., Huber, J., Johannesson, M., Kirchler, M., Almenberg, J., Altmejd, A., Chan, T., Heikensten, E., Holzmeister, F., Imai, T., Isaksson, S., Nave, G., Pfeiffer, T., Razen, M., & Wu, H. (2016). Evaluating replicability of laboratory experiments in economics. *Science*, *351*(6280), 1433–1436.

Camerer, C. F., Dreber, A., Holzmeister, F., Ho, T.-H., Huber, J., Johannesson, M., Kirchler, M., Nave, G., Nosek, B. A., Pfeiffer, T., Altmejd, A., Buttrick, N., Chan, T., Chen, Y., Forsell, E., Gampa, A., Heikensten, E., Hummer, L., Imai, T., ... Wu, H. (2018). Evaluating the replicability of social science experiments in Nature and Science between 2010 and 2015. *Nature Human Behaviour*, *2*(9), 637–644.

Chen, Y. T., & Witten, D. M. (2022). Selective inference for k-means clustering. *arXiv preprint*. https://doi.org/10.48550/arXiv.2203.15267

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*(8), 790–799.

Clauset, A., Newman, M. E., & Moore, C. (2004). Finding community structure in very large networks. *Physical Review E*, *70*(6), 066111.

Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(5), 603–619.

Curtis, C., Shah, S. P., Chin, S.-F., Turashvili, G., Rueda, O. M., Dunning, M. J., Speed, D., Lynch, A. G., Samarajiwa, S., Yuan, Y., Gräf, S., Ha, G., Haffari, G., Bashashati, A., Russell, R., McKinney, S., METABRIC Group, Langerød, A., Green, A., ... Aparicio, S. (2012). The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, *486*, 346–352.

Dehghani, M., Tay, Y., Gritsenko, A. A., Zhao, Z., Houlsby, N., Diaz, F., Metzler, D., & Vinyals, O. (2021). The benchmark lottery. *arXiv preprint*. https://doi.org/10.48550/arXiv.2107.07002

Dohlman, A. B., Mendoza, D. A., Ding, S., Gao, M., Dressman, H., Iliev, I. D., Lipkin, S. M., & Shen, X. (2021). The cancer microbiome atlas: A pan-cancer comparative analysis to distinguish tissue-resident microbiota from contaminants. *Cell Host & Microbe*, *29*(2), 281–298.

Duan, R., Gao, L., Gao, Y., Hu, Y., Xu, H., Huang, M., Song, K., Wang, H., Dong, Y., Jiang, C., Zhang, C., & Jia, S. (2021). Evaluation and comparison of multi-omics data integration methods for cancer subtyping. *PLoS Computational Biology*, *17*, e1009224.

Duò, A., Robinson, M. D., & Soneson, C. (2018). A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000Research*, *7*, 1141.

Eijkelboom, F., Fokkema, M., Lau, A., & Verheijen, L. (2022). [Re] Reproduction study of variational fair clustering. *ReScience C*, *8*(2), #14.

Errington, T. M., Mathur, M., Soderberg, C. K., Denis, A., Perfito, N., Iorns, E., & Nosek, B. A. (2021). Investigating the replicability of preclinical cancer biology. *eLife*, *10*, e71601.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 226–231.

Everitt, B. S., Landau, S., Leese, M., & Stahl, D. (2011). *Cluster analysis* (5th ed.). John Wiley & Sons.

Faust, K., Sathirapongsasuti, J. F., Izard, J., Segata, N., Gevers, D., Raes, J., & Huttenhower, C. (2012). Microbial co-occurrence relationships in the human microbiome. *PLoS Computational Biology*, *8*(7), e1002606.

Fred, A. L., & Jain, A. K. (2005). Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(6), 835–850.

Freeman, L. C. (1978). Centrality in social networks: Conceptual clarification. *Social Networks*, *1*(3), 215–239.

Friedman, J., & Alm, E. J. (2012). Inferring correlation networks from genomic survey data. *PLoS Computational Biology*, *8*(9), e1002687.

Fukunaga, K., & Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, *21*(1), 32–40.

Gabelica, M., Bojčić, R., & Puljak, L. (2022). Many researchers were not compliant with their published data sharing statement: Mixed-methods study. *Journal of Clinical Epidemiology*, *150*, 33–41.

Galili, T. (2015). dendextend: An R package for visualizing, adjusting, and comparing trees of hierarchical clustering. *Bioinformatics*, *31*(22), 3718–3720.

Gao, L. L., Bien, J., & Witten, D. (2022). Selective inference for hierarchical clustering. *Journal of the American Statistical Association*. https://doi.org/10.1080/01621459.2022.2116331

Garrido-Castro, A. C., Lin, N. U., & Polyak, K. (2019). Insights into molecular classifications of triple-negative breast cancer: Improving patient selection for treatment. *Cancer Discovery*, *9*(2), 176–198.

Gelman, A., & Loken, E. (2014). The statistical crisis in science: Data-dependent analysis—a "garden of forking paths"—explains why many statistically significant comparisons don't hold up. *American Scientist*, *102*(6), 460.

Germain, P.-L., Sonrel, A., & Robinson, M. D. (2020). pipeComp, a general framework for the evaluation of computational pipelines, reveals performant single cell RNA-seq preprocessing tools. *Genome Biology*, *21*, 227.

Gloor, G. B., Macklaim, J. M., Pawlowsky-Glahn, V., & Egozcue, J. J. (2017). Microbiome datasets are compositional: And this is not optional. *Frontiers in Microbiology*, *8*, 2224.

Goodman, S. N., Fanelli, D., & Ioannidis, J. P. (2016). What does research reproducibility mean? *Science Translational Medicine*, *8*(341), 341ps12.

Grabski, I. N., Street, K., & Irizarry, R. A. (2022). Significance analysis for clustering with single-cell RNA-sequencing data. *bioRxiv preprint*. https://doi.org/10.1101/2022.08.01.502383

Gundersen, O. E., & Kjensmo, S. (2018). State of the art: Reproducibility in artificial intelligence. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1), 1644–1651.

Hardwicke, T. E., Mathur, M. B., MacDonald, K., Nilsonne, G., Banks, G. C., Kidwell, M. C., Hofelich Mohr, A., Clayton, E., Yoon, E. J., Henry Tessler, M., Lenne, R. L., Altman, S., Long, B., & Frank, M. C. (2018). Data availability, reusability, and analytic reproducibility: Evaluating the impact of a mandatory open data policy at the journal Cognition. *Royal Society Open Science*, *5*(8), 180448.

Hardwicke, T. E., Serghiou, S., Janiaud, P., Danchev, V., Crüwell, S., Goodman, S. N., & Ioannidis, J. P. (2020). Calibrating the scientific ecosystem through meta-research. *Annual Review of Statistics and Its Application*, *7*, 11–37.

Head, M. L., Holman, L., Lanfear, R., Kahn, A. T., & Jennions, M. D. (2015). The extent and consequences of p-hacking in science. *PLoS Biology*, *13*(3), e1002106.

Hedges, L. V. (2019). The statistics of replication. *Methodology*, *15*, 3–14.

Held, L. (2020). A new standard for the analysis and design of replication studies. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *183*(2), 431–448.

Hennig, C. (2015). What are the true clusters? *Pattern Recognition Letters*, *64*, 53–62.

Hennig, C. (2022). An empirical comparison and characterisation of nine popular clustering methods. *Advances in Data Analysis and Classification*, *16*(1), 201–229.

Hennig, C., & Meila, M. (2015). Cluster analysis: An overview. In C. Hennig, M. Meila, F. Murtagh, & R. Rocci (Eds.), *Handbook of cluster analysis* (pp. 1–19). Chapman & Hall/CRC.

Hoffmann, S., Schönbrodt, F., Elsas, R., Wilson, R., Strasser, U., & Boulesteix, A.-L. (2021). The multiplicity of analysis strategies jeopardizes replicability: Lessons learned across disciplines. *Royal Society Open Science*, *8*(4), 201925.

Hofner, B., Schmid, M., & Edler, L. (2016). Reproducible research in statistics: A review and guidelines for the Biometrical Journal. *Biometrical Journal*, *58*(2), 416–427.

Hullman, J., Kapoor, S., Nanayakkara, P., Gelman, A., & Narayanan, A. (2022). The worst of both worlds: A comparative analysis of errors in learning from data in psychology and machine learning. *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, 335–348.

Hutson, M. (2018). Artificial intelligence faces reproducibility crisis. *Science*, *359*(6377), 725–726.

Ioannidis, J. P. (2005). Why most published research findings are false. *PLoS Medicine*, *2*(8), e124.

Jelizarow, M., Guillemot, V., Tenenhaus, A., Strimmer, K., & Boulesteix, A.-L. (2010). Over-optimism in bioinformatics: An illustration. *Bioinformatics*, *26*(16), 1990–1998.

Kapoor, S., & Narayanan, A. (2022). Leakage and the reproducibility crisis in ML-based science. *arXiv preprint*. https://doi.org/10.48550/arXiv.2207.07048

Klau, S., Martin-Magniette, M.-L., Boulesteix, A.-L., & Hoffmann, S. (2020). Sampling uncertainty versus method uncertainty: A general framework with applications to omics biomarker selection. *Biometrical Journal*, *62*(3), 670–687.

Kurtz, Z. D., Müller, C. L., Miraldi, E. R., Littman, D. R., Blaser, M. J., & Bonneau, R. A. (2015). Sparse and compositionally robust inference of microbial ecological networks. *PLoS Computational Biology*, *11*(5), e1004226.

Li, H. (2015). Microbiome, metagenomics, and high-dimensional compositional data analysis. *Annual Review of Statistics and Its Application*, *2*, 73–94.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, *28*(2), 129–137.

Loftus, M., Hassouneh, S. A.-D., & Yooseph, S. (2021). Bacterial community structure alterations within the colorectal cancer gut microbiome. *BMC Microbiology*, *21*, 98.

Lohmann, A., Astivia, O. L. O., Morris, T., & Groenwold, R. H. (2021). It's time! 10+ 1 reasons we should start replicating simulation studies. *PsyArXiv preprint*. https://doi.org/10.31234/osf.io/agsnt

Lovell, D., Pawlowsky-Glahn, V., Egozcue, J. J., Marguerat, S., & Bähler, J. (2015). Proportionality: A valid alternative to correlation for relative data. *PLoS Computational Biology*, *11*(3), e1004075.

Mangul, S., Martin, L. S., Hill, B. L., Lam, A. K.-M., Distler, M. G., Zelikovsky, A., Eskin, E., & Flint, J. (2019). Systematic benchmarking of omics computational tools. *Nature Communications*, *10*(1), 1393.

McDonald, D., Hyde, E., Debelius, J. W., Morton, J. T., Gonzalez, A., Ackermann, G., Aksenov, A. A., Behsaz, B., Brennan, C., Chen, Y., et al. (2018). American gut: An open platform for citizen science microbiome research. *mSystems*, *3*(3), e00031–18.

Meinshausen, N., & Bühlmann, P. (2006). High-dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, *34*(3), 1436–1462.

Mohar, B. (1991). The Laplacian spectrum of graphs. In Y. Alavi, G. Chartrand, O. R. Oellermann, & A. J. Schwenk (Eds.), *Graph theory, combinatorics, and applications* (pp. 871–898). Wiley.

Newman, M. E. (2004). Analysis of weighted networks. *Physical review E*, *70*(5), 056131.

Newman, M. E., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, *69*(2), 026113.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, 849–856.

Nießl, C., Herrmann, M., Wiedemann, C., Casalicchio, G., & Boulesteix, A.-L. (2022). Over-optimism in benchmark studies and the multiplicity of design and analysis options when interpreting their results. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *12*(2), e1441.

Nießl, C., Hoffmann, S., Ullmann, T., & Boulesteix, A.-L. (2022). Explaining the optimistic performance evaluation of newly proposed methods: A cross-design validation experiment. *arXiv preprint*. https://doi.org/10.48550/arXiv.2209.01885

Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D., Breckler, S. J., Buck, S., Chambers, C. D., Chin, G., Christensen, G., Contestabile, M., Dafoe, A., Eich, E., Freese, J., Glennerster, R., Goroff, D., Green, D. P., Hesse, B., Humphreys, M., ... Yarkoni, T. (2015). Promoting an open research culture. *Science*, *348*(6242), 1422–1425.

Nosek, B. A., Ebersole, C. R., DeHaven, A. C., & Mellor, D. T. (2018). The preregistration revolution. *Proceedings of the National Academy of Sciences*, *115*(11), 2600–2606.

Nosek, B. A., & Errington, T. M. (2020). What is replication? *PLoS Biology*, *18*(3), e3000691.

Nuzzo, R. (2015). How scientists fool themselves–and how they can stop. *Nature News*, *526*, 182–185.

Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science*, *349*(6251), aac4716.

Pastushenko, I., Brisebarre, A., Sifrim, A., Fioramonti, M., Revenco, T., Boumahdi, S., Van Keymeulen, A., Brown, D., Moers, V., Lemaire, S., De Clercq, S., Minguijón, E., Balsat, C., Sokolow, Y., Dubois, C., De Cock, F., Scozzaro, S., Sopena, F., Lanas, A., ... Blanpain, C. (2018). Identification of the tumour transition states occurring during EMT. *Nature*, *556*, 463–468.

Patel, C. J., Burford, B., & Ioannidis, J. P. (2015). Assessment of vibration of effects due to model specification can demonstrate the instability of observational associations. *Journal of Clinical Epidemiology*, *68*(9), 1046–1058.

Pawel, S., Kook, L., & Reeve, K. (2022). Pitfalls and potentials in simulation studies. *arXiv preprint*. https://doi.org/10.48550/arXiv.2203.13076

Pearce, D., & Rantala, V. (1983). New foundations for metascience. *Synthese*, *56*(1), 1–26.

Peschel, S., Müller, C. L., von Mutius, E., Boulesteix, A.-L., & Depner, M. (2021). NetCoMi: Network construction and comparison for microbiome data in R. *Briefings in Bioinformatics*, *22*(4), bbaa290.

Plesser, H. E. (2018). Reproducibility vs. replicability: A brief history of a confused terminology. *Frontiers in Neuroinformatics*, *11*, 76.

Prat, A., Pineda, E., Adamo, B., Galván, P., Fernández, A., Gaba, L., Dıez, M., Viladot, M., Arance, A., & Muñoz, M. (2015). Clinical implications of the intrinsic molecular subtypes of breast cancer. *The Breast*, *24*, S26–S35.

Quinn, T. P., Erb, I., Richardson, M. F., & Crowley, T. M. (2018). Understanding sequencing data as compositions: An outlook and review. *Bioinformatics*, *34*(16), 2870–2878.

Quinn, T. P., Richardson, M. F., Lovell, D., & Crowley, T. M. (2017). propr: An R-package for identifying proportionally abundant features using compositional data analysis. *Scientific Reports*, *7*, 16252.

Robinson, M. D., & Vitek, O. (2019). Benchmarking comes of age. *Genome Biology*, *20*, 205.

Rosenthal, R. (1979). The file drawer problem and tolerance for null results. *Psychological Bulletin*, *86*(3), 638–641.

Röttjers, L., & Faust, K. (2018). From hairballs to hypotheses–biological insights from microbial networks. *FEMS Microbiology Reviews*, *42*(6), 761–780.

Röttjers, L., & Faust, K. (2020). manta: A clustering algorithm for weighted ecological networks. *mSystems*, *5*(1), e00903–19.

Rougier, N. P., Hinsen, K., Alexandre, F., Arildsen, T., Barba, L. A., Benureau, F. C., Brown, C. T., De Buyl, P., Caglayan, O., Davison, A. P., et al. (2017). Sustainable computational science: The ReScience initiative. *PeerJ Computer Science*, *3*, e142.

Schoch, C. L., Ciufo, S., Domrachev, M., Hotton, C. L., Kannan, S., Khovanskaya, R., Leipe, D., Mcveigh, R., O'Neill, K., Robbertse, B., Sharma, S., Soussov, V., Sullivan, J. P., Sun, L., Turner, S., & Karsch-Mizrachi, I. (2020). NCBI Taxonomy: A comprehensive update on curation, resources and tools. *Database (Oxford)*, *2020*, baaa062.

Schooler, J. W. (2014). Metascience could rescue the 'replication crisis'. *Nature*, *515*, 9.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)*, *42*(3), 19.

Schweinsberg, M., Feldman, M., Staub, N., van den Akker, O. R., van Aert, R. C. M., van Assen, M. A. L. M., Liu, Y., Althoff, T., Heer, J., Kale, A., Mohamed, Z., Amireh, H., Venkatesh Prasad, V., Bernstein, A., Robinson, E., Snellman, K., Amy Sommer, S., Otner, S. M. G., Robinson, D., ... Luis Uhlmann, E. (2021). Same data, different conclusions: Radical dispersion in empirical results when independent analysts operationalize and test the same hypothesis. *Organizational Behavior and Human Decision Processes*, *165*, 228–249.

Shreiner, A. B., Kao, J. Y., & Young, V. B. (2015). The gut microbiome in health and in disease. *Current Opinion in Gastroenterology*, *31*(1), 69.

Silberzahn, R., Uhlmann, E. L., Martin, D. P., Anselmi, P., Aust, F., Awtrey, E., Bahník, Š., Bai, F., Bannard, C., Bonnier, E., Carlsson, R., Cheung, F., Christensen, G., Clay, R., Craig, M. A., Dalla Rosa, A., Dam, L., Evans, M. H., Flores Cervantes, I., ... Nosek, B. A. (2018). Many analysts, one data set: Making transparent how variations in analytic choices affect results. *Advances in Methods and Practices in Psychological Science*, *1*(3), 337–356.

Simmons, J. P., Nelson, L. D., & Simonsohn, U. (2011). False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, *22*(11), 1359–1366.

Simonsohn, U., Nelson, L. D., & Simmons, J. P. (2014). P-curve: A key to the file-drawer. *Journal of Experimental Psychology: General*, *143*(2), 534.

Simonsohn, U., Simmons, J. P., & Nelson, L. D. (2020). Specification curve analysis. *Nature Human Behaviour*, *4*(11), 1208–1214.

Sneath, P. H. (1957). The application of computers to taxonomy. *Microbiology*, *17*(1), 201–226.

Sokal, R. R., & Michener, C. D. (1958). A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, *38*, 1409–1438.

Sorensen, T. A. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. *Biologiske Skrifter*, *5*, 1–34.

Steegen, S., Tuerlinckx, F., Gelman, A., & Vanpaemel, W. (2016). Increasing transparency through a multiverse analysis. *Perspectives on Psychological Science*, *11*(5), 702–712.

Stefan, A., & Schönbrodt, F. (2022). Big little lies: A compendium and simulation of p-hacking strategies. *PsyArXiv preprint*. https://doi.org/10.31234/osf.io/xy2dk

Sterling, T. D. (1959). Publication decisions and their possible effects on inferences drawn from tests of significance—or vice versa. *Journal of the American Statistical Association*, *54*(285), 30–34.

Steyerberg, E. W., & Harrell, F. E. (2016). Prediction models need appropriate internal, internal–external, and external validation. *Journal of Clinical Epidemiology*, *69*, 245–247.

Strehl, A., & Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, *3*(Dec), 583–617.

Teule, T., Reints, N., Gerges, C. A., & Baanders, P. (2021). [Re] Deep fair clustering for visual learning. *ReScience C*, *7*(2), #4.

The Cancer Genome Atlas Network. (2012). Comprehensive molecular portraits of human breast tumours. *Nature*, *490*, 61–70.

Tsilimigras, M. C., & Fodor, A. A. (2016). Compositional data analysis of the microbiome: Fundamentals, tools, and challenges. *Annals of Epidemiology*, *26*(5), 330–335.

Ullmann, T., Beer, A., Hünemörder, M., Seidl, T., & Boulesteix, A.-L. (2022). Over-optimistic evaluation and reporting of novel cluster algorithms: An illustrative study. *Advances in Data Analysis and Classification*. https://doi.org/10.1007/s11634-022-00496-5

Ullmann, T., Hennig, C., & Boulesteix, A.-L. (2022). Validation of cluster analysis results on validation data: A systematic framework. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *12*(3), e1444.

Ullmann, T., Peschel, S., Finger, P., Müller, C. L., & Boulesteix, A.-L. (2023). Over-optimism in unsupervised microbiome analysis: Insights from network learning and clustering. *PLoS Computational Biology*, *19*(1), e1010820.

Van Dongen, S. M. (2000). *Graph clustering by flow simulation* (Doctoral dissertation).

Van Mechelen, I., Boulesteix, A.-L., Dangl, R., Dean, N., Guyon, I., Hennig, C., Leisch, F., & Steinley, D. (2018). Benchmarking in cluster analysis: A white paper. *arXiv preprint*. https://doi.org/10.48550/arXiv.1809.10496

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, *17*(4), 395–416.

Von Luxburg, U., Williamson, R. C., R.C., & Guyon, I. (2012). Clustering: Science or art? In I. Guyon, G. Dror, V. Lemaire, & G. Taylor (Eds.), *Proceedings of ICML Workshop on Unsupervised and Transfer Learning* (pp. 65–79).

Wasserstein, R. L., Schirm, A. L., & Lazar, N. A. (2019). Moving to a world beyond "$p < 0.05$". *The American Statistician*, *73*(sup1), 1–19.

Weber, L. M., Saelens, W., Cannoodt, R., Soneson, C., Hapfelmeier, A., Gardner, P. P., Boulesteix, A.-L., Saeys, Y., & Robinson, M. D. (2019). Essential guidelines for computational method benchmarking. *Genome Biology*, *20*, 125.

Weiss, Y. (1999). Segmentation using eigenvectors: A unifying view. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, *2*, 975–982.

Westphal, M., & Brannath, W. (2020). Evaluation of multiple prediction models: A novel view on model selection and performance assessment. *Statistical Methods in Medical Research*, *29*(6), 1728–1745.

Whitaker, K. (2016). Showing your working: A guide to reproducible neuroimaging analyses. https://doi.org/10.6084/m9.figshare.4244996.v1

Yoon, G., Carroll, R. J., & Gaynanova, I. (2020). Sparse semiparametric canonical correlation analysis for data of mixed types. *Biometrika*, *107*(3), 609–625.

Yoon, G., Gaynanova, I., & Müller, C. L. (2019). Microbial networks in SPRING – semiparametric rank-based correlation and partial correlation estimation for quantitative microbiome data. *Frontiers in Genetics*, *10*, 516.

Yoon, G., Müller, C. L., & Gaynanova, I. (2021). Fast computation of latent correlations. *Journal of Computational and Graphical Statistics*, *30*(4), 1249–1256.

Zhang, J. M., Kamath, G. M., & David, N. T. (2019). Valid post-clustering differential analysis for single-cell RNA-seq. *Cell Systems*, *9*(4), 383–392.

Zheng, C., Zheng, L., Yoo, J.-K., Guo, H., Zhang, Y., Guo, X., Kang, B., Hu, R., Huang, J. Y., Zhang, Q., Liu, Z., Dong, M., Hu, X., Ouyang, W., Peng, J., & Zhang, Z. (2017). Landscape of infiltrating T cells in liver cancer revealed by single-cell sequencing. *Cell*, *169*(7), 1342–1356.

A Contribution 1: "Validation of cluster analysis results on validation data: A systematic framework"

This chapter is a reprint of:

Ullmann, T., Hennig, C., & Boulesteix, A.-L. (2022). Validation of cluster analysis results on validation data: A systematic framework. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *12*(3), e1444

Copyright:

Author contributions:

T. Ullmann conceptualized the paper together with A.-L. Boulesteix (ideas, formulation of overarching aims, etc.), taking comments from C. Hennig into account. T. Ullmann was the lead for the methodology of the article; she conducted the literature review and developed the validation framework. C. Hennig and A.-L. Boulesteix provided comments for the latter task. The majority of the original draft was written by T. Ullmann, with the exception of Sections 2.4 ("Visual validation") and 3.4.3 ("Validating (Vis): Visual patterns") which were written by C. Hennig. A.-L. Boulesteix made several comments regarding the original draft that were included in the manuscript. The review and editing of the manuscript were performed jointly by all three authors. T. Ullmann wrote the supporting information containing an illustrative analysis for the validation of clustering results on validation data, taking comments from A.-L. Boulesteix and C. Hennig into account. All authors read and approved the final version of the article.

Received: 1 March 2021 | Revised: 16 November 2021 | Accepted: 27 November 2021

DOI: 10.1002/widm.1444

ADVANCED REVIEW

WIREs DATA MINING AND KNOWLEDGE DISCOVERY WILEY

Validation of cluster analysis results on validation data: A systematic framework

Theresa Ullmann[1] | **Christian Hennig**[2] | **Anne-Laure Boulesteix**[1]

[1]Institute for Medical Information Processing, Biometry and Epidemiology, Ludwig-Maximilians-Universität München, Munich, Germany

[2]Dipartimento di Scienze Statistiche "Paolo Fortunati", Universita di Bologna, Bologna, Italy

Correspondence
Theresa Ullmann, Institute for Medical Information Processing, Biometry and Epidemiology, Ludwig-Maximilians-Universität München, Marchioninistraße 15, 81377, Munich, Germany.
Email: tullmann@ibe.med.uni-muenchen.de

Funding information
Bundesministerium für Bildung und Forschung, Grant/Award Number: 01IS18036A; Deutsche Forschungsgemeinschaft, Grant/Award Number: BO3139/7-1

Edited by: Witold Pedrycz, Editor-in-Chief

Abstract

Cluster analysis refers to a wide range of data analytic techniques for class discovery and is popular in many application fields. To assess the quality of a clustering result, different cluster validation procedures have been proposed in the literature. While there is extensive work on classical validation techniques, such as internal and external validation, less attention has been given to validating and replicating a clustering result using a validation dataset. Such a dataset may be part of the original dataset, which is separated before analysis begins, or it could be an independently collected dataset. We present a systematic, structured review of the existing literature about this topic. For this purpose, we outline a formal framework that covers most existing approaches for validating clustering results on validation data. In particular, we review classical validation techniques such as internal and external validation, stability analysis, and visual validation, and show how they can be interpreted in terms of our framework. We define and formalize different types of validation of clustering results on a validation dataset, and give examples of how clustering studies from the applied literature that used a validation dataset can be seen as instances of our framework.

This article is categorized under:
Technologies > Structure Discovery and Clustering
Algorithmic Development > Statistics
Technologies > Machine Learning

KEYWORDS
cluster stability, cluster validation, clustering, independent data, replication

1 | INTRODUCTION

Cluster analysis refers to data analytic techniques for structure and class discovery. It is popular in a range of fields, for example, medicine, biology, market research, social science, and data compression. However, when conducting cluster analysis, researchers are confronted with an overwhelming number of existing methods. They must preprocess the data, choose a clustering algorithm, and set parameters, such as the number of clusters (Van Mechelen et al., 2018;

Zimmermann, 2020). It is often unclear a priori which choice should be made for the analysis, and even once a choice is made, it may remain unclear how good the quality of the resulting clustering is.

These problems have prompted the development of so-called *cluster validation* techniques, see Handl et al. (2005) and Hennig (2015a) for overviews. The literature distinguishes between internal validation (where the clustering is evaluated based on internal properties, such as compactness and separateness of the clusters) and external validation (where the clustering is evaluated by comparing the clusters with respect to one or more variables not used for clustering, e.g., a survival time or a true class membership). Less attention has been given to the validation and replication of clustering results on a *validation dataset*, for which we introduce a structured framework that summarizes the existing literature in a systematic manner. A validation dataset could be part of the original dataset, set apart before the start of the analysis, or it could be a separate dataset, obtained, for example, from a different study centre.

The idea of validating a clustering on another dataset is not new and has appeared in the methodological literature decades ago (Breckenridge, 1989; McIntyre & Blashfield, 1980). In applied literature involving cluster analysis, it is not uncommon for authors to validate their clustering results on new data, be it with the procedure of McIntyre and Blashfield (1980) or another method. To the best of our knowledge, these approaches have never been systematically structured and evaluated, and different validation strategies are scattered across different works and application fields. This contrasts with the abundant methodological literature devoted to validation in the context of *supervised* classification (or more generally, supervised learning). This contrast may be partly due to the fact that cluster analysis—as opposed to supervised classification—is often viewed as exploratory research. The validation of clustering results is rightly considered to be less straightforward than the validation of a prediction model because "true labels" are unknown (Von Luxburg et al., 2012). Indeed, it is difficult to define exactly what is meant by validating a clustering on validation data. Answering this question is the key aspect of our framework.

In this article, we aim to give a systematic review of the various strategies used in the literature for validating clustering results on validation data. These existing approaches are combined into a structured framework. In this framework, we define and formalize the concept of validation on a validation dataset. In particular, we demonstrate that many classical validation techniques, such as internal and external validation, stability analysis, and visual validation, can be linked to evaluation on validation data: using validation datasets does not replace these approaches; rather, classical validation can be combined with validation data. Moreover, we show how clustering studies from the applied literature that used a validation dataset can be classified into our framework.

Why do researchers consider validation and replication of clustering results on a validation dataset to be important? The answer is closely tied to the clustering aim, which could either be *inferential* or *descriptive*. We define these terms as follows:

- *Inferential clustering*: The objects being clustered form a sample drawn from an underlying population for which inference is of interest, rather than making statements about the specific objects in the original dataset.
- *Descriptive clustering*: The data form a fixed set of entities of specific interest, and statements such as objects 1, 5, and 99 form a cluster are of interest.

As an example of the difference between inferential and descriptive clustering, consider an $n \times p$ dataset including the expression levels (continuous values) of p genes for n patients suffering from a particular disease, see Figure 1.

On the one hand, it may be of interest to perform clustering analyses of the patients to see if there are subpopulations of patients with systematically different gene expressions. This would be *inferential clustering*. For example, researchers have frequently used gene expression data to detect distinct breast cancer subtypes (Burstein et al., 2015; Curtis et al., 2012; Kapp et al., 2006; Lehmann et al., 2011; Sørlie et al., 2003; Sotiriou et al., 2003). Such subtypes can have clinical implications and may guide targeted treatment (Garrido-Castro et al., 2019; Prat et al., 2015). On the other hand, an $n \times p$ gene expression dataset could also be used to perform clustering of the (fixed set of) p genes to see if there are groups of specific genes that behave similarly, which might suggest a similar function or involvement in a common molecular process. This is an example of *descriptive clustering*. For example, researchers have used cluster analysis to find different groups of cancer-related genes (Freudenberg et al., 2009; Yang et al., 2014; Zhang et al., 2014).

For both clustering aims, using validation data is of crucial importance. To illustrate this, we again use the gene expression example, see Figure 1. First, we consider inferential clustering of breast cancer patients. All the papers for breast cancer subtype detection cited above used validation datasets to confirm the results of their own analyses and/or to validate previously reported subtypes. Indeed, a clustering of cancer patients would not be of much use if it only held on a single dataset. Due to the inferential nature of the clustering, the researchers' aim is to better understand the

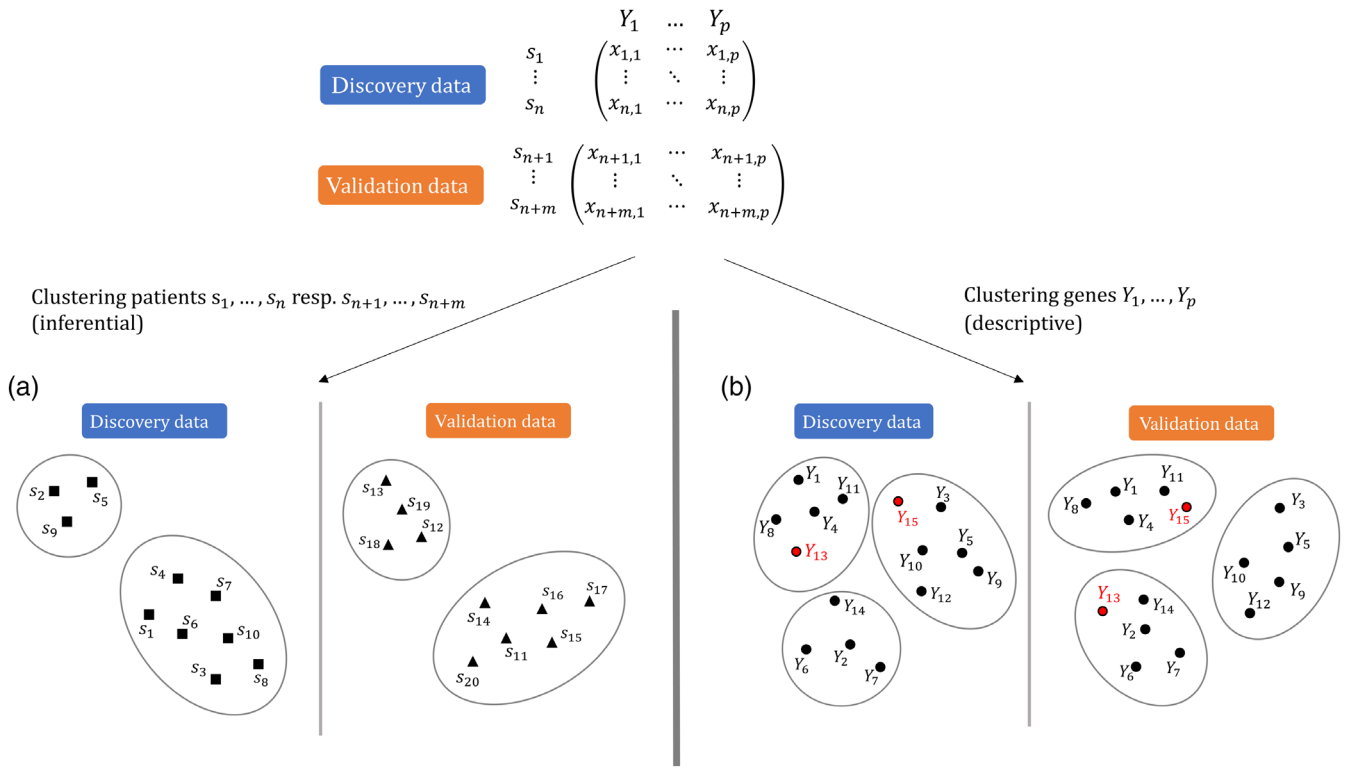


FIGURE 1 Schematic representation of clustering on a gene expression dataset. (a) Inferential clustering of the patients and (b) descriptive clustering of the genes. For illustration purposes, there are 10 patients $s_1,...,s_{10}$ in the discovery data, 10 patients $s_{11},...,s_{20}$ in the validation data, and 15 genes $Y_1,...,Y_{15}$. For inferential clustering, the objects to cluster (here, patients) are different between discovery and validation data, as indicated by using different symbols (squares vs. triangles). The two resulting clusterings nevertheless look somewhat similar: a smaller cluster on the top left, and a larger cluster on the bottom right. For descriptive clustering, the objects to cluster (here, genes $Y_1,...,Y_{15}$, marked by circles) remain the same across both datasets. However, their positions are slightly shifted in the validation data, because the gene expression values now stem from patients $s_{11},...,s_{20}$. Consequently, genes Y_{13} and Y_{15} (marked in red) are clustered differently on the validation data

disease and to find options for treatment with respect to the underlying population, for example, the population of *all* breast cancer patients. In particular, the clustering should not only hold for patients from a single hospital or a single country. To make sure that the clustering is not just an artifact of a single dataset, researchers thus use independently collected samples for validation, or at least split their dataset into discovery and validation sets.

Now consider the example of the descriptive clustering of cancer-related genes. While the set of genes is fixed, researchers typically want the gene clustering to hold more generally than only for the n specific patients. The genes' functions or involvement in molecular processes should reflect biological principles that hold for all patients with the particular cancer type, and researchers thus want to recover the clustering on datasets with other patients having the same disease. Again, validation datasets are used for this purpose. In this sense, descriptive clustering can have an inferential component, with the difference to "inferential clustering" (as defined above) being that the objects to be clustered are fixed and do not represent samples drawn from an underlying population. This has implications for choosing a suitable validation dataset and validation strategy, as will be discussed in more detail in Section 3 below.

Similar arguments about the importance of replicability and generalisability of clusterings results (as given for the example of gene expression data above) hold more generally for most cluster analysis applications, which can typically be classified as either inferential or descriptive clustering. For example, in market segmentation (inferential clustering of customers), the resulting clusters should be replicable such that managers can consistently market their products to the customer groups (Dolnicar & Leisch, 2010; Müller & Hamm, 2014). In text and keyword analysis, where words are clustered to reveal overarching topics (descriptive clustering), it is interesting to see whether topics stay stable on validation data, or whether some changes appear (Ding et al., 2001). Across different application fields, researchers usually want their results to be as generalizable as possible. Interesting properties of a clustering result should hold not only for a single specific dataset, but should also reappear when clustering validation data sampled from the same, or even

different distributions. Validating clusterings on validation data also enables researchers to evaluate results reported by other research teams. The confirmation of results on validation data is a vital part of research in general, and it has received considerable attention in recent years due to the so-called "replication crisis" (Hutson, 2018). In the context of classical hypothesis tests and effect estimates, many published results have turned out to be non-replicable, that is, they could not be confirmed on independent data [e.g., in psychology (Open Science Collaboration, 2015), cancer research (Begley & Ellis, 2012), or economics (Camerer et al., 2016)]. Replication is thus vital for assessing the credibility of scientific claims (Nosek & Errington, 2020). For cluster analysis, our article appears to be the first one to systematically review and discuss this topic.

Our framework, which is described in detail in Section 3, is based on the following two-step cluster analysis procedure (see also Figure 2):

1. *The primary cluster analysis and method selection step*: Using the original dataset or a part of it (in the following called "discovery data") a single clustering method is selected (where the "method" includes not only the choice of clustering algorithm, but also parameters such as the number of clusters and diverse pre/postprocessing steps), for example, via its performance with respect to internal/external validation indices.
2. *The validation step*: Important aspects of the clustering resulting from this method are validated on another dataset or the rest of the original dataset (in the following denoted by "validation data"). The validation data should be completely hidden from the method selection process of Step 1—analogously to the evaluation of supervised classifiers, where the selected model (including the chosen parameters) must be finally evaluated using validation data that was *not* used in any way for parameter tuning or model selection (Boulesteix et al., 2008; Simon et al., 2003).

The "important aspects" of the clustering that are checked in Step 2 usually depend on the research question and the field of application. Consider again the above example of clustering cancer patients, based on expression levels of cancer-related genes, for the purpose of finding subtypes of that disease. In this context, the following properties might be relevant aspects of the clustering:

- Suppose that Step 1 has resulted in two clusters. One cluster is much larger than the other, with about 80% percent of the patients in this cluster. One might be interested in whether this pattern of one large cluster and one smaller cluster can be replicated in Step 2.
- Assume it is found that the clustering chosen in Step 1 is related to survival time, that is, the patients' survival times differ depending on which cluster they belong to. Can this finding be replicated in Step 2 for patients in the validation data?

In the literature, the term "cluster validation" is sometimes used to refer to the use of validation techniques as a tool to compare different clusterings and select the most appropriate. This use of terminology would place validation within Step 1. But when validation techniques are used as selection tool, it is still an open issue whether the results generalize to new data, and this is addressed by Step 2.

The phrase "cluster validation" also appears in the literature about *benchmarking* of clustering methods (Boulesteix & Hatz, 2017; Van Mechelen et al., 2018; Zimmermann, 2020). A benchmarking study is a systematic comparison of different clustering *methods* on a class of data distributions or datasets. Validation techniques may be used to compare different methods. Benchmark studies thus analyze the "validity" of clustering methods and provide general guidance on which method to use. In contrast, our review considers the validation of specific results of applied clustering studies.

This article is structured as follows: in Section 2, we give an overview of the different uses of the term "validation" and perspectives on validity found in the clustering literature. We then present our validation framework in detail in Section 3. In Section 4, we demonstrate in an exemplary manner how clustering studies from the applied literature can

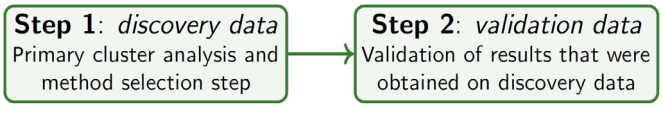


FIGURE 2 Two-step procedure for validating clustering results

be sorted into the framework. Section 5 contains a final discussion. In the Supporting Information, we present an illustration of the discussed validation strategies using openly available real-world data, where the data analysis is performed with thoroughly commented R code.

2 | DIFFERENT PERSPECTIVES ON "VALIDITY" IN CLUSTER ANALYSIS

We identified four approaches that address the validity of clusterings in the literature: (1) the comparison of "true" cluster labels with inferred clusters, (2) internal and external validity indices, (3) stability analyses, and (4) visual validation. These four approaches are briefly reviewed in the following subsections. An additional approach, hypothesis testing, is briefly discussed in Section 5. Internal and external validation, stability, and visual validation form the building blocks of our framework, see Section 3.

2.1 | Recovery of "true" clusters and analogies to the validity of supervised classification models

According to this perspective, a clustering of a dataset is "valid" if it corresponds to the "true" cluster structure in the data. Correspondingly, a clustering method is called "valid" if it can recover the "true" clusters in the data (Breckenridge, 1989; Milligan & Cooper, 1987). A related view is presented in the paper of Dougherty et al. (2007), which shows a connection to the term "validity" in the context of supervised classification. For supervised classification models, the validation of a classifier relates to estimating the *prediction error* on a test set, that is, how well the classifier can predict the known "true" labels of the instances in the test set. Dougherty et al. (2007) demonstrate that this approach can be transferred to cluster analysis. However, this requires datasets with *known* cluster labels. Yet, in practice, cluster analysis is usually applied to real datasets for which the "true" cluster labels of the data points are unknown. Note that even in the rare case of a cluster analysis performed on a dataset with given "true" cluster labels, these may not be unique, and there might be other equally legitimate cluster structures in the data, which can be even more interesting and useful as a result of the analysis than the one previously know (see Färber et al., 2010; Hennig, 2015b). When validating a clustering on validation data, the validation step used in supervised classification usually cannot be mimicked. The idea of Dougherty et al. (2007) thus mainly makes sense in the context of benchmark studies comparing clustering methods using simulated data with known "true" cluster labels. The ability of the methods to recover the true clusters may then be used as a performance criterion. To evaluate clusterings in applied studies, other options for validation are needed.

2.2 | Internal and external validation

In the absence of "true" cluster labels, assessing "cluster validity" often uses so-called internal indices or external information—leading to the terms "internal validation" and "external validation," respectively.

- *Internal validation* uses only the data that was used for clustering. Typically, internal validation consists of calculating an index that is supposed to measure how well the clustering fits the data (Halkidi et al., 2015). Such indices often exploit the proximity structure of the data, for example, by measuring the homogeneity and/or the separation of the clusters. Examples are the Average Silhouette Width index (Kaufman & Rousseeuw, 2009) and the Caliński–Harabasz index (Caliński & Harabasz, 1974). These indices combine measurements of the homogeneity and the separation of a clustering into a single value, in order to balance a small within-cluster heterogeneity and a large between-clusters heterogeneity. There are also indices that measure only isolated aspects of a clustering (e.g., only the homogeneity or only the separation of the clusters), see Akhanli and Hennig (2020).
- *External validation* makes use of additional (external) information that was *not* used for clustering. For example, when clustering a cancer gene expression dataset, one may use the survival time of patients to determine whether the clustering of patients based on gene expression can predict survival. The term "external validation" also encompasses the recovery of previously known "true labels" as presented in Section 2.1.

2.3 | Stability

Many authors consider *stability* to be a crucial aspect of cluster validity. The idea is that a good clustering method should yield similar partitions when applied to multiple datasets drawn from the same data distribution (Ben-David et al., 2006; Von Luxburg, 2010). In this spirit, a specific clustering of a single real dataset may be considered as validated if the clusterings obtained from datasets generated from the same data distribution are similar. There are several methods of generating multiple datasets to emulate the data distribution of the dataset to be analyzed, for example, by drawing subsamples from the original dataset (Hennig, 2007).

Stability analysis dates back to McIntyre and Blashfield (1980), Morey et al. (1983), and Breckenridge (1989). These authors considered the replicability of a clustering result on a validation dataset. To generate the validation dataset, the original data is split into two halves (by splitting along the objects to be clustered for inferential clustering, or by splitting across the variables of the dataset for descriptive clustering). This is followed by assessing whether the clustering obtained in the first half can be replicated in the second half. For descriptive clustering, because the objects in the two halves are the same, replicability can be assessed directly with a partition similarity index such as the Adjusted Rand Index (ARI; Hubert & Arabie, 1985; Rand, 1971), the Jaccard index (Jaccard, 1908), or the FM index (Fowlkes & Mallows, 1983). See Meila (2015) and Albatineh et al. (2006) for overviews of partition similarity indices. For inferential clustering, the objects to cluster are not the same in the two data halves, and thus the objects from the second half have to be classified into the clusters of the first half, before the clusterings can be compared with a partition similarity index (see Section 3.3 for details). Such stability analyses will indeed be a special case of the broader validation framework presented in Section 3.

In the decades that followed, however, the focus of stability analysis shifted away from this concept and more towards *method or model selection*. Like other validation techniques, stability analyses are used in Step 1 (see Figure 2) as a basis for the selection of a suitable clustering method and its parameters, such as the number of clusters (Ben-Hur et al., 2002; Bertrand & Mufti, 2006; Dolnicar & Leisch, 2010; Dudoit & Fridlyand, 2002; Fang & Wang, 2012; Fu & Perry, 2020; Lange et al., 2004; Levine & Domany, 2001; Monti et al., 2003; Tibshirani & Walther, 2005; Wang, 2010). In these approaches, stability analysis selects the clustering method that is most stable over multiple subsamples. The subsamples are drawn without replacement or in a cross-validation manner, or are bootstrap samples drawn with replacement from the data. For example, different numbers of clusters k can be considered in turn, and the k that leads to the most stable clustering, or the smallest k that exceeds a stability threshold, can be chosen. These studies typically consider inferential clustering, such that the term "subsamples" refers to subsets of objects to be clustered. Some schemes require the comparison of clusterings on subsets of objects that consist of disjunct subsamples of the original dataset and thus have no overlap (e.g., Dudoit & Fridlyand, 2002; Fang & Wang, 2012; Lange et al., 2004; Tibshirani & Walther, 2005; Wang, 2010). This requires the aforementioned supervised classification step for classifying observations of one sample to the clusters of the other sample. However, the approaches could in principle be modified to also apply to descriptive clustering.

When splitting the dataset multiple times to determine the stability of a clustering method or parameter, eventually information from the whole dataset enters the method selection process. Thus putting aside a validation dataset that is only used *after* the method selection is advised. Even if a clustering is chosen by stability analysis on a discovery dataset, it is *not* guaranteed that this clustering can be validated on a validation dataset.

Stability analysis can also be combined with classical internal validation indices by checking whether internal validation indices have similar values for multiple clusterings calculated on subsamples of the data (Jain & Moreau, 1987), see also Dangl and Leisch (2020) for a related approach. This idea will also be part of our framework in Section 3.

2.4 | Visual validation

Cluster analysis is often exploratory without fixed predefined expectations from the user. Patterns in the data that qualify to be interpreted as clusters can have very diverse appearances. Some key characteristics of clusters, such as being areas of high density separated by areas of lower density, are difficult to translate into easily computable statistics. Furthermore, many clustering methods rely on model assumptions and cluster concepts, the appropriateness of which is hard to diagnose by means other than visual. This explains why visual validation is important in cluster analysis.

Clusters can be declared valid based on visualization if they correspond to clearly visible patterns in the data, or in some cases if the assumptions required for the chosen clustering method look valid.

Useful plots for visual cluster validation can be distinguished into:

1. General purpose data plots in which found clusters can be indicated by colors or glyphs, such as scatterplots, matrix plots, principal components biplots, multidimensional scaling, or parallel coordinates plots (Cook & Swayne, 2007, chapter 5). There are also projection pursuit approaches that generate "interesting" data projections, potentially showing clustering structure, without requiring the clustering as input (e.g., Tyler et al., 2009).
2. Plots set up to visualize a specific clustering, which can be further classified as:

a. Plots that visualize the original data directly, such as cluster heatmaps (Hahsler & Hornik, 2011; Wilkinson & Friendly, 2009) or projections to optimally discriminate clusters (Hennig, 2004).
b. Plots that visualize the clustering solution without representing the original observations directly such as dendrograms, silhouette plots, and neighborhood graphs (Leisch, 2008).

We refer to the Supporting Information for an illustration of some of these methods.

Plots that visualize the original data directly can be used to assess patterns in data space, although these plots come with either information loss by dimension reduction, or heavy reliance on aspects such as variable and observation ordering. The advantage of plots that optimize objective functions dependent on the clustering, such as discriminant projections or heatmaps with orderings determined by the clustering, is that they have better chances to bring out the data patterns corresponding to the clustering than general purpose plots. On the other hand, they may lead to an overoptimistic assessment of the validity of the clustering, or an interpretation of spurious patterns. Validation data that is kept separate from the beginning of the analysis may help to avoid overoptimism, see Section 3.4.

Some of the plots that do not represent the original observations directly can also be valuable for cluster validation. The silhouette plot accompanies the Average Silhouette Width index (Kaufman & Rousseeuw, 2009) and gives observation-wise information about the quality of assignment in the given clustering; dendrograms visualize the hierarchical merging process and can sometimes reveal issues, such as potentially meaningful clusters disappearing at higher levels of the hierarchy.

3 | A SYSTEMATIC FRAMEWORK FOR VALIDATING A CLUSTERING ON A VALIDATION DATASET

In this section, we present a systematic framework for validating a clustering on a validation dataset that includes many existing approaches from the literature as special cases and revisits them more formally. We also show how the validation methods that we reviewed in the last section are incorporated into the framework.

We first discuss what is meant by a "validation dataset" in Section 3.1. In Section 3.2, we give an overview of properties of a clustering result that may be validated on the validation set (these properties are strongly related to the classical validation procedures discussed in Section 2). In Section 3.3, we outline the distinction between method-based and result-based validation on a validation dataset. In Section 3.4, we combine the concepts of Sections 3.2 and 3.3 into an overview of strategies for validation on validation data. In Section 3.5, we discuss how to judge whether "successful" validation has been achieved.

3.1 | Validation datasets

The term "validation dataset" can refer to a dataset composed of independently collected data (e.g., collected by other researchers or in a different laboratory) which is similar enough to the original data for cluster evaluation to be possible. In practice, however, genuinely independent data is often not available. In this case, one might split a single dataset into a discovery and a validation set.

Apart from this consideration, the structure of the validation data depends on two further aspects: (a) The data for clustering can either be object by variable data or object by object proximity data (where the term "proximity" denotes

either similarities or dissimilarities), see Van Mechelen et al. (2018). Here, "objects" denote the entities which are to be clustered. (b) The aim of the clustering could either be inferential or descriptive, as defined in the introduction (Section 1).

For inferential clustering, the validation data consists of more objects to cluster. On the other hand, for descriptive clustering, validation data does *not* consist of more objects because the set of objects to be clustered is fixed. Consider the example of the $n \times p$ gene expression dataset as described in the introduction. This dataset can be understood as an object by variable dataset in two ways. For inferential clustering of the patients, the patients are the "objects" and the genes the "variables." A validation dataset consists of more patients. For descriptive clustering of the genes, now the genes constitute the "objects," and the patients are the "variables." A validation dataset consists of more variables, that is, again of more patients.

In Table 1, we give a general overview of the structure of the validation data, where we distinguish between inferential and descriptive clustering as well as between object by variable and object by object data.

If separately collected data is not available, and the dataset must be split into discovery and validation sets, a 50/50 split ratio is usually chosen. Indeed, we believe that this choice makes sense in most cases: validation strategies often require the number of data points in the validation set to not be too small when trying to validate certain properties obtained from the clustering on the discovery set. A similar argument has been made in the context of stability analysis (Lange et al., 2004).

3.2 | Clustering properties to be validated

In the literature, we identified four categories of properties of clusterings that researchers may want to validate.

(Int) Internal properties of the clusters (that turn up when clustering the discovery data), for example:

- descriptive measures of the clusters such as the values of the cluster centroids or the relative sizes of the clusters,
- the value of an internal validation index calculated for the clustering result, and
- subsets of variables that characterize the clusters.

(Ext) Associations of the clusters with external variables or agreement of the clustering with an externally known partition. Some examples:

- Clusters of cancer patients have different mean survival rates.
- A clustering of genes shows some agreement with known functional gene labels. For example, a clustering may be compatible with known partitions of the genes into functional categories. Less restrictively, some particular genes, of which this was previously expected, may be in the same cluster.

(Vis) Characteristics that can be assessed using visualization: do the clusters correspond to distinctive meaningful patterns in the data? Do the clusters look how they were supposed to look like? This could refer to model assumptions for the clustering method, or a priori hypotheses or requirements by the researcher.

TABLE 1 Structure of the validation data depending on inferential versus descriptive clustering and object by variable versus object by object data

	Inferential clustering	Descriptive clustering
Object by variable data	Validation data: further objects, same variables	Validation data: further variables, same objects
If a single dataset is split	Split performed along the objects	Split performed along the variables
Object by object data	Validation data: proximity matrix of further objects	Validation data: proximity matrix of same objects, but with proximities derived from another source (e.g., based on different underlying variables).
If a single dataset is split	Objects can be split into two disjoint sets, yielding two smaller proximity matrices (one representing the discovery data, the other the validation data).	Impossible to split proximity data directly into discovery and validation data, but may be possible to split underlying variables.

(Stab) Stability of cluster membership: Does cluster membership remain stable when the same method (algorithm, number of clusters, etc.) is applied to the validation data? Since the objects in the discovery and the validation set are disjunct in the case of inferential clustering, this involves supervised classification of objects of one dataset to clusters of the other dataset.

Most subsections of Section 2 correspond to a category in the above list, with the exception of Section 2.1 (recovery of "true" clusters). If the "true" cluster labels are indeed known, this can be considered as a part of (Ext).

3.3 | Method-based and result-based validation

The validation of a clustering on a specific dataset can refer either to the validity of the used clustering *method*, or the validity of the clustering *result* itself. While this distinction is often not made clear in the literature on classical validation procedures, it has important implications for how validation on a validation dataset is performed. We thus distinguish between *method-based* and *result-based* validation on validation data, as illustrated in Figure 3. In the following, we explain these terms in more detail.

We denote the discovery data by D_1 and the validation data by D_2. The clustering chosen on D_1 in Step 1 (method selection, see Figure 2) is called C_1. Given C_1, the validation dataset can be handled in two different ways:

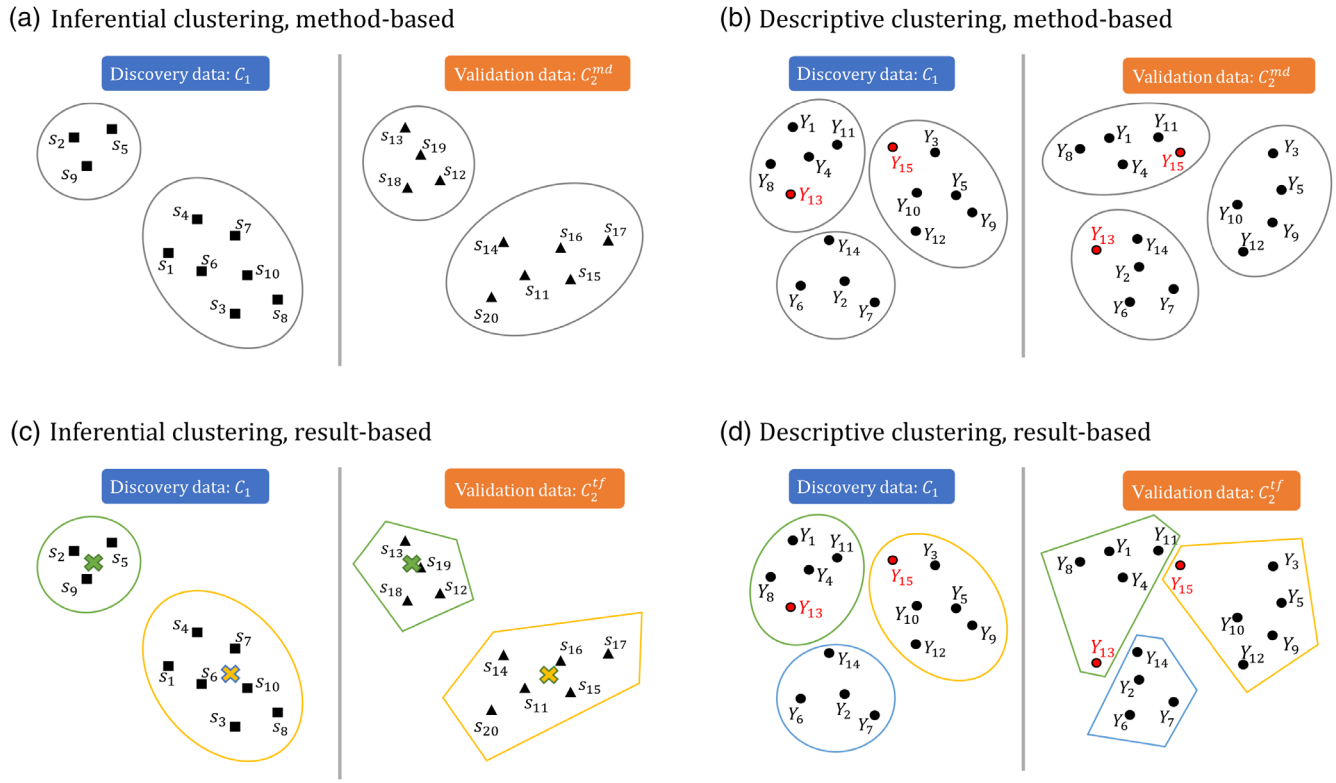


FIGURE 3 Method- and result-based validation for inferential and descriptive clustering. We use the same data example as in Figure 1. The top panel (a) and (b) (method-based validation) is from Figure 1. For inferential clustering (a), re-applying the clustering method to the validation data again detects a smaller cluster on the top left and a larger one on the bottom right. For descriptive clustering (b), the clustering C_2^{md} on the validation data groups the elements Y_{13} and Y_{15} (marked in red) differently than the clustering C_1 on the discovery data. The bottom panel (c) and (d) (result-based validation) illustrates the classification procedures that yield C_2^{tf}. The clusterings C_2^{tf} are depicted as polygons. The colors of the polygons match the corresponding clusters on the discovery data. For inferential clustering, nearest-centroid classification is depicted: the green and yellow crosses represent the centroids of C_1. The samples in the validation data are then assigned to the nearest centroid. In this particular example, the resulting clustering C_2^{tf} in (c) is equal to C_2^{md} in (a): in our terminology, the criterion (Stab) is perfectly fulfilled. For descriptive clustering (d), the most obvious way of transferring C_1 to the validation data is to set the cluster memberships in C_2^{tf} equal to those of C_1. In particular, the elements Y_{13} and Y_{15} are clustered as in C_1. Comparing C_2^{tf} with C_2^{md} in (b) shows that the cluster memberships are not perfectly stable according to criterion (Stab)

a. The same clustering method that yielded C_1 (i.e., same algorithm, same number of cluster k, etc.) can be applied to D_2, yielding a clustering C_2^{md} on D_2 ("md" for "method"). C_1 and C_2^{md} can then be compared with respect to aspects (Int), (Ext), or (Vis). We call this approach *method-based validation*. It puts a focus on the structural similarity of the clustering results as generated by the method.
b. Instead of applying the clustering method again, C_1 can be "transferred" to the validation data by using a supervised classifier to predict the cluster labels of the validation set (explained in more detail below). This results in a clustering C_2^{tf} on D_2 ("tf" for "transferred"). The transferred clustering can be compared to the original clustering C_1 with respect to aspects (Int), (Ext), or (Vis). We call this approach *result-based validation*. It puts a focus on whether the specific clustering result is also sensible for the validation data.

We now explain what we mean by "transferring" the clustering. For descriptive clustering, C_2^{tf} is simply C_1 (recall that for descriptive clustering, the objects to be clustered are the same for D_1 and D_2, and thus C_1 can immediately be considered to be a clustering of D_2). For inferential clustering, the objects to be clustered are different in the discovery and validation sets, so some proper "transfer" is required. This can be done using a supervised classifier (using the labeled discovery set (D_1, C_1) as "training set") to assign the objects in D_2 to the clusters in C_1 (Akhanli & Hennig, 2020; Lange et al., 2004). For example, one can calculate the centroids of the clusters in C_1, and then assign each sample in D_2 to its nearest centroid ("nearest-centroid classifier"). As C_2^{tf} is supposed to be an "extension" or "transfer" of the original clustering to the validation data, one should use a classifier that fits the assignment rule of the chosen clustering algorithm as closely as possible. The nearest-centroid classifier is suitable for k-means, which indeed clusters points by assigning them to the nearest centroid (Lloyd, 1982). For suitable classifiers for other clustering algorithms see Akhanli and Hennig (2020).

For (Stab) (stability of cluster membership), the clustering method needs to be applied again to D_2. We check whether the cluster memberships resulting from applying the method to the validation data are similar to the cluster memberships resulting from transferring the original clustering to the validation data. This combines (a) and (b).

3.4 | Overview of validation strategies

Table 2 combines the concepts of Sections 3.2 and 3.3 into an overview of strategies for validation on validation data. The precise choice of indices, plots, and so forth depends on the specific context of the analysis. We refer to Section 4 for illustrative examples from the applied literature.

Here are some considerations regarding the different strategies. The commented R code in the Supporting Information illustrates the following paragraphs with real-world datasets and concrete choices for indices and visualization tools.

3.4.1 | Validating (Int): Internal properties of the clustering

When applying result-based validation, the clusters of C_2^{tf} correspond to those of C_1. This makes the comparison easier. For method-based validation, the clusters of C_2^{md} are not automatically associated one-to-one with the clusters of C_1. Such an association is not needed when calculating internal indices that refer to a whole clustering, and comparing the index values between the clusterings on discovery and validation data. However, one may also be interested in comparing characteristics of specific clusters such as cluster centroids. In this case, there needs to be a matching of the clusters of C_2^{md} to the clusters of C_1, usually assuming that their number is the same. There are various methods to do this. For

TABLE 2 Strategies for validation on validation data

	Method-based validation	Result-based validation
	Compare C_1, C_2^{md} with respect to:	Compare C_1, C_2^{tf} with respect to:
(Int)	Internal properties	Internal properties
(Ext)	External associations	External associations
(Vis)	Visual properties	Visual properties
(Stab)	Compare C_2^{md}, C_2^{tf} with respect to cluster membership	

example, in centroid-based clustering one could match the centroids so that the sum of distances between centroids of matched clusters is minimal (Mirkin, 2005). Breckenridge (2000) suggests associating each cluster of C_2^{md} to a cluster of C_2^{tf} (e.g., by choosing the cluster association that maximizes the sum of the intersections of the clusters). The one-to-one cluster association of C_2^{tf} to C_1 can then be used to assign each cluster of C_2^{md} to one of C_1.

3.4.2 | Validating (Ext): Associations with external variables or agreement with externally known partitions

As for method-based validation of internal properties (Int), here too it may be necessary to match the clusters of C_2^{md} to those in C_1 and the remarks made above apply again. Note that this is not necessarily required. For example, testing whether the clusters are associated with an external variable, such as survival time, without interpreting the association of specific clusters, does not require matching.

For result-based validation of descriptive clustering, the partition C_2^{tf} is actually equal to C_1. This makes certain approaches such as testing an association between cluster membership and an external variable on both discovery and validation data meaningless.

3.4.3 | Validating (Vis): Visual patterns

Using the same variables for D_1 and D_2 as in inferential clustering, some plots such as scatterplots or parallel coordinates plots can visualize both C_2^{md} and C_2^{tf} in a straightforward manner comparable to C_1. Some other plots such as principal components biplots, other linear projection plots such as those in Hennig (2004), and multidimensional scaling require a selection of an optimal projection space for the dataset to be plotted. Although this could be done on the validation data, for inferential clustering, plotting the validation dataset on the projection space defined by the discovery dataset (and its clustering, if the projection space depends on it) allows for a more direct comparison. For linear projection methods, this requires a standard linear projection given the coordinate axes determined from D_1. For multidimensional scaling, there are techniques to embed new observations into the projection space defined by the original observations, for example, Gower (1968). For descriptive clustering, on the other hand, embedding the observations of D_2 in the space defined by D_1 is not informative as the points would be identical, so here an optimized projection space for D_2 must be found.

Some other plots, such as the silhouette plot and cluster heatmaps (as long as observations are ordered only by a partition rather than a full dendrogram), may benefit from matching clusters for determining their order, see the comments on internal validation (Int) in Section 3.4.1.

The results of visual validation are subjective, and although plots are reproducible given both discovery and validation datasets, the way the researcher arrives at a validity verdict will not be reproducible. Displaying the involved plots will give the reader the chance to form their own conclusions.

3.4.4 | Validating (Stab): Stability of cluster membership

Here one needs to compute both C_2^{tf} and C_2^{md}. These are then compared with an index for comparing partitions. The rationale behind this is as follows: cluster memberships in C_1 and C_2^{md} are compared to check whether repeated application of the clustering method leads to stable cluster memberships. For descriptive clustering, C_1 can be compared to C_2^{md} directly (here C_1 is equal to C_2^{tf}). For inferential clustering, C_1 and C_2^{md} cannot be compared directly because they are partitions of different sets of objects. Thus C_2^{tf} is used as a surrogate for C_1 on D_2. Different choices of a partition similarity index are possible, for example, the ARI, the Jaccard index, or the FM index (for overviews, see Meila, 2015; Albatineh et al., 2006).

3.5 | When is a clustering successfully validated?

Due to random variation, researchers will hardly ever achieve the exact same results on discovery and validation data. So far, there seem to be no systematic approaches for judging "validation success" in the context of validating clustering

results on validation data. In this section, we review the current status and outline which aspects would be interesting to study in further research.

The problem of defining "successful" validation does not only arise in cluster analysis, but generally in validation or replication studies. Here we consider "validation" to be the broader term, and "replication" as more specific, for which strategies of the validation framework can be used. "Replication" refers to using new data to re-assess scientific claims made in a previous publication (Nosek & Errington, 2020). The discussion about judging replication success is ongoing in the field of methodological research on replication studies, mostly in the context of hypothesis tests and effect estimates. For example, Hedges (2019) and Held (2020) argue that, when trying to replicate a hypothesis test (that was significant on the original data), it is not enough to check whether the test on the replication data is significant again. Actually, the binary distinction between significance and insignificance may not be helpful, for example, when comparing p values of 0.04 and 0.06 (given a significance level of 0.05). Rather, we should also check whether the effect estimate in the replication study provides evidence for the claim about the effect in the original study. Some clustering validation aspects are connected to significance tests, particularly testing for external associations in (Ext). The same caveats apply here regarding general replication of test results.

The consideration of differences between (internal or external) validity measurements on discovery and validation data, or the consideration of an index value for stability between discovery and validation sets, could in principle also be framed as a testing problem of a null hypothesis formalizing some kind of equality of structure. To our knowledge, this has not been performed yet and is left as a potential direction of future research. It can be expected that validation data results will not be quite as good due to selection bias originating from basing selection of the final clustering on results of the discovery data: the more different clustering algorithms or parameters are tried during the analysis on the discovery data, the more likely it is that one of them yields a satisfying result. If only the best result is chosen, this might be "overoptimistic" to some extent. In other words, the multiplicity of possible analysis strategies may hinder replicability (Hoffmann et al., 2021), see also the discussion in Section 5. Observing slightly worse values on the validation data is thus to be expected and does not necessarily mean that the validation has failed. However, if the results are severely worse, then this suggests problematic overoptimism on the discovery data.

As it stands, it must be acknowledged that the question "is validation successful?" cannot simply be answered with "yes" or "no". The validation dataset may deliver high or low agreement regarding various aspects (internal and external validity, stability, visual aspects) with what was found on the discovery data—where the clustering on the discovery data may already have been assessed as a weaker or stronger clustering in Step 1. For example, regarding an internal index, such as the Average Silhouette Width, it is of interest both whether the value is reasonably high on the discovery dataset alone, and whether the validation dataset supports whatever value was found on the discovery data. Guidelines or thresholds for interpreting index values are rarely given and in fact mostly arbitrary, so the researcher must rely on their understanding of the index, experience, and judgment.

4 | EXAMPLES FROM THE APPLIED LITERATURE

In this section, we review application studies that conducted cluster analysis on a discovery set and then validated the results with a validation set. Our aim is to demonstrate how these studies fit into the framework outlined above. Given the vast amount of applied cluster analysis studies, it is impossible to list every cluster study that used a validation set. Rather, we start by giving a short historical overview and then present some exemplary studies in Table 3.

The appearance of clustering studies that used a discovery and a validation set dates back to at least the 1960s. One of the first clustering studies that used a validation set was Goldstein and Linden (1969) who clustered patients with alcohol use disorder. In our terms, they performed method-based validation with respect to internal properties. Rogers and Linden (1973) provided an early implementation of stability-based validation, (Stab). They clustered college freshwomen based on personality features and used discriminant analysis as the classifier to derive C_2^{tf}. (Stab) was then presented more systematically by McIntyre and Blashfield (1980) and Breckenridge (1989).

In recent decades, many more clustering studies that use validation data have appeared. In Table 3, we list exemplary applied studies for the different validation types as outlined in Table 2. The studies are taken from our main field of expertise, that is, medicine and health science. Some of these studies used multiple aspects of the validation framework, but for the sake of illustration, we only list one validation type per study. We did not find an example for result-based validation of (Vis). In general, there appear to be few studies which performed validation of visual properties on

TABLE 3 Study examples for each validation type

Validation type and study	Clustering aim and validation motivation	Validation data	Cluster algorithm	Validation procedure
(Int) result-based: Kapp and Tibshirani (2007)	Inferential clustering of breast cancer patients based on microarray gene expression to validate breast cancer subtypes that were previously found by Sørlie et al. (2003).	Validation data consisted of independently collected samples from different countries.	hierarchical clustering	C_2^{tf} was derived via a variant of nearest-centroid classification (each sample in the validation data was assigned to the original cluster whose centroid had the maximum Pearson's correlation coefficient with the sample). C_2^{tf} was then evaluated with a newly introduced internal validation index (the "in-group proportion" IGP). This index was combined with a statistical test procedure that consists of generating centroids randomly placed in the data, classifying the samples of the validation data to these centroids to obtain clusterings $\widetilde{C}_2^{tf}$, and comparing the values of the internal index for the $\widetilde{C}_2^{tf}$'s with the index value for C_2^{tf}. The IGP was not applied to C_1.
(Int) method-based: De Bourdeaudhuij and Van Oost (1998)	Inferential clustering of adolescents to find clusters of health behavior (with respect to smoking, alcohol use, sleeping, food choice, BMI, and physical activity). Validation was used by the authors to replicate their own findings.	Validation data was a separately collected dataset.	hierarchical clustering	Clusters of C_1 and C_2^{md} were matched manually. Compared means of health behavior variables (centroids) between C_1 and C_2^{md} (e.g., the mean amount of smoking for cluster 1 was compared between C_1 and C_2^{md} and so on). Overall, the centroids were similar between both clusterings. In particular, both the most "healthy" and the most "unhealthy" cluster could be recovered from the validation data.
(Ext) result-based: Curtis et al. (2012)	Inferential clustering of breast cancer patients based on copy number and gene expression data to discover novel breast cancer subtypes. The authors chose result-based validation because in clinical practice, doctors would typically want to assign a new patient to a subtype, and the validity of such a procedure can be analyzed by classification of the validation samples to yield C_2^{tf} and then comparing C_2^{tf} to C_1.	Validation data was a second cohort from the same tumor banks.	iCluster (Shen et al., 2009)	C_2^{tf} was derived via nearest shrunken centroid classification (Tibshirani et al., 2003), which is a modification of nearest-centroid classification where the cluster centroids are shrunk towards the overall centroid. Comparison of C_1 and C_2^{tf} w.r.t. their associations with survival: a Cox proportional hazards models was fitted to the discovery data (respectively validation data), with the cluster memberships of C_1 (resp. C_2^{tf}) as covariates. The hazard ratios of the clusters were similar between the model for the discovery data and the model for the validation data.

(Continues)

TABLE 3 (Continued)

Validation type and study	Clustering aim and validation motivation	Validation data	Cluster algorithm	Validation procedure
(Ext) method-based: Freudenberg et al. (2009)	Descriptive clustering of cancer-related genes to find biologically meaningful clusters of co-expressed genes, which may help to elucidate biological pathways and generate hypotheses about transcriptional regulatory mechanisms. Authors performed validation to check the replicability of the clustering results.	Validation data was an independently collected breast cancer dataset.	hierarchical clustering combined with the CSIMM algorithm (Liu et al., 2006)	Each gene in the clustering was assigned a CLEAN score, a newly introduced external measure for agreement with previously known functional categories. The correlation between the gene CLEAN scores obtained with C_1 and C_2^{md} was calculated.
(Vis) method-based: Sweatt et al. (2019)	Inferential clustering of pulmonary arterial hypertension (PAH) patients based on blood proteomic profiles to find distinct PAH immune phenotypes. The underlying idea was that patient subgroups might express distinct patterns of inflammation in blood, and the detection of these groups may in turn help to develop tailored treatments in future studies. Validation was performed to assess whether the results generalize to other patients.	Validation data consisted of independently collected samples from a different country.	Consensus Clustering (Monti et al., 2003)	Clusters of C_1 and C_2^{md} were matched manually. Compared heatmaps and PCA plots for C_1 and C_2^{md} which were generated separately for discovery and validation data, no common projection space was used. The heatmaps and PCA plots were deemed to be similar between discovery and validation data.
(Stab): Bergström et al. (2001)	Inferential clustering of spinal pain patients based on the Multidimensional Pain Inventory, which is a battery of questionnaires where patients self-report their pain severity, pain-related interference in everyday life, etc. Previous studies had detected distinct subgroups of spinal pain patients with respect to how well the patients coped with their disease, which has implications for tailored treatment. The authors sought to find similar clusters in their data, and performed validation to assess the replicability of their own findings.	Validation data was an independently collected dataset.	k-means	C_2^{tf} was derived via nearest-centroid classification. The kappa coefficient (a partition similarity index) was used to compare C_2^{md} and C_2^{tf}. The resulting index value was 0.82, which was judged as indicating very good agreement.

a validation dataset in a thorough manner. We believe future studies would benefit from considering the procedures for (Vis), outlined above.

The studies cited in Table 3 mostly treat validation and discovery data *asymmetrically* (with the exception of Freudenberg et al., 2009, and Bergström et al., 2001). This is more obvious for result-based validation: the clustering C_1 is transferred to the validation data (and not the other way around). Method-based validation may appear more symmetric because the same method is applied to both discovery and validation data and the results are typically compared descriptively in a symmetric fashion. However, method-based validation can be asymmetric to the extent of which the validation data is kept apart from the method selection on the discovery data, and is only used later without model selection to validate the results on the discovery data. Asymmetry could be made more explicit by using a suitable test procedure to judge validation success (inspired by the methodological research on judging replication success, for example, Held (2020) advocates for an asymmetric approach when comparing the replication study to the original study), but as discussed in Section 3.5, such approaches do not seem to exist yet for cluster analysis.

Many studies in the literature do not strictly set apart the validation data during Step 1 (method selection). That is, these studies use the result of the validation on the validation data for method selection (e.g., Brennan et al., 2012; Jamison et al., 1988; Sinclair et al., 2005). In contrast, we have argued in the introduction and in Section 2.3 that for the purpose of validating a clustering result on validation data in the sense of our framework, method selection should be finished after Step 1.

Another validation variant is also frequently found in the literature (e.g., Ailawadi et al., 2001; Gruber et al., 2010; Homburg et al., 2008; Kaluza, 2000; Phinney et al., 2005): method selection is performed on the whole dataset, after which the data is split into two sets. The chosen cluster method is applied to the first set, and then validation on the second set (the validation data) is assessed. Successful "validation" may indicate a certain robustness or stability of the result, but in order to avoid overoptimism on the validation data, method selection should be constrained to the first part of the split dataset, and not be performed on the whole data according to our framework.

Other studies (e.g., Alexe et al., 2006) perform a procedure that appears similar to method-based validation: they split a dataset into two halves, use the first half as the discovery set, but obtain C_2^{md} by clustering discovery and validation data *together* (instead of only clustering the validation data), which again will likely yield more optimistic validation results than if C_2^{md} had been obtained based on the validation data only.

5 | DISCUSSION

We have presented a systematic framework for validating clusterings on a validation dataset that encompasses procedures known from the literature. This framework might help researchers to identify a suitable approach to validate their clustering results in future studies. However, the procedure cannot be performed in an "automated" manner. Rather, it requires substantial input from the researchers who must decide which validation criteria are important for them depending on the substantive context. Furthermore, specific indices and plots need to be chosen, as well as whether the amount of agreement between results on the discovery and validation datasets is assessed as sufficient. We have given hints about when some aspects may be of interest, but as every application is different, there are no clear rules. This holds for the clustering process in general: while cluster analysis is often interpreted as being able to find meaningful structure in the data "on its own", the choice of cluster concept and method requires thorough consideration by researchers (Akhanli & Hennig, 2020; Hennig, 2015b). The same is true for our validation framework.

Performing validation on the validation data adds some computational complexity to the cluster analysis. However, the overall complexity is often less than twice the complexity that would result from only analyzing the discovery data: frequently method selection is performed on the discovery data, and this possibly time-consuming process is not applied to the validation data.

Regarding the choice of validation data, a validation dataset could be obtained by splitting the original dataset, or it could be a separately collected dataset. On one hand, if the validation dataset and the discovery dataset are obtained by splitting an originally collected dataset, it is unclear whether a successful validation allows for generalization to data from other sources. Moreover, this reduces the size of the data and can make it more difficult to find meaningful cluster structure in the data. On the other hand, if the validation data have been independently collected (potentially coming from a different distribution) and the validation fails, it can be difficult to determine whether this is due to the clustering not being meaningful, or due to systematic differences between discovery and validation data. Conversely, if validation *is* successful, then this is all the more encouraging, because it suggests that the clustering result may be valid in a more general context.

Notably, the validation of clustering results on a validation dataset may also allow detection of "overoptimism" due to "overfitting" effects: when researchers try different clustering algorithms or parameters during the analysis, they can use classical internal and external validation methods to choose a single clustering out of these. However, the more clustering methods tried, the more likely it is that one of them yields a satisfying result by chance. Consequently, the reported results may be less reliable than they seem, similarly to the results of multiple tests if no adjustment is performed. While this is well-understood in the context of multiple testing, this is less so in the context of clustering. Repeating the same cluster analysis on another dataset is a sensible approach to ensure that seemingly satisfactory results are not (solely) the product of such overfitting effects.

In future work, it would be interesting to study further aspects of cluster validation in relation to validation data use. *Hypothesis testing* is an approach to cluster validation that we have not embedded in our framework. For example, one can test if a clustering result is significantly "better" than clusterings generated by the same method on homogeneous datasets (for an overview, see Huang et al., 2015). This can involve internal validation indices (Dubes, 1993; Gordon, 1998; Halkidi et al., 2002; Hennig & Lin, 2015) or stability analysis (Bertrand & Mufti, 2006; Dudoit & Fridlyand, 2002; John et al., 2020; Smith & Dubes, 1980). We do not know of work where hypothesis testing for cluster validation has involved validation data, but it could be of interest to derive distributions under suitable null hypotheses for statistics that are evaluated on validation data.

In conclusion, our hope for this framework is to improve the interpretation of clustering studies that use validation data, and to stimulate the use of validation sets in cluster analysis.

ACKNOWLEDGMENTS

We thank Anna Jacob and Alethea Charlton for making valuable language corrections. This work has been partially supported by the German Federal Ministry of Education and Research (BMBF) [grant number 01IS18036A to Anne-Laure Boulesteix (Munich Center of Machine Learning)] and the German Research Foundation [grant number BO3139/7-1 to Anne-Laure Boulesteix]. The authors of this work take full responsibility for its content.

CONFLICT OF INTEREST

The authors have declared no conflicts of interest for this article.

AUTHOR CONTRIBUTIONS

Theresa Ullmann: Conceptualization (equal); methodology (lead); writing – original draft (lead); writing – review and editing (equal). **Christian Hennig:** Conceptualization (supporting); methodology (supporting); supervision (supporting); writing – original draft (supporting); writing – review and editing (equal). **Anne-Laure Boulesteix:** Conceptualization (equal); funding acquisition (lead); methodology (supporting); supervision (lead); writing – original draft (supporting); writing – review and editing (equal).

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Theresa Ullmann https://orcid.org/0000-0003-1215-8561
Christian Hennig https://orcid.org/0000-0003-1550-5637
Anne-Laure Boulesteix https://orcid.org/0000-0002-2729-0947

RELATED WIREs ARTICLE

Method evaluation, parameterization, and result validation in unsupervised data mining: A critical survey

REFERENCES

Ailawadi, K. L., Neslin, S. A., & Gedenk, K. (2001). Pursuing the value-conscious consumer: Store brands versus national brand promotions. *Journal of Marketing*, *65*(1), 71–89.

Akhanli, S. E., & Hennig, C. (2020). Comparing clusterings and numbers of clusters by aggregation of calibrated clustering validity indexes. *Statistics and Computing*, *30*(5), 1523–1544.

Albatineh, A. N., Niewiadomska-Bugaj, M., & Mihalko, D. (2006). On similarity indices and correction for chance agreement. *Journal of Classification*, *23*(2), 301–313.

Alexe, G., Dalgin, G. S., Ramaswamy, R., DeLisi, C., & Bhanot, G. (2006). Data perturbation independent diagnosis and validation of breast cancer subtypes using clustering and patterns. *Cancer Informatics*, *2*, 243–227.

Begley, C. G., & Ellis, L. M. (2012). Raise standards for preclinical cancer research. *Nature*, *483*(7391), 531–533.

Ben-David, S., Von Luxburg, U., & Pál, D. (2006). A sober look at clustering stability. In *International conference on computational learning theory* (pp. 5–19). Springer.

Ben-Hur, A., Elisseeff, A., & Guyon, I. (2002). A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing*, *7*, 6–17.

Bergström, G., Bodin, L., Jensen, I. B., Linton, S. J., & Nygren, A. L. (2001). Long-term, non-specific spinal pain: Reliable and valid subgroups of patients. *Behaviour Research and Therapy*, *39*(1), 75–87.

Bertrand, P., & Mufti, G. B. (2006). Loevinger's measures of rule quality for assessing cluster stability. *Computational Statistics & Data Analysis*, *50*(4), 992–1015.

Boulesteix, A.-L., & Hatz, M. (2017). Benchmarking for clustering methods based on real data: A statistical view. In F. Palumbo, A. Montanari, & M. Vichi (Eds.), *Data science—Innovative developments in data analysis and clustering* (pp. 73–82). Springer.

Boulesteix, A.-L., Strobl, C., Augustin, T., & Daumer, M. (2008). Evaluating microarray-based classifiers: An overview. *Cancer Informatics*, *6*, 77–97.

Breckenridge, J. N. (1989). Replicating cluster analysis: Method, consistency, and validity. *Multivariate Behavioral Research*, *24*(2), 147–161.

Breckenridge, J. N. (2000). Validating cluster analysis: Consistent replication and symmetry. *Multivariate Behavioral Research*, *35*(2), 261–285.

Brennan, T., Breitenbach, M., Dieterich, W., Salisbury, E. J., & Van Voorhis, P. (2012). Women's pathways to serious and habitual crime: A person-centered analysis incorporating gender responsive factors. *Criminal Justice and Behavior*, *39*(11), 1481–1508.

Burstein, M. D., Tsimelzon, A., Poage, G. M., Covington, K. R., Contreras, A., Fuqua, S. A., Savage, M. I., Osborne, C. K., Hilsenbeck, S. G., Chang, J. C., Mills, G. B., Lau, C. C., & Brown, P. H. (2015). Comprehensive genomic analysis identifies novel subtypes and targets of triple-negative breast cancer. *Clinical Cancer Research*, *21*(7), 1688–1698.

Caliński, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, *3*(1), 1–27.

Camerer, C. F., Dreber, A., Forsell, E., Ho, T.-H., Huber, J., Johannesson, M., Kirchler, M., Almenberg, J., Altmejd, A., Chan, T., Heikensten, E., Holzmeister, F., Imai, T., Isaksson, S., Nave, G., Pfeiffer, T., Razen, M., & Wu, H. (2016). Evaluating replicability of laboratory experiments in economics. *Science*, *351*(6280), 1433–1436.

Cook, D., & Swayne, D. F. (2007). *Interactive and dynamic graphics for data analysis with R and GGobi*. Springer.

Curtis, C., Shah, S. P., Chin, S.-F., Turashvili, G., Rueda, O. M., Dunning, M. J., Speed, D., Lynch, A. G., Samarajiwa, S., & Yuan, Y. (2012). The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, *486*(7403), 346–352.

Dangl, R., & Leisch, F. (2020). Effects of resampling in determining the number of clusters in a data set. *Journal of Classification*, *37*, 558–583.

De Bourdeaudhuij, I., & Van Oost, P. (1998). Family characteristics and health behaviours of adolescents and families. *Psychology and Health*, *13*(5), 785–803.

Ding, Y., Chowdhury, G. G., & Foo, S. (2001). Bibliometric cartography of information retrieval research by using co-word analysis. *Information Processing & Management*, *37*(6), 817–842.

Dolnicar, S., & Leisch, F. (2010). Evaluation of structure and reproducibility of cluster solutions using the bootstrap. *Marketing Letters*, *21*(1), 83–101.

Dougherty, E. R., Hua, J., & Bittner, M. L. (2007). Validation of computational methods in genomics. *Current Genomics*, *8*(1), 1–19.

Dubes, R. C. (1993). Cluster analysis and related issues. In C. H. Chen, L. F. Pau, & P. S. P. Wang (Eds.), *Handbook of pattern recognition and computer vision* (pp. 3–32). World Scientific Publishing Company.

Dudoit, S., & Fridlyand, J. (2002). A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology*, *3*(7), research0036.1–0036.21.

Fang, Y., & Wang, J. (2012). Selection of the number of clusters via the bootstrap method. *Computational Statistics & Data Analysis*, *56*(3), 468–477.

Färber, I., Günnemann, S., Kriegel, H.-P., Kröger, P., Müller, E., Schubert, E., Seidl, T., & Zimek, A. (2010). On using class-labels in evaluation of clusterings. In *MultiClust: 1st international workshop on discovering, summarizing and using multiple clusterings held in conjunction with KDD*, Washington, DC.

Fowlkes, E. B., & Mallows, C. L. (1983). A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, *78*(383), 553–569.

Freudenberg, J. M., Joshi, V. K., Hu, Z., & Medvedovic, M. (2009). Clean: Clustering enrichment analysis. *BMC Bioinformatics*, *10*(1), 234.

Fu, W., & Perry, P. O. (2020). Estimating the number of clusters using cross-validation. *Journal of Computational and Graphical Statistics*, *29*(1), 162–173.

Garrido-Castro, A. C., Lin, N. U., & Polyak, K. (2019). Insights into molecular classifications of triple-negative breast cancer: Improving patient selection for treatment. *Cancer Discovery*, *9*(2), 176–198.

Goldstein, S. G., & Linden, J. D. (1969). Multivariate classification of alcoholics by means of the MMPI. *Journal of Abnormal Psychology*, *74*(6), 661–669.

Gordon, A. D. (1998). Cluster Validation. In C. Hayashi, K. Yajima, H. Bock, N. Ohsumi, Y. Tanaka, & Y. Baba (Eds.), *Data science, classification, and related methods. Proceedings of the fifth conference of the International Federation of Classification Societies (IFCS-96)* (pp. 22–39). Springer.

Gower, J. C. (1968). Adding a point to vector diagrams in multivariate analysis. *Biometrika*, *55*, 582–585.

Gruber, M., Heinemann, F., Brettel, M., & Hungeling, S. (2010). Configurations of resources and capabilities and their performance implications: An exploratory study on technology ventures. *Strategic Management Journal*, *31*(12), 1337–1356.

Hahsler, M., & Hornik, K. (2011). Dissimilarity plots: A visual exploration tool for partitional clustering. *Journal of Computational and Graphical Statistics*, *20*(2), 335–354.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2002). Cluster validity methods: Part I. *ACM SIGMOD Record*, *31*(2), 40–45.

Halkidi, M., Vazirgiannis, M., & Hennig, C. (2015). Method-independent indices for cluster validation and estimating the number of clusters. In C. Hennig, M. Meila, F. Murtagh, & R. Rocci (Eds.), *Handbook of cluster analysis* (pp. 616–639). Chapman and Hall/CRC.

Handl, J., Knowles, J., & Kell, D. B. (2005). Computational cluster validation in post-genomic data analysis. *Bioinformatics*, *21*(15), 3201–3212.

Hedges, L. V. (2019). The statistics of replication. *Methodology*, *15*, 3–14.

Held, L. (2020). A new standard for the analysis and design of replication studies. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *183*(2), 431–448.

Hennig, C. (2004). Asymmetric linear dimension reduction for classification. *Journal of Computational and Graphical Statistics*, *13*, 930–945.

Hennig, C. (2007). Cluster-wise assessment of cluster stability. *Computational Statistics & Data Analysis*, *52*(1), 258–271.

Hennig, C. (2015a). Clustering strategy and method selection. In C. Hennig, M. Meila, F. Murtagh, & R. Rocci (Eds.), *Handbook of cluster analysis* (pp. 703–730). Chapman & Hall/CRC.

Hennig, C. (2015b). What are the true clusters? *Pattern Recognition Letters*, *64*, 53–62.

Hennig, C., & Lin, C.-J. (2015). Flexible parametric bootstrap for testing homogeneity against clustering and assessing the number of clusters. *Statistics and Computing*, *25*(4), 821–833.

Hoffmann, S., Schönbrodt, F., Elsas, R., Wilson, R., Strasser, U., & Boulesteix, A.-L. (2021). The multiplicity of analysis strategies jeopardizes replicability: Lessons learned across disciplines. *Royal Society Open Science*, *8*(4), 201925.

Homburg, C., Jensen, O., & Krohmer, H. (2008). Configurations of marketing and sales: A taxonomy. *Journal of Marketing*, *72*(2), 133–154.

Huang, H., Liu, Y., Hayes, D. N., Nobel, A., Marron, J. S., & Hennig, C. (2015). Significance testing in clustering. In C. Hennig, M. Meila, F. Murtagh, & R. Rocci (Eds.), *Handbook of cluster analysis* (pp. 336–357). Chapman and Hall/CRC.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*(1), 193–218.

Hutson, M. (2018). Artificial intelligence faces reproducibility crisis. *Science*, *359*(6377), 725–726.

Jaccard, P. (1908). Nouvelles recherches sur la distribution florale. *Bulletin de la Societe Vaudoise des Sciences Naturelles*, *44*, 223–270.

Jain, A. K., & Moreau, J. V. (1987). Bootstrap technique in cluster analysis. *Pattern Recognition*, *20*(5), 547–568.

Jamison, R. N., Rock, D. L., & Parris, W. C. (1988). Empirically derived symptom checklist 90 subgroups of chronic pain patients: A cluster analysis. *Journal of Behavioral Medicine*, *11*(2), 147–158.

John, C. R., Watson, D., Russ, D., Goldmann, K., Ehrenstein, M., Pitzalis, C., Lewis, M., & Barnes, M. (2020). M3C: Monte Carlo reference-based consensus clustering. *Scientific Reports*, *10*(1), 1–14.

Kaluza, G. (2000). Changing unbalanced coping profiles-a prospective controlled intervention trial in worksite health promotion. *Psychology and Health*, *15*(3), 423–433.

Kapp, A. V., Jeffrey, S. S., Langerød, A., Børresen-Dale, A.-L., Han, W., Noh, D.-Y., Bukholm, I. R., Nicolau, M., Brown, P. O., & Tibshirani, R. (2006). Discovery and validation of breast cancer subtypes. *BMC Genomics*, *7*(1), 231.

Kapp, A. V., & Tibshirani, R. (2007). Are clusters found in one dataset present in another dataset? *Biostatistics*, *8*(1), 9–31.

Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: An introduction to cluster analysis*. John Wiley & Sons.

Lange, T., Roth, V., Braun, M. L., & Buhmann, J. M. (2004). Stability-based validation of clustering solutions. *Neural Computation*, *16*(6), 1299–1323.

Lehmann, B. D., Bauer, J. A., Chen, X., Sanders, M. E., Chakravarthy, A. B., Shyr, Y., & Pietenpol, J. A. (2011). Identification of human triple-negative breast cancer subtypes and preclinical models for selection of targeted therapies. *The Journal of Clinical Investigation*, *121*(7), 2750–2767.

Leisch, F. (2008). Visualizing cluster analysis and finite mixture models. In C.-H. Chen, W. Härdle, & A. Unwin (Eds.), *Handbook of data visualization* (pp. 561–587). Springer.

Levine, E., & Domany, E. (2001). Resampling method for unsupervised estimation of cluster validity. *Neural Computation*, *13*(11), 2573–2593.

Liu, X., Sivaganesan, S., Yeung, K. Y., Guo, J., Bumgarner, R. E., & Medvedovic, M. (2006). Context-specific infinite mixtures for clustering gene expression profiles across diverse microarray dataset. *Bioinformatics*, *22*(14), 1737–1744.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, *28*(2), 129–137.

McIntyre, R. M., & Blashfield, R. K. (1980). A nearest-centroid technique for evaluating the minimum-variance clustering procedure. *Multivariate Behavioral Research*, *15*(2), 225–238.

Meila, M. (2015). Criteria for comparing Clusterings. In C. Hennig, M. Meila, F. Murtagh, & R. Rocci (Eds.), *Handbook of cluster analysis* (pp. 640–657). Chapman and Hall/CRC.

Milligan, G. W., & Cooper, M. C. (1987). Methodology review: Clustering methods. *Applied Psychological Measurement*, *11*(4), 329–354.

Mirkin, B. (2005). *Clustering for data mining: A data recovery approach*. CRC Press.

Monti, S., Tamayo, P., Mesirov, J., & Golub, T. (2003). Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, *52*(1), 91–118.

Morey, L. C., Blashfield, R. K., & Skinner, H. A. (1983). A comparison of cluster analysis techniques within a sequential validation framework. *Multivariate Behavioral Research*, *18*(3), 309–329.

Müller, H., & Hamm, U. (2014). Stability of market segmentation with cluster analysis–a methodological approach. *Food Quality and Preference*, *34*, 70–78.

Nosek, B. A., & Errington, T. M. (2020). What is replication? *PLoS Biology*, *18*(3), e3000691.

Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science*, *349*(6251), aac4716.

Phinney, J. S., Dennis, J. M., & Gutierrez, D. M. (2005). College orientation profiles of Latino students from low socioeconomic backgrounds: A cluster analytic approach. *Hispanic Journal of Behavioral Sciences*, *27*(4), 387–408.

Prat, A., Pineda, E., Adamo, B., Galván, P., Fernández, A., Gaba, L., Dez, M., Viladot, M., Arance, A., & Muñoz, M. (2015). Clinical implications of the intrinsic molecular subtypes of breast cancer. *The Breast*, *24*, S26–S35.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, *66*(336), 846–850.

Rogers, G., & Linden, J. D. (1973). Use of multiple discriminant function analysis in the evaluation of three multivariate grouping techniques. *Educational and Psychological Measurement*, *33*(4), 787–802.

Shen, R., Olshen, A. B., & Ladanyi, M. (2009). Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics*, *25*(22), 2906–2912.

Simon, R., Radmacher, M. D., Dobbin, K., & McShane, L. M. (2003). Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. *Journal of the National Cancer Institute*, *95*(1), 14–18.

Sinclair, R. R., Tucker, J. S., Cullen, J. C., & Wright, C. (2005). Performance differences among four organizational commitment profiles. *Journal of Applied Psychology*, *90*(6), 1280.

Smith, S. P., & Dubes, R. (1980). Stability of a hierarchical clustering. *Pattern Recognition*, *12*(3), 177–187.

Sørlie, T., Tibshirani, R., Parker, J., Hastie, T., Marron, J. S., Nobel, A., Deng, S., Johnsen, H., Pesich, R., & Geisler, S. (2003). Repeated observation of breast tumor subtypes in independent gene expression data sets. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(14), 8418–8423.

Sotiriou, C., Neo, S.-Y., McShane, L. M., Korn, E. L., Long, P. M., Jazaeri, A., Martiat, P., Fox, S. B., Harris, A. L., & Liu, E. T. (2003). Breast cancer classification and prognosis based on gene expression profiles from a population-based study. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(18), 10393–10398.

Sweatt, A. J., Hedlin, H. K., Balasubramanian, V., Hsi, A., Blum, L. K., Robinson, W. H., Haddad, F., Hickey, P. M., Condliffe, R., & Lawrie, A. (2019). Discovery of distinct immune phenotypes using machine learning in pulmonary arterial hypertension. *Circulation Research*, *124*(6), 904–919.

Tibshirani, R., Hastie, T., Narasimhan, B., & Chu, G. (2003). Class prediction by nearest shrunken centroids, with applications to DNA microarrays. *Statistical Science*, *18*(1), 104–117.

Tibshirani, R., & Walther, G. (2005). Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, *14*(3), 511–528.

Tyler, D. E., Critchley, F., Dümbgen, L., & Oja, H. (2009). Invariant co-ordinate selection (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *71*, 549–592.

Van Mechelen, I., Boulesteix, A.-L., Dangl, R., Dean, N., Guyon, I., Hennig, C., Leisch, F. & Steinley, D. (2018). Benchmarking in cluster analysis: A white paper. arXiv preprint arXiv:1809.10496.

Von Luxburg, U. (2010). *Clustering stability: An overview*. Now Publishers Inc.

Von Luxburg, U., Williamson, R.C., & Guyon, I. (2012). Clustering: Science or art? In Guyon, I., Dror, G., Lemaire, V., and Taylor, G., editors, Proceedings of ICML Workshop on Unsupervised and Transfer Learning, *volume 27 of* Proceedings of Machine Learning Research, pages 65–79, . PML Research Press.

Wang, J. (2010). Consistent selection of the number of clusters via crossvalidation. *Biometrika*, *97*(4), 893–904.

Wilkinson, L., & Friendly, M. (2009). The history of the cluster heat map. *The American Statistician*, *63*(2), 179–184.

Yang, Y., Han, L., Yuan, Y., Li, J., Hei, N., & Liang, H. (2014). Gene co-expression network analysis reveals common system-level properties of prognostic genes across cancer types. *Nature Communications*, *5*(1), 1–9.

Zhang, Q., Burdette, J. E., & Wang, J.-P. (2014). Integrative network analysis of TCGA data for ovarian cancer. *BMC Systems Biology*, *8*(1), 1–18.

Zimmermann, A. (2020). Method evaluation, parameterization, and result validation in unsupervised data mining: A critical survey. *WIREs Data Mining and Knowledge Discovery*, *10*(2), e1330.

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

How to cite this article: Ullmann, T., Hennig, C., & Boulesteix, A.-L. (2022). Validation of cluster analysis results on validation data: A systematic framework. *WIREs Data Mining and Knowledge Discovery*, *12*(3), e1444. https://doi.org/10.1002/widm.1444

Illustrative analysis for the validation of clustering results on validation data

Theresa Ullmann, Christian Hennig, Anne-Laure Boulesteix

Contents

1 Background

We present two illustrative examples for cluster validation on a validation dataset, accompanying the paper "Validation of cluster analysis results on validation data: A systematic framework" (2021) by Theresa Ullmann, Christian Hennig and Anne-Laure Boulesteix. This illustration is intended for readers of our paper who are familiar with R, but are new to cluster validation on validation datasets.

In the first example we consider inferential clustering of the Wisconsin Breast Cancer dataset (Street, Wolberg, and Mangasarian 1993). The second example presents descriptive clustering of a microbiome dataset obtained from human intestinal tracts (Lahti et al. 2014). For both examples, we split the data into discovery and validation sets. On the discovery data, we perform method selection and evaluate the resulting clustering with respect to different validation criteria. We then check whether the results are replicated on the validation dataset. Thus we demonstrate how the strategies given in Table 2 of the paper can be applied to both example datasets.

Our analyses are written in tutorial style and only serve illustrative purposes. We try to keep the analyses as simple as possible. For more refined studies of the example datasets, see the original sources cited above.

While the analyses for the Winsconsin Breast Cancer and the microbiome data already involve visual validation, we also use a simple toy example to further illustrate visualisation tools in the last section of this

document. We perform inferential clustering of the Iris dataset (Anderson 1935) and generate various plots to compare the clusterings on discovery and validation data.

2 Inferential clustering of the Wisconsin Breast Cancer dataset

The Wisconsin Breast Cancer dataset is a popular and publicly available dataset that is often used for performance evaluation of classifiers. Here we will use it for inferential clustering of patients.

To download the dataset from the UCI Machine Learning Repository, use the following link:

https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.data

Save the file "wdbc.data" in your working directory. Then read in the data:

```
data = read.csv("wdbc.data", header = FALSE)

str(data)
```

```
## 'data.frame':    569 obs. of  32 variables:
##  $ V1 : int  842302 842517 84300903 84348301 84358402 843786 844359 84458202 844981 84501001 ...
##  $ V2 : chr  "M" "M" "M" "M" ...
##  $ V3 : num  18 20.6 19.7 11.4 20.3 ...
##  $ V4 : num  10.4 17.8 21.2 20.4 14.3 ...
##  $ V5 : num  122.8 132.9 130 77.6 135.1 ...
##  $ V6 : num  1001 1326 1203 386 1297 ...
##  $ V7 : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
##  $ V8 : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
##  $ V9 : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
##  $ V10: num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
##  $ V11: num  0.242 0.181 0.207 0.26 0.181 ...
##  $ V12: num  0.0787 0.0567 0.06 0.0974 0.0588 ...
##  $ V13: num  1.095 0.543 0.746 0.496 0.757 ...
##  $ V14: num  0.905 0.734 0.787 1.156 0.781 ...
##  $ V15: num  8.59 3.4 4.58 3.44 5.44 ...
##  $ V16: num  153.4 74.1 94 27.2 94.4 ...
##  $ V17: num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
##  $ V18: num  0.049 0.0131 0.0401 0.0746 0.0246 ...
##  $ V19: num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
##  $ V20: num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
##  $ V21: num  0.03 0.0139 0.0225 0.0596 0.0176 ...
##  $ V22: num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
##  $ V23: num  25.4 25 23.6 14.9 22.5 ...
##  $ V24: num  17.3 23.4 25.5 26.5 16.7 ...
##  $ V25: num  184.6 158.8 152.5 98.9 152.2 ...
##  $ V26: num  2019 1956 1709 568 1575 ...
##  $ V27: num  0.162 0.124 0.144 0.21 0.137 ...
##  $ V28: num  0.666 0.187 0.424 0.866 0.205 ...
##  $ V29: num  0.712 0.242 0.45 0.687 0.4 ...
##  $ V30: num  0.265 0.186 0.243 0.258 0.163 ...
##  $ V31: num  0.46 0.275 0.361 0.664 0.236 ...
##  $ V32: num  0.1189 0.089 0.0876 0.173 0.0768 ...
```

```
data[1:10,1:10]
```

```
##          V1 V2    V3    V4     V5     V6      V7      V8      V9     V10
## 1    842302  M 17.99 10.38 122.80 1001.0 0.11840 0.27760 0.30010 0.14710
```

```
## 2    842517  M 20.57 17.77 132.90 1326.0 0.08474 0.07864 0.08690 0.07017
## 3  84300903  M 19.69 21.25 130.00 1203.0 0.10960 0.15990 0.19740 0.12790
## 4  84348301  M 11.42 20.38  77.58  386.1 0.14250 0.28390 0.24140 0.10520
## 5  84358402  M 20.29 14.34 135.10 1297.0 0.10030 0.13280 0.19800 0.10430
## 6    843786  M 12.45 15.70  82.57  477.1 0.12780 0.17000 0.15780 0.08089
## 7    844359  M 18.25 19.98 119.60 1040.0 0.09463 0.10900 0.11270 0.07400
## 8  84458202  M 13.71 20.83  90.20  577.9 0.11890 0.16450 0.09366 0.05985
## 9    844981  M 13.00 21.82  87.50  519.8 0.12730 0.19320 0.18590 0.09353
## 10 84501001  M 12.46 24.04  83.97  475.9 0.11860 0.23960 0.22730 0.08543
```

The data consists of 569 samples (patients). Each patient underwent a breast mass biopsy. The following variables are given: 1) the ID number of each patient, 2) the class label information, namely the clinical diagnosis for the breast mass (B for benign, M for malignant), and 3) 30 real-valued featured that describe different characteristics of the cells that were obtained during the biopsy. For more details, see the description of the dataset on the UCI Machine Learning Repository.

We put the class labels into a separate vector, because we will not use this information for the clustering itself, and only use it at a later point for external validation. Then we remove the patient ID numbers, which we do not need for our analysis.

```
labels = data[,2]
data = data[,-c(1,2)]
```

The data now looks as follows:

```
data[1:10, 1:10]
```

```
##       V3    V4     V5     V6      V7      V8      V9     V10    V11     V12
## 1  17.99 10.38 122.80 1001.0 0.11840 0.27760 0.30010 0.14710 0.2419 0.07871
## 2  20.57 17.77 132.90 1326.0 0.08474 0.07864 0.08690 0.07017 0.1812 0.05667
## 3  19.69 21.25 130.00 1203.0 0.10960 0.15990 0.19740 0.12790 0.2069 0.05999
## 4  11.42 20.38  77.58  386.1 0.14250 0.28390 0.24140 0.10520 0.2597 0.09744
## 5  20.29 14.34 135.10 1297.0 0.10030 0.13280 0.19800 0.10430 0.1809 0.05883
## 6  12.45 15.70  82.57  477.1 0.12780 0.17000 0.15780 0.08089 0.2087 0.07613
## 7  18.25 19.98 119.60 1040.0 0.09463 0.10900 0.11270 0.07400 0.1794 0.05742
## 8  13.71 20.83  90.20  577.9 0.11890 0.16450 0.09366 0.05985 0.2196 0.07451
## 9  13.00 21.82  87.50  519.8 0.12730 0.19320 0.18590 0.09353 0.2350 0.07389
## 10 12.46 24.04  83.97  475.9 0.11860 0.23960 0.22730 0.08543 0.2030 0.08243
```

The labels are stored separately:

```
labels[1:10]
```

```
##  [1] "M" "M" "M" "M" "M" "M" "M" "M" "M" "M"
```

```
table(labels)
```

```
## labels
##   B   M
## 357 212
```

Next, we split the data randomly into discovery and validation sets. Because we want to perform inferential clustering, we split along the samples. Setting the random seed is important for reproducibility of the results. Alternatively, we could shuffle the whole dataset and choose the first n/2 samples as discovery set and the remaining samples as the validation set. (The shuffling would be important for this alternative because otherwise, a pre-existing ordering of the samples might cause artifical differences between discovery and validation sets.)

```r
set.seed(123)
n = 0.5 * nrow(data)
discov_samples = sample(nrow(data), size = n)
valid_samples = sample(setdiff(1:nrow(data), discov_samples), n)

discov_data = data[discov_samples,]
valid_data = data[valid_samples,]
```

As preprocessing step, we standardize all features to zero mean and unit variance, such that they are on the same scale in Euclidean space (which is recommended for the cluster algorithms we will use later). Note that we do this separately for discovery and validation data, to keep the information from the validation set apart from the analysis on the discovery data. If we had standardized the features on the full dataset and only split the data later, we would have obtained slightly different feature values. This is a first hint that preprocessing must be carefully combined with the data splitting. We will discuss this issue in more detail for the descriptive clustering of the microbiome data below.

```r
discov_data = as.data.frame(scale(discov_data))
valid_data = as.data.frame(scale(valid_data))
```

2.1 Method selection and internal validation

Now we are ready to cluster the patients in the discovery data. First, we must decide which method to use. At the start of the analysis, researchers are often unsure which algorithm to use, and which number of clusters to choose. We try out both k-means and spectral clustering (for the latter, we load the `kernlab` package), and consider numbers of clusters between 2 to 10. As selection criterion, we use an internal validation index, namely the ASW, i.e., the Average Silhouette Width (Kaufman and Rousseeuw 2009), as calculated by the `cluster` package. That is, we will select the combination of clustering algorithm and number of clusters that has the highest ASW.

The ASW is calculated as $\frac{1}{n}\sum_{i=1}^{n} s(i)$, where $s(i)$ denotes the individual silhouette value of sample i (here: patient i). Each silhouette value ranges between -1 and 1. The higher the value, the more similar the sample is to its own cluster relative to its distance to the other clusters. Consequently, the higher the ASW, the better the overall clustering quality according to this criterium. For details on the calculation, see the documentation of `cluster::silhouette`.

To start with the method selection, we first generate the ASW values for all method combinations.

```r
library(cluster)
library(kernlab)
```

```r
# the vectors will store the ASW values
asw_kmeans = numeric(9)
asw_spectral = numeric(9)

# the lists will store the clustering results
cluster_kmeans = vector(mode = "list", length = 9)
cluster_spectral = vector(mode = "list", length = 9)

dist_matrix_discov = dist(discov_data, method = "euclidean")

for (k in 2:10) {
  cluster_kmeans[[k-1]] = kmeans(discov_data, centers = k)
  cluster_spectral[[k-1]] = kernlab::specc(x = as.matrix(discov_data), centers = k)

  asw_kmeans[k-1] = mean(cluster::silhouette(cluster_kmeans[[k-1]]$cluster,
                                             dist_matrix_discov)[,3])
```

```
  asw_spectral[k-1] = mean(cluster::silhouette(cluster_spectral[[k-1]]@.Data,
                                               dist_matrix_discov)[,3])

}
```

We look at the ASW values for both cluster algorithms:

```
asw_kmeans
```

```
## [1] 0.3534330 0.3310786 0.1716651 0.1717592 0.1505005 0.1315227 0.1226142
## [8] 0.1140580 0.1136161
```

```
asw_spectral
```

```
## [1] 0.35231282 0.28348073 0.24752441 0.09805734 0.08567037 0.05841300 0.07360816
## [8] 0.07250569 0.07335270
```

We see that the "best" clustering method is given by k-means clustering with k = 2 clusters (although spectral clustering with k = 2 is nearly as good). Thus we fix the resulting clustering as C_1.

```
best_k = 2
C_1 = cluster_kmeans[[1]]$cluster

table(C_1)
```

```
## C_1
##   1   2
## 199  85
```

```
asw_discov = asw_kmeans[1]
asw_discov
```

```
## [1] 0.353433
```

The ASW of approximately 0.353 indicates a moderate clustering performance.

Given that we have "optimised" the method combination to the discovery data, the ASW value might still be slightly overoptimistic (as discussed in Sections 3.5 and 5 in the paper). We thus check whether the ASW result can be replicated on the validation data. As described in the paper, we can perform either method-based or result-based validation on the validation data.

For method-based validation, to obtain the clustering C_2^{md} on the validation data, we have to apply the clustering method chosen on the discovery data to the validation data.

```
cluster_kmeans_valid = kmeans(valid_data, centers = best_k)
C_2_md = cluster_kmeans_valid$cluster
table(C_2_md)
```

```
## C_2_md
##   1   2
## 100 184
```

The ASW is then calculated for C_2^{md}:

```
dist_matrix_valid = dist(valid_data, method = "euclidean")
asw_valid_md = mean(cluster::silhouette(C_2_md, dist_matrix_valid)[,3])
asw_valid_md
```

```
## [1] 0.3429625
```

We see that the ASW for C_2^{md} is very similar to the ASW on the discovery data. So for the case of method-based validation, there does not seem to be problematic overoptimism. We next check what happens

in the case of result-based validation.

To obtain the clustering C_2^{tf} for result-based validation, we have to transfer the clustering C_1 to the validation data. Since C_1 was generated with k-means, which clusters points by assigning them to the nearest centroid, it is natural to use the nearest-centroid classifier to assign the validation data samples to the clusters of C_1. We write a function `closest.cluster` that performs this assignment.

```
centroids_discov = cluster_kmeans[[1]]$centers

closest.cluster = function(x) {
  cluster.dist = apply(centroids_discov, 1, function(y) sqrt(sum((x-y)^2)))
  return(which.min(cluster.dist)[1])
}

C_2_tf = apply(valid_data, MARGIN = 1, FUN = closest.cluster)
table(C_2_tf)
```

```
## C_2_tf
##   1   2
## 194  90
```

Now the ASW is calculated for C_2^{tf}.

```
asw_valid_tf = mean(cluster::silhouette(C_2_tf, dist_matrix_valid)[,3])

print(asw_valid_tf)
```

```
## [1] 0.3505059
```

Again, the ASW value is very similar to the value on the discovery data.

In our next steps, we will apply further validation criteria to the clustering on the discovery data and check whether the results also hold on the validation data. We do not use these criteria to perform method selection again, i.e., we use the clusterings C_1, C_2^{md} and C_2^{tf} from above. We start with visual validation and then proceed to external validation and stability analysis.

2.2 Visual validation

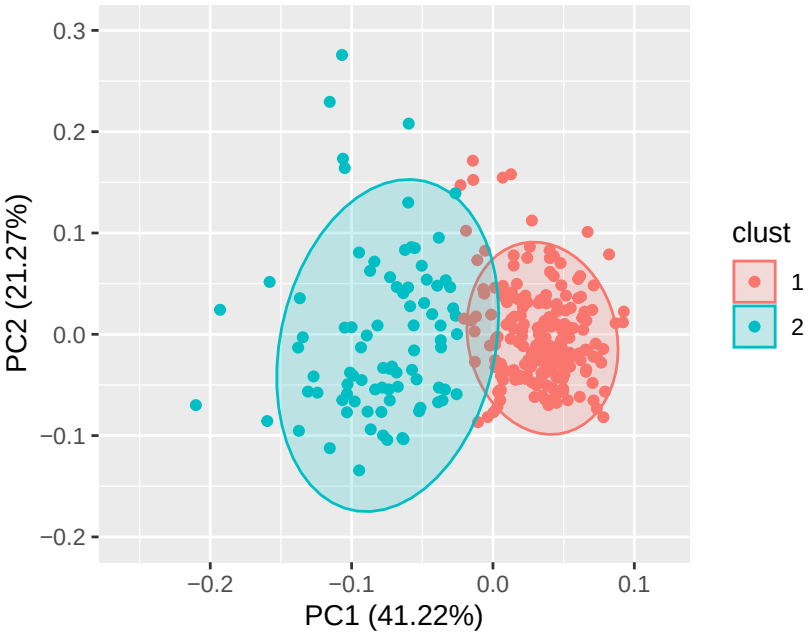
Here we generate different plots for visualising the clusterings.

2.2.1 PCA plots

First, we consider principal component analysis (PCA) plots, for which we use the `ggplot2` and `ggfortify` packages. We display the discovery data in the first two principal components, colour samples according to their cluster membership in C_1, and draw ellipses around the centers of the clusters (see `ggplot2::stat_ellipse()` for details). By setting the option `loadings = TRUE` in `autoplot` we could also turn the PCA plot into a principal components biplot which additionally shows the variable loadings.

```
library(ggplot2)
library(ggfortify)
library(gridExtra)
```

```
discov_data2 = discov_data
discov_data2$clust = as.factor(C_1)
discov.pca = prcomp(discov_data)
autoplot(discov.pca, data = discov_data2, colour = "clust",
         group_by = "clust", frame = TRUE, frame.type = "t") +
  xlim(-0.26, 0.12) + ylim(-0.2, 0.3)
```



The PCA plot shows that the two clusters are marked by higher vs. lower values on the first principal component. However, there is no clearly visible separation between the clusters, as demonstrated by the overlapping ellipses.

Now we want to compare the PCA plot for the discovery data with the PCA plot for the validation data. As noted in Section 3.4 of the paper, to allow for a more direct comparison, it makes sense to plot the validation dataset on the projection space defined by the discovery dataset (instead of calculating the PCA anew for the validation data). Therefore we project the validation data onto the PCs of `discov.pca` (via scaling and rotating).

```
valid_scale = scale(valid_data, center = discov.pca$center)
valid_projection = valid_scale %*% discov.pca$rotation
valid.pca = discov.pca
valid.pca$x = valid_projection
```

We need one more step before we can display the PCA plots for discovery and validation data next to each other. We want the colours of the clusterings to match. That is, since cluster 1 of C_1 is depicted in red in the plot above, the "corresponding" cluster of the clustering on the validation data should also be coloured in red. Therefore we need to match the clusters on the validation data to the clusters on the discovery data. As explained in Section 3.4 of the paper, the clusters of C_2^{tf} are automatically matched to the clusters of C_1 due to the transfer process, i.e., cluster 1 (resp. 2) of C_2^{tf} corresponds to cluster 1 (resp. 2) of C_1. For matching the clusters of C_2^{md} to those of C_1, we calculate the distances between the cluster centroids with the `proxy` package.

```
library(proxy)
```

```
centroids_valid = cluster_kmeans_valid$centers
rownames(centroids_valid) = c("C_2_md: clust 1", "C_2_md: clust 2")
# centroids_discov were already calculated above
rownames(centroids_discov) = c("C_1: clust 1", "C_1: clust 2")

proxy::dist(centroids_valid, centroids_discov)
```

```
##                  C_1: clust 1 C_1: clust 2
```

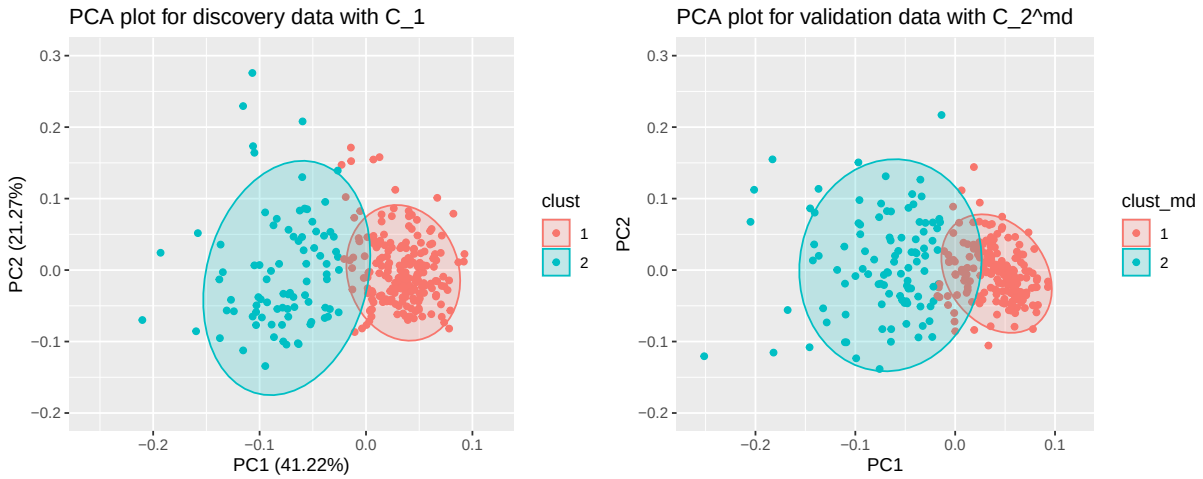
```
## C_2_md: clust 1    6.3397063    0.6951449
## C_2_md: clust 2    0.4766867    7.0257728
```

The centroid of cluster 1 (resp. 2) of C_1 is closer to the centroid of cluster 2 (resp. 1) of C_2^{md}. Therefore we rename the cluster names of C_2^{md} from (1,2) into (2,1).

```
C_2_md_renamed = C_2_md
C_2_md_renamed[C_2_md == 1] = 2
C_2_md_renamed[C_2_md == 2] = 1
```

Now we can use the renamed clustering to finally display the PCA plots for C_1 vs. C_2^{md}:

```
valid_data2 = valid_data
valid_data2$clust_md = as.factor(C_2_md_renamed)
valid_data2$clust_tf = as.factor(C_2_tf)

p1 = autoplot(discov.pca, data = discov_data2, colour = "clust",
         group_by = "clust", frame = TRUE, frame.type = "t",
         main = "PCA plot for discovery data with C_1") +
  xlim(-0.26, 0.12) + ylim(-0.2, 0.3)
p2 = autoplot(valid.pca, data = valid_data2, colour = "clust_md",
         group_by = "clust_md", frame = TRUE, frame.type = "t",
         main = "PCA plot for validation data with C_2^md") +
  xlim(-0.26, 0.12) + ylim(-0.2, 0.3) +
  labs(x = "PC1", y = "PC2")
grid.arrange(p1, p2, ncol=2)
```

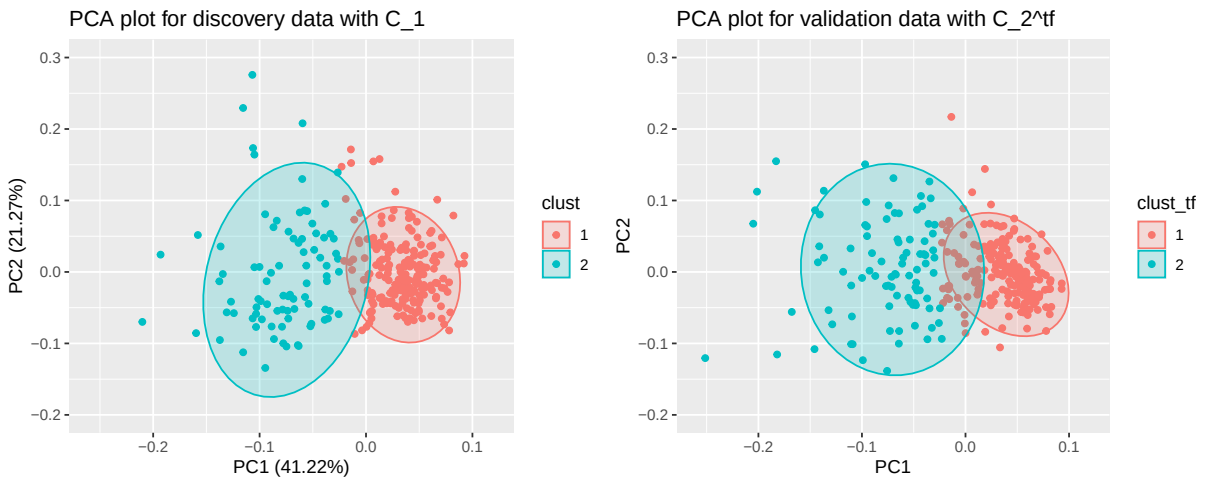


The clusterings C_1 and C_2^{md} look very similar: one larger cluster on the left, and a smaller, more compact cluster on the right, with some overlaps between both clusters.

Analogously, we display the PCA plots for the comparison of C_1 and C_2^{tf}:

```
p1 = autoplot(discov.pca, data = discov_data2, colour = "clust",
         group_by = "clust", frame = TRUE, frame.type = "t",
         main = "PCA plot for discovery data with C_1") +
  xlim(-0.26, 0.12) + ylim(-0.2, 0.3)
p2 = autoplot(valid.pca, data = valid_data2, colour = "clust_tf",
         group_by = "clust_tf", frame = TRUE, frame.type = "t",
         main = "PCA plot for validation data with C_2^tf") +
  xlim(-0.26, 0.12) + ylim(-0.2, 0.3) +
```

```
  labs(x = "PC1", y = "PC2")
grid.arrange(p1, p2, ncol=2)
```

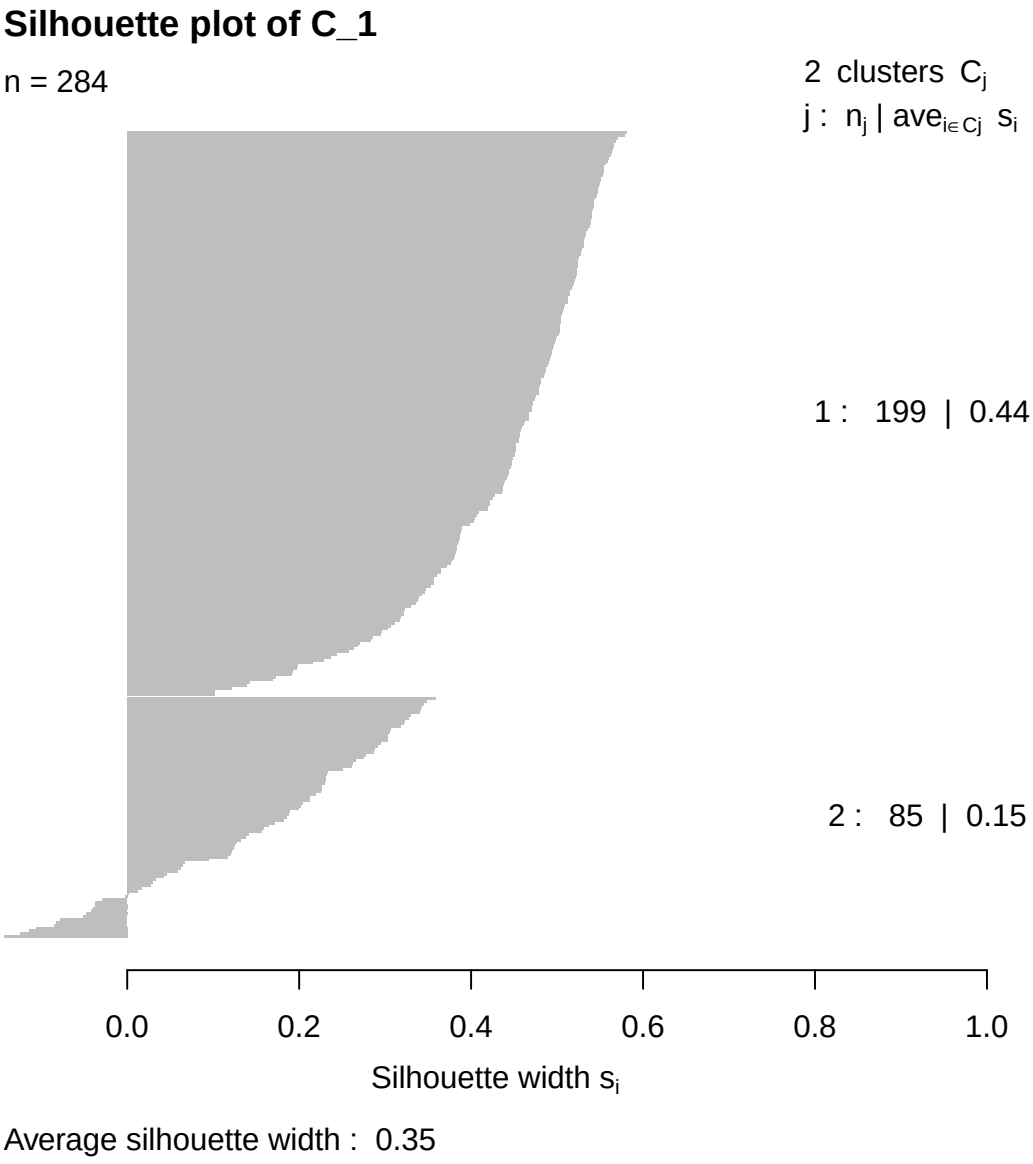


Again, we see that C_1 and C_2^{tf} look quite similar in this visual representation.

2.2.2 Silhouette plots

Next, we consider silhouette plots. Recall the Average Silhouette Width (ASW) that we used above. As described, it is calculated as the mean over all individual silhouette values $s(i)$. Instead of aggregating the values in this way, we can also display all individual values in a plot. The `cluster` package generates such plots. The samples are sorted first by their cluster membership, and then by the magnitude of their silhouette values. Here is the silhouette plot for the clustering C_1 on the discovery data.

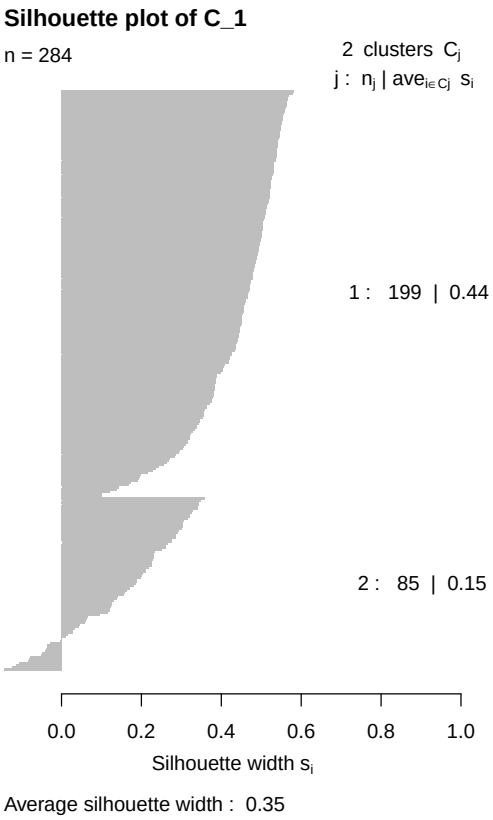
```
plot(cluster::silhouette(C_1, dist_matrix_discov), nmax = 80, cex.names = 1,
     main = "Silhouette plot of C_1")
```

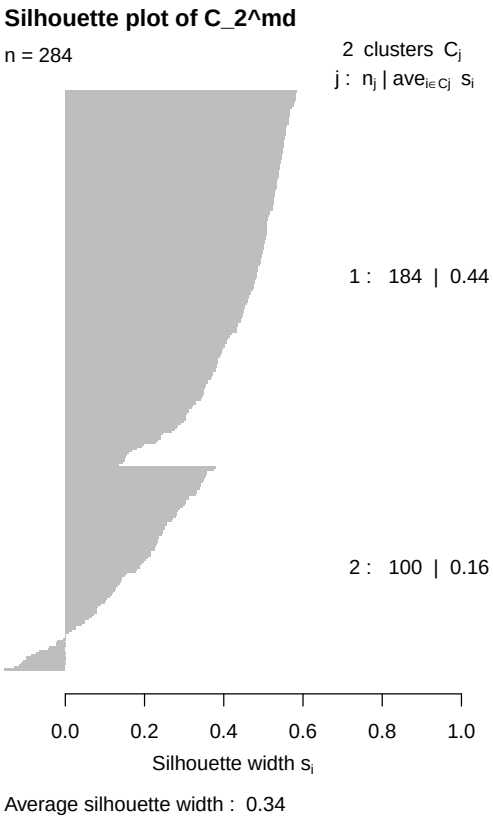


The plot gives us some information that we did not know from the ASW value alone. The first cluster has a better clusterwise ASW of 0.44, while the second cluster has a worse quality, with a clusterwise ASW of only 0.15, and several samples actually having negative silhouette values.

We now use the Silhouette plots to compare the clusterings C_2^{md} and C_2^{tf} with C_1. Because the silhouette plots show the silhouette values ordered by clusters, it is advisable to match the clusters on the validation data to the clusters on the discovery data (just as we did for the PCA plots), to ensure that the corresponding clusters are shown next to each other. We therefore use the renamed clustering `C_2_md_renamed` from above to compare the silhouette plot of C_2^{md} to the plot for C_1:

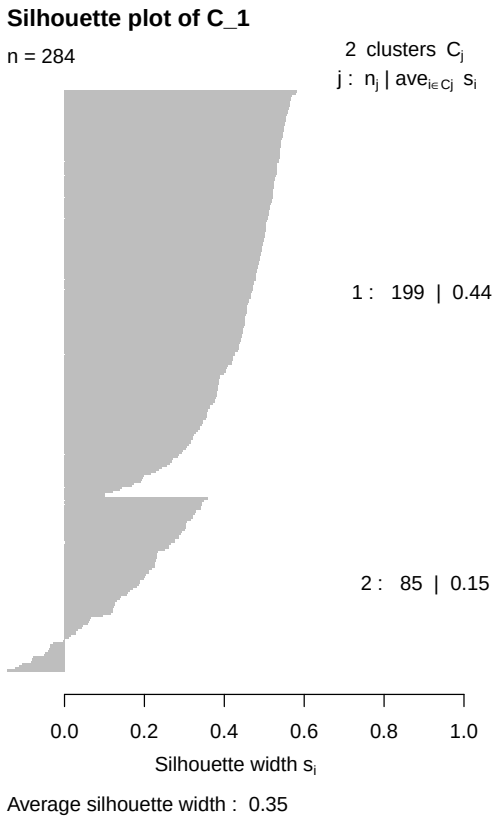
```
par(mfrow = c(1,2))
plot(cluster::silhouette(C_1, dist_matrix_discov), nmax = 80, cex.names = 1,
     main = "Silhouette plot of C_1")
plot(cluster::silhouette(C_2_md_renamed, dist_matrix_valid), nmax = 80, cex.names = 1,
     main = "Silhouette plot of C_2^md")
```

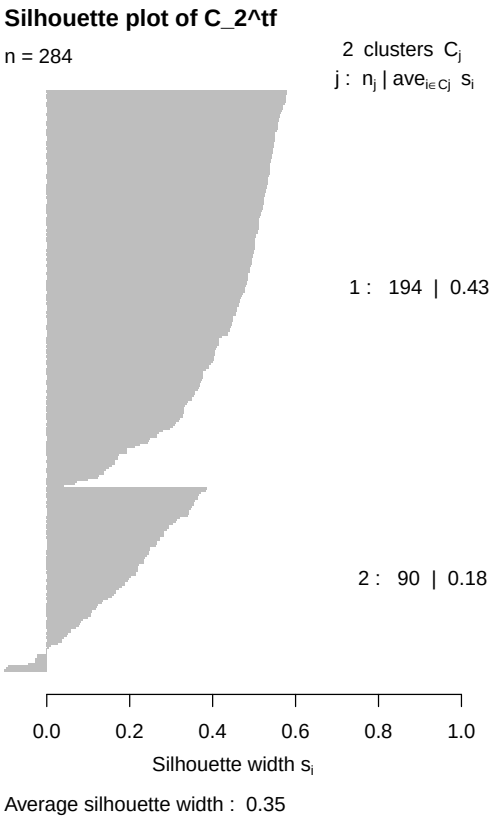




We can also compare the silhouettes of C_1 to those of C_2^{tf}:

```
par(mfrow = c(1,2))
plot(cluster::silhouette(C_1, dist_matrix_discov), nmax = 80, cex.names = 1,
     main = "Silhouette plot of C_1")
plot(cluster::silhouette(C_2_tf, dist_matrix_valid), nmax = 80, cex.names = 1,
     main = "Silhouette plot of C_2^tf")
```





In both cases, the silhouette plots for the validation data look very similar to the plot for the discovery data.

2.3 External validation

Here we use the "true" class labels (clinical diagnoses) that we have put aside in the `labels` vector before the start of the cluster analysis. The Adjusted Rand Index (ARI) (Rand 1971; Hubert and Arabie 1985) is calculated with the package `mclust` to determine the agreement of the cluster labels with the "true" labels.

```
library(mclust)

# "true" labels for the discovery data
classes_discov = as.numeric(as.factor(labels[discov_samples]))

adjustedRandIndex(C_1, classes_discov)
```

```
## [1] 0.6599149
```

The ARI value of about 0.660 shows that while the clustering on the discovery data is not perfectly aligned with the clinical diagnosis, the agreement is still notably better than chance. Let us check what happens on the validation data.

```
# "true" labels for the validation discovery data
classes_valid = as.numeric(as.factor(labels[valid_samples]))

adjustedRandIndex(C_2_md, classes_valid)
```

```
## [1] 0.6539913
```

```
adjustedRandIndex(C_2_tf, classes_valid)
```

```
## [1] 0.6308588
```

The ARI values for both method-based and result-based validation are quite similar to the ARI value on the discovery data.

2.4 Stability

Finally, we calculate the stability of cluster membership between discovery and validation set. As defined in our paper, we have to compare C_2^{md} with C_2^{tf}. Again, we use the ARI for this purpose.

```
adjustedRandIndex(C_2_md, C_2_tf)
```

```
## [1] 0.8621116
```

The ARI has a value of about 0.862, which indicates a high, but not perfect stability.

2.5 Summary

Overall, in this particular example and with respect to the validation criteria used, we can speak of successful validation of the clustering results on the validation data. The index values for the clusterings on the validation data were (nearly) as good as for the clustering on the discovery data, and the plots looked very similar. This is perhaps not that surprising, given that we have split a rather large dataset into two parts. The samples appear to not vary that much between discovery and validation data. This might be different when splitting smaller datasets, or if the validation data is an independently collected dataset.

3 Descriptive clustering of the atlas1006 microbiome dataset

We will now present an example for descriptive clustering, namely the clustering of microbes. Such clusterings can generate hypotheses about which microbes interact with each other. We use the `atlas1006` dataset (Lahti et al. 2014) which is publicly available in the `microbiome` R package. The dataset consists of gut microbiome samples from over 1000 adults, and contains 130 genus-like bacterial groups. That is, for each adult, a faecal sample was collected and it was counted how many microbes from each bacterial group appeared in the sample. The term "genus-like group" means that each group contains several bacterial species and corresponds roughly to a bacterial genus. Our aim is to cluster these bacterial groups/genera into higher-level clusters. If you are new to the topic of taxonomy, the Wikipedia article on taxonomic ranks contains some useful examples: https://en.wikipedia.org/wiki/Taxonomic_rank

First, we load the necessary packages: along with the `microbiome` package, we also load the `phyloseq` package which offers general utilities and preprocessing functions for microbiome data, as well as the `SpiecEasi` package which we later use for data normalization. The `SpiecEasi` package can be installed from github with the `devtools::install_github` command.

```
#library(devtools)
#devtools::install_github("zdk123/SpiecEasi")

library(phyloseq)
library(SpiecEasi)
library(microbiome)
```

Next, we load the data into our environment and take a first look:

```
data(atlas1006)

atlas1006
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 130 taxa and 1151 samples ]
## sample_data() Sample Data:       [ 1151 samples by 10 sample variables ]
## tax_table()   Taxonomy Table:    [ 130 taxa by 3 taxonomic ranks ]
```

```
otu_table(atlas1006)[1:5,1:5]
```

```
## OTU Table:          [5 taxa and 5 samples]
##                      taxa are rows
##                              Sample-1 Sample-2 Sample-3 Sample-4 Sample-5
## Actinomycetaceae                    0        0        0        0        0
## Aerococcus                          0        0        0        0        0
## Aeromonas                           0        0        0        0        0
## Akkermansia                        21       36      475       61       34
## Alcaligenes faecalis et rel.        1        1        1        2        1
```

```
tax_table(atlas1006)[1:5,]
```

```
## Taxonomy Table:     [5 taxa by 3 taxonomic ranks]:
##                              Phylum            Family
## Actinomycetaceae             "Actinobacteria"  "Actinobacteria"
## Aerococcus                   "Firmicutes"      "Bacilli"
## Aeromonas                    "Proteobacteria"  "Proteobacteria"
## Akkermansia                  "Verrucomicrobia" "Verrucomicrobia"
## Alcaligenes faecalis et rel. "Proteobacteria"  "Proteobacteria"
##                              Genus
## Actinomycetaceae             "Actinomycetaceae"
## Aerococcus                   "Aerococcus"
## Aeromonas                    "Aeromonas"
## Akkermansia                  "Akkermansia"
## Alcaligenes faecalis et rel. "Alcaligenes faecalis et rel."
```

The `atlas1006` data is saved as a `phyloseq` object. This means that it consists of three different subdatasets: First, the OTU count table, which can be extracted with `otu_table()`. This is the dataset that we will use for clustering. The OTU table shows how often each of the 130 bacterial groups (taxa) appears in each sample, as described above. For example, we see that bacteria from the group "Akkermansia" appeared 21 times in the first sample.

The sample data contains some more information on the samples, but we will not need this for our analysis.

The taxonomy table (extracted via `tax_table()`) shows the higher taxonomic levels of the bacterial groups. For example, the group "Akkermansia" belongs to the family "Verrucomicrobia" and to the phylum with the same name. We will later use this information for external validation.

3.1 Preprocessing and split into discovery and validation sets

Microbiome data typically requires several preprocessing steps, which is the most involved part of our present example. After preprocessing is finished, the clustering and validation procedures are relatively easy. If you are less interested in the preprocessing steps, you may skip ahead towards the end of this subsection. However, the subsection is instructive because it demonstrates how preprocessing must be carefully combined with the split into discovery and validation sets.

A common preprocessing procedure for microbiome data is to keep only samples for which enough reads (i.e., overall bacterial counts, also called sequencing depth) are available (samples with low sequencing depth indicate low measurement quality). Here, we keep the samples with a minimum sequencing depth of more than 10000 reads.

```
sequencing_depths = colSums(otu_table(atlas1006))
atlas1006_filt = prune_samples(sequencing_depths > 10000, atlas1006)
atlas1006_filt
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 130 taxa and 712 samples ]
## sample_data() Sample Data:       [ 712 samples by 10 sample variables ]
## tax_table()   Taxonomy Table:    [ 130 taxa by 3 taxonomic ranks ]
```

We see that 712 samples remain. The number of taxa stays of course constant.

Now we are ready to split the data into discovery and validation sets. Note that in the OTU table displayed above, the objects to be clustered (the bacteria) are in the rows, and the samples are in the columns. Since our aim is to generate a descriptive clustering of the bacterial groups, they have to stay constant between discovery and validation data. Therefore, we will split along the columns of the dataset (i.e., the samples). We then save discovery and validation sets as `phyloseq` objects.

```
set.seed(123)
ncols = ncol(otu_table(atlas1006_filt))
n = 0.5 * ncols
discov_samples = sample(ncols, size = n)
valid_samples = sample(setdiff(1:ncols, discov_samples), n)

discov_phyloseq = phyloseq(otu_table(atlas1006_filt)[,discov_samples],
                           tax_table(atlas1006_filt))
valid_phyloseq = phyloseq(otu_table(atlas1006_filt)[,valid_samples],
                          tax_table(atlas1006_filt))
```

We now continue with preprocessing. It is common to remove rare taxa which occur very infrequently across samples. That is, we keep only the taxa which have counts > 0 in at least 10% of the samples. Why did we not already perform this step above, when we were filtering the samples? This has to do with the split into discovery and validation data, and illustrates why one must be careful when combining the split with preprocessing steps:

The sample filtering was performed sample-wise, i.e., with a calculation across taxa for each individual sample. No information between samples was exchanged. We could have also performed this step after the split into discovery and validation data, but then we might have ended up with different sizes of the discovery and validation sets.

For the filtering of taxa, however, information *between* samples is used. For each bacterial group, the row in the OTU table is traversed to look for samples where no counts are present. That is, it makes a difference whether one performs this step before or after splitting the dataset into two sample sets. As in our previous example for inferential clustering, we try to avoid using information from the validation data for the analysis on the discovery data. Therefore, we perform the filtering of taxa on the discovery data, which gives us the final set of objects to be clustered. On the validation set, we keep the same taxa. This means, of course, that there might be taxa in the validation set which do *not* have counts > 0 in at least 10% of the validation samples (although this is not the case in this particular example).

```
taxa_counts_discov = rowSums(sign(otu_table(discov_phyloseq)))
discov_phyloseq = prune_taxa(taxa_counts_discov > 0.1 * n, discov_phyloseq)
valid_phyloseq = prune_taxa(taxa_counts_discov > 0.1 * n, valid_phyloseq)
discov_data = otu_table(discov_phyloseq)
valid_data = otu_table(valid_phyloseq)

discov_phyloseq
```

```
## phyloseq-class experiment-level object
```

```
## otu_table()   OTU Table:         [ 117 taxa and 356 samples ]
## tax_table()   Taxonomy Table:    [ 117 taxa by 3 taxonomic ranks ]
```

```
valid_phyloseq
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 117 taxa and 356 samples ]
## tax_table()   Taxonomy Table:    [ 117 taxa by 3 taxonomic ranks ]
```

We see that 117 bacterial groups have remained.

For the cluster analysis below we will use hierarchical clustering, which requires a distance or dissimilarity matrix as input. We would like to calculate this dissimilarity matrix based on the Pearson correlations between the bacterial taxa. However, the OTU count tables in its current state cannot be used for calculating the correlations, given that we are not in Euclidean space. Instead, we have positive integer-valued counts which sum up to the fixed sequencing depths. In microbiome analysis, this is known as the "compositional nature" of the data.

We thus have to use a transformation to move the data points into Euclidean space. A popular choice is Aitchison's centered log ratio (clr) transformation (Aitchison 1982), which is implemented in the `SpiecEasi` package. As the clr cannot deal with zeros in the data, we first have to add a pseudo count of 1. Note that the clr transform does only perform calculations *inside* samples, and not *between* samples. Thus, we could also have applied this transformation *before* the split into discovery and validation sets.

```
discov_data = discov_data + 1
discov_data = SpiecEasi::clr(x.f = discov_data, mar = 2, base = exp(1))

valid_data = valid_data + 1
valid_data = SpiecEasi::clr(x.f = valid_data, mar = 2, base = exp(1))
```

The transformed OTU count tables now look as follows:

```
discov_data[1:5, 1:5]
```

```
##                              Sample-635 Sample-687  Sample-318 Sample-778
## Actinomycetaceae             -2.4305869 -2.4908494 -2.53827082  -1.929597
## Akkermansia                  -0.1280018  0.3423639  0.79393369   2.297236
## Alcaligenes faecalis et rel. -1.3319747 -1.7977022 -1.43965853  -1.524132
## Allistipes et rel.            3.2461669  2.7131573 -0.05336417   2.002228
## Anaerofustis                 -2.4305869 -2.4908494 -2.53827082  -2.622745
##                              Sample-340
## Actinomycetaceae              -2.844564
## Akkermansia                    0.869008
## Alcaligenes faecalis et rel.  -1.745952
## Allistipes et rel.             2.082690
## Anaerofustis                  -2.844564
```

```
valid_data[1:5, 1:5]
```

```
##                              Sample-1159  Sample-508 Sample-399 Sample-1124
## Actinomycetaceae             -2.41622177 -2.28793888  -2.015841   -2.613590
## Akkermansia                   0.06868488 -0.20849734   1.119654    1.717144
## Alcaligenes faecalis et rel. -1.72307459  0.01464621  -1.610376   -1.920443
## Allistipes et rel.            0.87961510  1.82293498   3.328883    3.305304
## Anaerofustis                 -2.41622177 -2.28793888  -2.708988   -2.613590
##                              Sample-220
## Actinomycetaceae              -2.638416
## Akkermansia                    1.894183
## Alcaligenes faecalis et rel.  -1.539804
```

```
## Allistipes et rel.             1.872443
## Anaerofustis                  -2.638416
```

As the clr is a monotone transformation (sample-wise), higher values indicate higher original OTU counts.

3.2 Method selection and internal validation

We perform hierarchical clustering of the bacterial groups based on the discovery data. To generate the dissimilarity matrix required for hierarchical clustering, we first calculate the Pearson correlations between the bacteria:

```
cor_mat_discov = stats::cor(t(discov_data), use = "complete.obs", method = "pearson")
```

To transform the correlation matrix (which represents similarities) into a *dis*similarity matrix, we apply the signed distance $\sqrt{0.5(1-r_{ij})}$ (see e.g. Peschel et al. (2021) for more detailed explanations).

```
diss_mat_discov = sqrt(0.5 * (1-cor_mat_discov))
diss_mat_discov[1:5,1:5]
```

```
##                             Actinomycetaceae Akkermansia
## Actinomycetaceae                   0.0000000   0.7599526
## Akkermansia                        0.7599526   0.0000000
## Alcaligenes faecalis et rel.       0.6901934   0.7090702
## Allistipes et rel.                 0.8269252   0.6947596
## Anaerofustis                       0.6233327   0.7380582
##                             Alcaligenes faecalis et rel. Allistipes et rel.
## Actinomycetaceae                               0.6901934          0.8269252
## Akkermansia                                    0.7090702          0.6947596
## Alcaligenes faecalis et rel.                   0.0000000          0.6853242
## Allistipes et rel.                             0.6853242          0.0000000
## Anaerofustis                                   0.7080303          0.7532644
##                             Anaerofustis
## Actinomycetaceae               0.6233327
## Akkermansia                    0.7380582
## Alcaligenes faecalis et rel.   0.7080303
## Allistipes et rel.             0.7532644
## Anaerofustis                   0.0000000
```

Now we use the resulting dissimilarity matrix for hierarchical clustering. As in our previous example, we use the ASW internal validation index to determine the "best" number of clusters, with k ranging from 2 to 10.

```
sw = numeric(9)

h_clust_discov = hclust(as.dist(diss_mat_discov), method = "average")

for (k in 2:10) {
  cluster_hierarch = cutree(h_clust_discov, k = k)

  sw[k-1] = mean(cluster::silhouette(cluster_hierarch, diss_mat_discov)[,3])

}

best_k = which(sw == max(sw)) + 1
best_k
```

```
## [1] 4
```

```
sw_discov = max(sw)
C_1 = cutree(h_clust_discov, k = best_k)
table(C_1)
```

```
## C_1
##  1  2  3  4
## 42 15 28 32
```

```
print(sw_discov)
```

```
## [1] 0.133544
```

It turns out that $k = 4$ is the best choice according to the ASW. Note, however, that the value of the ASW is rather low even for this best case.

For method-based validation, we apply the hierarchical clustering with $k = 4$ to the validation data. Beforehand, we calculate the correlations and then the dissimilarity matrix.

```
cor_mat_valid = stats::cor(t(valid_data), use = "complete.obs", method = "pearson")

diss_mat_valid = sqrt(0.5 * (1-cor_mat_valid))

h_clust_valid = hclust(as.dist(diss_mat_valid), method = "average")
C_2_md = cutree(h_clust_valid, k = best_k)
sw_valid = mean(cluster::silhouette(C_2_md, diss_mat_valid)[,3])

table(C_2_md)
```

```
## C_2_md
##  1  2  3  4
## 54 15 16 32
```

```
print(sw_valid)
```

```
## [1] 0.1430595
```

We see that the ASW of C_2^{md} is very similar to the ASW for C_1.

For result-based validation, the transferring process from C_1 to C_2^{tf} is easier than for inferential clustering: because the objects to cluster remain constant between discovery and validation data, we can simply set $C_1 = C_2^{tf}$.

```
C_2_tf = C_1

sw_valid = mean(cluster::silhouette(C_2_tf, diss_mat_valid)[,3])

table(C_2_tf)
```

```
## C_2_tf
##  1  2  3  4
## 42 15 28 32
```

```
print(sw_valid)
```

```
## [1] 0.1288389
```

Again, the ASW value is rather similar to the result on the discovery data.

We move on to other validation strategies on the validation data: stability analysis, visual validation and external validation.

3.3 Stability

How similar are the partitions of the 117 fixed bacterial groups given by C_2^{md} and $C_2^{tf} = C_1$? As before, we use the Adjusted Rand Index (ARI) with the `mclust` package.

```
library(mclust)
adjustedRandIndex(C_2_md, C_2_tf)
```

```
## [1] 0.7536271
```

The ARI value of about 0.754 shows reasonable stability between cluster membership on the discovery and validation data, but the partitions are definitely not identical. Several bacterial taxa are grouped differently based on discovery vs. validation data. To further analyse this, we cross-tabulate the cluster memberships.

```
table(C_1, C_2_md)
```

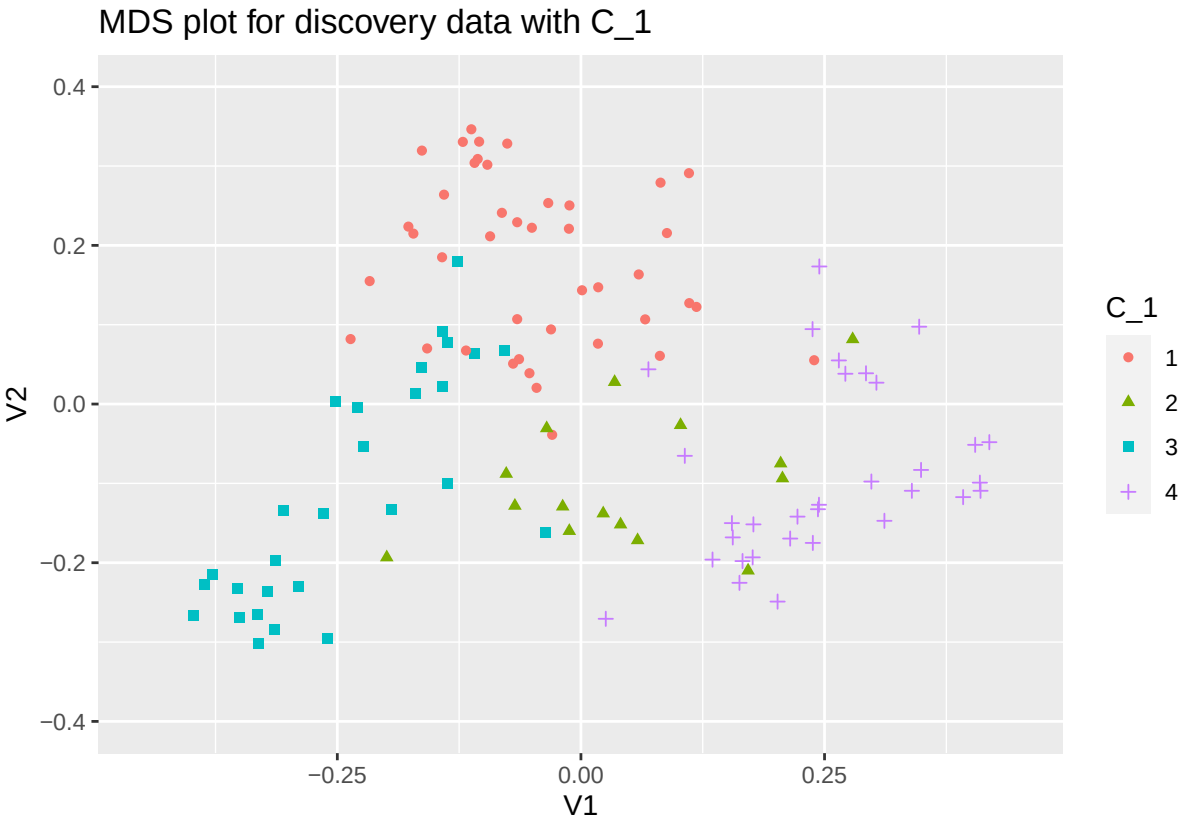
```
##    C_2_md
## C_1  1  2  3  4
##   1 42  0  0  0
##   2  0 15  0  0
##   3 12  0 16  0
##   4  0  0  0 32
```

Clusters 2 and 4 perfectly match across C_1 and C_2^{md}. The differences are in clusters 1 and 3. Cluster 1 of C_2^{md} contains not only elements from cluster 1 of C_1, but also some elements from cluster 3 of C_1. We can also see this with visualisation tools.

3.4 Visual validation

Multidimensional scaling (MDS) is a visualisation technique that takes a distance/dissimilarity matrix as input and display the objects in a lower-dimensional space such that the distances between the objects are preserved as well as possible. We can thus use this method in combination with our dissimilarity matrices to visualise the bacterial groups and their clustering. Cluster membership is indicated by different colours and shapes.

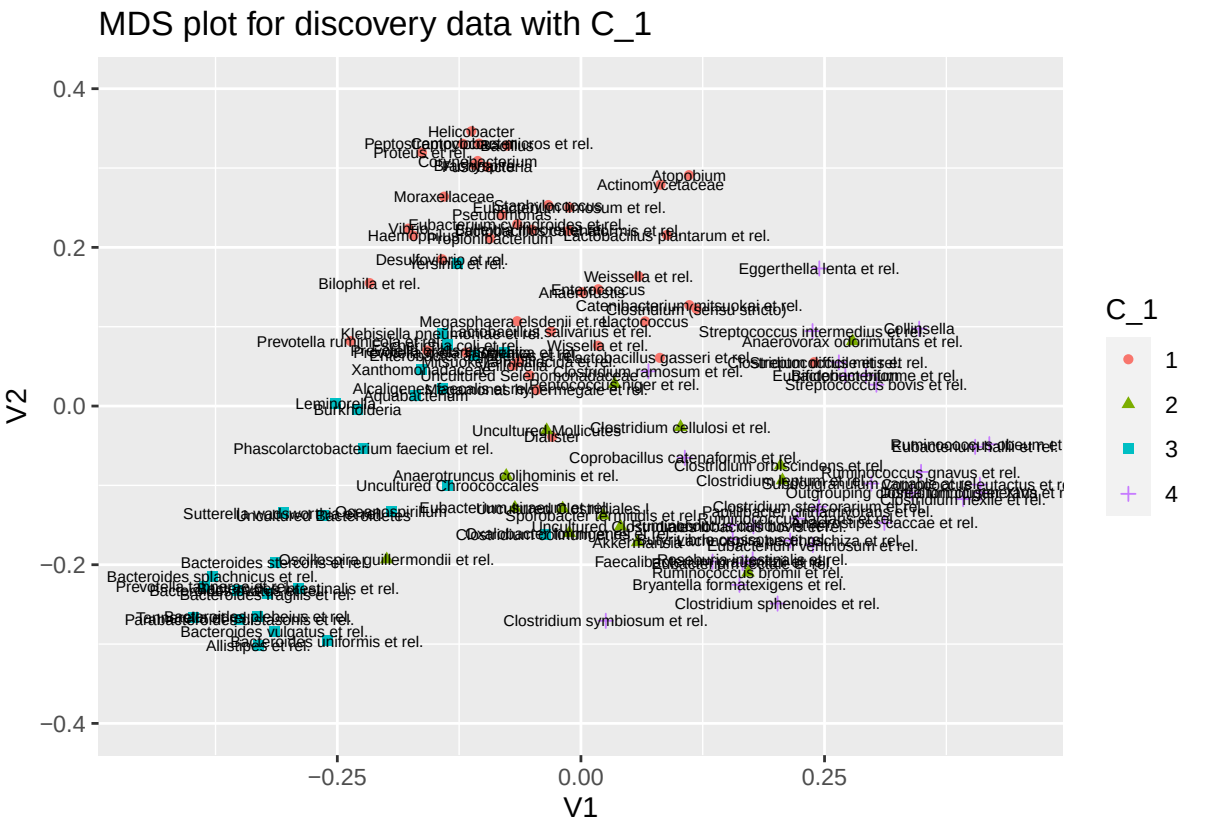
```
library(ggplot2)

mds_discov = cmdscale(diss_mat_discov, k = 2, eig = TRUE)
discov_data_plot = as.data.frame(mds_discov$points)
discov_data_plot$C_1 = as.factor(C_1)

ggplot(discov_data_plot, aes(V1, V2)) +
  geom_point(aes(colour = C_1, shape = C_1)) + xlim(-0.45, 0.45) + ylim(-0.4, 0.4) +
  labs(title = "MDS plot for discovery data with C_1")
```



The clusters are clearly not well separated (which also explains the rather low value of the ASW).

Showing the taxa labels might help to better interpret the clusters and generate biological hypotheses. Here, the resulting plot is rather confusing because of the large number of labels. With suitable taxa abbreviations, however, the plot could be rendered more readable, although we will not pursue this here.

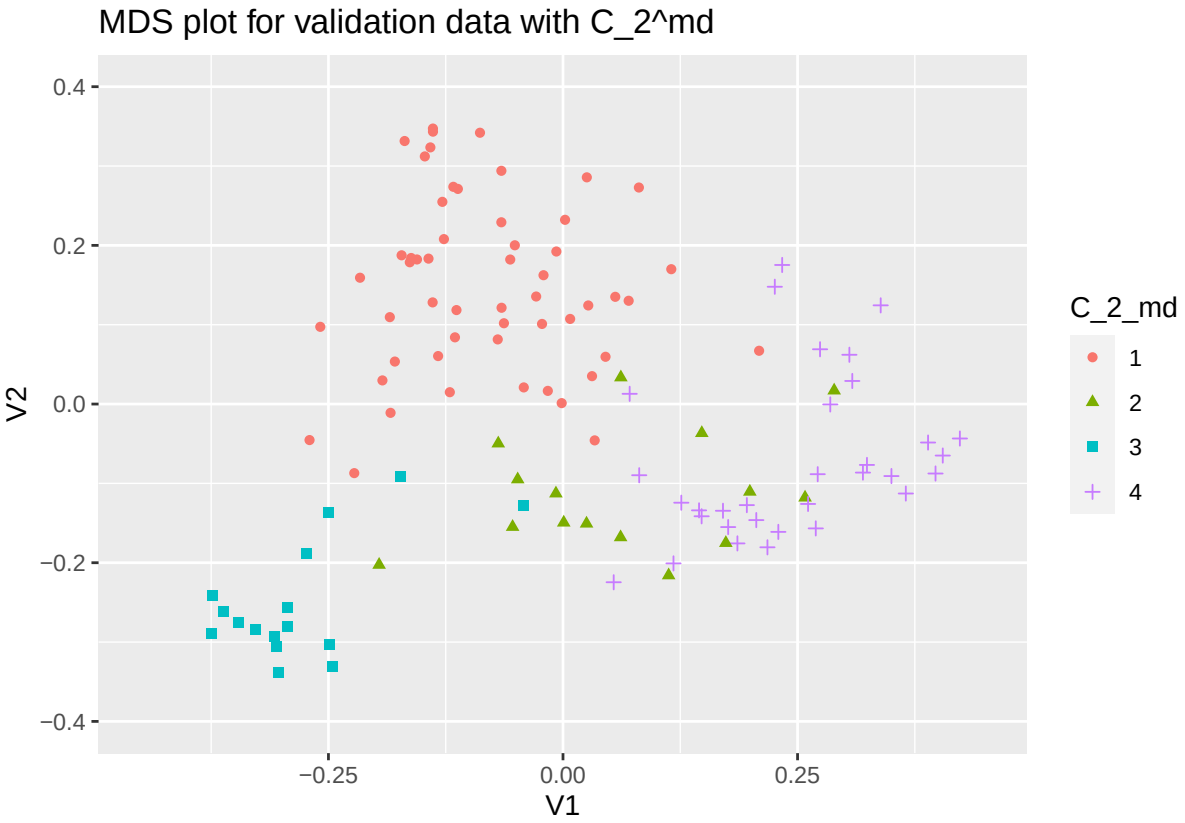
```
ggplot(discov_data_plot, aes(V1, V2, label = rownames(discov_data_plot))) +
  geom_point(aes(colour = C_1, shape = C_1)) +
  geom_text(size = 2) + xlim(-0.45, 0.45) + ylim(-0.4, 0.4) +
  labs(title = "MDS plot for discovery data with C_1")
```



Next, we want to compare this plot with MDS visualisations on the validation data. In the example for inferental clustering above, we embedded the validation data points into the projection space defined by the PCA on the discovery data. Embedding procedures are also possible for MDS. However, as written in the paper, this would not be informative in the case of descriptive clustering, as the points (here: bacterial groups) would be identical. We thus perform MDS anew for the validation data.

We start with method-based validation, where we want to indicate the cluster memberships as given by C_2^{md} in the MDS plot. As above, we would like to match the clusters of C_2^{md} to those of C_1, to ensure that "corresponding" clusters have the same colours and shapes in both plots. Since the clusters were generated with hierarchical clustering, the centroid matching that we used in the above example for inferential clustering is not necessarily appropriate here. Instead, we recall from the stability analysis that cluster 2 and 4 are perfectly matched, and that clusters 1 and 3 of C_2^{md} roughly "correspond" to clusters 1 and 3 of C_1. That is, we do not have to rename the clusters in C_2^{md}.

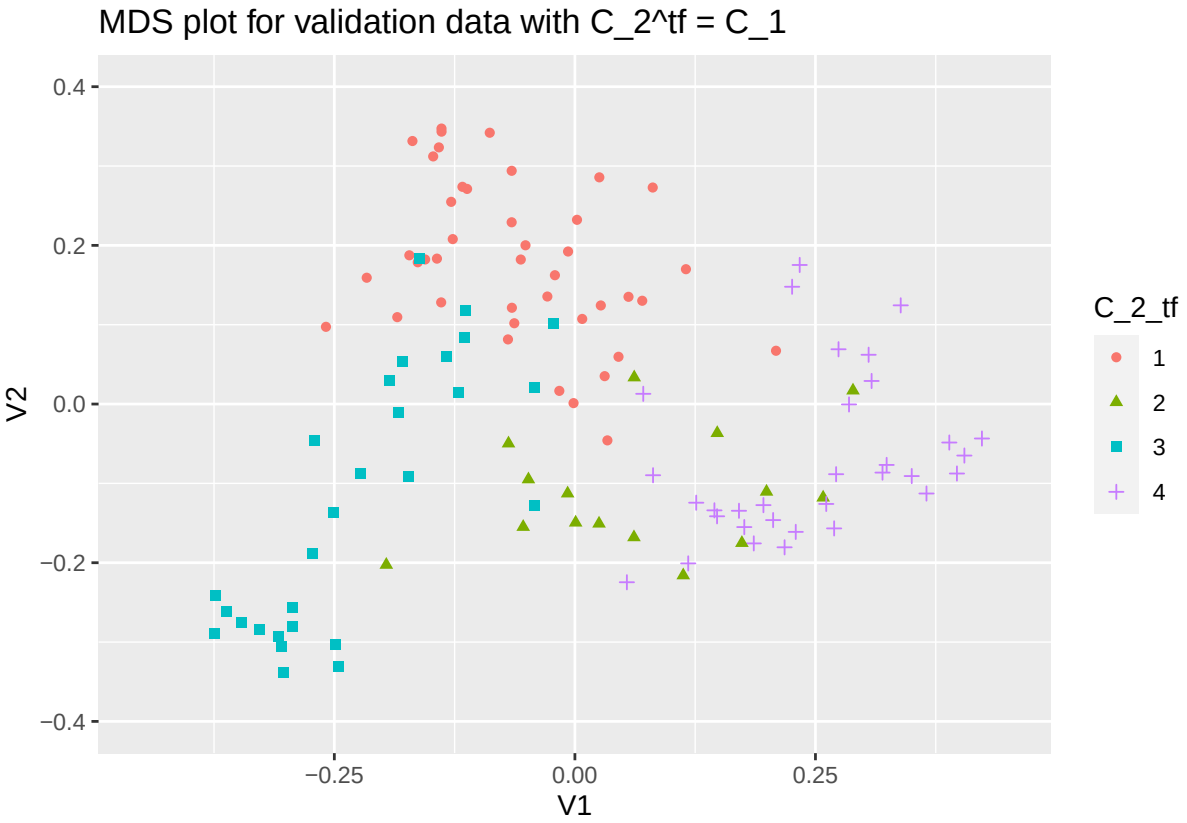
```
mds_valid = cmdscale(diss_mat_valid, k = 2, eig = TRUE)
valid_data_plot = as.data.frame(mds_valid$points)
valid_data_plot$C_2_md = as.factor(C_2_md)
valid_data_plot$C_2_tf = as.factor(C_2_tf)

ggplot(valid_data_plot, aes(V1, V2)) +
  geom_point(aes(colour = C_2_md, shape = C_2_md)) + xlim(-0.45, 0.45) +  ylim(-0.4, 0.4) +
  labs(title = "MDS plot for validation data with C_2^md")
```



Similar patterns can be seen as in the plot for the discovery data. Optionally, taxa labels or numbers could be displayed to see how the positions of the taxa have shifted on the validation data.

Again, we consider the MDS plot for the validation data, but this time cluster colours and shapes are indicated according to $C_2^{tf} = C_1$.

```
ggplot(valid_data_plot, aes(V1, V2)) +
  geom_point(aes(colour = C_2_tf, shape = C_2_tf)) + xlim(-0.45, 0.45) + ylim(-0.4, 0.4) +
  labs(title = "MDS plot for validation data with C_2^tf = C_1")
```



As we could already see in the stability analysis, the comparison of the two plots for the validation data shows that cluster 1 of C_2^{md} contains some taxa that were previously sorted into cluster 3 of C_1.

3.5 External validation

Finally, we use external validation to evaluate our results. Our aim was to cluster the bacterial genus-like groups into higher-level clusters. The taxonomy table of our phyloseq objects contains external information on such higher-level groups, namely the grouping into taxonomic families and - several levels above - into taxonomic phyla. We can thus check whether our clustering on the discovery data aligns with the partition into families and phyla, and whether this can be replicated on the validation data. For the latter, we only perform method-based validation, i.e., compare C_2^{md} with C_1. Result-based validation does not make sense here, as $C_2^{tf} = C_1$.

```
labels = tax_table(discov_phyloseq)
# the taxonomy table is the same for the validation data

# agreement with the partition into families:
adjustedRandIndex(C_1, as.numeric(as.factor(labels[,2])))
```

```
## [1] 0.2264815
```

```
adjustedRandIndex(C_2_md, as.numeric(as.factor(labels[,2])))
```

```
## [1] 0.2541615
```

```
# agreement with the partition into phyla:
adjustedRandIndex(C_1, as.numeric(as.factor(labels[,1])))
```

```
## [1] 0.1535982
```

```
adjustedRandIndex(C_2_md, as.numeric(as.factor(labels[,1])))
```

```
## [1] 0.1432907
```

Overall, our clusterings are notably different from the partitions into families and phyla, although the agreement is better than chance. The ARI values are similar for both discovery and validation data.

3.6 Summary

In this example, the clusterings on discovery and validation data had similarities, but also some differences with respect to cluster membership. This could also be seen in the MDS plots. Regarding internal and external validation, the clusterings showed similar quality. External validation indicated that bacterial genera from different families and phyla were grouped together, and might possibly interact with each other. Of course, more thorough analyses would be required to further examine such hypotheses.

4 Visual validation plots for the Iris data

Here we focus on visual validation and use the Iris dataset (Anderson 1935) as a simple toy example to generate principal components plots, cluster heatmaps and silhouette plots.

First, load the necessary packages. To install the `ComplexHeatmap` package, you have to use the `BiocManager` package.

```
# library(BiocManager)
# BiocManager::install("ComplexHeatmap")

library(ggplot2)
library(ggfortify)
library(gridExtra)
library(ComplexHeatmap)
library(cluster)
library(factoextra)
```

The Iris dataset is already contained in R. We remove the species labels, then shuffle and scale the data. There are 150 flower samples, and four variables were measured for each sample: sepal length, sepal width, petal length and petal width.

```
set.seed(123)
iris2 = iris[,1:4]
iris2 = iris2[sample(nrow(iris2)),]
iris2 = as.data.frame(scale(iris2))
head(iris2)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width
## 14   -1.86378030  -0.1315388   -1.5056946  -1.4422448
## 50   -1.01843718   0.5567457   -1.3357516  -1.3110521
## 118   2.24217198   1.7038865    1.6665739   1.3128014
## 43   -1.74301699   0.3273175   -1.3923993  -1.3110521
## 150   0.06843254  -0.1315388    0.7602115   0.7880307
## 148   0.79301235  -0.1315388    0.8168591   1.0504160
```

We split the dataset into discovery and validation sets, where we split along the samples (because our goal is inferential clustering of the flower samples). As we have already shuffled the dataset, we can choose the first 75 samples as discovery set and the remaining samples as the validation set.

```
discov_data = iris2[1:75,]
valid_data = iris2[76:150,]
```

Now we apply k-means clustering with $k = 3$ and 10 random starts to the discovery set. We reorder the label names of the resulting clustering such that cluster 1 is the first cluster on the left hand side in the PCA plot below.

```
cluster_kmeans_discov = kmeans(discov_data, centers = 3, nstart = 10)
C_1 = cluster_kmeans_discov$cluster

C_1_renamed = C_1
C_1_renamed[C_1 == 1] = 2
C_1_renamed[C_1 == 2] = 1
C_1_renamed[C_1 == 3] = 3
C_1 = C_1_renamed

centroids_discov = cluster_kmeans_discov$centers[c(2, 1, 3),]
```

We also apply k-means clustering to the validation data to yield C_2^{md}. In the present example, we will only compare C_1 to C_2^{md} (method-based validation), and will not consider C_2^{tf} (result-based validation).

```
cluster_kmeans_valid = kmeans(valid_data, centers = 3, nstart = 10)
C_2_md = cluster_kmeans_valid$cluster
centroids_valid = cluster_kmeans_valid$centers
```

Now we calculate the PCA for the discovery data:

```
discov.pca = prcomp(discov_data)
```

We want to compare the PCA plot for the discovery data with the PCA plot for the validation data. As in the example involving the Wisconsin breast cancer dataset above, we will plot the validation dataset on the projection space defined by the discovery dataset. Thus we project the validation data onto the PCs of `discov.pca` via scaling and rotating.

```
valid_scale = scale(valid_data, center = discov.pca$center)
valid_projection = valid_scale %*% discov.pca$rotation
valid.pca = discov.pca
valid.pca$x = valid_projection
```

When we compare C_1 to C_2^{md} with the PCA plots, we want the colours of the clusterings to match. We use the same strategy as for the Wisconsin breast cancer example: to match the clusters of C_2^{md} to the clusters of C_1, we calculate the distances between the cluster centroids with the `proxy` package.

```
library(proxy)
rownames(centroids_valid) = c("C_2_md: clust 1", "C_2_md: clust 2", "C_2_md: clust 3")
rownames(centroids_discov) = c("C_1: clust 1", "C_1: clust 2", "C_1: clust 3")

proxy::dist(centroids_valid, centroids_discov)
```

```
##                 C_1: clust 1 C_1: clust 2 C_1: clust 3
## C_2_md: clust 1    2.8315445    0.5149277    2.3656479
## C_2_md: clust 2    3.8153211    1.3702816    0.5359275
## C_2_md: clust 3    0.2905475    3.1705558    4.0137996
```

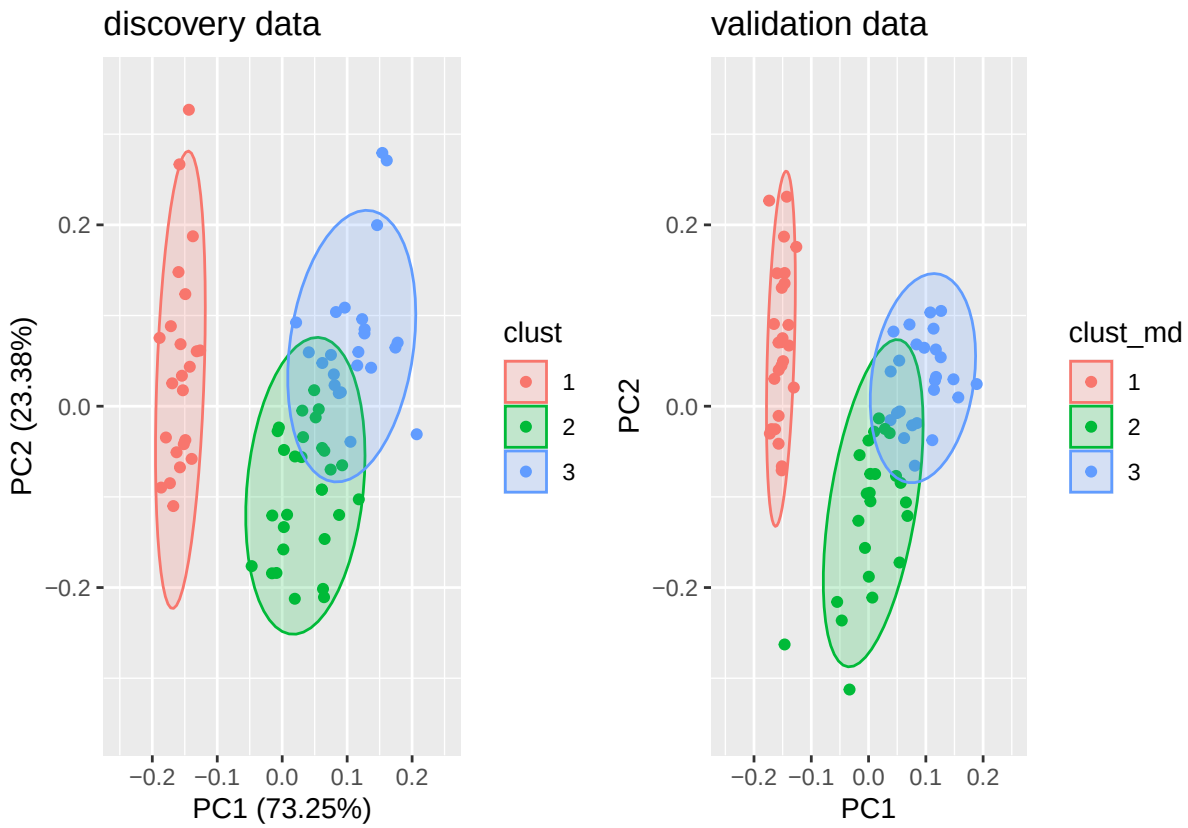
We rename the cluster labels of C_2^{md} according to the cluster correspondence with C_1.

```
C_2_md_renamed = C_2_md
C_2_md_renamed[C_2_md == 1] = 2
C_2_md_renamed[C_2_md == 2] = 3
C_2_md_renamed[C_2_md == 3] = 1
```

Now we can display the PCA plots for C_1 vs. C_2^{md}, using the `ggfortify` package. As before, samples are

coloured according to their cluster membership, and ellipses are drawn around the centers of the clusters (see `ggplot2::stat_ellipse()` for details).

```
discov_data$clust = as.factor(C_1)

valid_data$clust_md = as.factor(C_2_md_renamed)

p1 = autoplot(discov.pca, data = discov_data, colour = "clust",
              group_by = "clust", frame = TRUE, frame.type = "t",
              main = "discovery data") + xlim(-0.25, 0.25) + ylim(-0.35, 0.35)
p2 = autoplot(valid.pca, data = valid_data, colour = "clust_md",
              group_by = "clust_md", frame = TRUE, frame.type = "t",
              main = "validation data") + xlim(-0.25, 0.25) + ylim(-0.35, 0.35) +
  labs(x = "PC1", y = "PC2")
p_pca = grid.arrange(p1, p2, ncol=2)
```

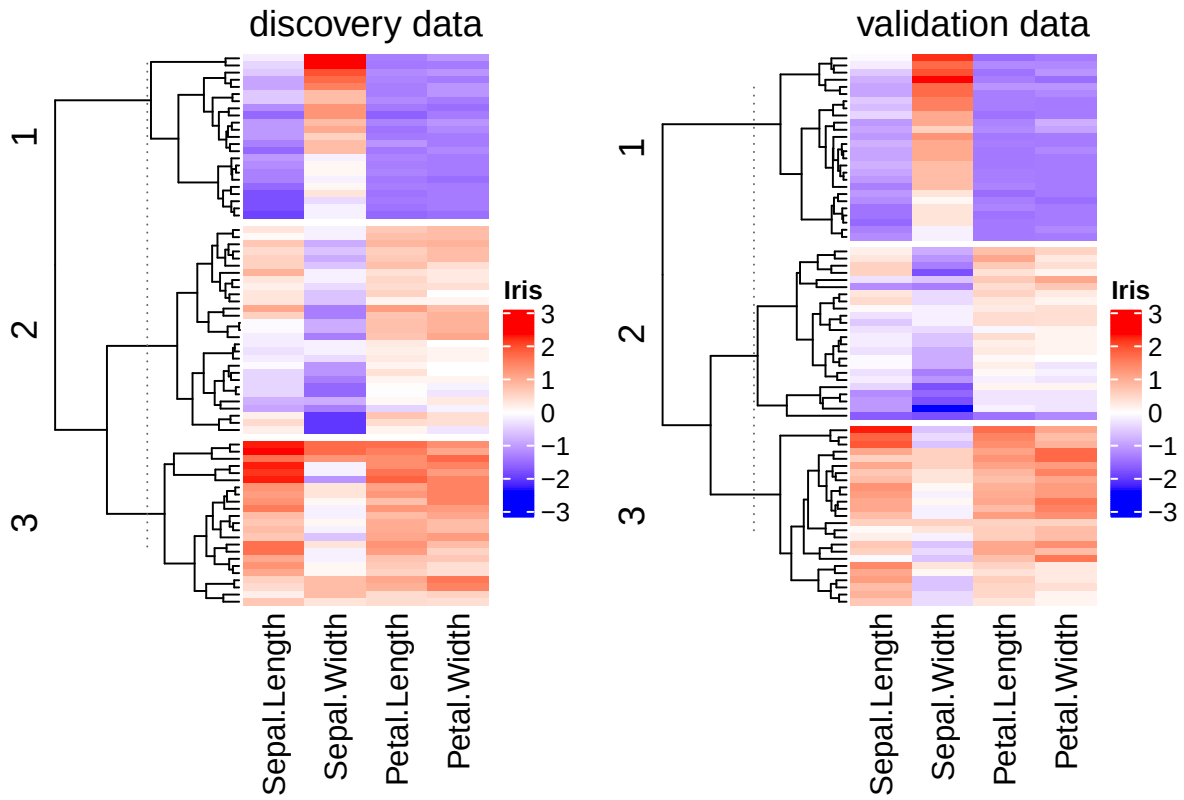


It can be seen that the first cluster is notably separated from the other two clusters along the first principal component. Clusters 2 and 3 show some overlap. This is similar for both discovery and validation data.

Using the package `ComplexHeatmap`, we generate the cluster heatmaps. For discovery and validation sets separately, the samples are first ordered by cluster membership according to the k-means clustering. Next, for each individual cluster in turn, hierarchical clustering is applied to the samples in the respective cluster to generate a dendrogram, resulting in three dendrograms overall. These dendrograms are connected by a parent dendrogram (left to the dashed line), which is generated based on the centroids of the three clusters. Here, because cluster 1 is more dissimilar to the other two, it is split from clusters 2 and 3 right at the head of the parent dendrogram, while clusters 2 and 3 are split one level below. After the samples have been ordered in this way, the variable values are indicated by colours.

For more details, see `?ComplexHeatmap::Heatmap` and Chapter 2 in the package reference manual: https://jokergoo.github.io/ComplexHeatmap-reference/book/a-single-heatmap.html

```
library(circlize)
col_fun = colorRamp2(c(-2.3, 0, 2.3), c("blue", "white", "red"))

p1 = grid.grabExpr(draw(Heatmap(as.matrix(discov_data[,1:4]),
                                name = "Iris", col = col_fun, row_split = discov_data$clust,
                                show_row_names = FALSE, row_dend_width = unit(2.5, "cm"),
                                cluster_columns = FALSE,
                                column_title = "discovery data")))

p2 = grid.grabExpr(draw(Heatmap(as.matrix(valid_data[,1:4]),
                                name = "Iris", col = col_fun, row_split = valid_data$clust_md,
                                show_row_names = FALSE, row_dend_width = unit(2.5, "cm"),
                                cluster_columns = FALSE,
                                column_title = "validation data")))

p_heatmap = grid.arrange(p1, p2, ncol=2)
```



Again, this visualizes the difference of the first cluster to the other two clusters: For both discovery and validation data, cluster 1 is marked by higher values of sepal width and lower values of sepal length, petal length and petal width.

Finally, we use the `factoextra` package to generate the silhouette plots to compare C_1 and C_2^{md}. This is an alternative to using `plot(cluster::silhouette())` as we did for the breast cancer example above. The principle remains the same: The samples are sorted first by their cluster membership, and then by the

magnitude of their silhouette values. Here, the clusters are marked by the same colours as in the PCA plot. The overall average silhouette widths are indicated by red dashed lines.

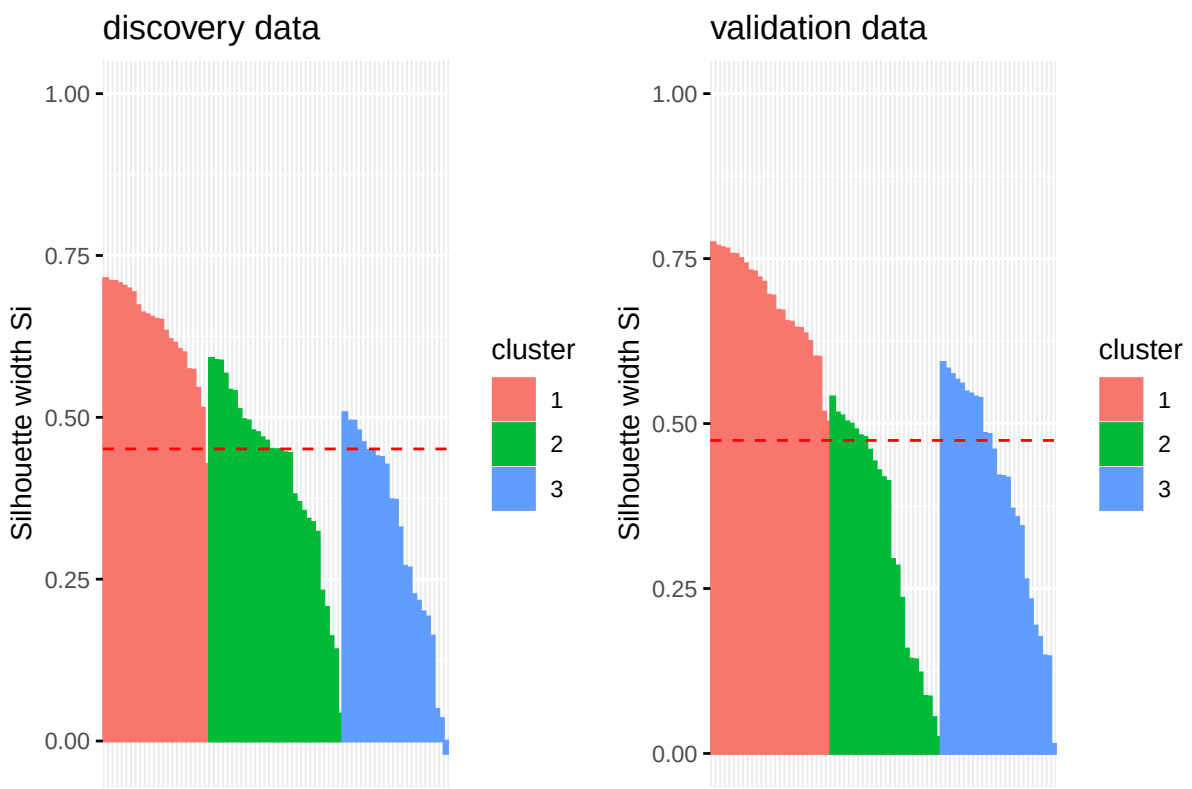
```
sil_discov = cluster::silhouette(as.numeric(discov_data$clust), dist(discov_data[,1:4]))
sil_valid = cluster::silhouette(as.numeric(valid_data$clust_md), dist(valid_data[,1:4]))

p1 = fviz_silhouette(sil_discov) + labs(title = "discovery data")
```

```
##   cluster size ave.sil.width
## 1       1   23          0.63
## 2       2   29          0.41
## 3       3   23          0.32
```

```
p2 = fviz_silhouette(sil_valid) + labs(title = "validation data")
```

```
##   cluster size ave.sil.width
## 1       1   26          0.68
## 2       2   24          0.33
## 3       3   25          0.40
```

```
p_silh = grid.arrange(p1, p2, ncol = 2)
```



Cluster 1 generally has higher silhouette values than clusters 2 and 3, indicating that cluster 1 is more cohesive and separated than the other two. Again, this is similar for both discovery and validation data. On the discovery dataset, cluster 2 has a slightly higher average silhouette width than cluster 3; on the validation dataset, it is the other way around.

References

Aitchison, John. 1982. "The Statistical Analysis of Compositional Data." *Journal of the Royal Statistical Society: Series B (Methodological)* 44 (2): 139–60.

Anderson, Edgar. 1935. "The Irises of the Gaspé Peninsula." *Bulletin of the American Iris Society* 59: 2–5.

Hubert, Lawrence, and Phipps Arabie. 1985. "Comparing Partitions." *Journal of Classification* 2 (1): 193–218.

Kaufman, Leonard, and Peter J. Rousseeuw. 2009. *Finding Groups in Data: An Introduction to Cluster Analysis*. Hoboken, NJ: John Wiley & Sons.

Lahti, Leo, Jarkko Salojärvi, Anne Salonen, Marten Scheffer, and Willem M. De Vos. 2014. "Tipping Elements in the Human Intestinal Ecosystem." *Nature Communications* 5 (1): 1–10.

Peschel, Stefanie, Christian L. Müller, Erika von Mutius, Anne-Laure Boulesteix, and Martin Depner. 2021. "NetCoMi: Network Construction and Comparison for Microbiome Data in R." *Briefings in Bioinformatics* 22 (4): bbaa290.

Rand, William M. 1971. "Objective Criteria for the Evaluation of Clustering Methods." *Journal of the American Statistical Association* 66 (336): 846–50.

Street, W. Nick, William H. Wolberg, and Olvi L. Mangasarian. 1993. "Nuclear Feature Extraction for Breast Tumor Diagnosis." In *IS&T/Spie 1993 International Symposium on Electronic Imaging: Science and Technology*, 1905:861–70. San Jose, CA: International Society for Optics; Photonics.

B Contribution 2: "Over-optimism in unsupervised microbiome analysis: Insights from network learning and clustering"

This chapter is a reprint of:

Ullmann, T., Peschel, S., Finger, P., Müller, C. L., & Boulesteix, A.-L. (2023). Over-optimism in unsupervised microbiome analysis: Insights from network learning and clustering. *PLoS Computational Biology*, *19*(1), e1010820

Copyright:

Author contributions:

T. Ullmann conceptualized the paper together with C. L. Müller and A.-L. Boulesteix (ideas, formulation of overarching aims, etc.), based on a pilot study designed by P. Finger and A.-L. Boulesteix. T. Ullmann designed the methodology of the article, with all other authors providing input and support. T. Ullmann analyzed the data and wrote the R code for this purpose, with support from S. Peschel. T. Ullmann and S. Peschel generated figures for visualizing the study design and the results. The original draft of the manuscript was written by T. Ullmann. All authors contributed to the review and editing of the manuscript. All authors read and approved the final version of the article.

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Over-optimism in unsupervised microbiome analysis: Insights from network learning and clustering

Theresa Ullmann[1,2]*, **Stefanie Peschel**[3,4], **Philipp Finger**[1], **Christian L. Müller**[4,5,6☯], **Anne-Laure Boulesteix**[1,2☯]

1 Institute for Medical Information Processing, Biometry, and Epidemiology, Ludwig-Maximilians-Universität München, München, Germany, **2** Munich Center for Machine Learning (MCML), München, Germany, **3** Institute for Asthma and Allergy Prevention, Helmholtz Zentrum München, Neuherberg, Germany, **4** Department of Statistics, Ludwig-Maximilians-Universität München, München, Germany, **5** Institute of Computational Biology, Helmholtz Zentrum München, Neuherberg, Germany, **6** Center for Computational Mathematics, Flatiron Institute, New York, New York, United States of America

☯ These authors contributed equally to this work.
* tullmann@ibe.med.uni-muenchen.de

Citation: Ullmann T, Peschel S, Finger P, Müller CL, Boulesteix A-L (2023) Over-optimism in unsupervised microbiome analysis: Insights from network learning and clustering. PLoS Comput Biol 19(1): e1010820. https://doi.org/10.1371/journal.pcbi.1010820

Editor: Luis Pedro Coelho, Fudan University, CHINA

Received: July 19, 2022

Accepted: December 15, 2022

Published: January 6, 2023

Data Availability Statement: The AGP dataset can be accessed on Zenodo at https://doi.org/10.5281/zenodo.6652711. The source code used to produce the results and analyses presented in this manuscript is available on Github at https://github.com/thullmann/overoptimism-microbiome.

Funding: This work has been partially supported by the German Federal Ministry of Education and Research (BMBF, www.bmbf.de) [grant number 01IS18036A to A.-L. B. (Munich Center of Machine Learning)] and the German Research Foundation

Abstract

In recent years, unsupervised analysis of microbiome data, such as microbial network analysis and clustering, has increased in popularity. Many new statistical and computational methods have been proposed for these tasks. This multiplicity of analysis strategies poses a challenge for researchers, who are often unsure which method(s) to use and might be tempted to try different methods on their dataset to look for the "best" ones. However, if only the best results are selectively reported, this may cause over-optimism: the "best" method is overly fitted to the specific dataset, and the results might be non-replicable on validation data. Such effects will ultimately hinder research progress. Yet so far, these topics have been given little attention in the context of unsupervised microbiome analysis. In our illustrative study, we aim to quantify over-optimism effects in this context. We model the approach of a hypothetical microbiome researcher who undertakes four unsupervised research tasks: clustering of bacterial genera, hub detection in microbial networks, differential microbial network analysis, and clustering of samples. While these tasks are unsupervised, the researcher might still have certain expectations as to what constitutes interesting results. We translate these expectations into concrete evaluation criteria that the hypothetical researcher might want to optimize. We then randomly split an exemplary dataset from the American Gut Project into discovery and validation sets multiple times. For each research task, multiple method combinations (e.g., methods for data normalization, network generation, and/or clustering) are tried on the discovery data, and the combination that yields the best result according to the evaluation criterion is chosen. While the hypothetical researcher might only report this result, we also apply the "best" method combination to the validation dataset. The results are then compared between discovery and validation data. In all four research tasks, there are notable over-optimism effects; the results on the validation data set are worse compared to the discovery data, averaged over multiple random splits into discovery/validation data. Our study thus highlights the importance of validation and replication

(DFG, www.dfg.de) [grant number BO3139/7-1 to A.-L. B.]. The authors of this work take full responsibility for its content. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

in microbiome analysis to obtain reliable results and demonstrates that the issue of over-optimism goes beyond the context of statistical testing and fishing for significance.

Author summary

Microbiome research focuses on communities of microbes, for example, those living in the human gut. To identify the structure of such communities, constructing microbial networks that represent associations between different microbes has become popular. The microbial associations are often further analyzed by applying cluster algorithms, i.e., researchers try to find groups (clusters) of microbes that are strongly associated with each other. Likewise, researchers are also interested in finding clusters of samples that are similar in bacterial compositions, often referred to as enterotypes. To produce broader and more reliable insights, networks and clustering results that have been constructed based on one specific dataset should generalize to other datasets as well. However, this may be compromised by the large number of statistical methods available for network learning and clustering. Due to uncertainty about which method to use, researchers might try multiple approaches on their dataset and pick the method which yields the "best" result (e.g., the network that has the highest number of strongly connected microbes). When many such methods are tried, the "best" method may be overly fitted to the specific dataset at hand, and the result may not generalize to new data. Our study demonstrates such over-optimism effects and gives recommendations for detecting and/or avoiding over-optimistic bias. We aim to generate greater awareness around this issue and to increase reliability of future microbiome studies.

This is a *PLOS Computational Biology* Methods paper.

1 Introduction

The popularity of microbiome research has surged in recent decades. Many hypotheses about the human microbiome, as well as the microbiome of other species or in various environments, are postulated and tested each year. At the same time, new statistical and computational methods for analyzing microbiome data are continually introduced. Microbiome analysis has yielded exciting results, leading to high hopes for new treatment and prevention options in medicine [1, 2].

In such a fast-moving and promising research field, validation is of vital importance to ensure the reliability of new results. Yet such practices may sometimes be neglected in favor of chasing new hypotheses. There is a certain danger of *over-optimism* in the field: New and exciting results might turn out to be non-replicable, i.e., they cannot be confirmed in studies with independent data. While a discussion about validation and replication has emerged in microbiome research in recent years [3, 4], it is not as advanced as in other fields such as psychology, where the so-called "replication crisis" has received considerable attention [5]. There is a lack of studies which illustrate the validation process in microbiome analysis and quantify over-optimism and (non)replicability. In particular, scant attention has been given to these topics in relation to *unsupervised* microbiome data analysis, e.g., network analysis and clustering.

In the present paper, we take a step toward filling this gap. We illustrate how over-optimism can arise in unsupervised microbiome analysis using four unsupervised "research tasks" as examples: clustering bacterial genera, finding hubs in microbial networks, differential network analysis, and clustering samples. The underlying idea is to model the approach of a "hypothetical researcher" who has these research tasks in mind and is confronted with a variety of methods to choose from. Due to uncertainty about the appropriate method to apply in the present case, the researcher might be tempted to try different analysis strategies and pick the "optimal result" for each task. We quantify the over-optimistic bias that can arise out of choosing the "best" method in this way, by validating the optimized results on validation data (which we will define shortly). Our primary interest does not lie in any of the four specific research tasks, but rather in demonstrating the importance of validation and the necessity of avoiding questionable research practices. Through this illustrative study, we aim to raise awareness for these topics in microbiome analysis.

We now explain our usage of the terms "over-optimism", "validation", and "replication". Broadly speaking, over-optimism may result from two sources of *multiplicity*: a) multiplicity of (tested) hypotheses or b) multiplicity of analysis strategies. It is well known that *multiple testing* (i.e., testing multiple hypotheses on a dataset) can lead to false-positive results due to the accumulation of the type I-error probability. Such problems may appear in microbiome research, e.g., when testing many associations of microbiome-related variables with health-related variables and only reporting the significant results [3]. However, even when considering only a single hypothesis, the *multiplicity of analysis strategies* [6]—which we focus on in this paper—may lead to varied results and the potential for selectively reporting only the best ones. Researchers must make several choices about their analysis strategy (a mechanism known as "researcher degrees of freedom", [7]), including data preprocessing (e.g., normalization) and statistical analysis in a narrower sense. Often, multiple analysis strategies are possible and sensible, which leads to *method uncertainty* [8] because it is not necessarily clear which analysis choice is the best one. In microbiome analysis, for example, a large number of methods for estimating and analysing microbial association networks exists [9], from which the researcher must choose.

In such situations, there is a temptation for the researcher to try different methods and then pick the one that yields the best result. This approach might be considered sensible: Finding the "best" method for the data appears to be a natural goal. However, when the number of tried methods is high, there is a substantial danger of "overfitting" the analysis to the present dataset. The best-performing method might thus perform well on the data currently used, but perhaps not as well on a validation dataset due to sampling variability—in other words, the optimized result cannot be (fully) validated or replicated on the validation data. Here, we define "replication" as applying the same methods of a study to new data [4]; see [10] for a more extensive discussion of the concept of replication. "Validation", as we use it, is a broader term: A result is reappraised on a validation dataset, which may be either genuinely new data, or a dataset obtained by splitting the original data into two parts (discovery and validation data) [11]. We use the latter approach in our study.

The connection between the multiplicity of analysis strategies and over-optimism is occasionally mentioned in the literature, mostly in relation to significance testing [12]. For example, it is well known that trying different analysis choices can make it easier to find a statistically significant result [7, 13]. If the researcher does this in an intentional manner (i.e., tweaking the analysis choices sequentially until a "significant" p-value is reached), this is called *p-hacking* [14]. However, over-optimistic bias might also appear without conscious "hacking": A researcher may try different methods with the best intentions but then proceed to *selective reporting* (reporting only the method that yields the best result). Additionally, such effects do

not only pertain to significance testing, but may appear whenever the result of a statistical analysis is quantified (e.g., with a performance measure or an index value).

In this paper we focus on over-optimism in the context of unsupervised microbiome analysis, outside of the classical setting of significance testing. We illustrate the over-optimistic effects caused by the multiplicity of analysis strategies in combination with selective reporting, as quantified by the subsequent validation of the optimistic results. As exemplary data, we use OTU count data from the American Gut Project (AGP) obtained with 16S amplicon sequencing [15]. It is well known that technical variation in amplicon sequencing (e.g., batch effects with respect to different labs or different machines) or using different methods for clustering sequences to obtain OTUs may lead to variation in the generation of the OTU count data and the results of subsequent statistical analysis [4, 16–18]. In the present work, however, we focus on the multiplicity of the statistical analysis methods (starting from the processed OTU count table), which has received somewhat less attention than multiplicity stemming from different technical methods. Recently, some studies have highlighted that different statistical analysis methods or modeling strategies may yield inconsistent results, namely in the context of microbiome-disease association modeling [19], microbiome differential abundance methods [20], and analyzing microbiome intervention design studies [21]. In contrast to these studies, a) we focus on the multiplicity of *unsupervised* statistical methods, i.e., methods for network learning and clustering, and b) our main goal is not to compare the results of different methods, but rather to quantify over-optimism effects that stem from picking the "best" result. The range of the statistical methods we consider includes 1) normalization to make read counts comparable across samples and to account for compositionality (if required by the subsequent analysis steps), 2) estimation of microbial networks, sample networks, and (dis)similarity matrices, and 3) methods to further process the network/(dis)similarity information such as clustering.

The key idea of our illustrative study consists of splitting the whole dataset into a discovery and a validation set, trying out different methods for each of these three analysis steps on the discovery data, choosing the combination of methods that yields the best result on the discovery data according to an evaluation criterion, and applying this combination to the validation data to check whether the evaluation criterion takes a similar value. Fig 1 gives an overview of this approach, which we now describe in more detail.

We use four exemplary "research tasks" to illustrate the effects of the multiplicity of analysis strategies. Imagine a researcher who wishes to perform an unsupervised analysis of microbiome data. Even though the analysis is unsupervised and might be performed for exploratory purposes, the researcher usually has some hopes for the results. While these expectations could be vague at first, the researcher might eventually focus on a concrete evaluation criterion that represents these hopes in order to judge the results. The researcher tries different statistical methods and chooses the method that yields the best result according to the evaluation criterion. We now detail the four research tasks, the hopes that our hypothetical researcher might have, and the concrete evaluation criteria they might use (and which we therefore choose for our illustrative study):

1. **Clustering of bacterial genera:** Bacterial genera can be clustered based on their associations such that highly associated genera are likely to belong to the same cluster. Hence, the assignment of two genera to the same cluster indicates shared variation over the samples, which in turn might suggest a shared functionality. We assume that the hypothetical researchers hopes to find a clustering of bacterial genera that yields good agreement with the taxonomic categorization of the genera into families. As concrete evaluation criterion, we choose the Adjusted Rand Index (ARI, [22]), a measure for comparing two partitions, normalized for chance agreement. The ARI ranges in $[-1, 1]$, with higher values indicating

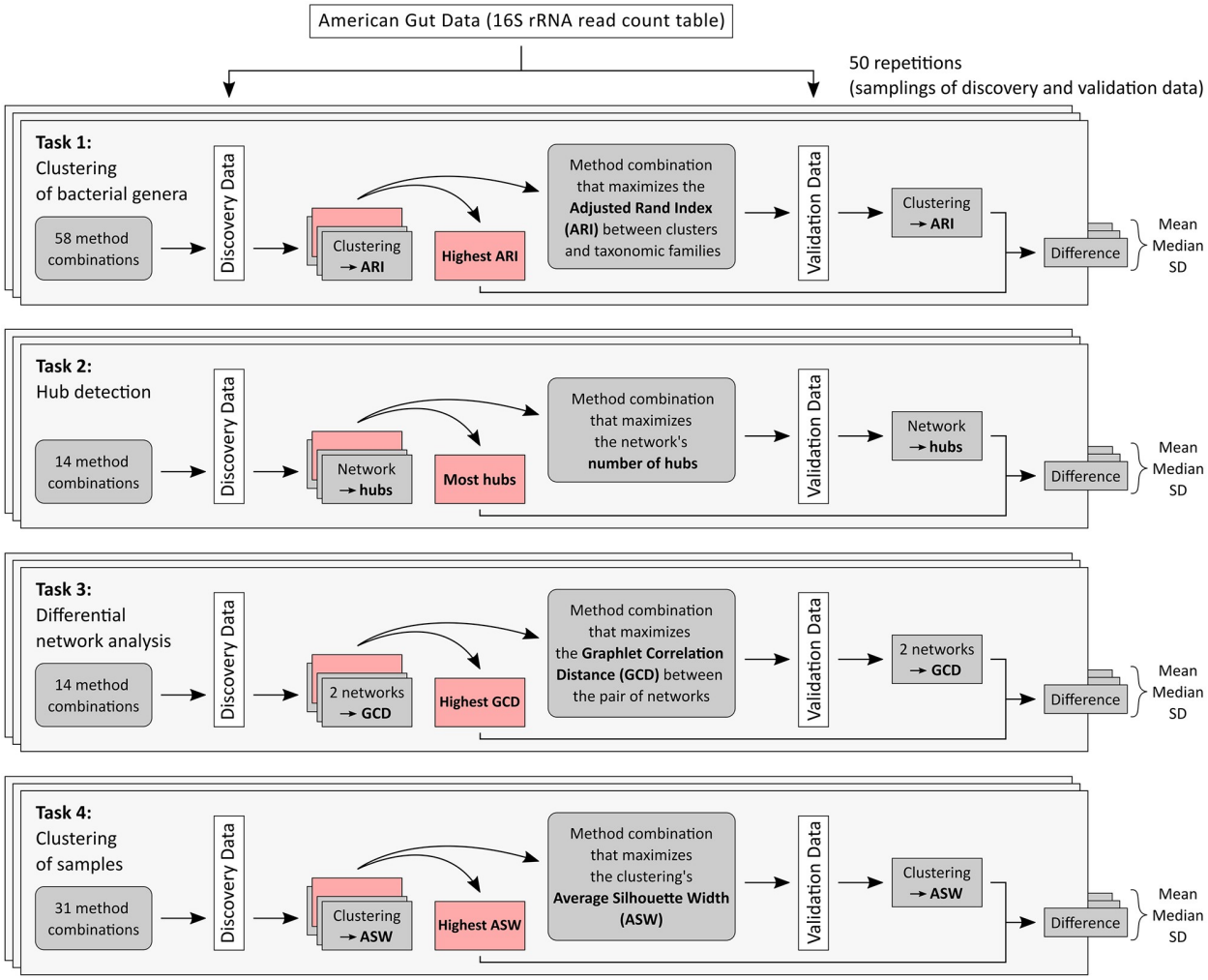


Fig 1. Graphical overview of our study. The process of drawing 50 samplings of discovery and validation data is repeated for different sample sizes: $n \in \{100, 250, 500, 1000, 4000\}$ for tasks 1 and 2, $n \in \{100, 250, 500\}$ for task 3, and $n \in \{100, 250, 500, 1000, 3500\}$ for task 4.

https://doi.org/10.1371/journal.pcbi.1010820.g001

higher similarity of the partitions. In this research task, one partition is given by the clustering as calculated by the researcher, the other one by the taxonomic categorization of the genera into families. The higher the ARI (i.e., the closer to 1), the more similar the calculated clustering is to the taxonomic categorization, which indicates a "better" clustering. While it is typically not realistic to find a clustering that is *perfectly* aligned with the taxonomic categorization (i.e., where the ARI is equal to 1), some agreement with the taxonomy is often considered as a good property of a bacterial clustering [23]. While we perform the clustering at the genus level, the same logic would apply at any taxonomic level. This remark also holds for the other research tasks.

2. **Hub detection:** A researcher might hope to find a microbial network with interesting keystone taxa (also called "microbial hubs"), i.e., highly connected taxa which are assumed to have a strong impact on the rest of the network. Detecting and analyzing keystone taxa in order to better understand microbial interactions has become popular in recent years [24–26]. Taxa that are identified as hubs based on network centrality measures (see Section 4.3.2

for details) are not automatically *biologically* important keystone taxa [25]. Still, hub detection can serve as a starting point to carry out further analyses about the role of the detected hubs [27]. For example, a recent study [28] analyzed microbiome data from aquatic environments where many microbes are "unknown taxa", i.e., uncharacterized. The authors generated microbial networks and performed hub detection. Frequently, the detected hubs were unknown taxa, which in turn serves to prioritize these specific taxa for further analyses.

In our illustrative example, we assume that our hypothetical researcher is interested in generating as many interesting hypotheses and directions for further research as possible. Therefore, we assume that the researcher chooses a method that yields a relatively high number of hubs, to maximize the "hubbiness" of the network. Thus, the number of hubs is used as the concrete evaluation criterion. Of course, other criteria to choose an "interesting" network with hubs are also feasible.

3. **Differential network analysis:** Microbiome researchers are often interested in the effects of treatments, such as antibiotics, on the gut microbial community (see, e.g., [29, 30] for background). When generating microbial association networks for two groups (one for persons who did not take antibiotics in the last year, and one for persons who took antibiotics in the last month), a researcher might expect that the networks (as proxies for microbial community structure) potentially change. As concrete evaluation criterion we measure the dissimilarity between the networks with the Graphlet Correlation Distance (GCD) between the networks [31]. The method that yields the largest GCD between the two networks is chosen. The GCD has been used in previous studies to compare microbial networks [32–34].

4. **Clustering of samples:** The three previous research tasks are all based on associations between microbes. In contrast, the fourth task focuses on similarities between *samples* (individuals). The goal is to find a clustering of samples such that samples within the same cluster have a similar bacterial composition, while the composition differs between samples of different clusters. This task is inspired by the popular concept of "enterotypes". In 2011, a study [35] argued that individuals can be clustered into three distinct groups which represent different gut microbiome types (enterotypes). Whether enterotypes truly exist (and if they do, how many there are) has since become a topic of controversial discussion [36–40]. Some studies have already noted that using different methods for clustering the samples (e.g., different methods for calculating the similarities between the samples) may lead to different enterotype results [37, 41]. However, to the best of our knowledge, the relation between the multiplicity of analysis strategies and over-optimism has not yet been explicitly studied. For this exemplary research task, we assume that the hypothetical researcher is interested in finding enterotypes in the AGP dataset. As concrete evaluation criterion, we use the Average Silhouette Width (ASW [42]). The ASW is a cluster validation index that measures the homogeneity as well as the separation of the clusters. The index ranges in $[-1, 1]$, with higher values indicating a better clustering. The ASW has been previously used in enterotype studies to evaluate the quality of sample clusterings [35, 37, 41].

For each of the four research tasks, we imitate our "hypothetical researcher" by trying different methods (i.e., methods for estimating microbial networks, calculating similarities between samples, and/or clustering) and looking for the best result. The hypothetical researcher might stop at this point, and only report the best result according to the respective criterion. In contrast, we are interested in whether the best result can be confirmed on *validation data*: The result obtained by the "best" method on the discovery data (i.e., the "best" ARI, number of hubs, GCD, or ASW, respectively) is compared with the result obtained by this

method on the validation data. The discovery and validation datasets are obtained by randomly sampling two disjoint subsets from the full AGP dataset, a process which is repeated multiple times.

Note that our analysis serves only illustrative purposes to study over-optimism effects. It is not our aim to systematically evaluate or compare the chosen method combinations. Moreover, we do not claim that researchers typically apply multiple methods to a dataset as systematically as we do this here, nor that they "optimize" for the best method with malicious intent. Nevertheless, during a longer research process, researchers will often try multiple methods on a dataset, and even if this happens with the best intentions, it might still cause over-optimism effects.

So far, we have spoken of imitating the behavior of a single hypothetical researcher or research team. Our study might also be interpreted as modeling the behavior of *multiple* research teams. Each team tries a different analysis strategy and only the team with the "best" result is able to publish their findings (e.g., due to publication bias).

We present the results of our analysis in Section 2. Section 3 contains a discussion. In Section 4, we give a detailed overview of the exemplary dataset, our study design, and the different statistical methods that we applied to the discovery data.

2 Results

2.1 Quantifying over-optimism effects

For each research task, we drew discovery and validation sets (each with sample size n) of varying sizes: $n \in \{100, 250, 500, 1000, 4000\}$ for the first two research tasks, $n \in \{100, 250, 500\}$ for the third research task, and $n \in \{100, 250, 500, 1000, 3500\}$ for the fourth research task. For the third task, the maximal sample size was reduced due to the required information about antibiotics usage. For the fourth task, the maximal sample size was 3500 instead of 4000 because only samples from adults were kept for the analysis. More details are given in Section 4.2.

For each n, the process of drawing discovery and validation sets was repeated 50 times. As sampling variability decreases with increasing n, the performances of a method on both discovery and validation data should become more and more similar. We thus expected over-optimistic effects to decrease with increasing n.

For each research task, we applied multiple method combinations to the discovery data. For the first three research tasks which were based on microbial associations, this involved *normalization methods* (clr [43], mclr [44], and VST [45]), *association estimation* (Pearson correlation, Spearman correlation, latentcor [46], SPRING [44], and proportionality [47]), *sparsification* (*t*-test, threshold method, and neighborhood selection), and, for the first research task, *clustering* (hierarchical clustering, spectral clustering [48], fast greedy modularity optimization [49], the Louvain method for community detection [50], and manta [51]). For the fourth research task where samples were clustered based on their similarities, we applied *normalization methods* (clr, mclr, and VST), *similarity calculation* (Aitchison distance [52], Euclidean distance, compositional Kullback-Leibler divergence (cKLD) [53], and Bray-Curtis dissimilarity [54]), *sparsification* (threshold method, *K*-nearest neighbors), and *clustering* (Dirichlet multinomial mixtures (DMM) [55], spectral clustering, partitioning around medoids (PAM) [56], fast greedy modularity optimization, and the Louvain method for community detection). Detailed descriptions of the combinations are given in Section 4.3.

Supplementary figures in the Supporting Information S1, S2, S3 and S4 Text show the results of applying the different method combinations to the discovery data for the varying sample sizes (task 1: Fig A-J in S1 Text, task 2: Fig A-E in S2 Text, task 3: Fig A-C in S3 Text, task 4: Fig A-J in S4 Text). Notably, there is some change in the selected "best" method

combination with respect to sample size. In particular, the performance of the sparsification methods is dependent on the sample size. These results are discussed in detail in S1, S2, S3 and S4 Text.

Our main interest lies in choosing the method combination that yields the maximum value of the evaluation criterion (ARI, number of hubs, GCD, and ASW) on the discovery data, applying it to the validation data, and checking whether the values of the evaluation criteria can be validated. Over-optimism is indicated if the value of the evaluation criterion is lower on the validation data compared to the result on the discovery data. Exemplary results for $n = 250$ are shown in Fig 2 (research tasks 1 & 2) and Fig 3 (research tasks 3 & 4). The corresponding figures for all other sample sizes n are given in the Supporting Information (task 1: Fig K-O in S1 Text, task 2: Fig F-J in S2 Text, task 3: Fig D-F in S3 Text, task 4: Fig K-O in S4 Text).

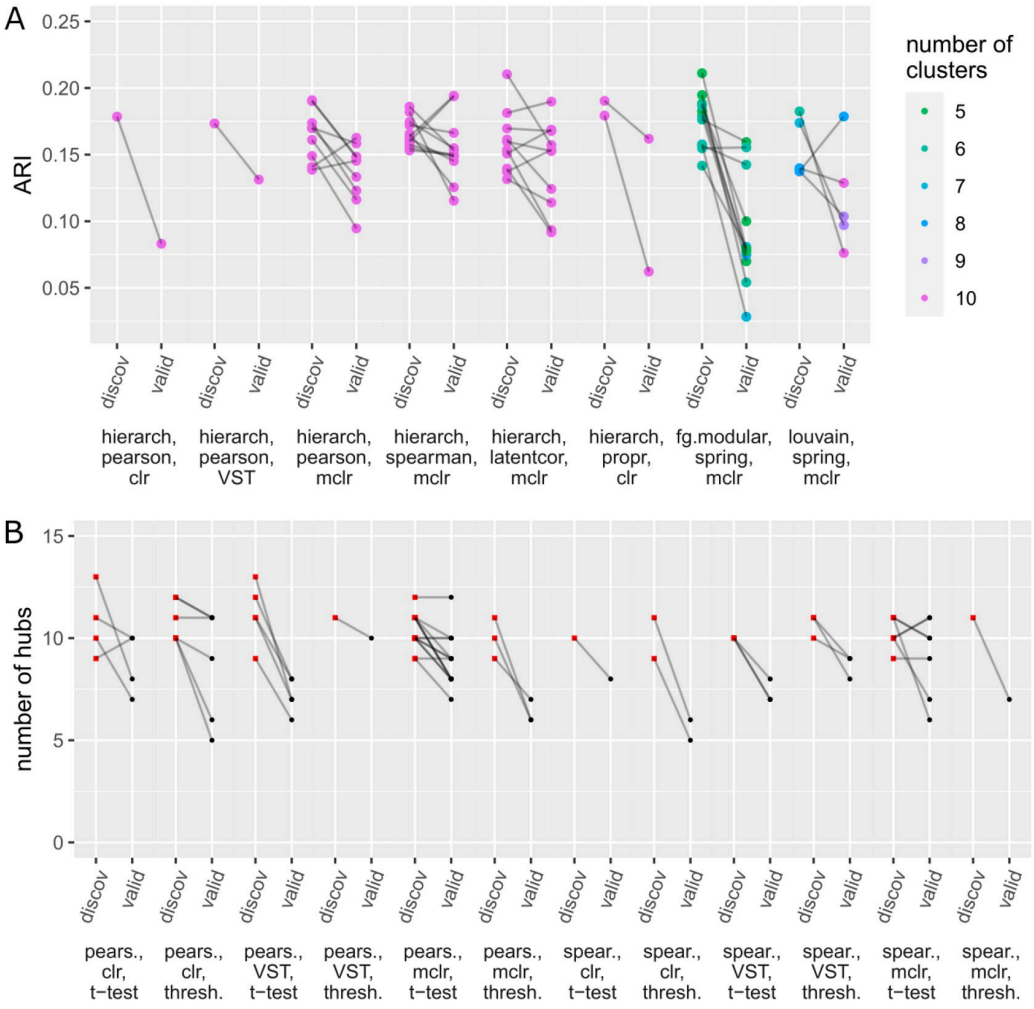


Fig 2. Research tasks 1 & 2: For $n = 250$, values of the evaluation criteria resulting from the "best" method combinations on the discovery data are compared to the corresponding results on the validation data. On the x-axis, the method combinations that performed best in at least one of the 50 samplings are shown. For each of the 50 samplings, the value of the evaluation criterion on the discovery data (belonging to the best method combination) and the corresponding value on the validation data are connected by a line, resulting in 50 lines overall. As the lines are slightly transparent, overlapping lines appear in a darker shade. a) ARI values for the task of clustering bacterial genera, b) numbers of hubs for the hub detection task.

https://doi.org/10.1371/journal.pcbi.1010820.g002

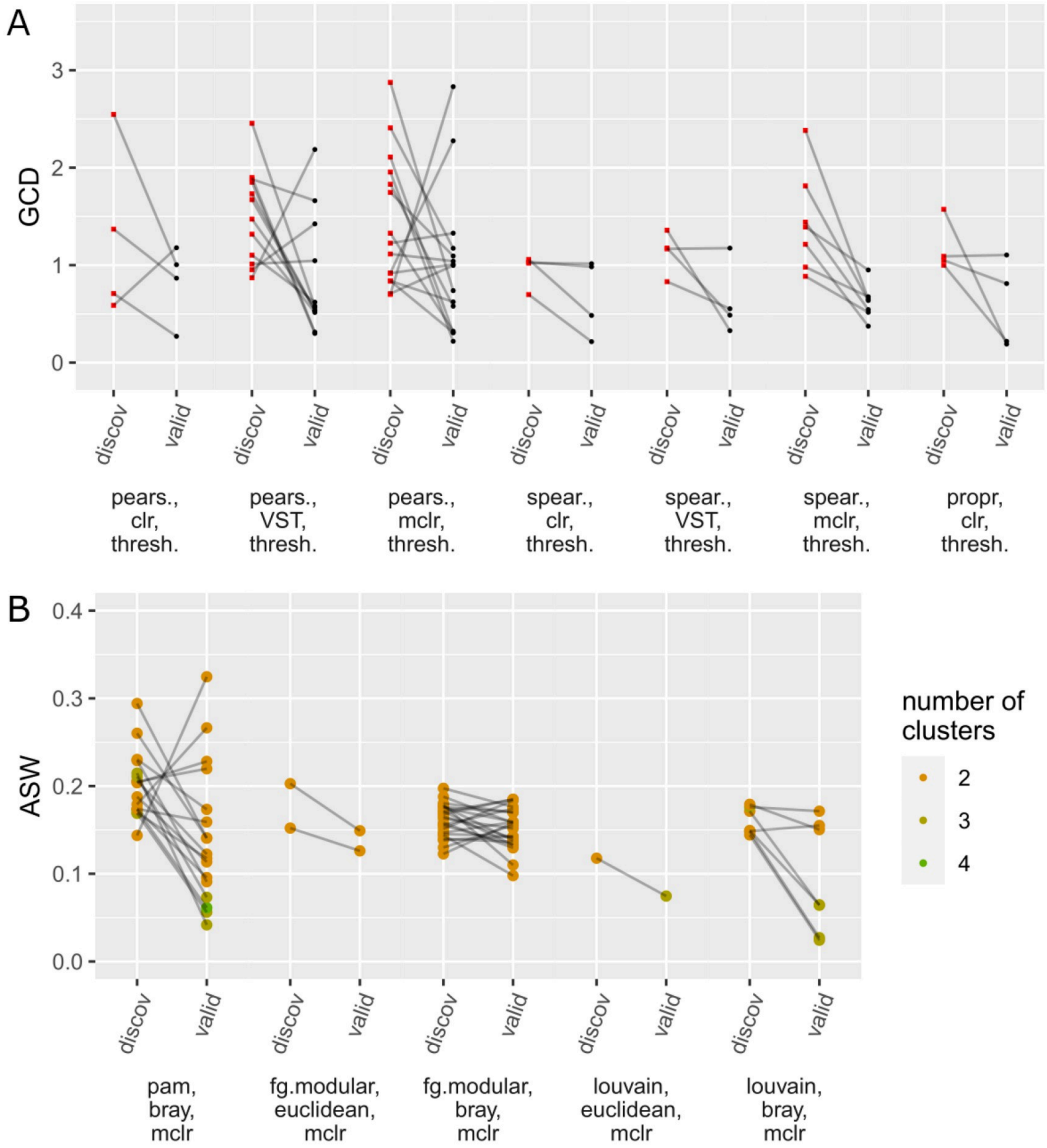


Fig 3. Research tasks 3 & 4: Analogously to Fig 2 (see the description there), values of the evaluation criteria are compared between discovery and validation data for $n = 250$. a) GCD values for the differential network analysis task, b) ASW values for the task of clustering samples.

https://doi.org/10.1371/journal.pcbi.1010820.g003

On the *x*-axis, only the method combinations that performed best in at least one of the 50 samplings are shown (that is, *not all* tried method combinations; the method combinations that did not perform best in at least one of the samplings do not appear in the plot because these were never applied to the validation data). For each sampling, the value of the evaluation criterion on the discovery data (belonging to the best method combination) and the corresponding value on the validation data are connected by a line. For the first and fourth task, the dots representing the ARI/ASW values are colored according to the number *k* of clusters in the respective clustering result. Details about the procedures for determining *k* are given in Sections 4.3.1 and 4.3.4. For the other two research tasks, the results are shown as red squares for the discovery data and black dots for the validation data.

Table 1. For research tasks 1 and 2: Mean, median, and standard deviation (over 50 samplings of discovery/validation data) of the difference (both unscaled and scaled) between the value of the evaluation criterion on the validation data and the corresponding value on the discovery data. Additionally, the effect size (mean divided by standard deviation) is reported. ARI_{discov} denotes the best ARI on the discovery data and ARI_{valid} the ARI resulting from the corresponding method combination on the validation data. The quantities $\#hubs_{discov}$, $\#hubs_{valid}$ (number of hubs) are defined analogously.

Research task 1: clustering of bacterial genera

	$ARI_{valid} - ARI_{discov}$				$\frac{ARI_{valid} - ARI_{discov}}{ARI_{discov}}$			
n	mean	median	sd	mean/sd	mean	median	sd	mean/sd
100	-0.054	-0.046	0.044	-1.22	-30.0%	-26.7%	24.1%	-1.24
250	-0.039	-0.035	0.046	-0.84	-22.0%	-21.8%	26.3%	-0.84
500	-0.038	-0.037	0.038	-1.01	-21.4%	-20.6%	21.6%	-0.99
1000	-0.042	-0.035	0.037	-1.13	-23.7%	-20.1%	20.3%	-1.16
4000	-0.035	-0.033	0.035	-1.00	-19.0%	-18.3%	18.8%	-1.01

Research task 2: hub detection

	$\#hubs_{valid} - \#hubs_{discov}$				$\frac{hubs_{valid} - hubs_{discov}}{hubs_{discov}}$			
n	mean	median	sd	mean/sd	mean	median	sd	mean/sd
100	-2.44	-3	2.35	-1.04	-21.6%	-24.0%	21.3%	-1.02
250	-2.18	-2	1.78	-1.22	-20.5%	-20.0%	16.5%	-1.24
500	-2.12	-2	1.88	-1.13	-20.8%	-20.0%	17.9%	-1.16
1000	-1.64	-2	1.52	-1.08	-16.3%	-18.2%	15.4%	-1.06
4000	-1.12	-1	1.32	-0.85	-11.5%	-11.1%	13.8%	-0.83

https://doi.org/10.1371/journal.pcbi.1010820.t001

The lines point downwards in most cases, i.e., the results for the validation data are usually slightly worse than for the discovery data. This indicates over-optimism effects. To further quantify these effects, Table 1 (tasks 1 & 2) and Table 2 (tasks 3 & 4) show the mean, median, and standard deviation of the difference as well as the scaled difference between the value of the evaluation criterion on the validation data and the value on the discovery data (over the 50 samplings of discovery/validation data). While it might be interesting to test the differences between discovery and validation data for significance (to assess whether the results on the

Table 2. For research tasks 3 and 4: Mean, median, and standard deviation (over 50 samplings of discovery/validation data) of the difference (both unscaled and scaled) between the value of the evaluation criterion on the validation data and the corresponding value on the discovery data. Additionally, the effect size (mean divided by standard deviation) is reported. GCD_{discov} denotes the largest GCD on the discovery data and GCD_{valid} the GCD resulting from the corresponding method combination on the validation data. The quantities ASW_{discov}, ASW_{valid} (average silhouette width) are defined analogously.

Research task 3: differential network analysis

	$GCD_{valid} - GCD_{discov}$				$\frac{GCD_{valid} - GCD_{discov}}{GCD_{discov}}$			
n	mean	median	sd	mean/sd	mean	median	sd	mean/sd
100	-0.481	-0.463	0.829	-0.58	-25.0%	-30.0%	55.5%	-0.45
250	-0.555	-0.516	0.856	-0.65	-26.7%	-52.8%	72.5%	-0.37
500	-0.305	-0.417	0.605	-0.50	-18.6%	-45.1%	63.5%	-0.29

Research task 4: clustering of samples

	$ASW_{valid} - ASW_{discov}$				$\frac{ASW_{valid} - ASW_{discov}}{ASW_{discov}}$			
n	mean	median	sd	mean/sd	mean	median	sd	mean/sd
100	-0.055	-0.043	0.088	-0.63	-20.0%	-22.6%	35.0%	-0.57
250	-0.036	-0.027	0.065	-0.55	-18.3%	-16.0%	36.8%	-0.50
500	-0.020	-0.017	0.041	-0.48	-10.6%	-10.1%	24.7%	-0.43
1000	-0.019	-0.002	0.039	-0.48	-11.2%	-1.6%	25.5%	-0.44
3500	-0.010	-0.010	0.017	-0.58	-7.2%	-8.0%	13.3%	-0.54

https://doi.org/10.1371/journal.pcbi.1010820.t002

validation data are "significantly worse"), a suitable procedure for that purpose has not yet been proposed, to the best of our knowledge, and would need to be explored in further work. For cluster analysis, challenges related to this issue have been recently discussed [11]. Instead of calculating *p*-values, we report the "effect size" (mean divided by standard deviation) in Tables 1 and 2.

As expected, the means and medians of the differences are negative for all four research tasks and all sample sizes, demonstrating that the results on the discovery data were somewhat over-optimistic. The effect sizes (mean divided by standard deviation) are notable for all research tasks, albeit slightly smaller for the third and fourth research task. We now discuss the behavior of the average differences over the varying sample sizes n in more detail for each research task in turn.

Research task 1 (clustering of bacterial genera): The average absolute decline of the ARI on the validation data is not drastic, but when considering the scaled difference, the ARI is reduced on the validation data by about 20–30% on average. Note that the absolute value of the mean/median ARI difference (both unscaled and scaled) is largest for $n = 100$, and smallest for $n = 4000$. This fits with our previously mentioned hypothesis that over-optimism effects are less pronounced when n is large. However, between 100 and 4000, there is no clear linearly decreasing tendency in the absolute mean/median ARI differences. Moreover, there is no clear tendency with respect to the effect sizes.

Research task 2 (hub detection): The absolute values of the means and medians of the differences tend to decrease with increasing sample size. Again, this fits with our hypothesis that the over-optimistic bias decreases with increasing n. This tendency also largely holds for the effect sizes, although the absolute value of the effect size is slightly larger at $n = 250$ compared to $n = 100$, due to the larger standard deviation at $n = 100$.

Research task 3 (differential network analysis): The absolute values of the means and medians do not monotonically decrease with increasing n: for $n = 250$, these are slightly larger than for $n = 100$. This is perhaps due to the fact that the sampling variability is still rather large at $n = 250$. At $n = 500$, however, the over-optimism effect appears to decrease, as evidenced by the drops in the absolute values of the average differences (both unscaled and scaled). For even higher sample sizes, we would expect to see a continuing decline of the over-optimistic bias, although we cannot confirm this due to the limited data availability.

Research task 4 (clustering of samples): Similar to the first research task, the average absolute decline of the evaluation criterion (here, the ASW) on the validation data is not drastic. When considering the relative decline, the ASW values decrease on the validation data by about 20% on average for smaller sample sizes. Over-optimistic bias tends to be less pronounced for larger sample sizes. With respect to the median differences and effect sizes, the bias slightly increases again at the largest sample size of $n = 3500$, but the mean and median differences are quite small.

We not only analyzed the relation of over-optimistic bias with the sample size, but also expected over-optimistic bias to decrease if fewer method combinations were tried. To investigate this hypothesis, we repeated our analyses with a reduced number of method combinations: five instead of 58 for the first research task, three instead of 14 for the second and third research tasks, and five instead of 31 for the fourth research task. The chosen subsets of combinations as well as the results are described in detail in the Supporting Information S5 Text. The means and medians of the differences mostly remain negative for the different research tasks and sample sizes (indicating that some over-optimistic bias still exists), but as expected, the absolute values of the mean/median differences as well as the effect sizes tend to be smaller. This supports our hypothesis that over-optimistic bias is more pronounced the more method

Table 3. Mean, median, and standard deviation of ARI_{stab}, i.e., the ARI between the clusterings of bacterial genera on discovery and validation data, over 50 samplings of discovery/validation data.

	ARI_{stab}		
n	**mean**	**median**	**sd**
100	0.361	0.329	0.111
250	0.509	0.491	0.166
500	0.604	0.574	0.168
1000	0.600	0.568	0.166
4000	0.763	0.792	0.140

https://doi.org/10.1371/journal.pcbi.1010820.t003

combinations are tried. Of course, the exact amount of over-optimistic bias still depends on the chosen (subset of) method combinations.

2.2 Additional stability analyses

While our main focus was to compare the "best" result on the discovery data to the corresponding result on the validation data (with respect to the evaluation criteria), we also report some additional stability results for the first two research tasks to further demonstrate that the methods do not necessarily yield stable results on discovery vs. validation data. For the task of clustering bacterial genera, we compared the clusterings on discovery vs. validation data with the ARI (while the agreement with the taxonomic categorization was ignored). This measure is denoted as ARI_{stab}. The results are reported in Table 3. For the hub detection task, we compared the sets of hubs on discovery vs. validation data with the Jaccard index (on the genus level) and cosine similarity index (on the family level), as reported in Table 4. The indices are described in more detail in Section 4.3.

For the clustering task, Table 3 shows that for smaller sample sizes, the mean ARIs are rather far away from 1, which indicates notable differences between the clusterings of the bacteria based on discovery vs. validation data. The clusterings tend to become more similar with increasing sample size, but even for $n = 4000$, the mean ARI of about 0.8 indicates that the clusterings are still different to some extent. This shows that the chosen clustering on the discovery data is not necessarily stable regarding cluster memberships when the result is validated on the validation data.

For the hub detection task, Table 4 demonstrates that the sets of hubs can be quite different between discovery and validation data, as measured with the Jaccard index (which ranges between 0 and 1). For smaller sample sizes, the similarity is particularly small. The Jaccard values increase with increasing sample size, but even at $n = 4000$, a mean value of about 0.7 shows that there are still notable dissimilarities between the sets of hubs. For the similarity on *family*

Table 4. Mean, median, and standard deviation (over 50 samplings of discovery/validation data) of a) the Jaccard index which compares the set of hubs obtained on the discovery data with the set of hubs on the validation data, and b) the cosine similarity which compares these sets of hubs, but on the level of families.

	Jaccard			Cosine similarity		
n	**mean**	**median**	**sd**	**mean**	**median**	**sd**
100	0.236	0.250	0.109	0.881	0.911	0.112
250	0.359	0.357	0.119	0.922	0.955	0.078
500	0.443	0.429	0.116	0.948	0.969	0.060
1000	0.546	0.538	0.139	0.946	0.974	0.068
4000	0.709	0.727	0.147	0.975	0.984	0.026

https://doi.org/10.1371/journal.pcbi.1010820.t004

level, we expected higher values (given that two hubs from the same family which differ on the genus level are counted as not equal for the Jaccard index and as equal for the cosine similarity). Indeed, the values of the cosine similarity (which ranges between -1 and 1), are generally quite high. Therefore, if one only interprets the hubs on family level (e.g., with respect to typical functions of the bacterial families), there is less danger of instability between discovery and validation data, compared to an interpretation on genus level.

We repeated the stability analyses with reduced numbers of tried methods combinations as described in the previous section. The results are reported in the Supporting Information S5 Text. Overall, the stability results are rather similar to the ones obtained with the full sets of method combinations.

3 Discussion

We have quantified over-optimism effects resulting from the multiplicity of analysis strategies coupled with selective reporting, using four exemplary microbiome research questions. Our results indicate an over-optimistic bias for all four research tasks. That is, when choosing the "best" method on the discovery data according to the maximization of an evaluation criterion, this criterion then tends to attain lower ("worse") values on the validation data when the same method is applied. The exact size of the over-optimistic bias depends on the research task and sample size. Generally speaking, the over-optimistic bias tends to be more pronounced at smaller sample sizes, although the relation between sample size and optimistic bias is not always strictly monotonically decreasing in our analyses. Moreover, the over-optimistic bias also depends on the number of tried method combinations. When we tried fewer combinations, we still detected some over-optimistic bias, but the bias was less pronounced.

Additional stability analyses for the first two research tasks have illustrated that clustering solutions and sets of hubs—which have been yielded by a method on discovery data—do not necessarily remain stable when the same method is applied to validation data.

In summary, our study has demonstrated that the issue of over-optimism and instability of results goes beyond the context of statistical testing and fishing for significance, and pertains to unsupervised analysis strategies as well.

The number of tried method combinations in the analyses with all combinations (58 for the clustering of bacterial genera, 14 for hub detection and differential network analysis, 31 for the clustering of samples) may seem quite large for a single researcher to attempt. However, we would argue that these numbers are not that unrealistic. The method combinations are not independent of each other. Rather, the combinations are obtained by varying methods along the analysis pipeline (e.g., the type of sparsification). Modern software packages make it very easy to quickly switch from one method choice to another. Moreover, as mentioned in the introduction, our study might also be interpreted as modeling the behavior of *multiple* research teams. Large public datasets, such as the AGP data, are studied by many researchers. While a single researcher or research team might only try a few analysis strategies, the strategies tried by multiple teams could sum up to a much larger number.

In order to quantify over-optimism, we deliberately split a single dataset into two parts instead of using an independent dataset as validation data. With the latter approach, we could not have determined whether worse performance on the validation data indeed stemmed from the multiplicity of analysis strategies combined with selective reporting (which is the focus of our work), or was simply due to substantial differences between discovery and validation data (e.g., different populations). Of course, beyond the context of our study, using external data is generally important to check the validity and generalizability of results.

A constraint of our study is that for each research task in turn, we translated expectations of the "hypothetical researcher" into a single fixed evaluation criterion. Of course, researchers might have various expectations and thus multiple criteria in mind. On the one hand, it is likely more difficult for researchers to find a result that is simultaneously good with respect to *multiple* criteria, thus potentially reducing over-optimism effects. On the other hand, considering multiple criteria might allow researchers to pick one or a few criteria based on obtaining good results. This constitutes another source of multiplicity (adding to the sources of multiplicity considered in the present study), which in turn might increase over-optimistic bias. It would be interesting to analyze the effects of considering multiple criteria in future work.

Over-optimism can lead to unreliable results and might ultimately hinder research progress. We now discuss some strategies which may help researchers avoid over-optimistic bias in their application studies.

As illustrated by our analyses with a reduced number of method combinations, over-optimistic bias tends to decrease if fewer methods are tried. Therefore, the first option is to reduce the multiplicity of analysis strategies *before* the start of the analysis. Researchers should carefully consider which method is most suitable for their application. Here, guidance from *neutral comparison studies* can be relevant. Such studies compare existing methods (instead of introducing a novel method), and the authors of the study are neutral, i.e., they do not have a vested interest in a particular method showing better performance than the others and are as a group approximately equally familiar with all considered methods. We refer to [57, 58] for a more detailed discussion of this concept. It would be desirable if more neutral comparison studies were published in the context of methodological research on microbiome analysis. For example, two recent studies already provide such a welcome effort in the context of microbial differential abundance testing [20, 59], and guidelines for benchmarking microbiome analysis methods have been proposed as well [60].

An additional strategy is *preregistration* of the researchers' analysis plan. Preregistering refers to defining the research hypotheses and analysis plan, and posting this plan to a registry, *before* observing the results. This concept has gained plenty of attention in recent years [61]. Once their analysis plan is registered, researchers might shy away from trying many other analysis strategies and selectively reporting only the best results.

However, preregistration might not always be possible or sensible: for example, in exploratory research, researchers typically cannot pin down the exact analysis strategy in advance, and trying out different methods sequentially is quite natural [4]. Indeed, unsupervised analysis methods, on which we have focused in our study, are often used for exploratory purposes. In such cases, when the multiplicity of analysis strategies cannot be avoided, researchers should honestly report that their study is exploratory and that multiple methods were tried. They should not present their analyses as if a single analysis pipeline was fixed in advance, nor should they report only the "best" results.

In general, we would advise researchers to use validation data to validate their results whenever possible. While we have included validation data in our study to quantify over-optimism effects, researchers can also use validation data in their applied research, to check whether the best results on the discovery data still hold on the validation data. This is particularly relevant when the multiplicity of possible analysis strategies cannot be reduced beforehand, e.g., in the absence of relevant neutral comparison studies for the methods of interest. For the topic of cluster analysis(research tasks 1 and 4), different strategies for validating clustering results on validation data have been previously discussed in detail [11]. More awareness for the importance of validation data has also emerged in microbiome research (see, e.g., in the context of supervised analysis [62, 63] and large-scale cohort studies [64]). Using validation data does not directly *prevent* over-optimism on the discovery data, but helps to *detect* over-optimistic

results. The evaluation on the validation data can be considered as a more realistic assessment of the quality of the result, thus correcting for over-optimistic bias.

Sometimes, validation data is not available, e.g., because the dataset is too small to be split into discovery and validation sets, and a suitable independent validation set does not exist. For such cases, it would be interesting to find other indicators of potential over-optimism. Researchers might check, for instance, whether the results from the different tested methods coincide. Similarity of the results indicates robustness with respect to method choice. However, lack of robustness does not automatically imply that the results (or the "best" result) will also be over-optimistic, in the sense that they cannot be validated on validation data. Vice versa, if the results are robust, it is not entirely clear to which extent this is an indicator of nonexistent or small over-optimistic bias (although a reduced extent of over-optimism might be somewhat likely because obtaining very similar results would not allow researchers to pick a single result that is notably better than the other ones). It might be interesting to study the relation between robustness and replicability on validation data in further work.

The present study does not aim at systematically evaluating the performance of any chosen method combination. In particular, we do not give recommendations about which methods to use. In future research, it might be interesting to explore whether the design used in this study could be adapted to method evaluation and comparison. More precisely, one might repeatedly sample discovery and validation datasets as in our study, and evaluate methods based on whether they a) have a good performance on the discovery data and b) have a similar performance on the validation data, i.e., do not tend to overfit to the discovery data.

In summary, we hope that our study helps raise awareness of the important problem of over-optimism in microbiome research, and that it motivates more widespread implementation of strategies to avoid over-optimistic bias. If researchers adhere to good research practices, the results of microbiome analyses will likely become more reliable and replicable in the future.

4 Materials and methods

4.1 Dataset

We used data from the American Gut Project [15], a large citizen-science initiative. The project collected (mainly) fecal samples from participants in the United States, United Kingdom, and Australia. The researchers also collected metadata on the participants, e.g., health status, disease history, and lifestyle variables. Bacterial abundances were obtained using high-throughput amplicon sequencing, targeting the V4 region of the 16S rRNA marker gene with subsequent variant calling.

We downloaded an OTU count table for unrarefied bacterial fecal samples (dating from 2017) from the project website http://ftp.microbio.me/AmericanGut/ag-2017-12-04/, together with metadata about the samples. The OTU count table originally contained $p = 35511$ OTUs and $N = 15148$ samples. Following [23], we performed three preprocessing steps: 1) removing samples with a sequencing depth of less than 10000 counts, 2) removing OTUs which were present in less than 30% of the remaining samples, 3) removing 10% of the remaining samples, namely the samples with a sequencing depth under the 10%-percentile. The resulting OTU count table comprises $p = 531$ OTUs and $N = 9631$ samples.

For all four research tasks, the analysis was performed on the taxonomic rank of genera, to which the data were agglomerated. OTUs with unknown genus were assigned their own individual genus, which resulted in $p = 323$ genera overall.

4.2 Sampling of discovery and validation datasets

We obtained discovery and validation datasets by randomly sampling two disjoint subsets from the full AGP dataset. For each research task, the process of sampling discovery and validation data was performed along the *samples* of the AGP data (i.e., the subjects), not along the bacteria. This is because in each task, the bacteria formed a fixed set of entities of specific interest. This set thus remained constant for both discovery and validation data. For clustering, this is discussed in more detail in [11].

Discovery and validation sets (each with sample size n) were drawn of varying sizes: $n \in \{100, 250, 500, 1000, 4000\}$ for the first two research tasks (clustering of bacterial genera and hub detection), $n \in \{100, 250, 500\}$ for the third research task (differential network analysis), and $n \in \{100, 250, 500, 1000, 3500\}$ for the fourth research task (clustering of samples). For differential network analysis, the maximal sample size was reduced because we only considered samples that did not take antibiotics in the last year as well as samples that took antibiotics in the last month. There were 6901 samples that fulfilled these criteria. Moreover, the sampling was stratified according to antibiotics use; for discovery and validation data each, we drew $n/2$ samples that did not take antibiotics in the last year and $n/2$ samples that took antibiotics in the last month. Because there are only 544 persons who took antibiotics in the last month, the maximum n is reduced to 500. For sample clustering, the maximum n is 3500 instead of 4000 because we only kept samples from adults between ages 20–65 (7145 samples overall). We focused on this age group because previous studies have shown that the composition of the gut microbiome varies across age [65–67], with potentially more extreme "enterotypes" in children and the elderly [40, 68].

4.3 Methods for unsupervised microbiome analysis

In this section, we discuss which method combinations were applied to the discovery data, and how the results were evaluated on the validation data.

4.3.1 Research task 1: Clustering bacterial genera. We varied different steps of the cluster analysis process, resulting in 58 method combinations that were tried on the discovery data. In this section we explain how the 58 combinations were obtained.

We used cluster algorithms from two categories. Algorithms from the first category are based on (dis)similarity matrices: hierarchical clustering and spectral clustering [48]. Algorithms from the second category are based on networks with weighted edges: fast greedy modularity optimization [49], the Louvain method for community detection [50], and the manta algorithm [51].

To generate either (dis)similarity matrices or weighted networks, associations $(r_{ij})_{i,j}$ between the microbes must be calculated. Beforehand, often zero handling and normalization of the data are required. Table 5 gives an overview of the method combinations used for calculating the associations r_{ij} for later use in (dis)similarity based clustering, i.e., for generating (dis)similarity matrices which will later be used as input for hierarchical and spectral clustering. We used four different association measures. The first ones are the Pearson and Spearman correlations, which require normalization to account for compositionality. Here we used either the centered log-ratio transformation (clr, [43]), the modified clr transformation (mclr, [44]), or the variance-stabilizing transformation (VST, [45]). As the clr and VST methods cannot handle zeros in the count data, a pseudo count of 1 was added to the count data before normalizing with these methods (mclr, on the other hand, can deal with zeros). Apart from the Pearson and Spearman correlations, we used the semi-parametric rank-based correlation, which is based on estimating the latent correlation matrix of a truncated Gaussian copula model (latentcor, [46, 69]). Since the latentcor method requires normalized counts that are

Table 5. Method combinations for generating microbial associations, which are then transformed into (dis)similarity matrices. The (dis)similarity matrices were used as input for hierarchical and spectral clustering.

Zero handling	Normalization	Association estimation
pseudo	clr	Pearson
pseudo	VST	Pearson
none	mclr	Pearson
pseudo	clr	Spearman
pseudo	VST	Spearman
none	mclr	Spearman
none	mclr	latentcor
pseudo	clr	proportionality

https://doi.org/10.1371/journal.pcbi.1010820.t005

strictly non-negative, it was only combined with the mclr transformation. The final association measure is proportionality [47, 70]. Proportionality is a compositionally aware method that measures associations between log-ratio transformed variables [47]. We thus used the clr transformation as proposed in [47] and replaced zero counts by a pseudo count.

Table 6 shows the method combinations used for calculating the associations r_{ij} for later use in network-based clustering. That is, these methods were used for generating weighted networks. The method combinations are very similar to the methods in Table 5 for generating (dis)similarity matrices. Indeed, weighted networks are also based on (dis)similarity matrices, but the generation contains an additional sparsification step, as explained below. Again, the Pearson and Spearman correlations were used with the respective normalization and/or zero handling methods. We also used the SPRING method [44], which combines the latentcor correlation estimation with sparse graphical modeling techniques, namely by using the neighborhood selection technique [71] for sparse estimation of partial correlations. Finally, we used the proportionality measure.

To generate a weighted network, the associations r_{ij} (which are usually different from zero) were not directly used as an adjacency matrix—otherwise, the network would be dense. Therefore, the associations r_{ij} were transformed into sparsified values r^*_{ij} by setting some r^*_{ij} to zero to indicate that i and j are not connected, $r^*_{ij} = r_{ij}$ otherwise. For sparsification of the Pearson and

Table 6. Method combinations for generating weighted microbial association networks. The networks were used as input for fast greedy modularity optimization, Louvain community detection, and manta.

Zero handling	Normalization	Association estimation	Sparsification
pseudo	clr	Pearson	*t*-test
pseudo	clr	Pearson	threshold
pseudo	VST	Pearson	*t*-test
pseudo	VST	Pearson	threshold
none	mclr	Pearson	*t*-test
none	mclr	Pearson	threshold
pseudo	clr	Spearman	*t*-test
pseudo	clr	Spearman	threshold
pseudo	VST	Spearman	*t*-test
pseudo	VST	Spearman	threshold
none	mclr	Spearman	*t*-test
none	mclr	Spearman	threshold
none	mclr	SPRING	neighborhood selection
pseudo	clr	proportionality	threshold

https://doi.org/10.1371/journal.pcbi.1010820.t006

Spearman correlations r_{ij}, we used either Student's t-test or the threshold method. The former sets $r^*_{ij} = 0$ if the association r_{ij} is not significantly different from 0 according to the t-test. The p-values were adjusted for multiple testing via the local false discovery rate [72]. For the threshold method, we set $r^*_{ij} = 0$ if $r_{ij} < c$ for some fixed threshold value c (we use $c = 0.15$ which gave reasonable results in preliminary analyses, not shown). For the proportionality measure, we used threshold sparsification. SPRING already comes with inbuilt sparsification given by the neighborhood selection method.

After calculating the associations as in Tables 5 and 6, they were then transformed as follows (the pipeline and notations are taken from [9]):

a. For (dis)similarity based clustering (Table 5): A dissimilarity matrix $D = (d_{ij})$ for hierarchical clustering is calculated via $d_{ij} = \sqrt{0.5(1 - r_{ij})}$. A similarity matrix $S = (s_{ij})$ for spectral clustering is obtained by setting $s_{ij} = 1 - d_{ij}$.

b. For network-based clustering (Table 6): A weighted network is constructed as follows. For the edges ij with $r^*_{ij} \neq 0$ (i.e., the edges that remain after sparsification), the distances d_{ij} and similarities s_{ij} are calculated as in a). Finally, the weighted network is represented as an adjacency matrix $A = (a_{ij})$ with $a_{ij} = s_{ij}$ for ij with $r^*_{ij} \neq 0$, and $a_{ij} = 0$ otherwise.

The (dis)similarity matrices and networks were then used as input for clustering. For hierarchical and spectral clustering, we fixed the number of clusters at $k = 10$, which was inspired by the ten different taxonomic classes in the data. Also, $k = 10$ tends to yield better ARI results than ks lower than ten (preliminary analysis, not shown). ks higher than ten were not tried because we aimed to emulate a researcher who wants to find an interpretable, handy clustering (there are 34 different taxonomic families, but 34 clusters are not easily interpretable). The other clustering algorithms all have inbuilt mechanisms for determining k. Forcing k to be 10 for these methods generally did not improve the results (not shown). However, k can be indirectly influenced via the sparsification: The sparser the network, the more clusters tend to be found. This is one of the reasons we set the threshold for threshold sparsification at $c = 0.15$ because this value generally yielded sufficiently high ks to find good results, but only rarely ks that are so high that the clusters are difficult to interpret.

Overall, the method combinations yielded 58 different clustering results on the discovery data: 16 based on (dis)similarity clustering (eight rows in Table 5 times two cluster algorithms), and 42 based on network clustering (fourteen rows in Table 6 times three cluster algorithms). The best one out of the 58 clustering results was chosen, i.e., the clustering with the highest ARI regarding the taxonomic categorization into families. The corresponding method combination was applied to the validation data. The ARI between the clustering on the validation data and the taxonomic categorization was computed and compared with the best ARI on the discovery data. If the ARI on the validation data was lower, this was an indication that the best ARI on the discovery data was over-optimistic.

As an additional stability analysis, we compared the chosen clustering on the discovery data with the clustering on the validation data, again using the ARI.

4.3.2 Research task 2: Hub detection. Here, we wanted to generate sparse weighted microbial association networks. For this purpose, we used the same methods as in Table 6. Thus, 14 method combinations were tried on the discovery data.

For hub detection in the resulting networks, hubs were defined as nodes that have the highest degree, betweenness, and closeness centrality [25]. More precisely, we determined the hubs as the nodes with centrality values above the 95% empirical quantile, for each of the three centrality measures simultaneously. The centralities are defined as follows [73]: The degree

centrality denotes the number of adjacent nodes. The betweenness centrality measures the fraction of times a node lies on the shortest path between all other nodes. The closeness centrality of a node is the reciprocal of the sum of shortest paths between this node and all other nodes. All centrality measures were normalized to be comparable between networks of different sizes (see [9] for details). The centralities were only calculated for the largest connected component of each network (i.e., the largest subgraph of the network in which all nodes are connected); centrality values of nodes in the disconnected component were set to zero. We assumed that "hubs" in small parts of the network that are disconnected from the majority of the nodes are of less interest to researchers. Moreover, the betweenness and closeness centrality depend on shortest paths, which are not well-defined for nodes in different unconnected sub-graphs.

After applying the 14 method combinations and calculating the hubs for each resulting network, the method combination that yielded the highest number of hubs was chosen. If there were multiple method combinations that attained the maximal number of hubs, we chose the combination that yielded higher mean centrality values of the hubs. More specifically, for each set of hubs that corresponds to a method combination, the mean values of the three centrality measures were calculated over the hubs. Then for each centrality measure separately, the sets of hubs were ranked according to these mean values. Finally, the set of hubs (and thus the corresponding method combination) that yielded the highest mean rank over all three centrality measures was chosen.

The "best" method combination was then applied to the validation data. The number of hubs in the microbial network on the validation data was calculated and compared with the highest number of hubs on the discovery data. Over-optimism was indicated if the number of hubs was lower on the validation data.

Additionally, we reported the similarity of the sets of hubs determined on the discovery vs. validation data with the Jaccard index [74]: let H_{discov}, H_{valid} be the sets of hubs for the discovery resp. validation data, then

$$\mathrm{Jacc}(H_{discov}, H_{valid}) = \frac{|H_{discov} \cap H_{valid}|}{|H_{discov} \cup H_{valid}|}.$$

The Jaccard index takes values in [0, 1], and is closer to 1 the more similar the sets are. The similarity between the sets of hubs was also assessed on the higher taxonomic level of families with the cosine similarity index. More precisely, assume that the hubs (genera) in the union $H_{discov} \cup H_{valid}$ belong to l distinct families overall. Let $\mathbf{f}^{(d)} = (f_1^{(d)}, \ldots, f_l^{(d)})$ be the family frequency vector for H_{discov}, that is, each entry $f_j^{(d)}$ counts how many hubs in H_{discov} belong to family j. Analogously, let $\mathbf{f}^{(v)}$ be the family frequency vector for H_{valid}. The vectors $\mathbf{f}^{(d)}$ and $\mathbf{f}^{(v)}$ are then compared with the cosine similarity index:

$$\cos\ \mathrm{sim}(\mathbf{f}^{(d)}, \mathbf{f}^{(v)}) = \frac{\sum_{j=1}^{l} f_j^{(d)} f_j^{(v)}}{\sqrt{\sum_{j=1}^{l} (f_j^{(d)})^2} \sqrt{\sum_{j=1}^{l} (f_j^{(v)})^2}}$$

The cosine similarity index ranges in [0, 1], with higher values indicating higher similarity.

4.3.3 Research task 3: Differential network analysis. As described in Section 4.2, the discovery and validation datasets each consisted of two halves: persons who did not take antibiotics in the last year ("non-antibiotics samples"), and persons who took antibiotics in the last month ("antibiotics samples"). The methods for generating weighted microbial association networks as in Table 6 were applied separately to the antibiotics and non-antibiotics samples of the discovery data.

The resulting networks were compared with the Graphlet Correlation Distance (GCD, [31]). This distance measures the similarity of the networks based on small induced subgraphs,

so-called graphlets. All graphlets composed of up to four nodes are considered, and the automorphism orbits of these graphlets are enumerated (orbits represent the "roles" that nodes can play in the graphlets). For each node in a given network, one can count how often the node participates in each graphlet at the respective orbits. Only 11 non-redundant orbits are considered here. Based on these orbit counts across all nodes, the 11×11 Spearman correlation matrix among the 11 orbits is calculated, which represents a robust and size independent network summary statistics. For comparing two networks, the Spearman correlation matrix is calculated for each network in turn. Then the Euclidean distance between the upper triangular parts of these matrices is calculated, resulting in the GCD.

In our study, the network generation method that yielded the largest GCD between the antibiotics network and the non-antibiotics network was chosen as the "best" one and applied to the antibiotics and non-antibiotics samples in the validation data. Again, the resulting networks were compared with the GCD. If the GCD on the validation data was smaller (i.e., the antibiotics vs. non-antibiotics networks were more similar than on the discovery data), this indicated over-optimism.

4.3.4 Research task 4: Clustering of samples. Similar to the first research task, both (dis)similarity-based and network-based cluster algorithms were applied to the discovery data (resulting in 31 clusterings overall). In contrast to the first task, dissimilarities between samples instead of microbes were calculated, and sample networks instead of microbial association networks were estimated (i.e., networks in which nodes correspond to samples, not taxa).

We considered partitioning around medoids (PAM) [56] as well as spectral clustering as instances of (dis)similarity-based clustering algorithms. We chose PAM since it has been frequently used in enterotype studies [35, 37, 41, 68]. For this research task, we excluded hierarchical clustering because this algorithm frequently resulted in clusters with nearly all samples contained in one cluster and only a few samples in other clusters (this phenomenon did not occur to the same extent in the clustering of bacterial genera). Presumably, researchers would be less interested in such clustering results.

From the category of network-based cluster algorithms, we chose fast greedy modularity optimization and the Louvain method for community detection. The manta algorithm was not chosen because it was explicitly developed for clustering taxa, not samples.

We also included clustering based on Dirichlet multinomial mixtures (DMM) [55]. In contrast to the cluster algorithms listed above, DMM does not require calculation of dissimilarities between samples and can be applied directly to the microbial count matrix. The DMM method has been used in several studies to detect enterotypes [55, 68, 75].

Table 7 presents the different methods for calculating dissimilarities $(d_{ij})_{i,j}$ between the samples, which are then used as input for PAM and spectral clustering. We used the Aitchison distance [52] which is defined as the Euclidean distance between clr-transformed compositions. We also combined the Euclidean distance with the VST and mclr normalization. Moreover,

Table 7. Method combinations for dissimilarity calculation. The dissimilarity matrices were used as input for PAM and spectral clustering.

Zero handling	Normalization	Association estimation
pseudo	clr	Aitchison
pseudo	VST	Euclidean
none	mclr	Euclidean
pseudo	fractions	cKLD
none	mclr	Bray-Curtis

https://doi.org/10.1371/journal.pcbi.1010820.t007

Table 8. Method combinations for generating weighted sample networks. The networks were used as input for fast greedy modularity optimization and Louvain community detection.

Zero handling	Normalization	Association estimation	Sparsification
pseudo	clr	Aitchison	threshold
pseudo	clr	Aitchison	*K*-NN
pseudo	VST	Euclidean	threshold
pseudo	VST	Euclidean	*K*-NN
none	mclr	Euclidean	threshold
none	mclr	Euclidean	*K*-NN
pseudo	fractions	cKLD	threshold
pseudo	fractions	cKLD	*K*-NN
none	mclr	Bray-Curtis	threshold
none	mclr	Bray-Curtis	*K*-NN

https://doi.org/10.1371/journal.pcbi.1010820.t008

we applied the compositional Kullback-Leibler divergence (cKLD) [53]. The cKLD measure is suitable for application on compositional data; thus, the counts are merely transformed into fractions (relative abundances) before the measure is applied. Finally, we applied the Bray-Curtis dissimilarity measure [54], which requires non-negative values as input and is therefore combined with the mclr normalization.

The dissimilarities d_{ij} were scaled to [0, 1], resulting in values d_{ij}^{scale} (see [9] for details). The scaled dissimilarities were used as input for PAM. Similarities s_{ij} for spectral clustering were obtained by setting $s_{ij} = 1 - d_{ij}^{scale}$.

For network-based clustering, the same methods for calculating dissimilarities as in Table 7 were used, but with an additional sparsification step. This is displayed in Table 8. The scaled dissimilarities d_{ij}^{scale} were transformed into sparsified values d_{ij}^{*}, either with the threshold method (by setting d_{ij}^{*} to 1, i.e., the maximum dissimilarity, if $d_{ij}^{scale} > 0.85$), or with the *K*-nearest neighbor method (each node is connected to the $K = 3$ nodes with minimum dissimilarity; if nodes *i* and *j* are not connected after this procedure, d_{ij}^{*} is set to 1). The weighted sample network is then represented as an adjacency matrix $A = (a_{ij})$ with $a_{ij} = s_{ij} = 1 - d_{ij}^{*}$, with $a_{ij} = 0$ for sparsified edges.

DMM clustering, fast greedy modularity optimization, and Louvain commmunity detection all have inbuilt mechanisms for determining the number of clusters *k*. For PAM and spectral clustering, we tried different values $k \in \{2, 3, \ldots, 10\}$ and chose the *k* that maximized the ASW of the clustering.

For calculating the ASW of a clustering, a corresponding dissimilarity matrix is required. For most clustering results, we used the dissimilarity matrix that was calculated one step before applying the cluster algorithm. The only exception are clustering results obtained by DMM which does not require prior calculation of dissimilarities. We calculated the ASW values for DMM clustering results based on the Bray-Curtis dissimilarity matrix since the authors of the DMM method used this dissimilarity measure to visualize their clustering results [55].

Overall, the considered method combinations led to 31 different clustering results on the discovery data: one based on DMM clustering, ten based on (dis)similarity clustering (five rows in Table 7 times two cluster algorithms), and 20 based on network clustering (ten rows in Table 8 times two cluster algorithms). The method combination that yielded the clustering with the highest ASW value was chosen and applied to the validation data, with over-optimistic bias indicated by lower ASW values on the validation data.

4.4 Technical implementation

All analyses were performed with R, version 4.0.4 and Python, version 3.6.13. Our fully reproducible code is available at https://github.com/thullmann/overoptimism-microbiome. (Dis) similarity matrices and weighted networks were generated with the R package NetCoMi [9]. Spectral clustering was performed with a previously published R implementation [23]. For fast greedy modularity optimization and the Louvain method for community detection, we used the R package igraph [76]. For clustering with manta, we accessed the Python implementation [51] with the reticulate interface for R [77]. We used the R package cluster [78] for PAM clustering, and the R package DirichletMultinomial [79] for DMM clustering. Orbit counts for the calculation of the GCD were generated with the R package orca [80].

Supporting information

S1 Text. Full results and plots for research task 1 (clustering of bacterial genera).
(PDF)

S2 Text. Full results and plots for research task 2 (hub detection).
(PDF)

S3 Text. Full results and plots for research task 3 (differential network analysis).
(PDF)

S4 Text. Full results and plots for research task 4 (clustering of samples).
(PDF)

S5 Text. Analyses with a reduced number of method combinations.
(PDF)

Acknowledgments

We thank Raphael Rehms for helpful comments about the Python code and Anna Jacob for valuable language corrections.

Author Contributions

Conceptualization: Theresa Ullmann, Christian L. Müller, Anne-Laure Boulesteix.

Formal analysis: Theresa Ullmann.

Funding acquisition: Christian L. Müller, Anne-Laure Boulesteix.

Methodology: Theresa Ullmann, Stefanie Peschel, Philipp Finger, Christian L. Müller, Anne-Laure Boulesteix.

Software: Theresa Ullmann, Stefanie Peschel.

Supervision: Christian L. Müller, Anne-Laure Boulesteix.

Validation: Theresa Ullmann.

Visualization: Theresa Ullmann, Stefanie Peschel.

Writing – original draft: Theresa Ullmann.

Writing – review & editing: Theresa Ullmann, Stefanie Peschel, Philipp Finger, Christian L. Müller, Anne-Laure Boulesteix.

References

1. Zmora N, Soffer E, Elinav E. Transforming medicine with the microbiome. Science Translational Medicine. 2019; 11(477):eaaw1815. https://doi.org/10.1126/scitranslmed.aaw1815 PMID: 30700573
2. Kuntz TM, Gilbert JA. Introducing the microbiome into precision medicine. Trends in Pharmacological Sciences. 2017; 38(1):81–91. https://doi.org/10.1016/j.tips.2016.10.001 PMID: 27814885
3. Kim D, Hofstaedter CE, Zhao C, Mattei L, Tanes C, Clarke E, et al. Optimizing methods and dodging pitfalls in microbiome research. Microbiome. 2017; 5(1):52. https://doi.org/10.1186/s40168-017-0267-5 PMID: 28476139
4. Schloss PD. Identifying and overcoming threats to reproducibility, replicability, robustness, and generalizability in microbiome research. mBio. 2018; 9(3):e00525–18. https://doi.org/10.1128/mBio.00525-18 PMID: 29871915
5. Open Science Collaboration. Estimating the reproducibility of psychological science. Science. 2015; 349(6251):aac4716. https://doi.org/10.1126/science.aac4716
6. Hoffmann S, Schönbrodt F, Elsas R, Wilson R, Strasser U, Boulesteix AL. The multiplicity of analysis strategies jeopardizes replicability: lessons learned across disciplines. Royal Society Open Science. 2021; 8:201925. https://doi.org/10.1098/rsos.201925 PMID: 33996122
7. Simmons JP, Nelson LD, Simonsohn U. False-positive psychology: undisclosed flexibility in data collection and analysis allows presenting anything as significant. Psychological Science. 2011; 22(11):1359–1366. https://doi.org/10.1177/0956797611417632 PMID: 22006061
8. Klau S, Martin-Magniette ML, Boulesteix AL, Hoffmann S. Sampling uncertainty versus method uncertainty: A general framework with applications to omics biomarker selection. Biometrical Journal. 2020; 62(3):670–687. https://doi.org/10.1002/bimj.201800309 PMID: 31099917
9. Peschel S, Müller CL, von Mutius E, Boulesteix AL, Depner M. NetCoMi: network construction and comparison for microbiome data in R. Briefings in Bioinformatics. 2020; 22(4):bbaa290. https://doi.org/10.1093/bib/bbaa290
10. Nosek BA, Errington TM. What is replication? PLoS Biology. 2020; 18(3):e3000691. https://doi.org/10.1371/journal.pbio.3000691 PMID: 32218571
11. Ullmann T, Hennig C, Boulesteix AL. Validation of cluster analysis results on validation data: A systematic framework. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2022; 12(3):e1444. https://doi.org/10.1002/widm.1444
12. Ioannidis JP. Why most published research findings are false. PLoS Medicine. 2005; 2(8):e124. https://doi.org/10.1371/journal.pmed.0020124 PMID: 16060722
13. Gelman A, Loken E. The statistical crisis in science. American Scientist. 2014; 102(6):460. https://doi.org/10.1511/2014.111.460
14. Head ML, Holman L, Lanfear R, Kahn AT, Jennions MD. The extent and consequences of p-hacking in science. PLoS Biology. 2015; 13(3):e1002106. https://doi.org/10.1371/journal.pbio.1002106 PMID: 25768323
15. McDonald D, Hyde E, Debelius JW, Morton JT, Gonzalez A, Ackermann G, et al. American gut: an open platform for citizen science microbiome research. Msystems. 2018; 3(3):e00031–18. https://doi.org/10.1128/mSystems.00031-18 PMID: 29795809
16. Sinha R, Abu-Ali G, Vogtmann E, Fodor AA, Ren B, Amir A, et al. Assessment of variation in microbial community amplicon sequencing by the Microbiome Quality Control (MBQC) project consortium. Nature Biotechnology. 2017; 35(11):1077–1086. https://doi.org/10.1038/nbt.3981 PMID: 28967885
17. Allali I, Arnold JW, Roach J, Cadenas MB, Butz N, Hassan HM, et al. A comparison of sequencing platforms and bioinformatics pipelines for compositional analysis of the gut microbiome. BMC Microbiology. 2017; 17(1):194. https://doi.org/10.1186/s12866-017-1101-8 PMID: 28903732
18. Clausen DS, Willis AD. Evaluating replicability in microbiome data. Biostatistics. 2021;kxab048 https://doi.org/10.1093/biostatistics/kxab048.
19. Tierney BT, Tan Y, Yang Z, Shui B, Walker MJ, Kent BM, et al. Systematically assessing microbiome–disease associations identifies drivers of inconsistency in metagenomic research. PLoS Biology. 2022; 20(3):1–18. https://doi.org/10.1371/journal.pbio.3001556 PMID: 35235560
20. Nearing JT, Douglas GM, Hayes MG, MacDonald J, Desai DK, Allward N, et al. Microbiome differential abundance methods produce different results across 38 datasets. Nature Communications. 2022; 13(1):1–16. https://doi.org/10.1038/s41467-022-28034-z
21. Khomich M, Måge I, Rud I, Berget I. Analysing microbiome intervention design studies: Comparison of alternative multivariate statistical methods. PLoS One. 2021; 16(11):1–20. https://doi.org/10.1371/journal.pone.0259973 PMID: 34793531

22. Hubert L, Arabie P. Comparing partitions. Journal of Classification. 1985; 2(1):193–218. https://doi.org/10.1007/BF01908075
23. Badri M, Kurtz ZD, Bonneau R, Müller CL. Shrinkage improves estimation of microbial associations under different normalization methods. NAR Genomics and Bioinformatics. 2020; 2(4):lqaa100. https://doi.org/10.1093/nargab/lqaa100 PMID: 33575644
24. Berry D, Widder S. Deciphering microbial interactions and detecting keystone species with co-occurrence networks. Frontiers in Microbiology. 2014; 5:219. https://doi.org/10.3389/fmicb.2014.00219 PMID: 24904535
25. Agler MT, Ruhe J, Kroll S, Morhenn C, Kim ST, Weigel D, et al. Microbial hub taxa link host and abiotic factors to plant microbiome variation. PLoS Biology. 2016; 14(1):e1002352. https://doi.org/10.1371/journal.pbio.1002352 PMID: 26788878
26. Banerjee S, Schlaeppi K, van der Heijden MG. Keystone taxa as drivers of microbiome structure and functioning. Nature Reviews Microbiology. 2018; 16(9):567–576. https://doi.org/10.1038/s41579-018-0024-1 PMID: 29789680
27. Röttjers L, Faust K. From hairballs to hypotheses–biological insights from microbial networks. FEMS Microbiology Reviews. 2018; 42(6):761–780. https://doi.org/10.1093/femsre/fuy030 PMID: 30085090
28. Zamkovaya T, Foster JS, de Crécy-Lagard V, Conesa A. A network approach to elucidate and prioritize microbial dark matter in microbial communities. The ISME Journal. 2021; 15(1):228–244. https://doi.org/10.1038/s41396-020-00777-x PMID: 32963345
29. Francino M. Antibiotics and the human gut microbiome: dysbioses and accumulation of resistances. Frontiers in microbiology. 2016; 6:1543. https://doi.org/10.3389/fmicb.2015.01543 PMID: 26793178
30. Costello EK, Stagaman K, Dethlefsen L, Bohannan BJ, Relman DA. The application of ecological theory toward an understanding of the human microbiome. Science. 2012; 336(6086):1255–1262. https://doi.org/10.1126/science.1224203 PMID: 22674335
31. Yaveroğlu ÖN, Malod-Dognin N, Davis D, Levnajic Z, Janjic V, Karapandza R, et al. Revealing the hidden language of complex networks. Scientific Reports. 2014; 4(1):1–9. https://doi.org/10.1038/srep04547 PMID: 24686408
32. Mahana D, Trent CM, Kurtz ZD, Bokulich NA, Battaglia T, Chung J, et al. Antibiotic perturbation of the murine gut microbiome enhances the adiposity, insulin resistance, and liver disease associated with high-fat diet. Genome Medicine. 2016; 8(1):1–20. https://doi.org/10.1186/s13073-016-0297-9 PMID: 27124954
33. Ruiz VE, Battaglia T, Kurtz ZD, Bijnens L, Ou A, Engstrand I, et al. A single early-in-life macrolide course has lasting effects on murine microbial network topology and immunity. Nature Communications. 2017; 8(1):1–14. https://doi.org/10.1038/s41467-017-00531-6 PMID: 28894149
34. Leung MH, Tong X, Wilkins D, Cheung HH, Lee PK. Individual and household attributes influence the dynamics of the personal skin microbiota and its association network. Microbiome. 2018; 6(1):1–15. https://doi.org/10.1186/s40168-018-0412-9 PMID: 29394957
35. Arumugam M, Raes J, Pelletier E, Le Paslier D, Yamada T, Mende DR, et al. Enterotypes of the human gut microbiome. Nature. 2011; 473:174–180. https://doi.org/10.1038/nature09944 PMID: 21508958
36. Jeffery IB, Claesson MJ, O'Toole PW, Shanahan F. Categorization of the gut microbiota: enterotypes or gradients? Nature Reviews Microbiology. 2012; 10(9):591–592. https://doi.org/10.1038/nrmicro2859 PMID: 23066529
37. Koren O, Knights D, Gonzalez A, Waldron L, Segata N, Knight R, et al. A guide to enterotypes across the human body: meta-analysis of microbial community structures in human microbiome datasets. PLoS Computational Biology. 2013; 9(1):e1002863. https://doi.org/10.1371/journal.pcbi.1002863 PMID: 23326225
38. Knights D, Ward TL, McKinlay CE, Miller H, Gonzalez A, McDonald D, et al. Rethinking "enterotypes". Cell Host & Microbe. 2014; 16(4):433–437. https://doi.org/10.1016/j.chom.2014.09.013 PMID: 25299329
39. Costea PI, Hildebrand F, Arumugam M, Bäckhed F, Blaser MJ, Bushman FD, et al. Enterotypes in the landscape of gut microbial community composition. Nature Microbiology. 2018; 3:8–16. https://doi.org/10.1038/s41564-017-0072-8 PMID: 29255284
40. Cheng M, Ning K. Stereotypes about enterotype: the old and new ideas. Genomics, Proteomics & Bioinformatics. 2019; 17(1):4–12. https://doi.org/10.1016/j.gpb.2018.02.004 PMID: 31026581
41. Wu GD, Chen J, Hoffmann C, Bittinger K, Chen YY, Keilbaugh SA, et al. Linking long-term dietary patterns with gut microbial enterotypes. Science. 2011; 334(6052):105–108. https://doi.org/10.1126/science.1208344 PMID: 21885731

42. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics. 1987; 20:53–65. https://doi.org/10.1016/0377-0427(87)90125-7

43. Aitchison J. The statistical analysis of compositional data. Journal of the Royal Statistical Society: Series B (Methodological). 1982; 44(2):139–160.

44. Yoon G, Gaynanova I, Müller CL. Microbial networks in SPRING—Semi-parametric rank-based correlation and partial correlation estimation for quantitative microbiome data. Frontiers in Genetics. 2019; 10:516. https://doi.org/10.3389/fgene.2019.00516 PMID: 31244881

45. Anders S, Huber W. Differential expression analysis for sequence count data. Genome Biology. 2010; 11:R106. https://doi.org/10.1186/gb-2010-11-10-r106 PMID: 20979621

46. Yoon G, Carroll RJ, Gaynanova I. Sparse semiparametric canonical correlation analysis for data of mixed types. Biometrika. 2020; 107(3):609–625. https://doi.org/10.1093/biomet/asaa007 PMID: 34621080

47. Lovell D, Pawlowsky-Glahn V, Egozcue JJ, Marguerat S, Bähler J. Proportionality: a valid alternative to correlation for relative data. PLoS Computational Biology. 2015; 11(3):e1004075. https://doi.org/10.1371/journal.pcbi.1004075 PMID: 25775355

48. Ng A, Jordan M, Weiss Y. On spectral clustering: analysis and an algorithm. Advances in Neural Information Processing Systems. 2001; 14:849–856.

49. Clauset A, Newman ME, Moore C. Finding community structure in very large networks. Physical Review E. 2004; 70(6):066111. https://doi.org/10.1103/PhysRevE.70.066111 PMID: 15697438

50. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment. 2008; 2008(10):P10008. https://doi.org/10.1088/1742-5468/2008/10/P10008

51. Röttjers L, Faust K. Manta: A clustering algorithm for weighted ecological networks. Msystems. 2020; 5(1):e00903–19. https://doi.org/10.1128/mSystems.00903-19 PMID: 32071163

52. Aitchison J. On criteria for measures of compositional difference. Mathematical Geology. 1992; 24(4): 365–379. https://doi.org/10.1007/BF00891269

53. Martín-Fernández JA, Bren M, Barceló-Vidal C, Pawlowsky-Glahn V. A measure of difference for compositional data based on measures of divergence. In: Proceedings of the Fifth Annual Conference of the International Association for Mathematical Geology. vol. 1; 1999. p. 211–215.

54. Bray JR, Curtis JT. An ordination of the upland forest communities of southern Wisconsin. Ecological Monographs. 1957; 27(4):326–349. https://doi.org/10.2307/1942268

55. Holmes I, Harris K, Quince C. Dirichlet multinomial mixtures: generative models for microbial metagenomics. PloS One. 2012; 7(2):e30126. https://doi.org/10.1371/journal.pone.0030126 PMID: 22319561

56. Kaufman L, Rousseeuw PJ. Finding Groups in Data. John Wiley & Sons, Ltd; 1990.

57. Boulesteix AL, Lauer S, Eugster MJ. A plea for neutral comparison studies in computational sciences. PloS One. 2013; 8(4):e61562. https://doi.org/10.1371/journal.pone.0061562 PMID: 23637855

58. Boulesteix AL, Wilson R, Hapfelmeier A. Towards evidence-based computational statistics: lessons from clinical research on the role and design of real-data benchmark studies. BMC Medical Research Methodology. 2017; 17:138. https://doi.org/10.1186/s12874-017-0417-2 PMID: 28888225

59. Wallen ZD. Comparison study of differential abundance testing methods using two large Parkinson disease gut microbiome datasets derived from 16S amplicon sequencing. BMC Bioinformatics. 2021; 22(1):1–29. https://doi.org/10.1186/s12859-021-04193-6 PMID: 34034646

60. Bokulich NA, Ziemski M, Robeson MS II, Kaehler BD. Measuring the microbiome: Best practices for developing and benchmarking microbiomics methods. Computational and Structural Biotechnology Journal. 2020; 18:4048–4062. https://doi.org/10.1016/j.csbj.2020.11.049 PMID: 33363701

61. Nosek BA, Ebersole CR, DeHaven AC, Mellor DT. The preregistration revolution. Proceedings of the National Academy of Sciences. 2018; 115(11):2600–2606. https://doi.org/10.1073/pnas.1708274114 PMID: 29531091

62. Wirbel J, Zych K, Essex M, Karcher N, Kartal E, Salazar G, et al. Microbiome meta-analysis and cross-disease comparison enabled by the SIAMCAT machine learning toolbox. Genome Biology. 2021; 22:93. https://doi.org/10.1186/s13059-021-02306-1 PMID: 33785070

63. Bien J, Yan X, Simpson L, Müller CL. Tree-aggregated predictive modeling of microbiome data. Scientific Reports. 2021; 11(1):1–13. https://doi.org/10.1038/s41598-021-93645-3 PMID: 34267244

64. Fromentin S, Forslund SK, Chechi K, Aron-Wisnewsky J, Chakaroun R, Nielsen T, et al. Microbiome and metabolome features of the cardiometabolic disease spectrum. Nature Medicine. 2022; 28:303–314. https://doi.org/10.1038/s41591-022-01688-4 PMID: 35177860

65. Palmer C, Bik EM, DiGiulio DB, Relman DA, Brown PO. Development of the human infant intestinal microbiota. PLoS Biology. 2007; 5(7):e177. https://doi.org/10.1371/journal.pbio.0050177 PMID: 17594176

66. Claesson MJ, Cusack S, O'Sullivan O, Greene-Diniz R, de Weerd H, Flannery E, et al. Composition, variability, and temporal stability of the intestinal microbiota of the elderly. Proceedings of the National Academy of Sciences. 2011; 108:4586–4591. https://doi.org/10.1073/pnas.1000097107 PMID: 20571116

67. Derrien M, Alvarez AS, de Vos WM. The gut microbiota in the first decade of life. Trends in Microbiology. 2019; 27(12):997–1010. https://doi.org/10.1016/j.tim.2019.08.001 PMID: 31474424

68. Zhong H, Penders J, Shi Z, Ren H, Cai K, Fang C, et al. Impact of early events and lifestyle on the gut microbiota and metabolic phenotypes in young school-age children. Microbiome. 2019; 7:2. https://doi.org/10.1186/s40168-018-0608-z PMID: 30609941

69. Yoon G, Müller CL, Gaynanova I. Fast computation of latent correlations. Journal of Computational and Graphical Statistics. 2021; 30(4):1249–1256. https://doi.org/10.1080/10618600.2021.1882468 PMID: 35280976

70. Quinn TP, Richardson MF, Lovell D, Crowley TM. propr: an R-package for identifying proportionally abundant features using compositional data analysis. Scientific Reports. 2017; 7(1):1–9. https://doi.org/10.1038/s41598-017-16520-0 PMID: 29176663

71. Meinshausen N, Bühlmann P. High-dimensional graphs and variable selection with the lasso. Annals of Statistics. 2006; 34(3):1436–1462. https://doi.org/10.1214/009053606000000281

72. Efron B. Local False Discovery Rates. Stanford University; 2005.

73. Freeman LC. Centrality in social networks conceptual clarification. Social networks. 1978; 1(3): 215–239. https://doi.org/10.1016/0378-8733(78)90021-7

74. Jaccard P. The distribution of the flora in the alpine zone. New Phytologist. 1912; 11(2):37–50. https://doi.org/10.1111/j.1469-8137.1912.tb05611.x

75. Ding T, Schloss PD. Dynamics and associations of microbial community types across the human body. Nature. 2014; 509(7500):357–360. https://doi.org/10.1038/nature13178 PMID: 24739969

76. Csárdi G, Nepusz T. The igraph software package for complex network research. InterJournal. 2006; Complex Systems:1695.

77. Ushey K, Allaire J, Tang Y. reticulate: interface to'Python'; 2022. Available from: https://rstudio.github.io/reticulate/.

78. Maechler M, Rousseeuw P, Struyf A, Hubert M, Hornik K. cluster: cluster analysis basics and extensions; 2022. Available from: https://CRAN.R-project.org/package=cluster.

79. Morgan M. DirichletMultinomial: Dirichlet-multinomial mixture model machine learning for microbiome data; 2022. Available from: https://www.bioconductor.org/packages/release/bioc/html/DirichletMultinomial.html.

80. Hočevar T, Demšar J. Computation of graphlet orbits for nodes and edges in sparse graphs. Journal of Statistical Software. 2016; 71(10):1–24.

S1: Full results and plots for research task 1 (clustering of bacterial genera)

List of Figures

Fig A-E show the results of applying different clustering methods to the discovery data for sample sizes $n \in \{100, 250, 500, 1000, 4000\}$. For each method combination on the x-axis, the resulting ARIs, which measure agreement with the taxonomic categorization into families, are summarized over the 50 samplings by boxplots. Outliers are indicated by black crosses. Additionally, all results are shown as colored dots, with the color indicating the number k of clusters in the respective clustering result. Results that were picked as the "best result" in one of the 50 samplings are marked by red square edges. For the network-based clustering methods with the networks generated by either the Pearson or Spearman correlation, the results for t-test and threshold sparsification are displayed together, i.e., 50*2 = 100 results are shown for these method combinations.

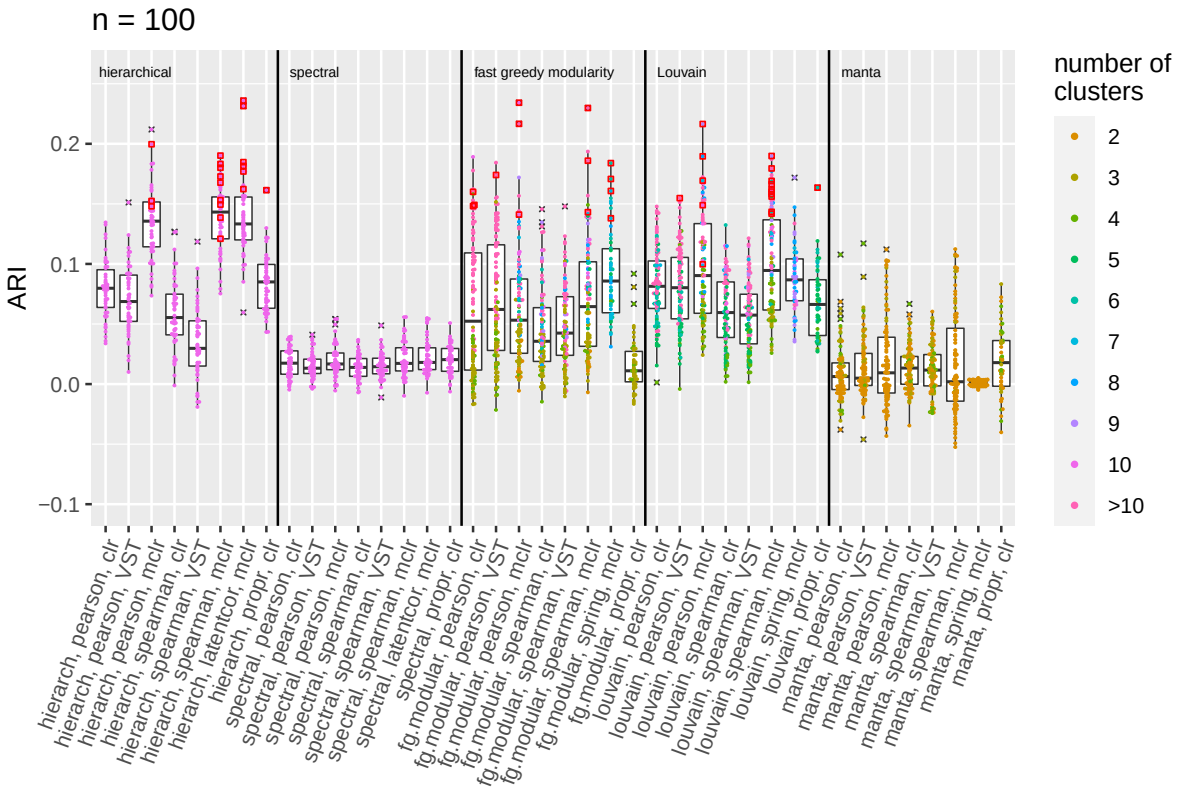


Fig A. Results for clustering bacterial genera on the discovery data, $n = 100$

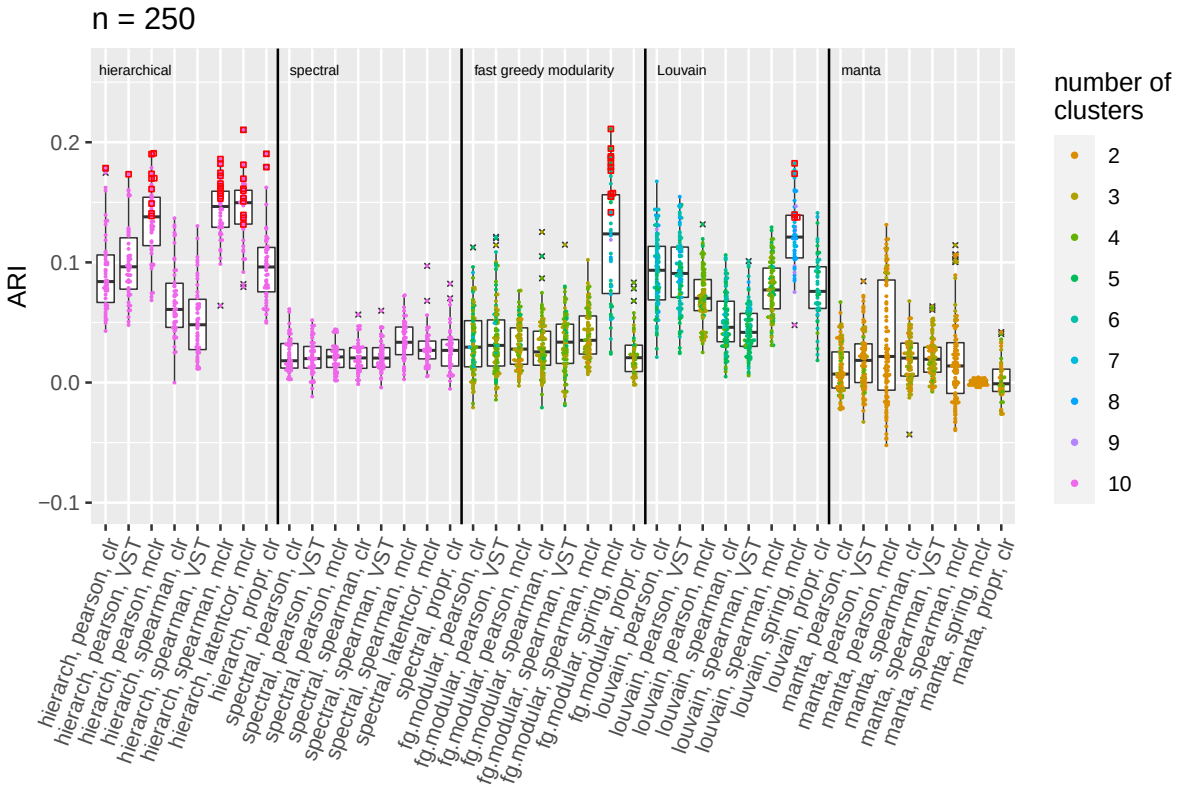


Fig B. Results for clustering bacterial genera on the discovery data, $n = 250$

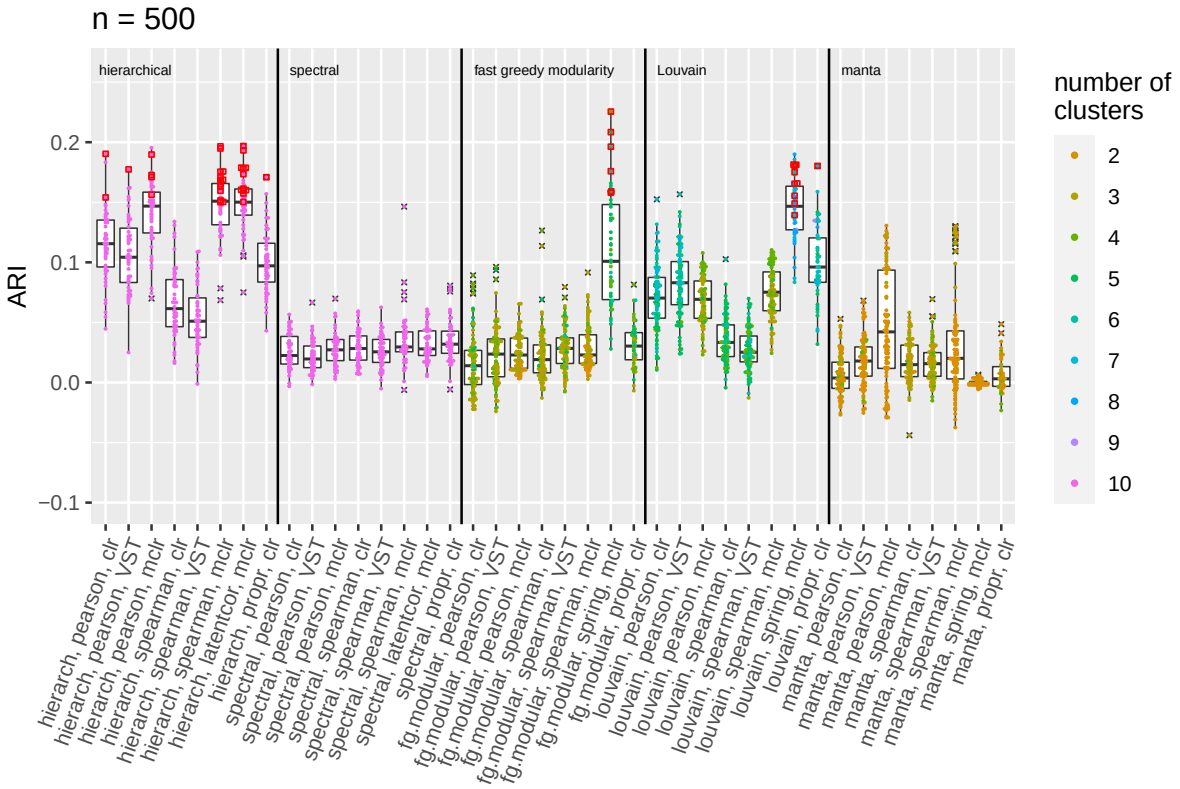


Fig C. Results for clustering bacterial genera on the discovery data, $n = 500$

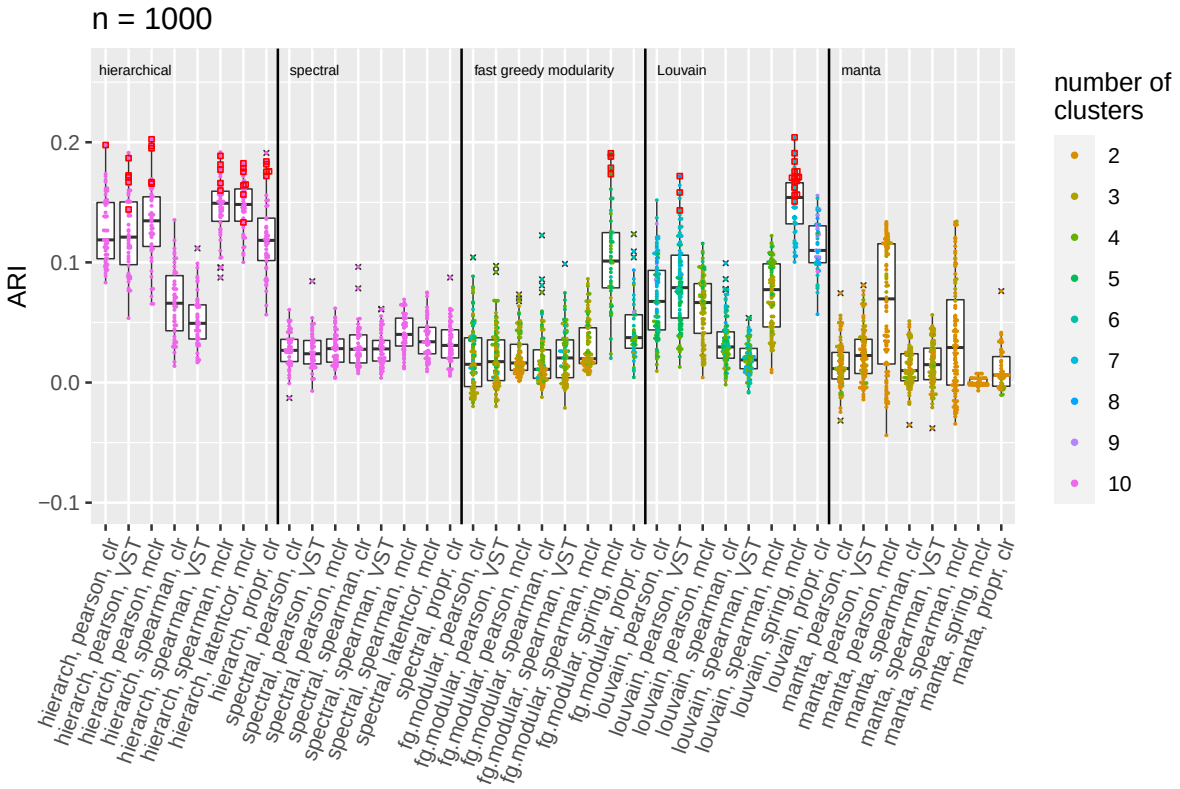


Fig D. Results for clustering bacterial genera on the discovery data, $n = 1000$

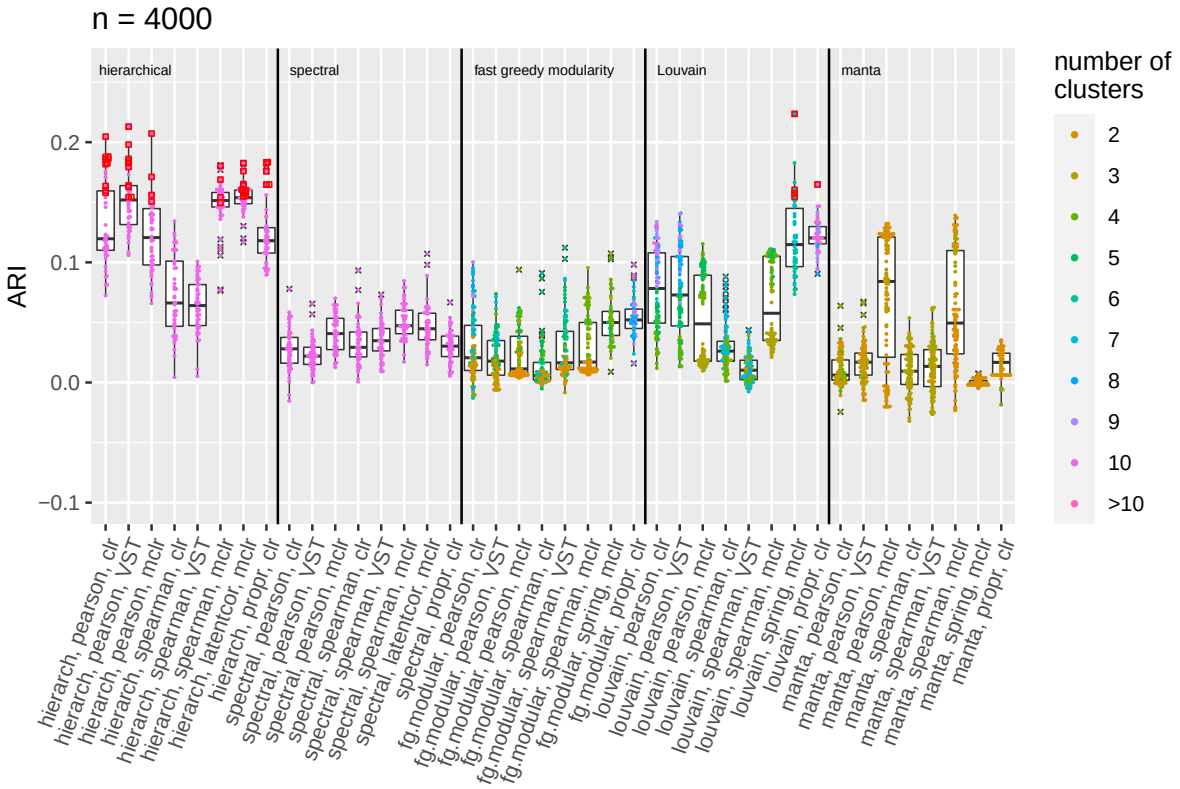


Fig E. Results for clustering bacterial genera on the discovery data, $n = 4000$

As can be seen in Fig A-E, the best ARI results stem either from hierarchical clustering, the Louvain method or fast greedy modularity optimization. Spectral clustering and manta are never selected. There is some change in the selected "best" methods with respect to sample size. For example, for $n = 100$, fast greedy modularity clustering performs well in several of the 50 samplings, but this cluster method does not yield very good ARI results for $n = 4000$. At $n = 4000$, hierarchical clustering is chosen as the best method in 45 of the 50 samplings.

In Fig F-J, results for the network-based clustering are shown separately for both sparsification methods (t-test and threshold). Results that were picked as the "best result" in one of the 50 samplings are marked by red square edges.

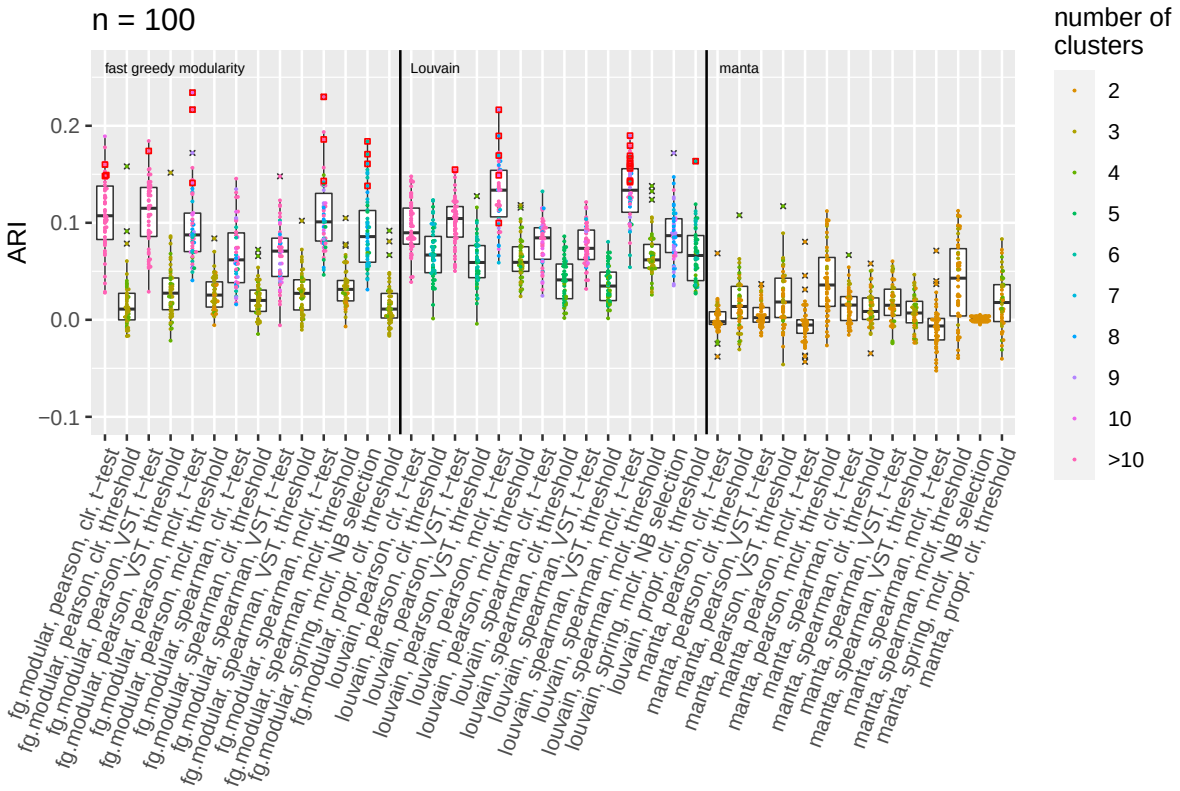


Fig F. Results for network-based clustering of bacterial genera on the discovery data, separated by sparsification methods, $n = 100$

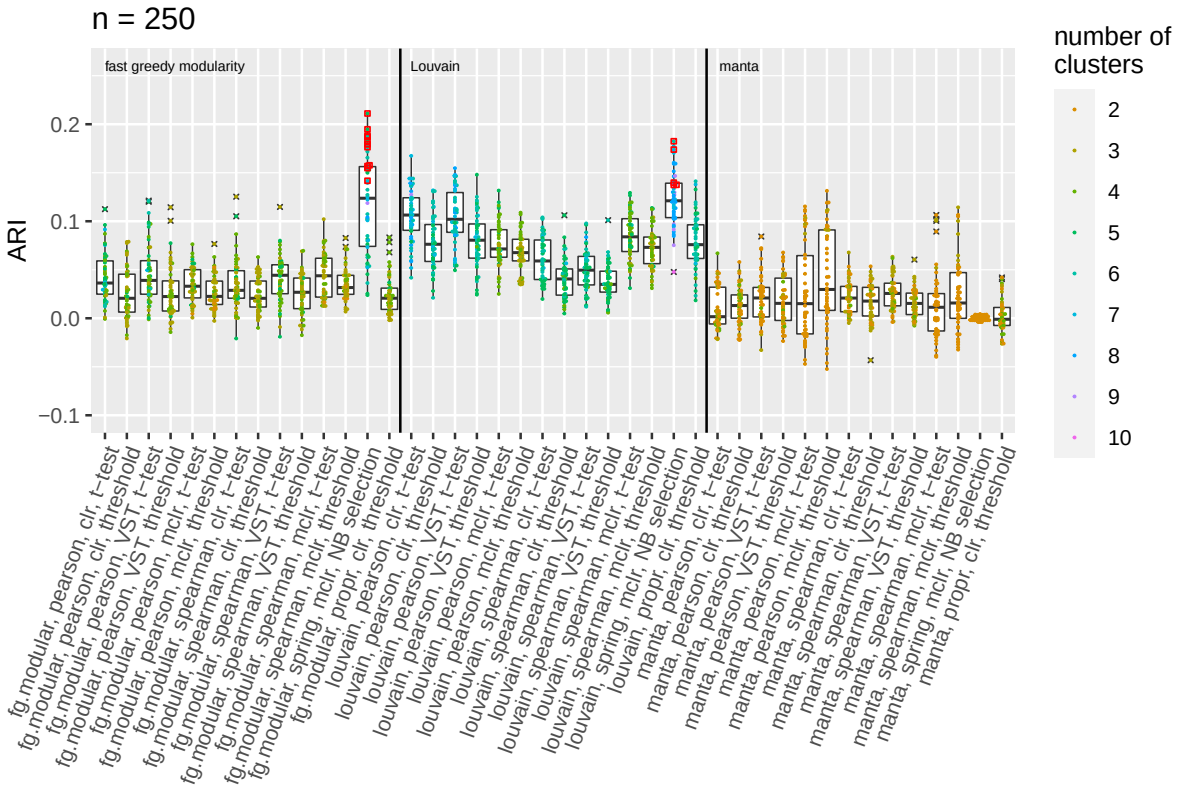


Fig G. Results for network-based clustering of bacterial genera on the discovery data, separated by sparsification methods, $n = 250$

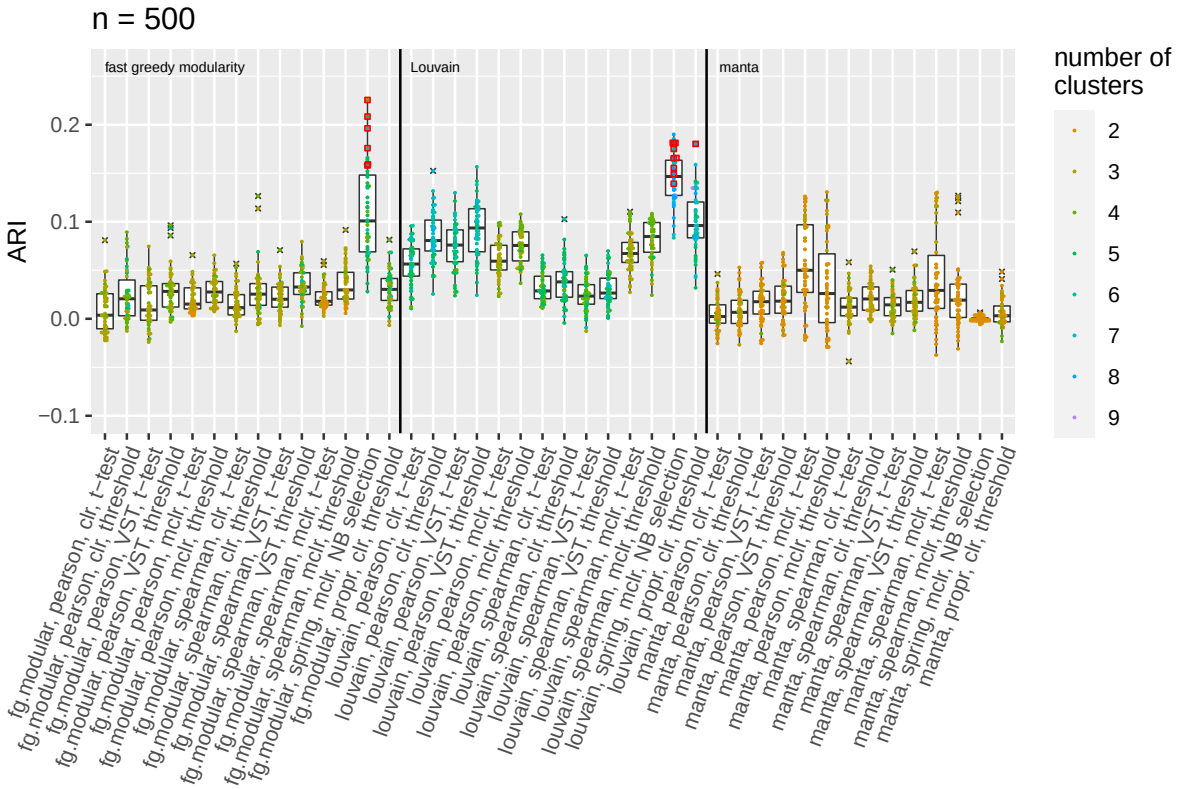


Fig H. Results for network-based clustering of bacterial genera on the discovery data, separated by sparsification methods, $n = 500$

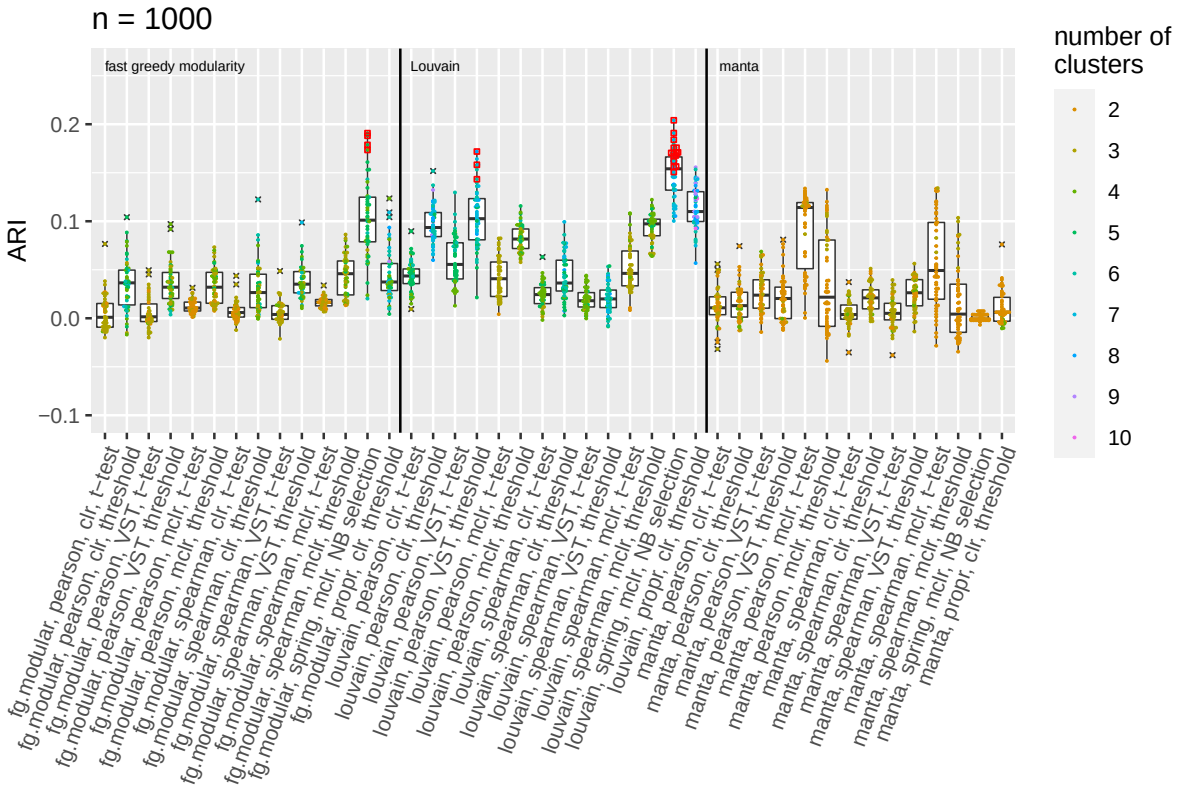


Fig I. Results for network-based clustering of bacterial genera on the discovery data, separated by sparsification methods, $n = 1000$

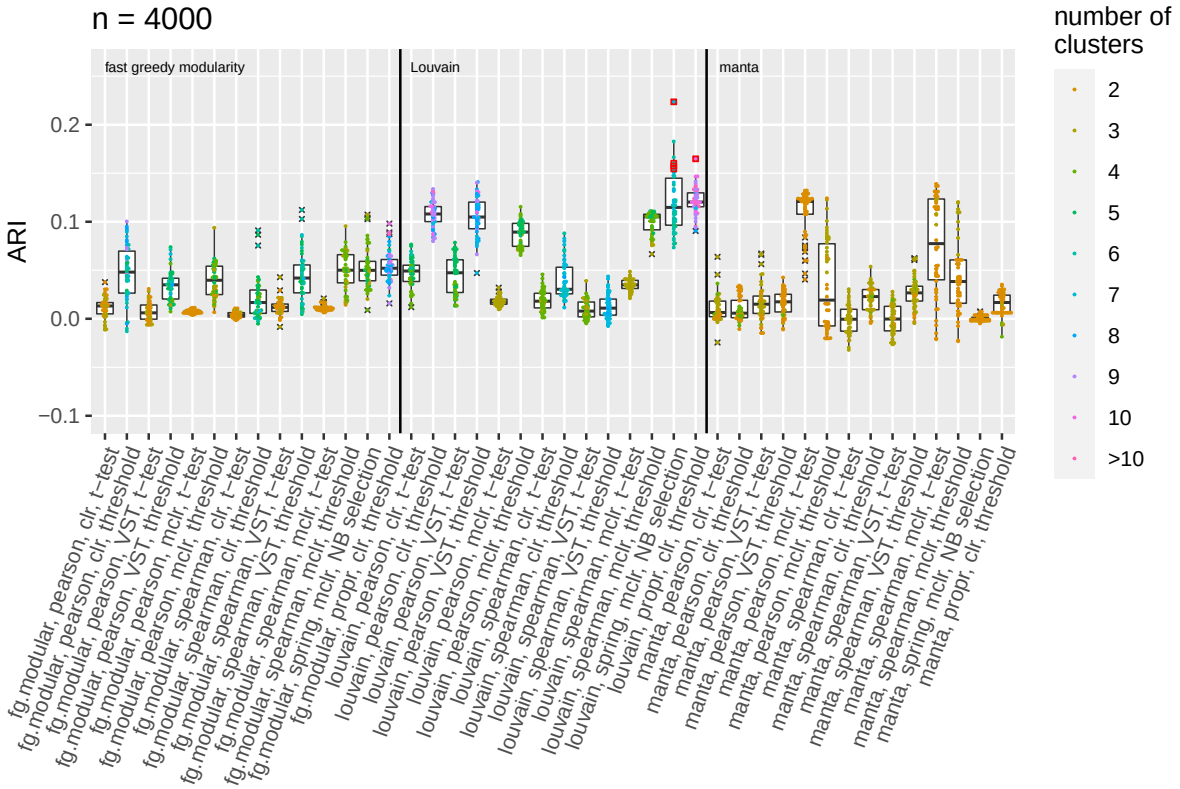


Fig J. Results for network-based clustering of bacterial genera on the discovery data, separated by sparsification methods, $n = 4000$

Our main interest lies in applying the "best" method to the validation data and checking whether the ARI result can be validated. The results are shown in Fig K-O. On the x-axis, the method combinations that were best in at least one of the 50 samplings are shown. The ARI values are shown as colored dots, with the color indicating the number k of clusters in the respective clustering result.

For each of the 50 samplings, the respective best method combination is applied to the validation data. The ARI value on the discovery data (belonging to the best method combination) and the corresponding ARI on the validation data are connected by lines. The lines point downwards in most cases, i.e., the results for the validation data are usually slightly worse than for the discovery data.

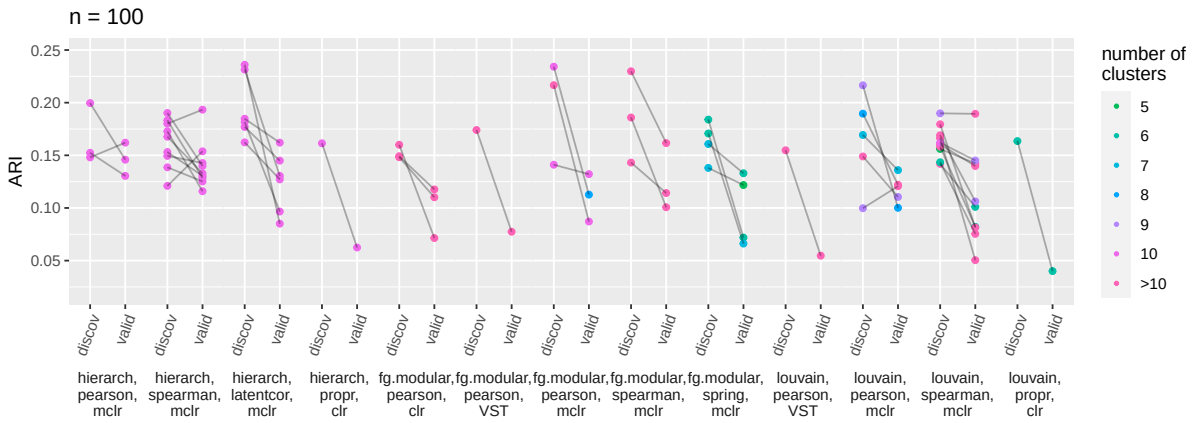


Fig K. Best ARIs for the clustering of bacterial genera on the discovery data, compared with the results on validation data, $n = 100$

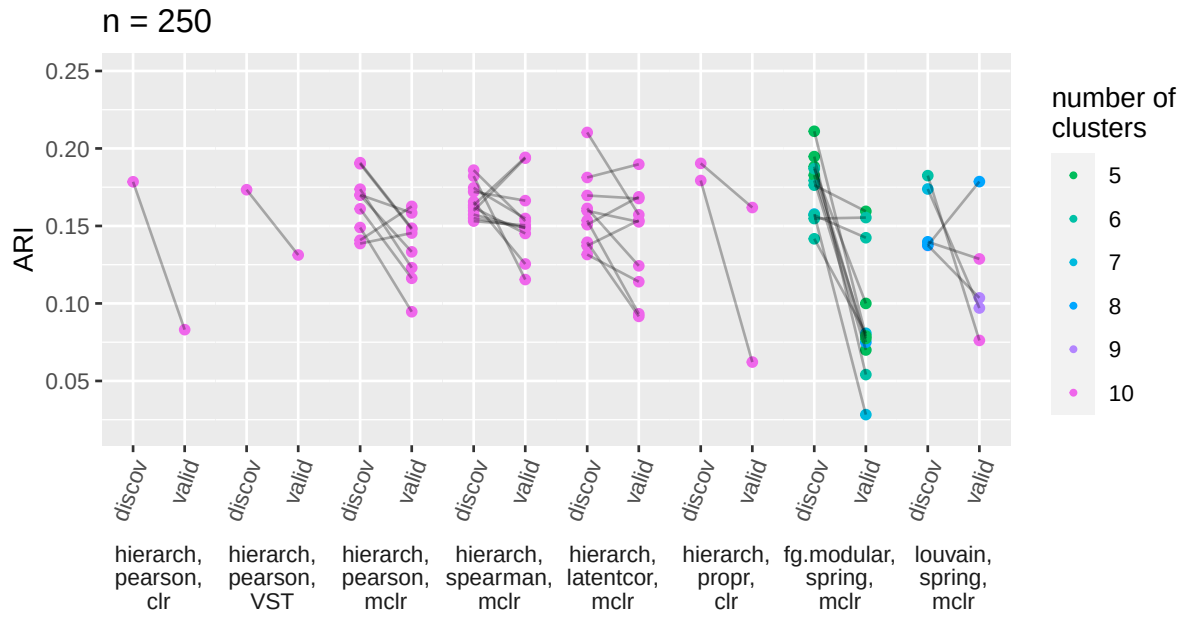


Fig L. Best ARIs for the clustering of bacterial genera on the discovery data, compared with the results on validation data, $n = 250$

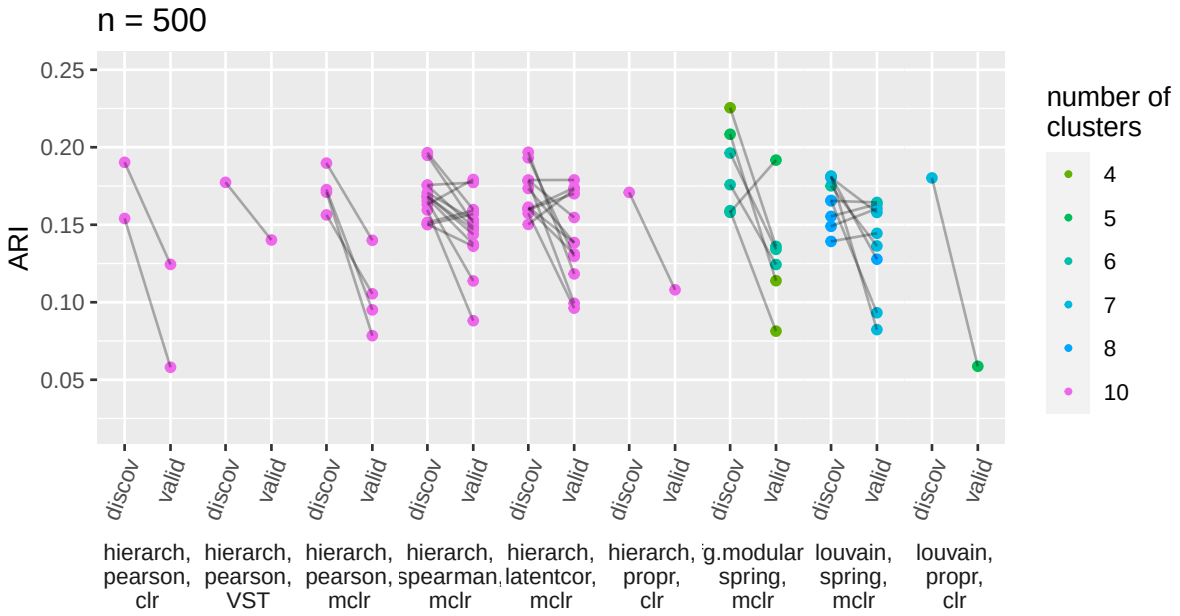


Fig M. Best ARIs for the clustering of bacterial genera on the discovery data, compared with the results on validation data, $n = 500$

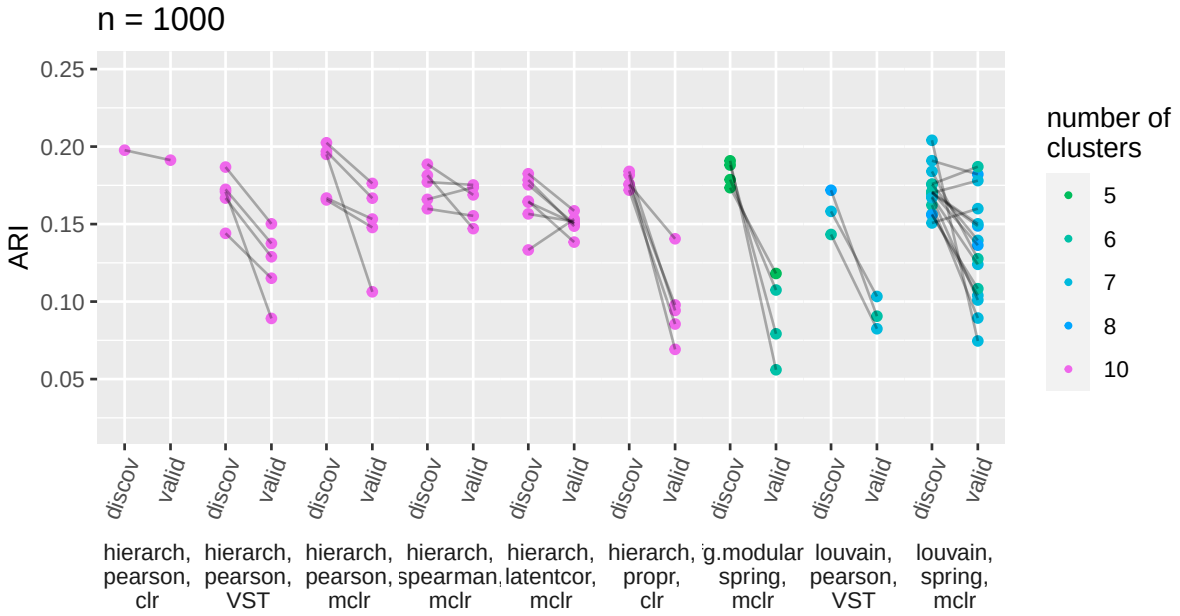


Fig N. Best ARIs for the clustering of bacterial genera on the discovery data, compared with the results on validation data, $n = 1000$

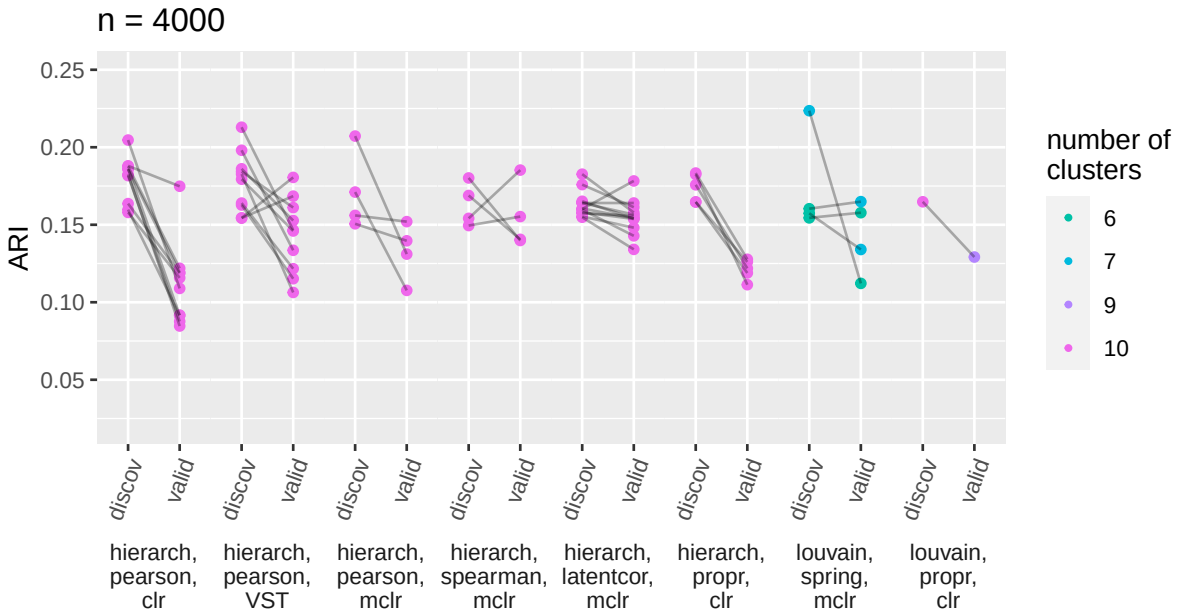


Fig O. Best ARIs for the clustering of bacterial genera on the discovery data, compared with the results on validation data, $n = 4000$

S2: Full results and plots for research task 2 (hub detection)

List of Figures

Fig A-E display the results of the hub detection applied to the discovery data. For each method combination on the x-axis, the 50 results obtained from 50 different discovery datasets are summarized as boxplots, indicating the number of detected hubs. Outliers are marked by black crosses. Results that were picked as the "best result" in one of the 50 samplings are marked by red squares.

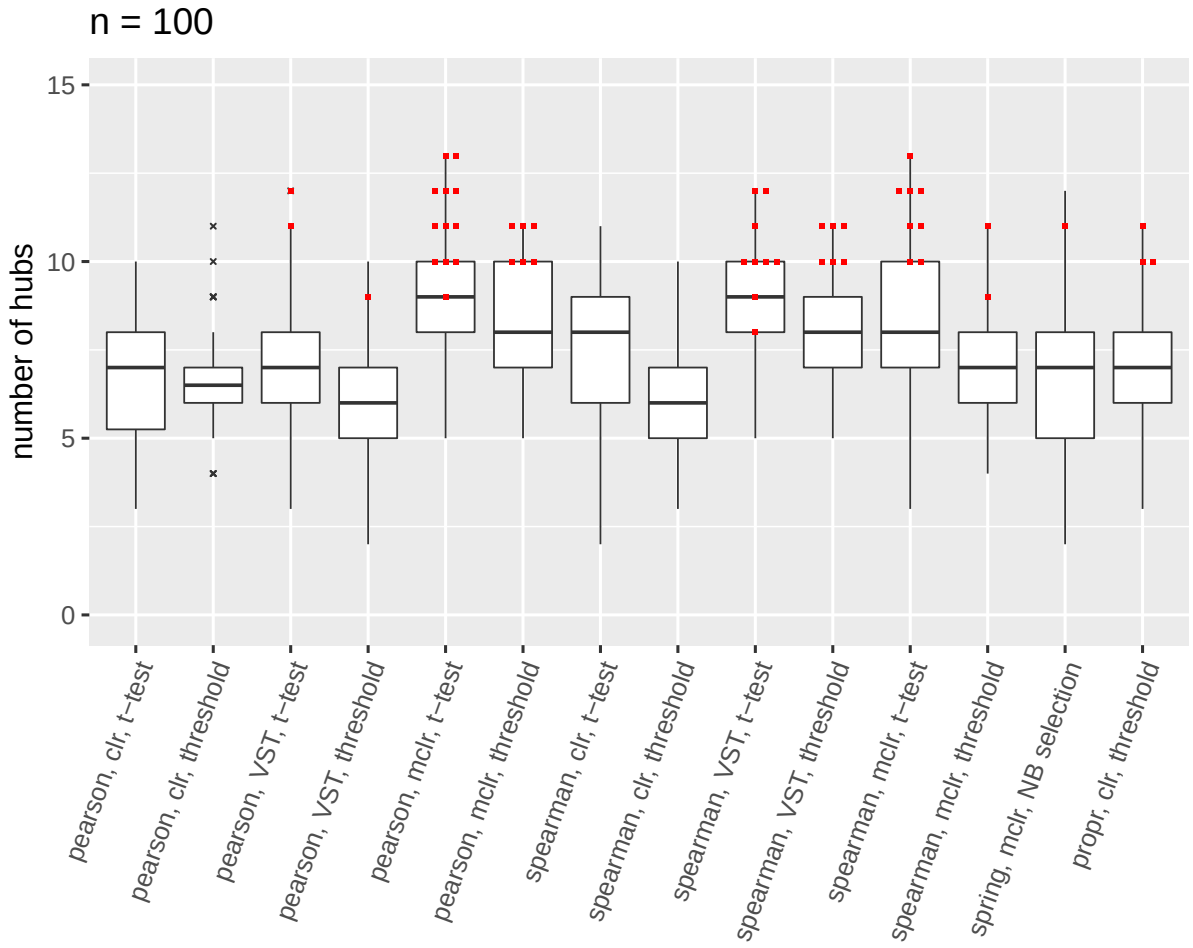


Fig A. Results for hub detection on the discovery data, $n = 100$

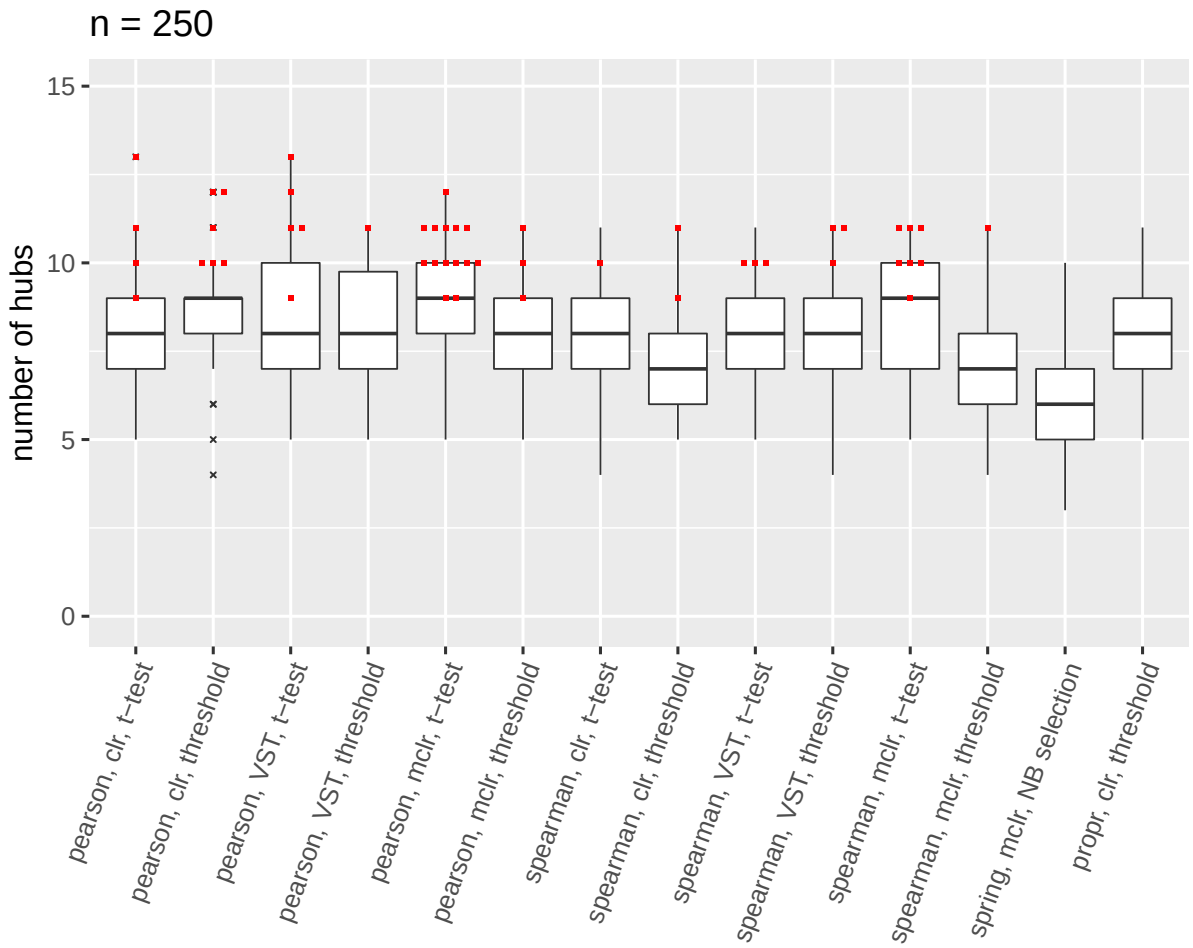


Fig B. Results for hub detection on the discovery data, $n = 250$

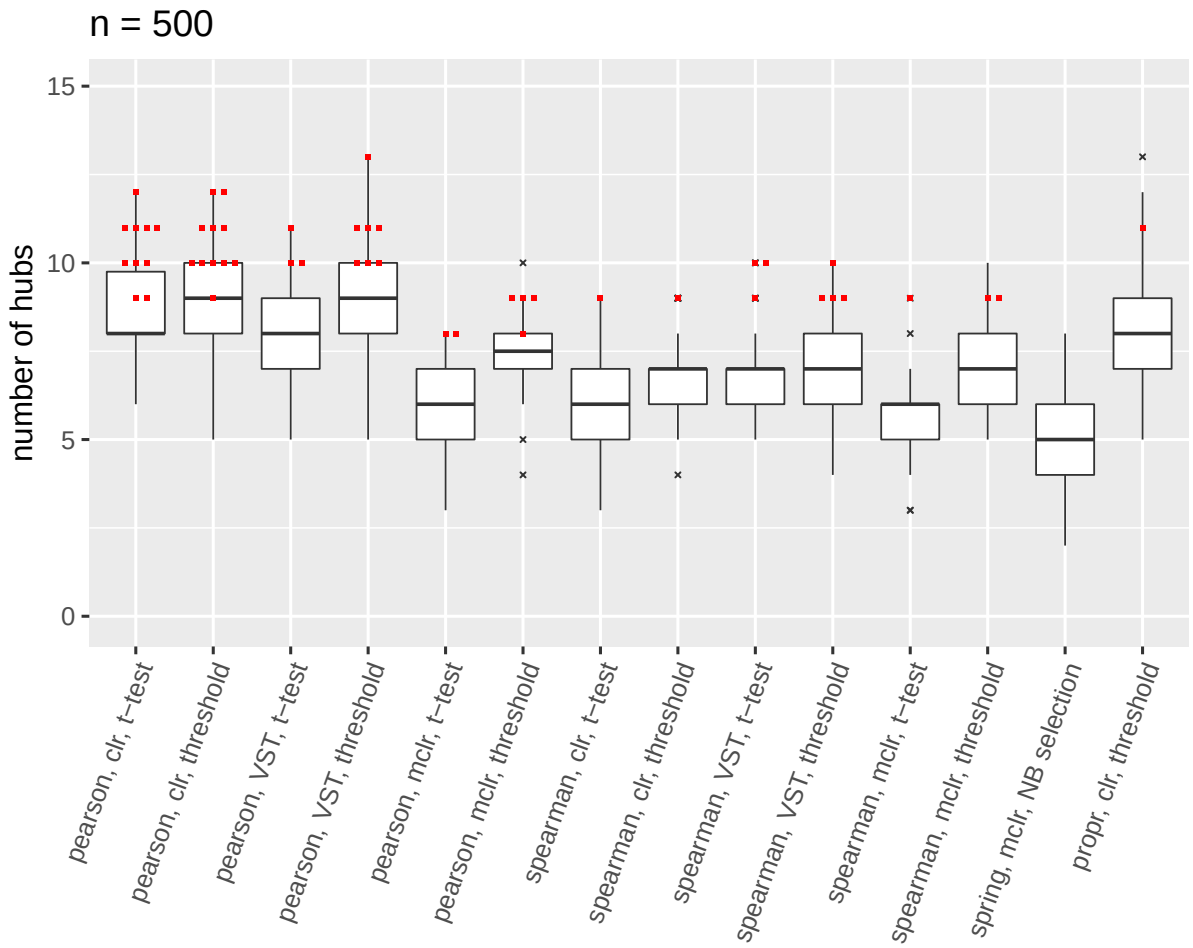


Fig C. Results for hub detection on the discovery data, $n = 500$

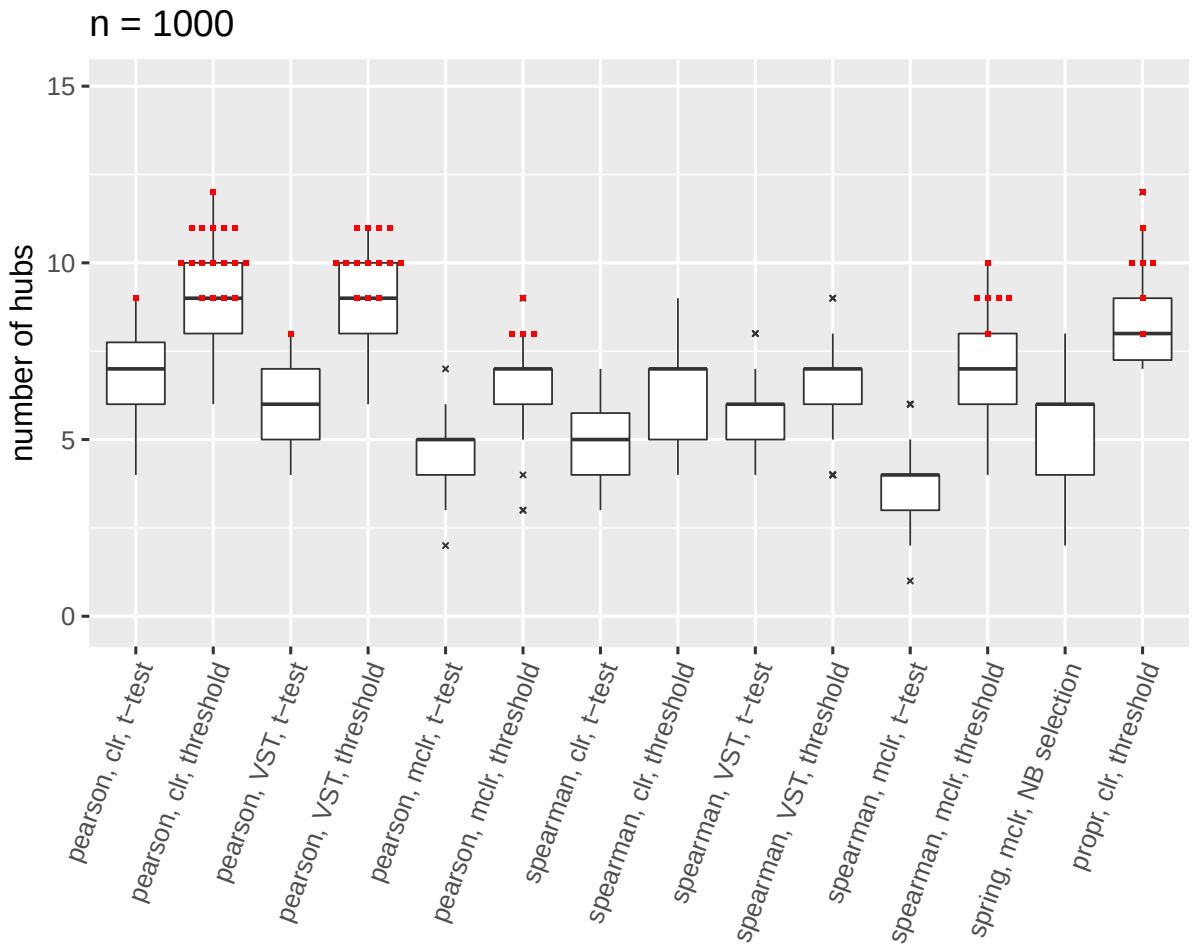


Fig D. Results for hub detection on the discovery data, $n = 1000$

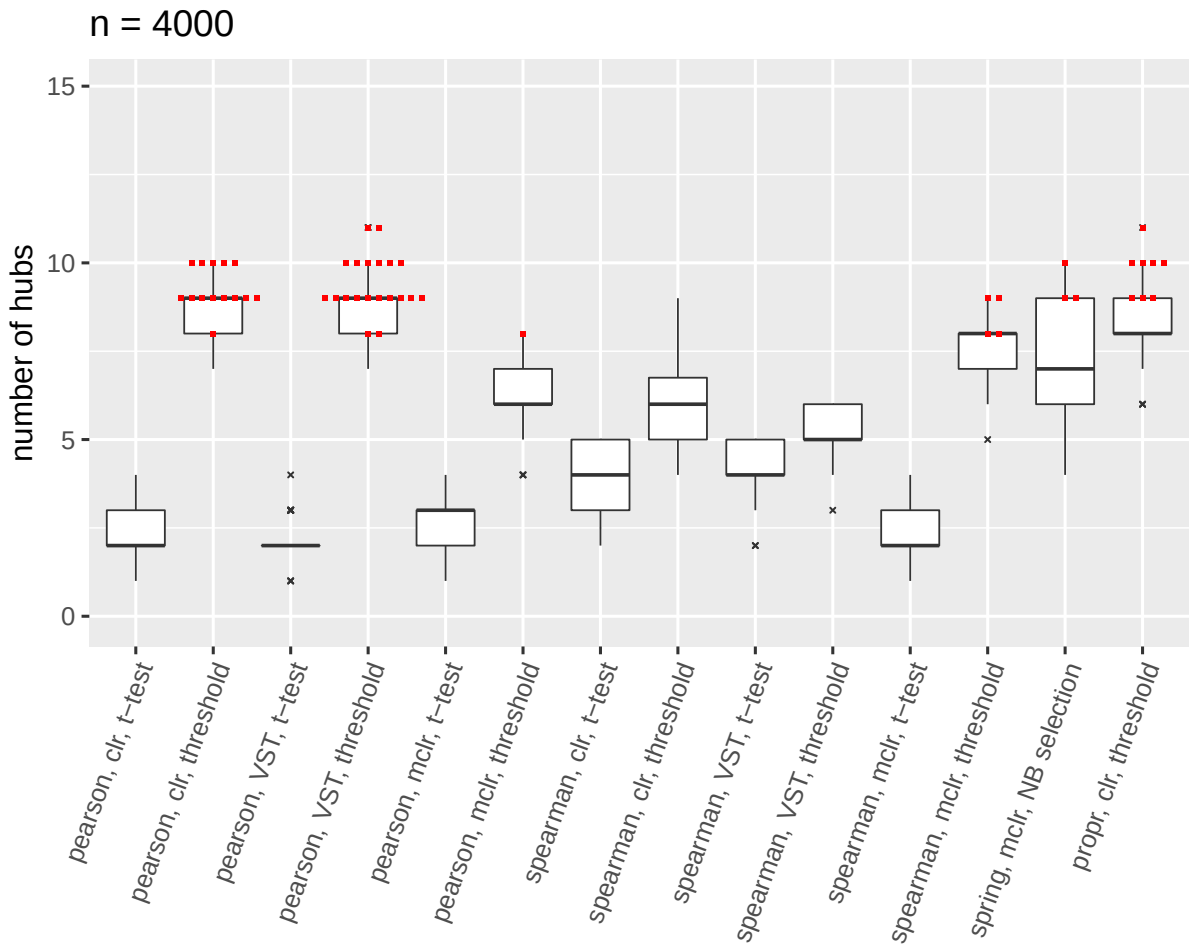


Fig E. Results for hub detection on the discovery data, $n = 4000$

There is not one single method combination that always yields the highest number of hubs. At $n = 100$, the best results are often found by Pearson correlation with mclr normalization, and Spearman correlation with VST or mclr normalization. With increasing sample size, Pearson correlation with clr or VST normalization frequently yields high number of hubs. As Fig D and Fig E show, for $n = 1000$ and $n = 4000$, sparsification of the network with the t-test generally leads to lower number of hubs compared to sparsification with the threshold method. At these sample sizes, the threshold method has a stronger sparsification effect than the t-test (given the chosen threshold of 0.15) and sparser networks tend to have more hubs for the chosen hub definition.

We consider the results of applying the chosen method combinations to the validation data. For each method combination that was chosen at least once as the "best" one, Fig F-J display the number of hubs obtained by the method on the discovery data vs. the number obtained by the same method on the validation data, where each square-dot combination corresponds to one of the 50 samplings. The results on discovery and validation data are connected by lines.

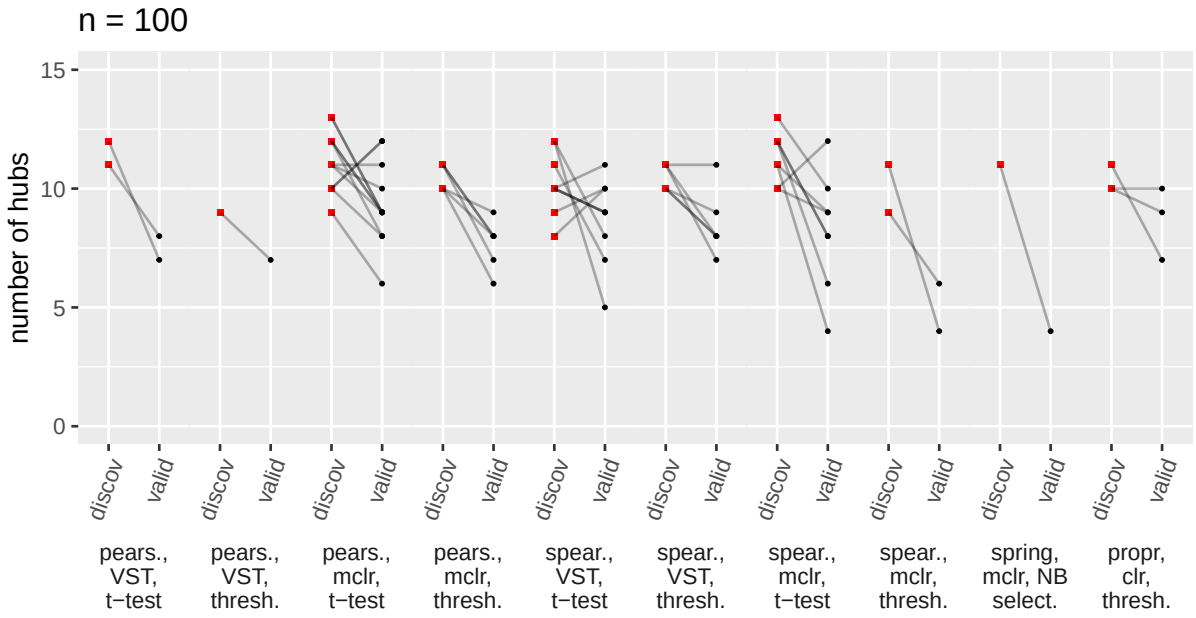


Fig F. Highest numbers of hubs for the hub detection on the discovery data, compared with the results on validation data, $n = 100$

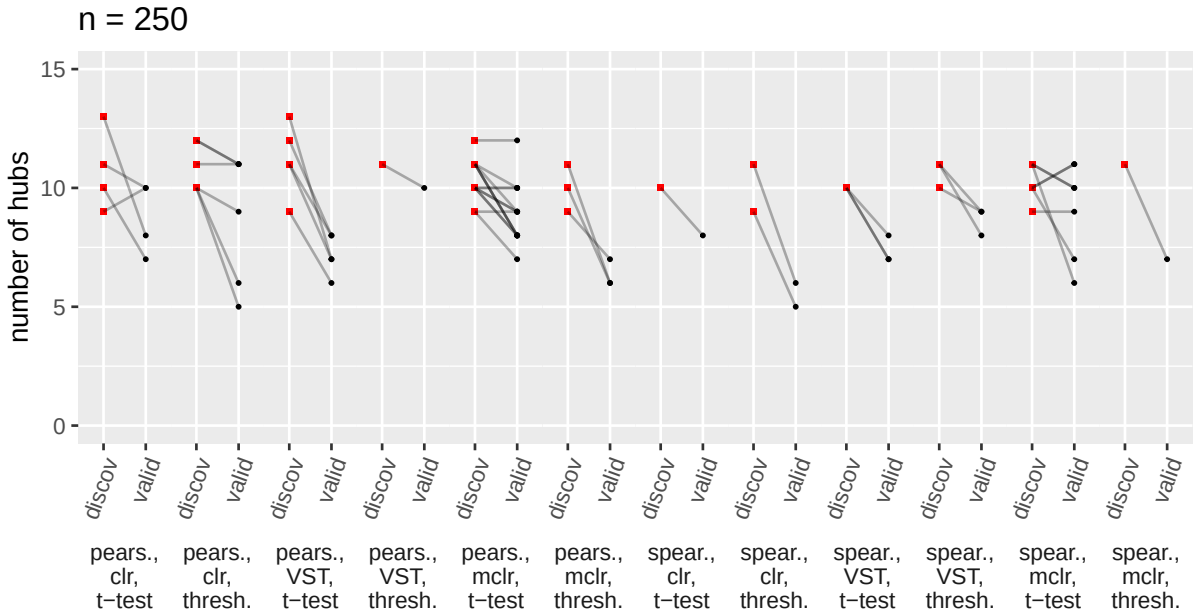


Fig G. Highest numbers of hubs for the hub detection on the discovery data, compared with the results on validation data, $n = 250$

Fig H. Highest numbers of hubs for the hub detection on the discovery data, compared with the results on validation data, $n = 500$

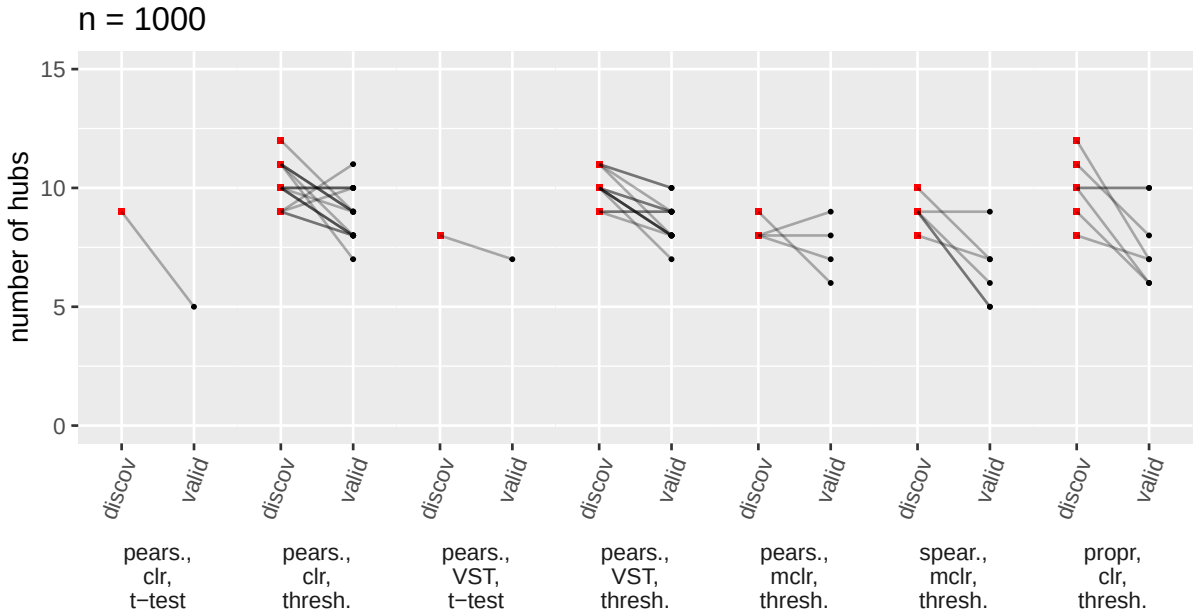


Fig I. Highest numbers of hubs for the hub detection on the discovery data, compared with the results on validation data, $n = 1000$

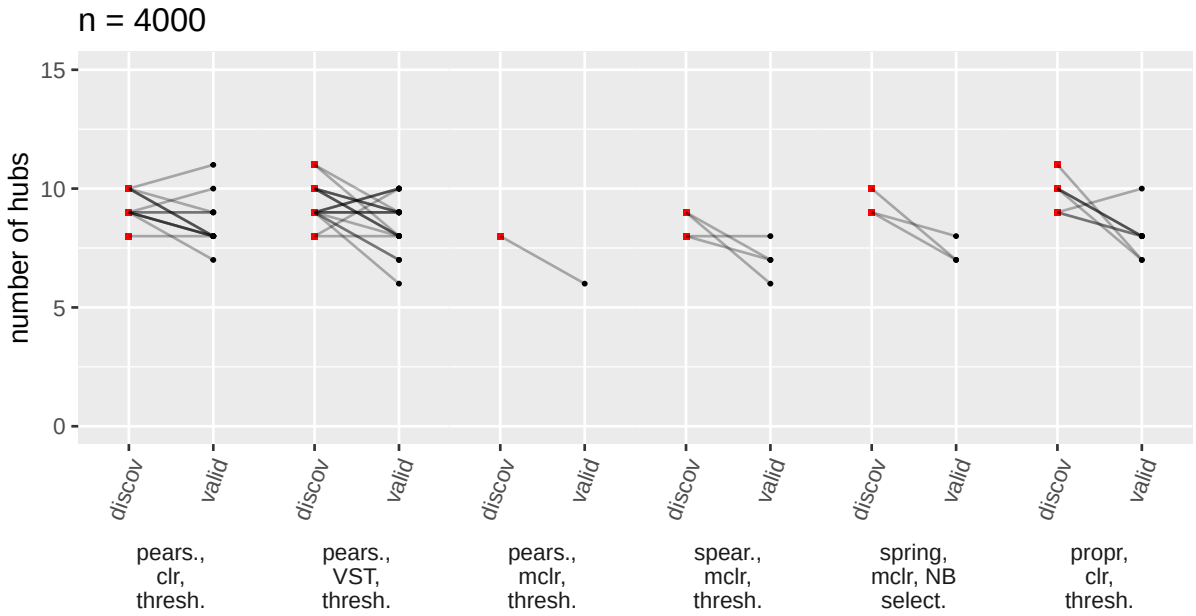


Fig J. Highest numbers of hubs for the hub detection on the discovery data, compared with the results on validation data, $n = 4000$

The lines point downwards in the majority of the 50 samplings, indicating worse results regarding the network's hubbiness on the validation data.

S3: Full results and plots for research task 3 (differential network analysis)

List of Figures

Fig A-C show the results of the differential network analysis on the discovery data over 50 samplings. Boxplots summarize the GCDs between the microbial network based on the non-antibiotics samples vs. the network based on the antibiotics samples. GCDs that were picked as the "best" results are marked by red squares.

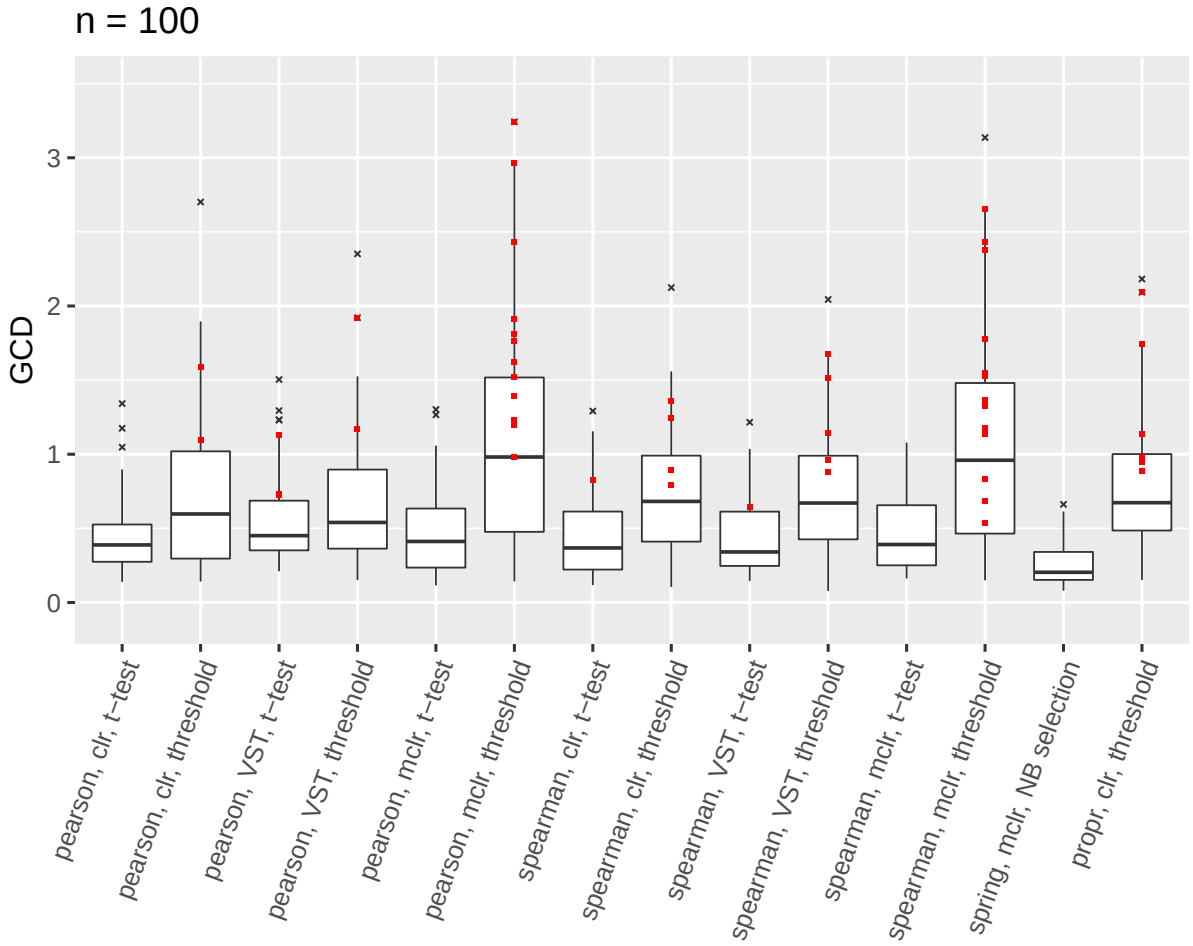


Fig A. Results for differential network analysis on the discovery data, $n = 100$

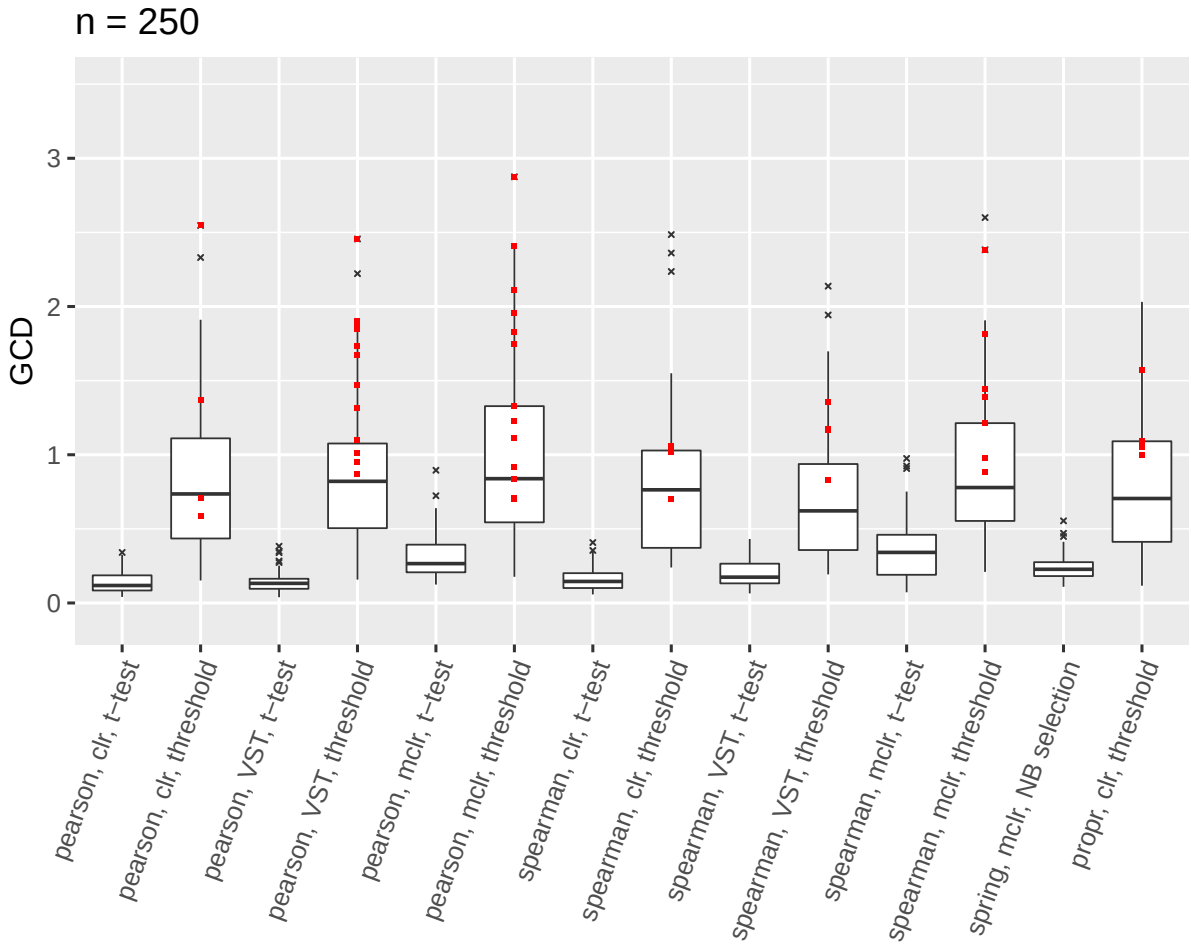


Fig B. Results for differential network analysis on the discovery data, $n = 250$

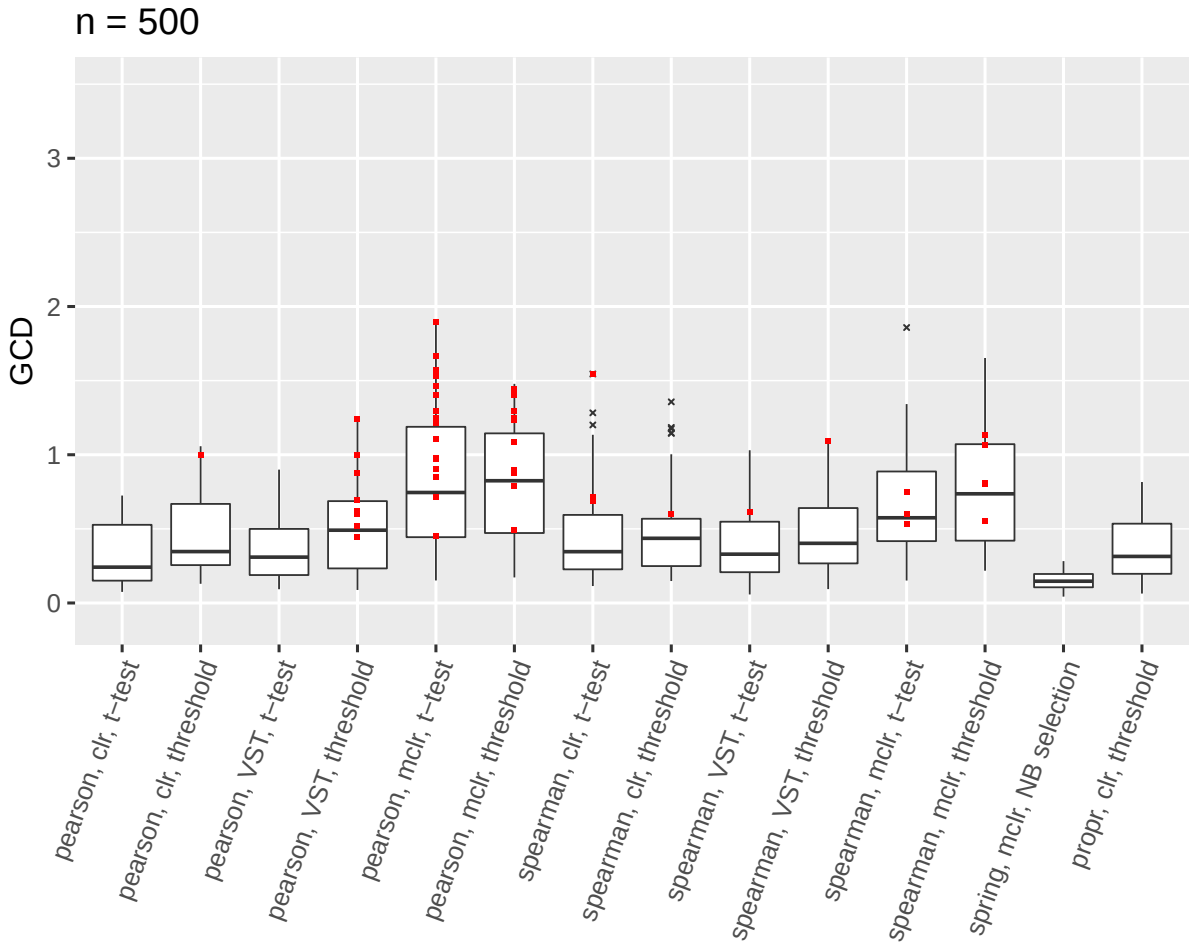


Fig C. Results for differential network analysis on the discovery data, $n = 500$

Similar to hub detection, there is no superior method combination that always leads to best results, i.e., highest GCD values. Notably, sparsification via t-test never leads to best results for $n = 100$ and $n = 250$, but only for $n = 500$. However, a general trend cannot be confirmed due to the limited sample size in this research task.

Fig D-F show the results of applying the best method combinations to the validation data and compare these to the results on the discovery data. Over-optimism is indicated by downward lines, which is the case in about 75% of the 50 samplings.

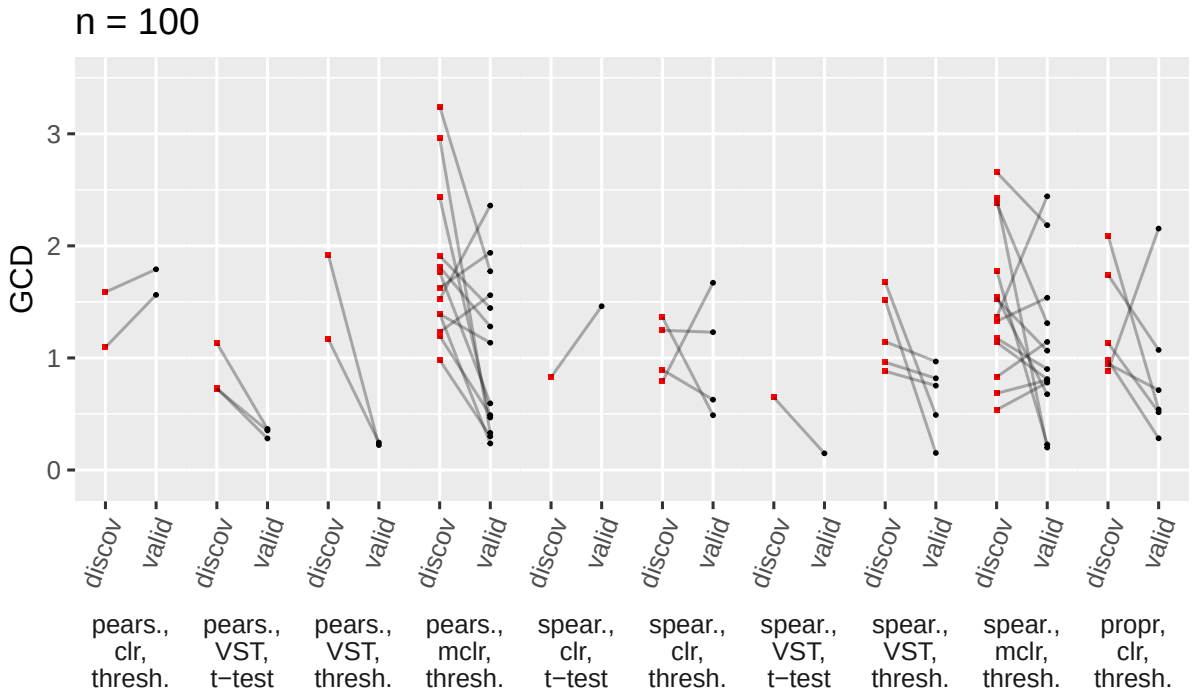


Fig D. Largest GCDs for the differential network analysis on the discovery data, compared with the results on validation data, $n = 100$

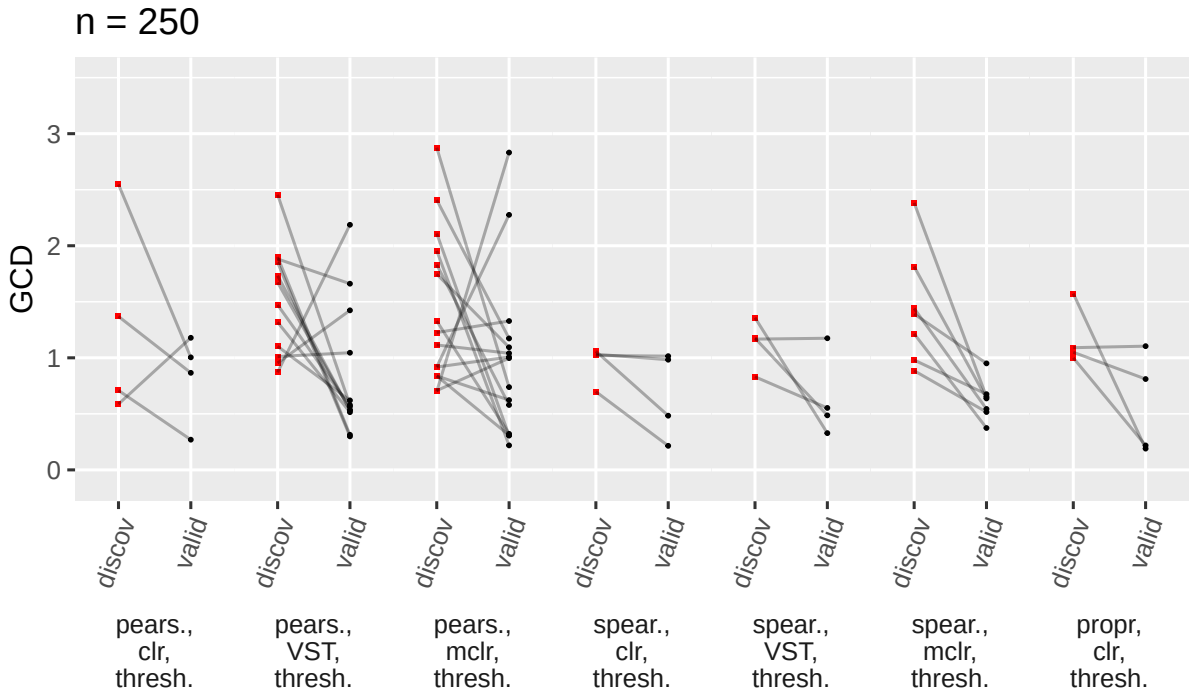


Fig E. Largest GCDs for the differential network analysis on the discovery data, compared with the results on validation data, $n = 250$

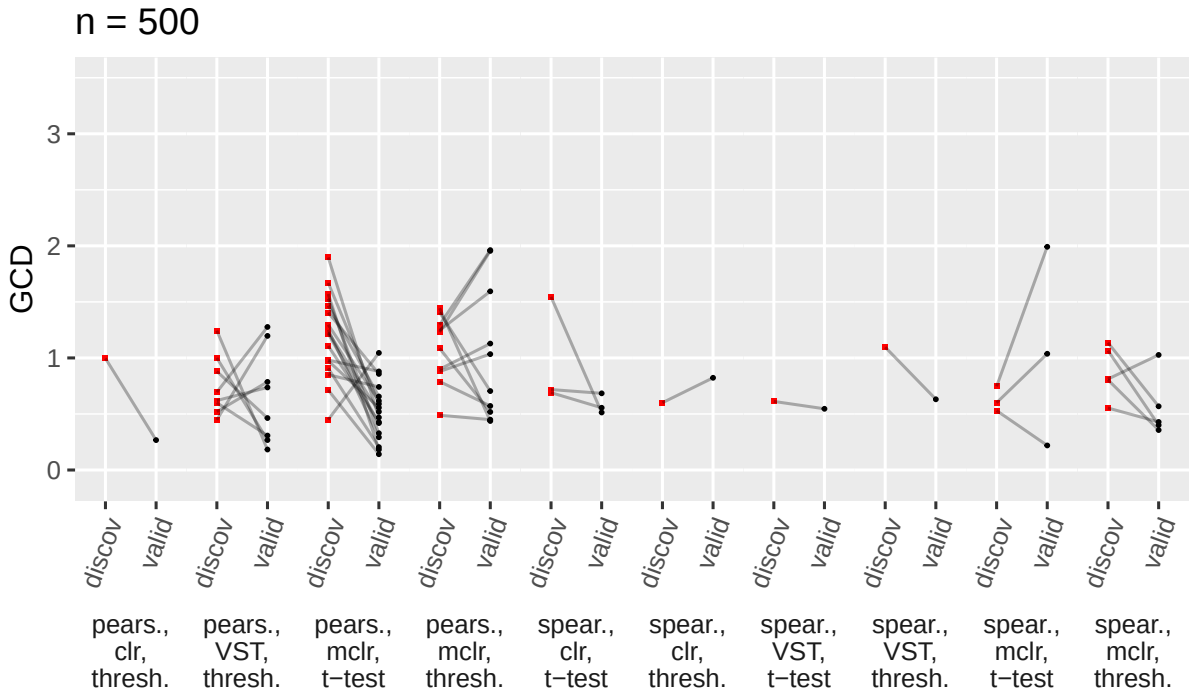


Fig F. Largest GCDs for the differential network analysis on the discovery data, compared with the results on validation data, $n = 500$

S4: Full results and plots for research task 4 (clustering of samples)

List of Figures

Fig A-E display the results of the sample clustering on the discovery data over 50 samplings. The ASW results are summarized by boxplots and are additionally shown as colored dots, with the color indicating the number k of clusters in the respective clustering result. Results picked as the "best result" in one of the 50 samplings are marked by red square edges. For the network-based clustering methods (fast greedy modularity optimization and the Louvain method), the results for threshold and K-nearest neighbor sparsification are displayed together, i.e., 50*2 = 100 results are shown for these method combinations.

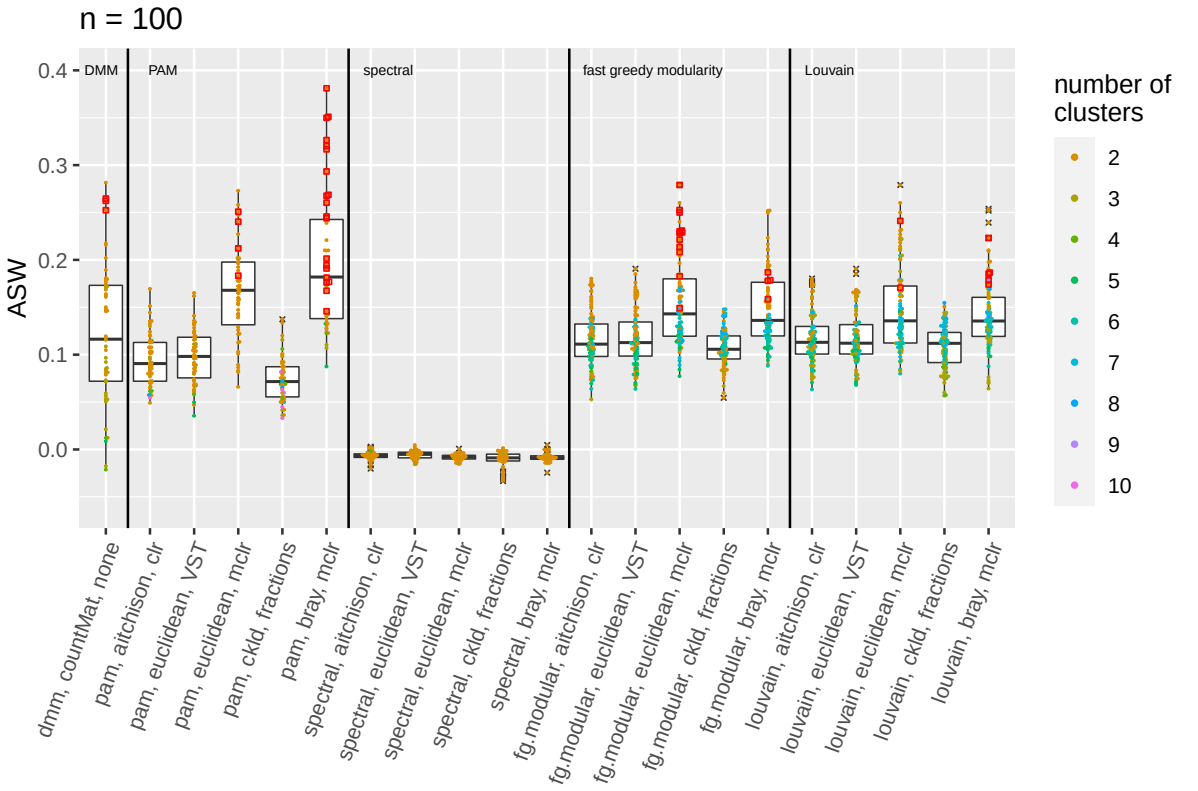


Fig A. Results for clustering samples on the discovery data, $n = 100$

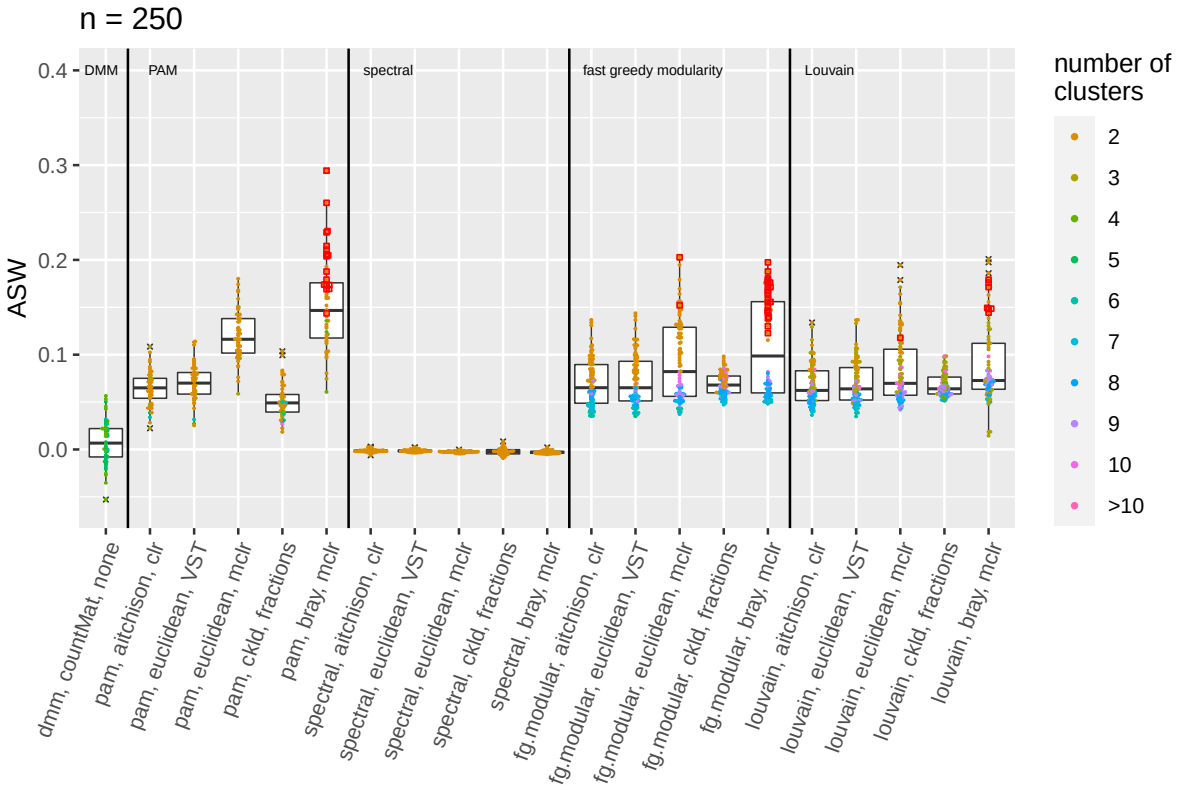


Fig B. Results for clustering samples on the discovery data, $n = 250$

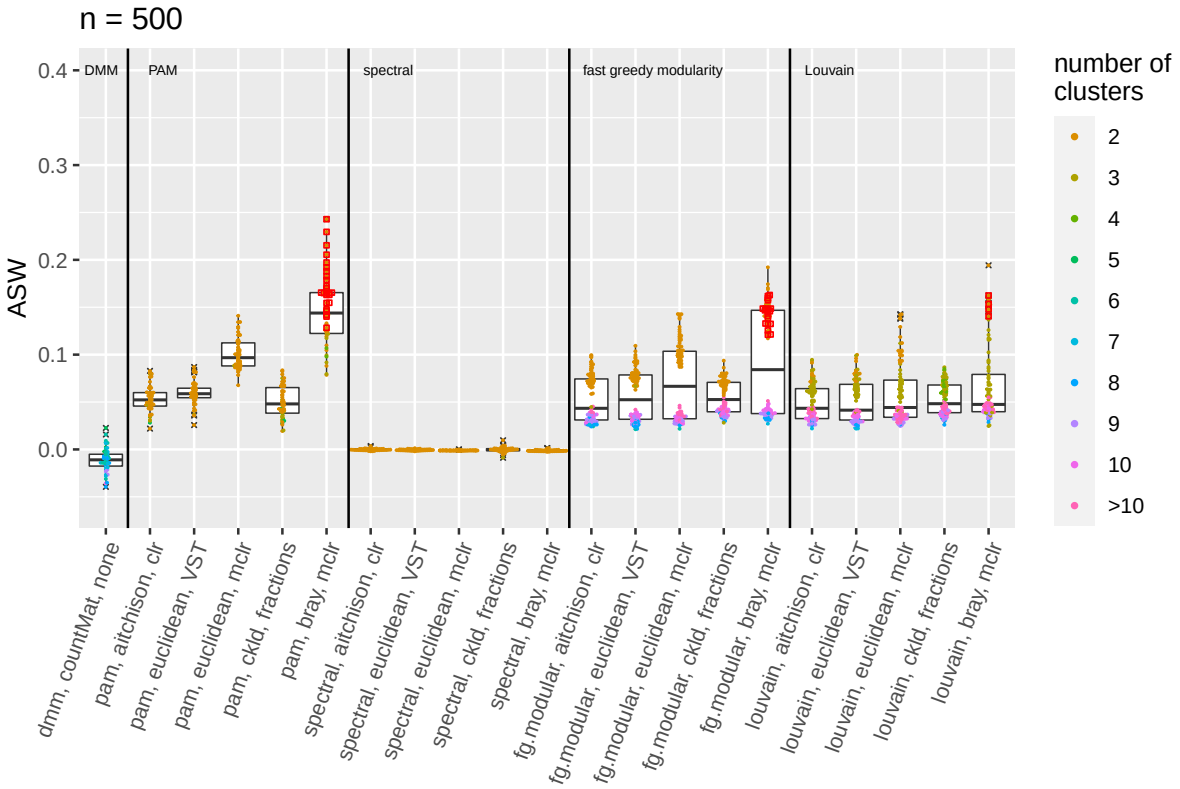


Fig C. Results for clustering samples on the discovery data, $n = 500$

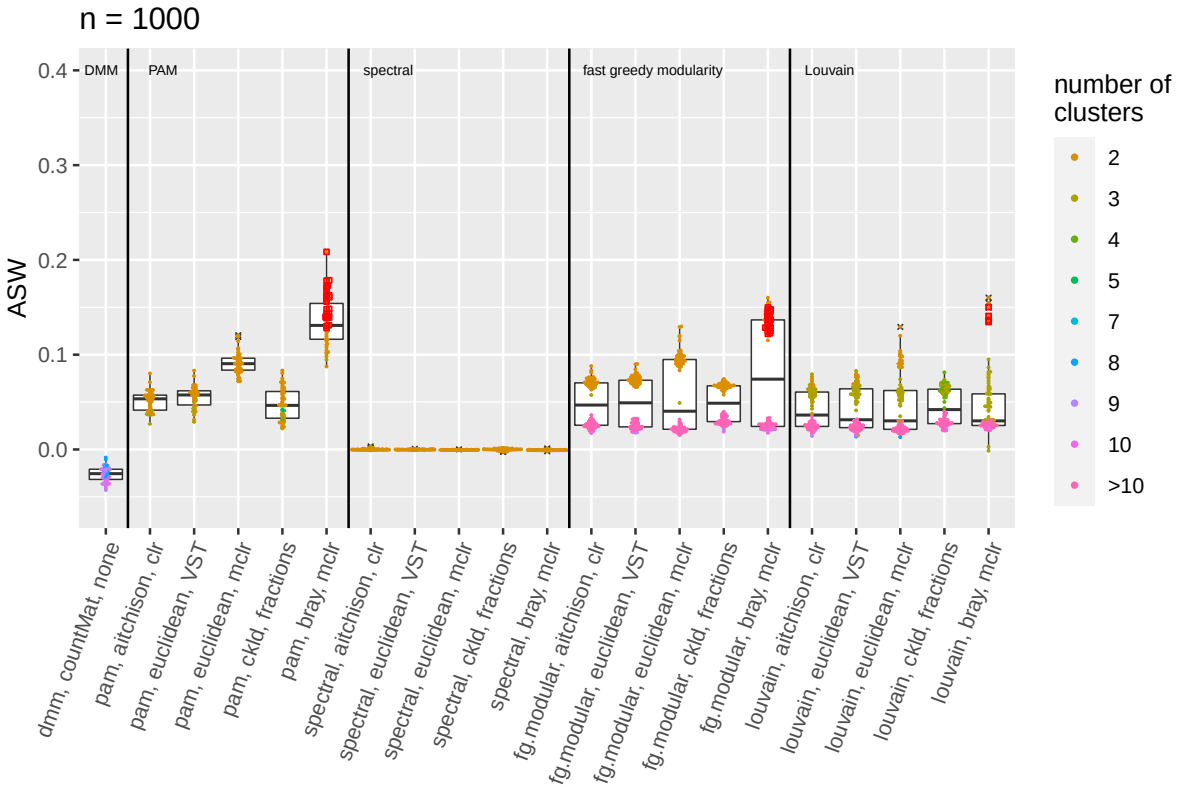


Fig D. Results for clustering samples on the discovery data, $n = 1000$

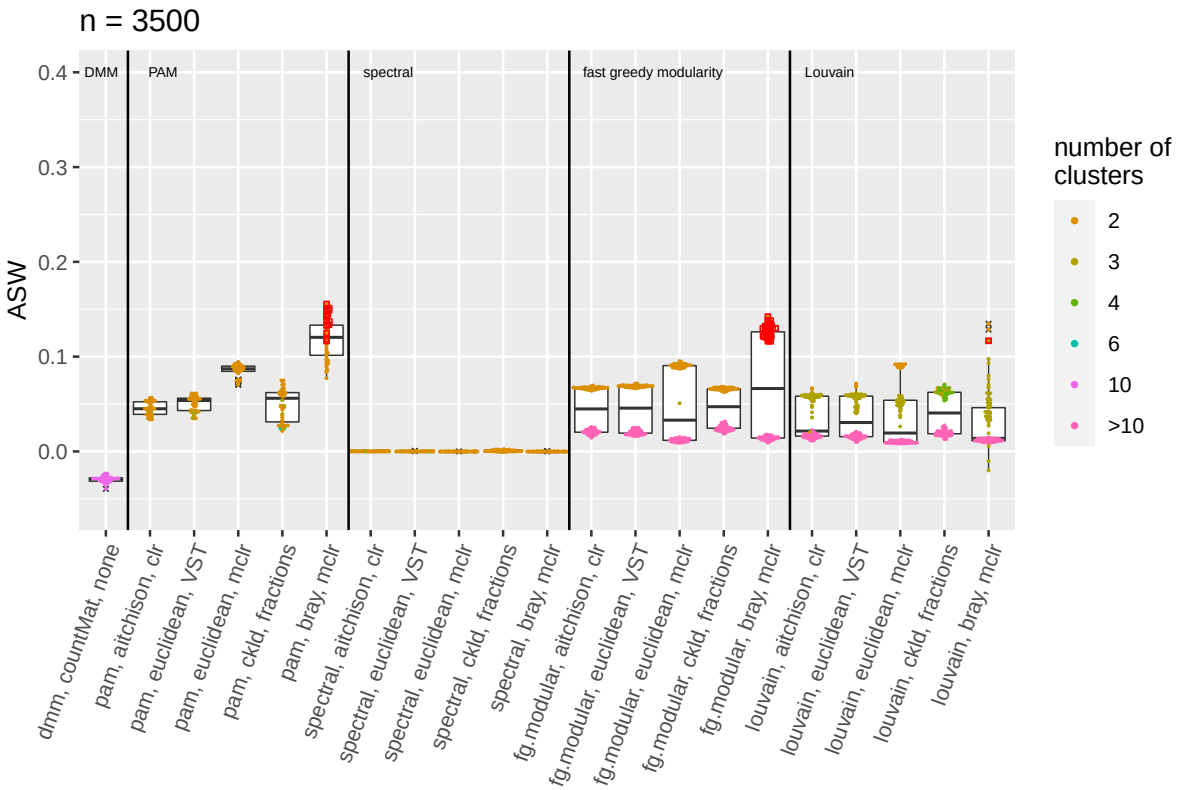


Fig E. Results for clustering samples on the discovery data, $n = 3500$

Overall, the ASW values are not particularly large, indicating at best a moderate quality of the clustering. Such results are not uncommon in enterotype research; for example, the original study about enterotypes [1] reported ASW values that were less than or equal to 0.25. From a sample size of $n = 250$ upwards, the number of clusters is mostly chosen as two or three, which fits with previous results from studies about enterotypes [1, 2, 3, 4]. Similar to the other three research tasks, there is not a single method combination that always yields the best results. PAM, fast greedy modularity optimization and the Louvain method are frequently chosen as the best clustering methods, often in combination with the Bray-Curtis dissimilarity and mclr normalization. DMM clustering performs reasonably well for $n = 100$, but does not yield good ASW values for the other sample sizes. Spectral clustering yields ASW values around zero for all sample sizes.

Fig F-J depict the results for the network-based clustering (fast greedy modularity optimization and the Louvain method) separately for both sparsification methods (threshold and K-nearest neighbors). Results that were picked as the "best result" in one of the 50 samplings are marked by red square edges. As the figures show, sparsification with the threshold method leads to smaller numbers of clusters and to larger ASW values. The threshold method has a weaker sparsification effect than the K-nearest neighbor method (given the chosen threshold of 0.85 and the number of nearest neighbors set to $K = 3$), and the cluster algorithms tend to find fewer clusters in denser (less sparse) networks. Similar to the previous research tasks, this demonstrates that network sparsification can have a notable effect on the final results.

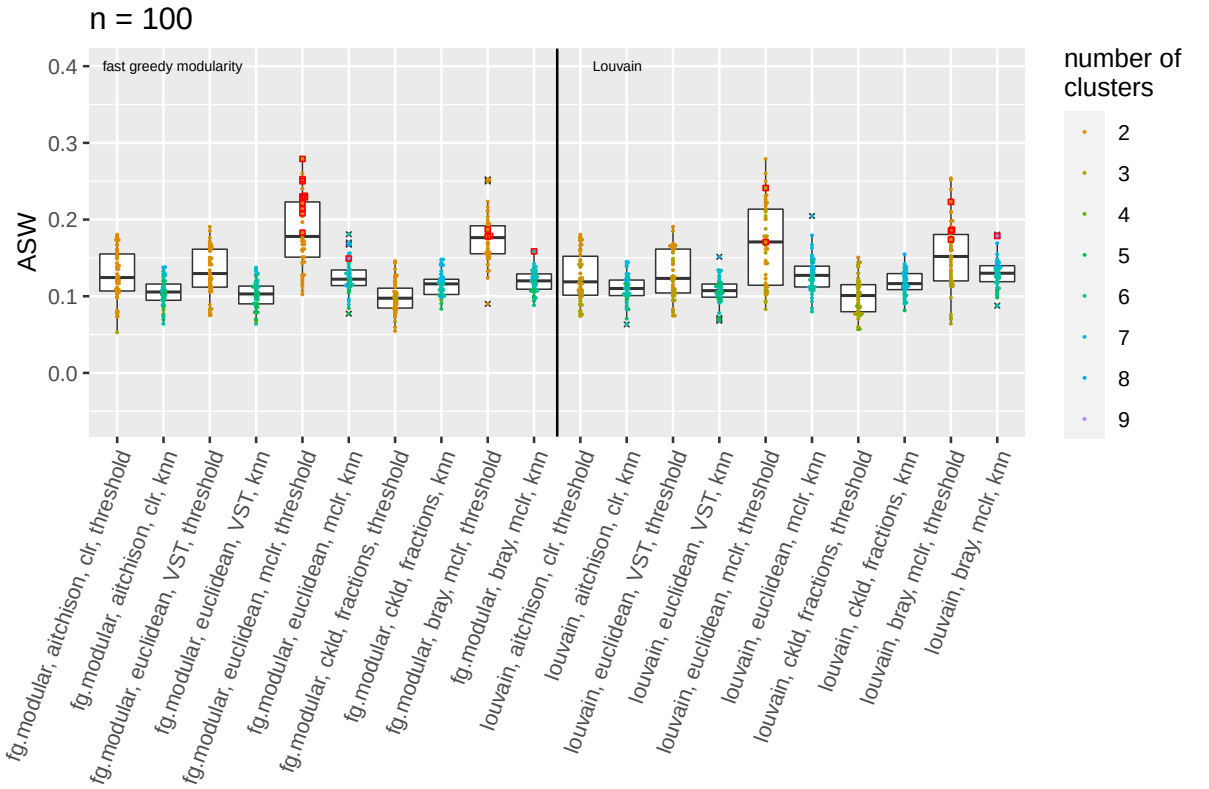


Fig F. Results for network-based clustering of samples on the discovery data, separated by sparsification methods, $n = 100$

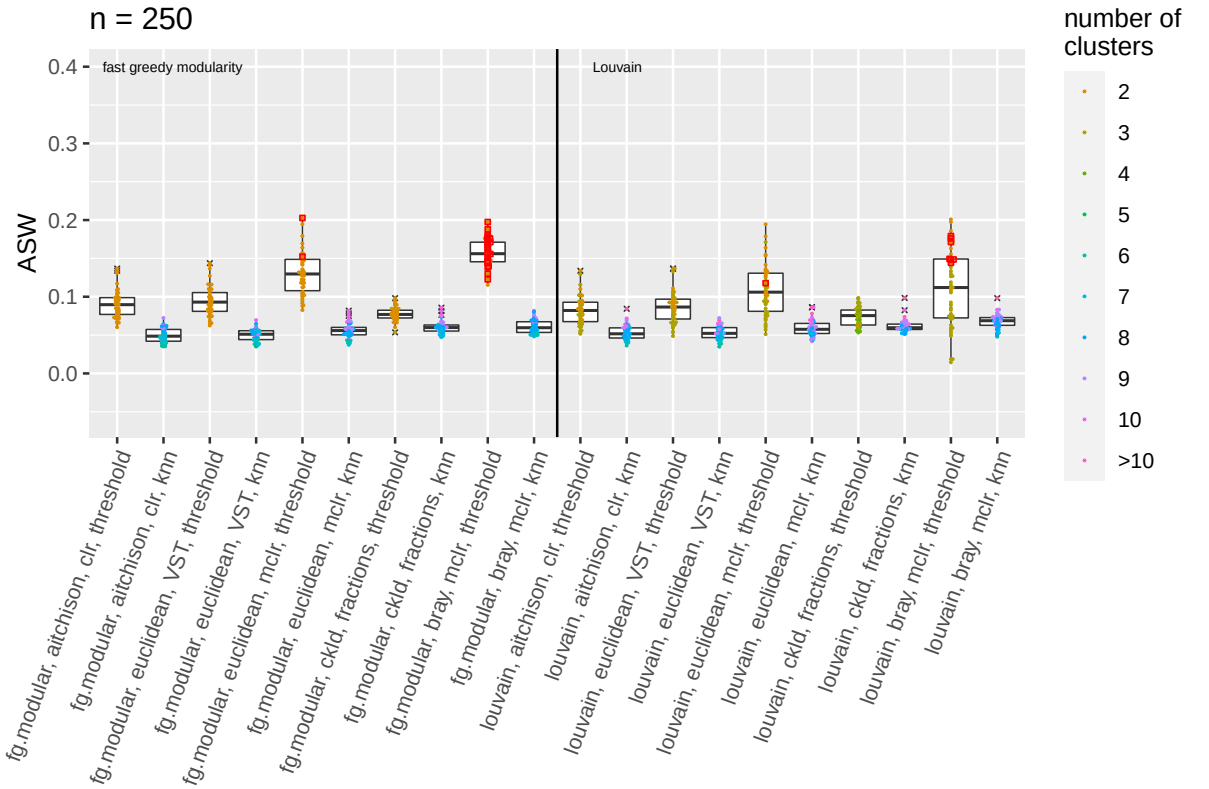


Fig G. Results for network-based clustering of samples on the discovery data, separated by sparsification methods, $n = 250$

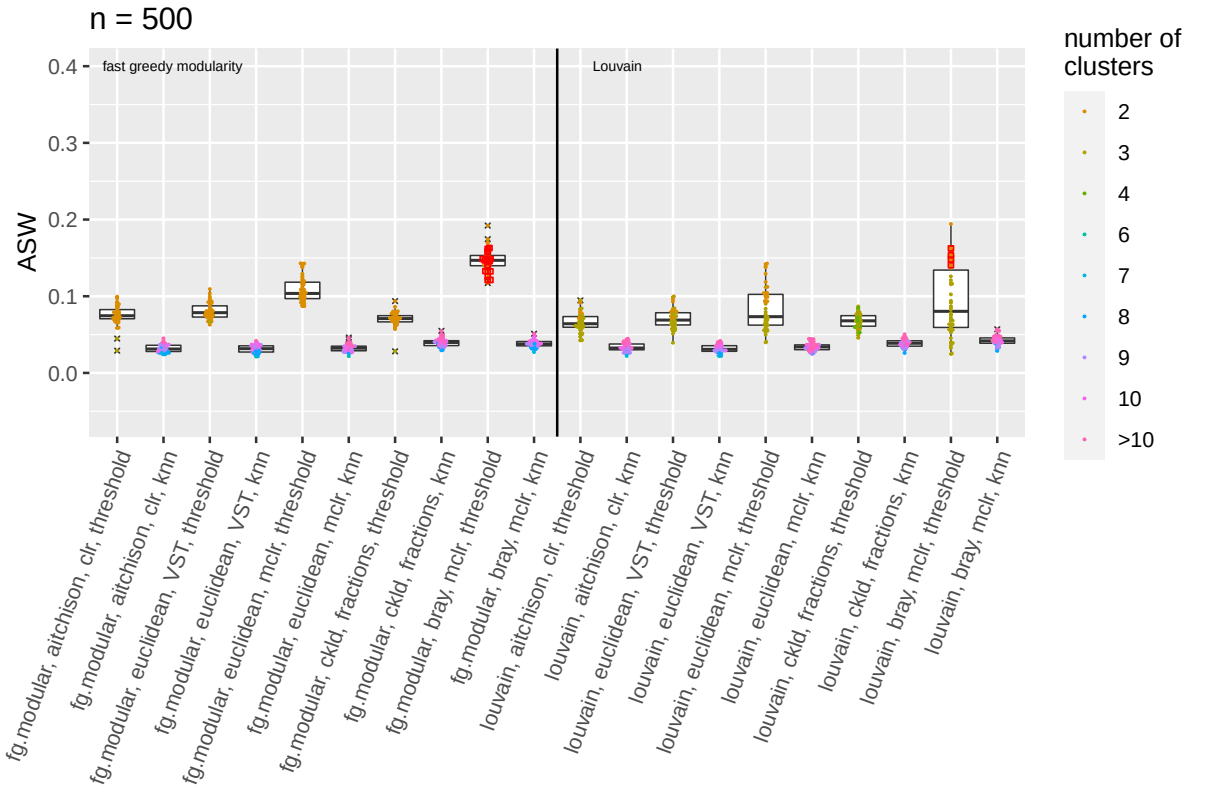


Fig H. Results for network-based clustering of samples on the discovery data, separated by sparsification methods, $n = 500$

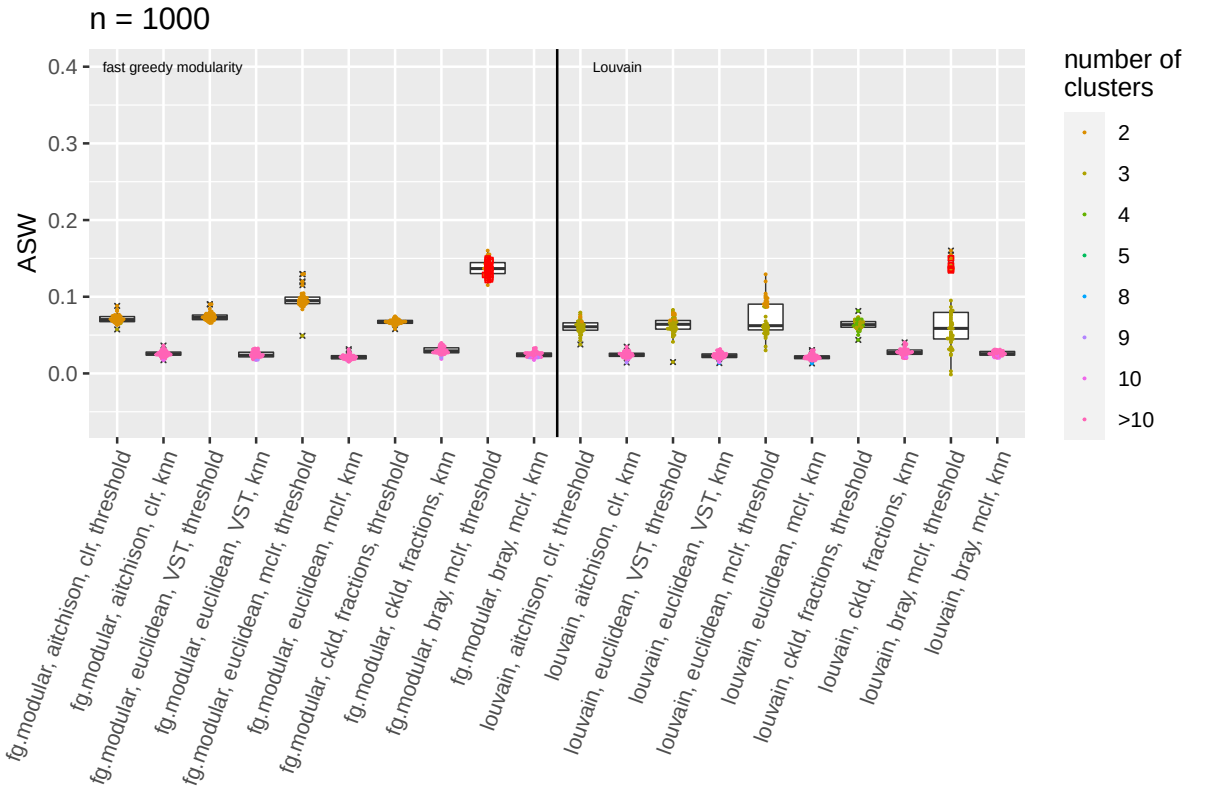


Fig I. Results for network-based clustering of samples on the discovery data, separated by sparsification methods, $n = 1000$

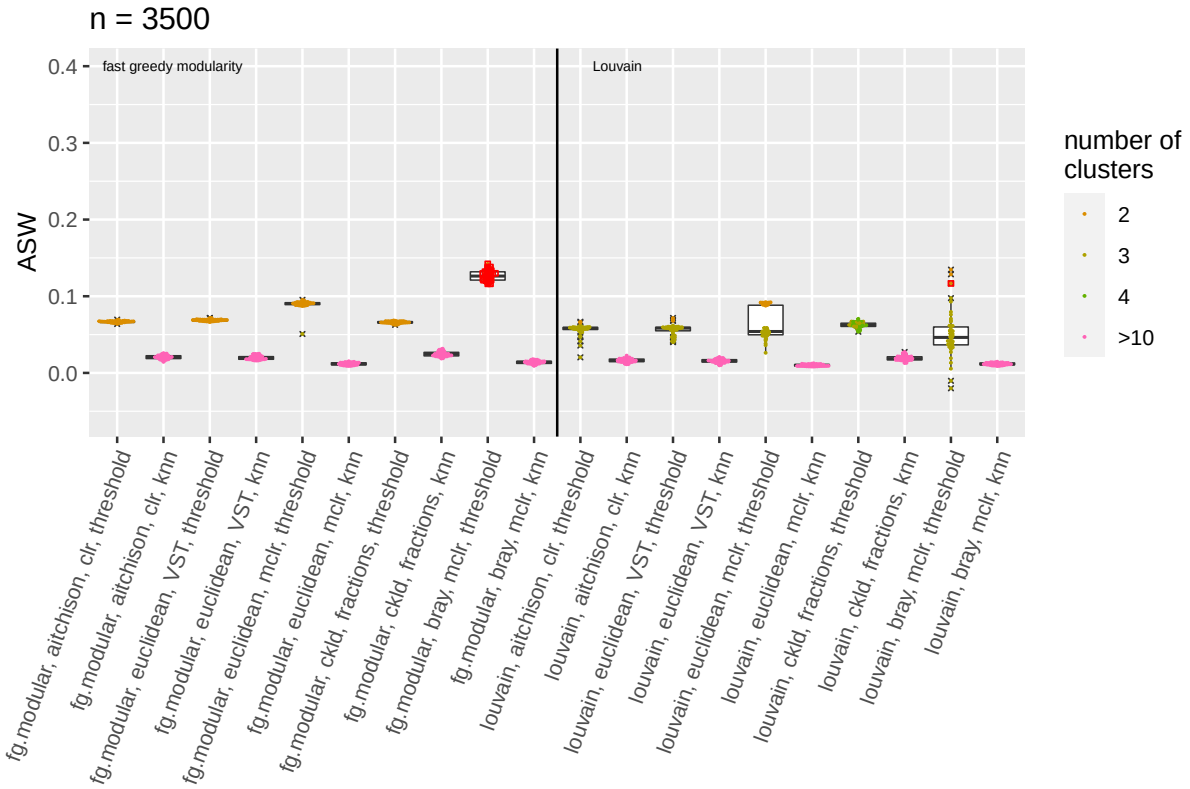


Fig J. Results for network-based clustering of samples on the discovery data, separated by sparsification methods, $n = 3500$

Fig K-O compare the ASW values resulting from the best method combinations on the discovery data to the corresponding ASW values on the validation data. Lines that point downwards indicate over-optimistic bias. For $n = 100$ and $n = 250$, this is the case in about 75% of the 50 samplings, for $n = 500$ and $n = 3500$, in about 67% of the samplings, and for $n = 1000$, in 54% of the samplings.

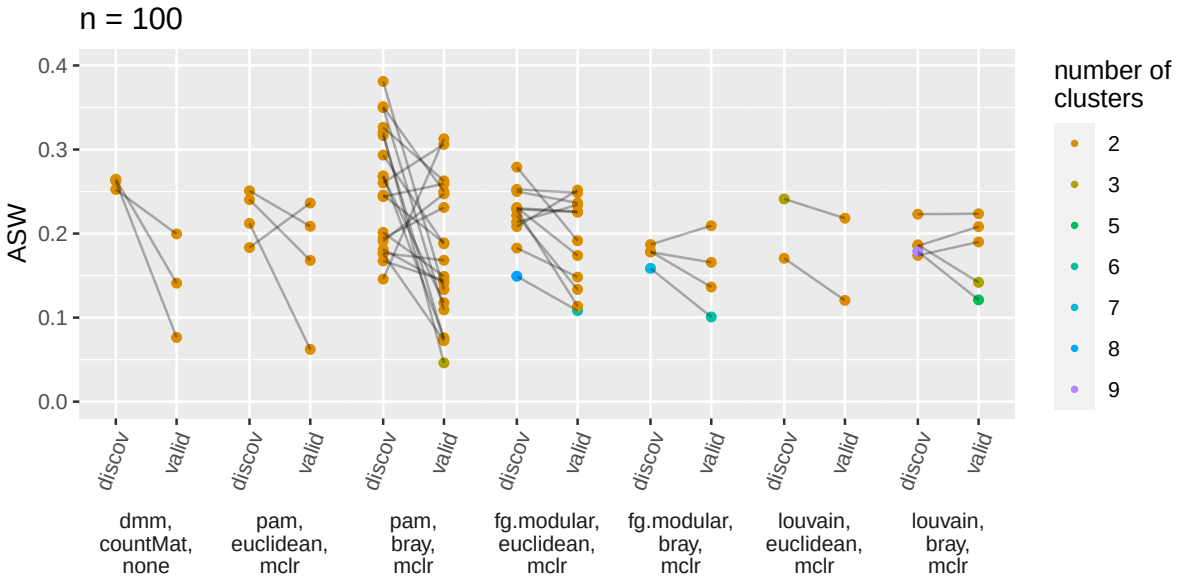


Fig K. Best ASWs for the clustering of samples on the discovery data, compared with the results on validation data, $n = 100$

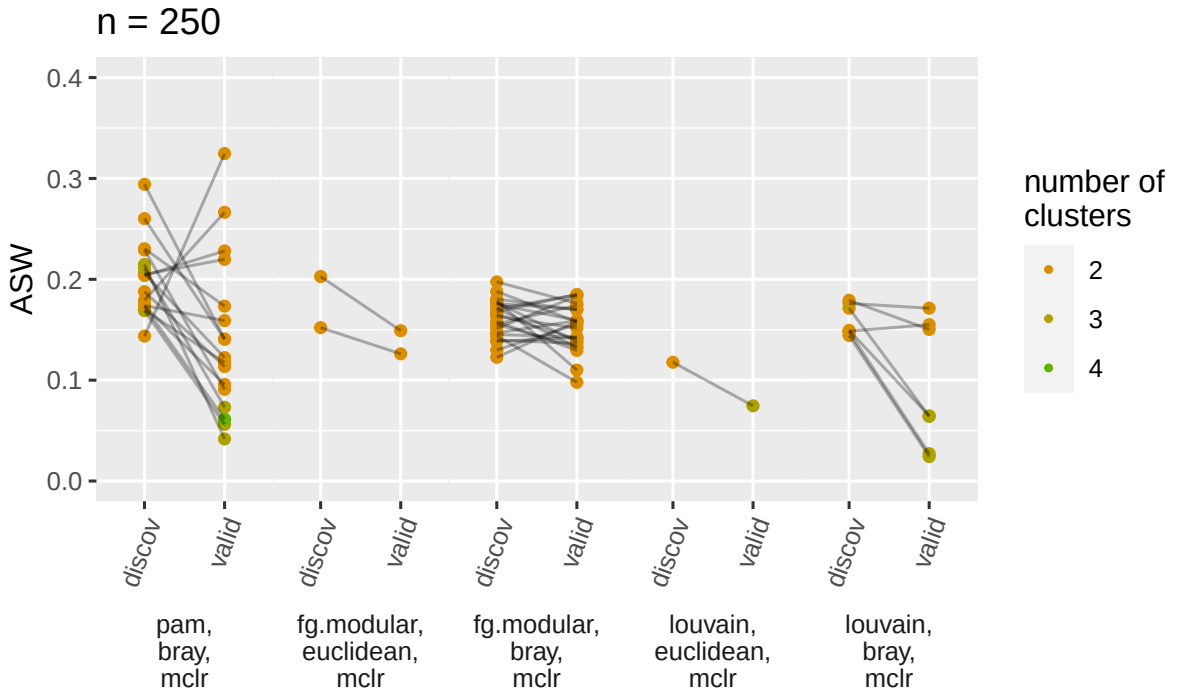


Fig L. Best ASWs for the clustering of samples on the discovery data, compared with the results on validation data, $n = 250$

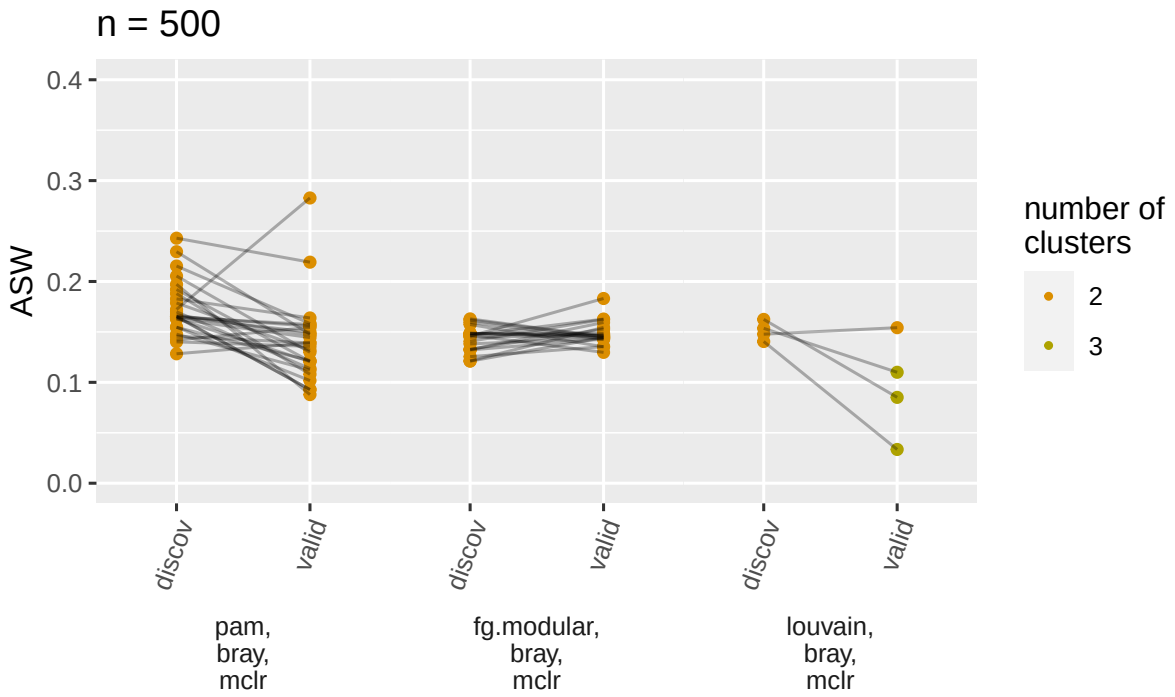


Fig M. Best ASWs for the clustering of samples on the discovery data, compared with the results on validation data, $n = 500$

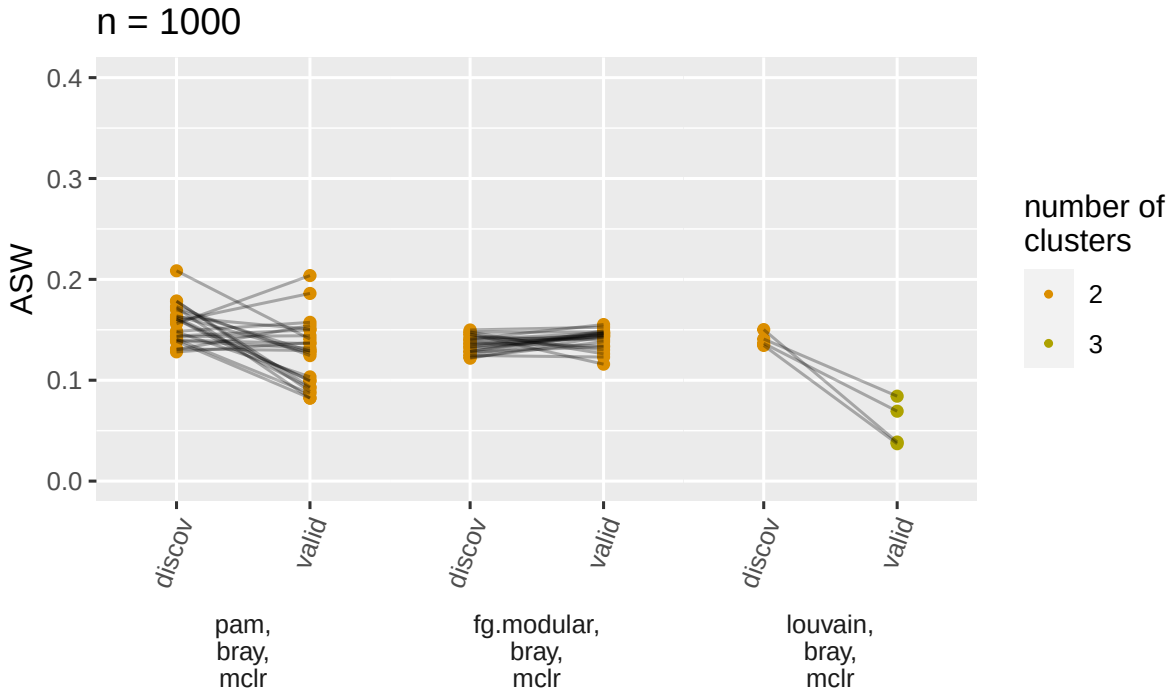


Fig N. Best ASWs for the clustering of samples on the discovery data, compared with the results on validation data, $n = 1000$

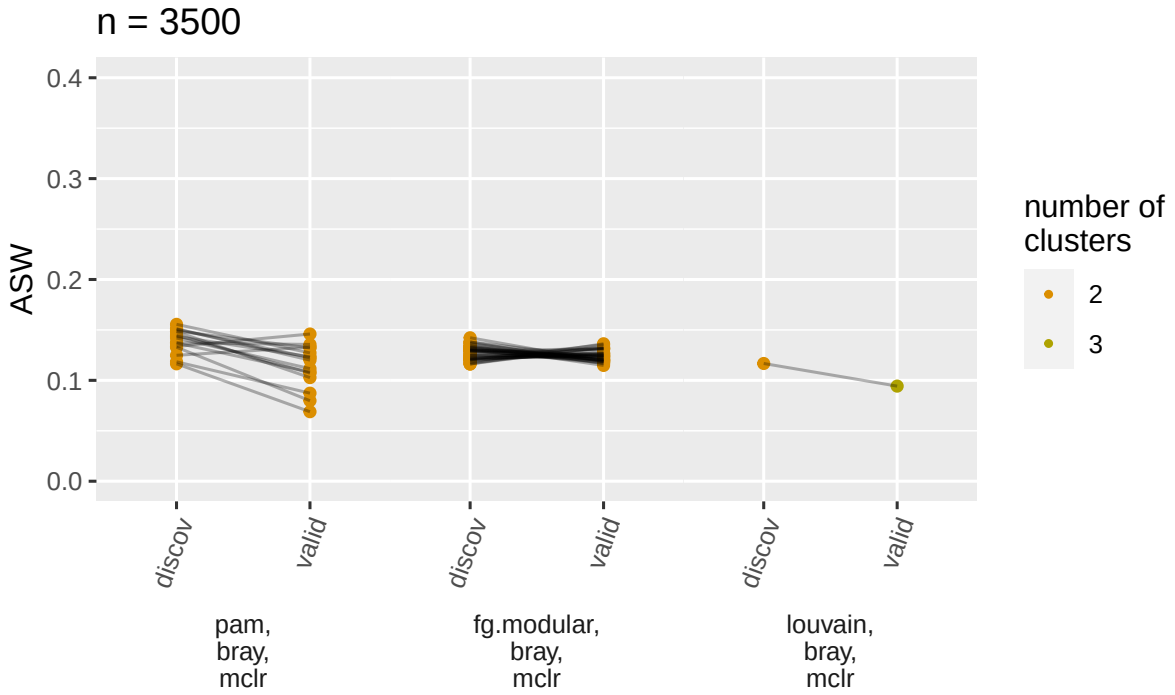


Fig O. Best ASWs for the clustering of samples on the discovery data, compared with the results on validation data, $n = 3500$

References

1. Arumugam M, Raes J, Pelletier E, Le Paslier D, Yamada T, Mende DR, et al. Enterotypes of the human gut microbiome. Nature. 2011;473:174–180.

2. Wu GD, Chen J, Hoffmann C, Bittinger K, Chen YY, Keilbaugh SA, et al. Linking long-term dietary patterns with gut microbial enterotypes. Science. 2011;334(6052):105–108.

3. Costea PI, Hildebrand F, Arumugam M, Bäckhed F, Blaser MJ, Bushman FD, et al. Enterotypes in the landscape of gut microbial community composition. Nature Microbiology. 2018;3:8–16.

4. Cheng M, Ning K. Stereotypes about enterotype: the old and new ideas. Genomics, Proteomics & Bioinformatics. 2019;17(1):4–12.

S5: Analyses with a reduced number of method combinations

We expected over-optimistic bias to decrease if fewer method combinations were tried. To investigate this hypothesis, we repeated our analyses with a reduced number of method combinations: 5 instead of 58 for the clustering of bacterial genera, 3 instead of 14 for hub detection and differential network analysis, and 5 instead of 31 for the clustering of samples.

The subsets of method combinations were chosen as follows:

Research task 1 (clustering of bacterial genera): The method of association estimation was fixed and only the type of cluster algorithm was varied (hierarchical clustering, spectral clustering [1], fast greedy modularity optimization [2], Louvain community detection [3], and manta [4]), leading to five method combinations overall. For (dis)similarity based clustering, association estimation was performed with the semi-parametric rank-based correlation (latentcor) [5, 6] combined with the mclr normalization. For network-based clustering, we used the SPRING method [7], which combines the latentcor correlation estimation with the neighborhood selection technique [8] for sparse estimation of partial correlations. The latentcor and SPRING methods were chosen because they are the most recently proposed methods and can be tentatively considered as "state of the art" among compositionally aware association estimation methods.

Research task 2 (hub detection): We chose three method combinations for network generation that represent three different classes of association estimation: Pearson correlation with clr normalization and sparsification via t-test (as an example of a simple method based on classical correlation estimation), the SPRING method (as a more advanced method that can estimate partial correlations), and the proportionality measure [9, 10] with clr normalization and sparsification via threshold (as an alternative approach that is not based on correlations).

Research task 3 (differential network analysis): The same three method combinations that were used in hub detection were selected.

Research task 4 (clustering of samples): Analogously to the first research task, the method for calculating dissimilarities between the samples was fixed and only the choice of cluster algorithm was varied, resulting in five method combinations. For DMM clustering [11], dissimilarities are not required. For the other cluster algorithms, dissimilarities were calculated with the Aitchison distance [12] which is a very well-known and popular method for this purpose. The dissimilarities were then used as input for PAM [13] and spectral clustering. Moreover, clustering with fast greedy modularity optimization and Louvain community detection was applied to the sparsified dissimilarities, where sparsification was performed with the K-nearest neighbor method.

The results are displayed in Tables A and B which have the same structure as Tables 1 and 2 in the main manuscript. They show the mean, median, and standard deviation of the difference as well as the scaled difference between the value of the evaluation criterion on the validation data and the value on the discovery data (over the 50 samplings of discovery/validation data). Additionally, the effect sizes (mean divided by standard deviation) are reported.

Research task 1: clustering of bacterial genera

	$ARI_{valid} - ARI_{discov}$				$\frac{ARI_{valid} - ARI_{discov}}{ARI_{discov}}$			
n	mean	median	sd	mean/sd	mean	median	sd	mean/sd
100	-0.024	-0.021	0.045	-0.53	-13.0%	-15.7%	28.4%	-0.46
250	-0.029	-0.012	0.051	-0.57	-15.9%	-8.1%	29.7%	-0.53
500	-0.019	-0.013	0.039	-0.49	-9.9%	-8.6%	23.1%	-0.43
1000	-0.030	-0.026	0.035	-0.86	-17.1%	-16.3%	19.4%	-0.88
4000	-0.014	-0.007	0.029	-0.48	-8.2%	-4.3%	17.6%	-0.47

Research task 2: hub detection

	$\#hubs_{valid} - \#hubs_{discov}$				$\frac{\#hubs_{valid} - \#hubs_{discov}}{\#hubs_{discov}}$			
n	mean	median	sd	mean/sd	mean	median	sd	mean/sd
100	-1.72	-1	2.47	-0.70	-18.2%	-14.3%	27.4%	-0.66
250	-0.70	-0.5	2.22	-0.32	-4.8%	-4.5%	25.9%	-0.19
500	-0.62	-1	1.94	-0.32	-4.8%	-9.5%	20.6%	-0.23
1000	-0.78	-1	1.97	-0.40	-7.3%	-11.1%	23.0%	-0.32
4000	-0.90	-1	1.61	-0.56	-9.4%	-11.1%	18.7%	-0.50

Table A. For research tasks 1 and 2: Mean, median, and standard deviation (over 50 samplings of discovery/validation data) of the difference (both unscaled and scaled) between the value of the evaluation criterion on the validation data and the corresponding value on the discovery data. Additionally, the effect size (mean divided by standard deviation) is reported. ARI_{discov} denotes the best ARI on the discovery data and ARI_{valid} the ARI resulting from the corresponding method combination on the validation data. The quantities $\#hubs_{discov}, \#hubs_{valid}$ (number of hubs) are defined analogously.

As Tables A and B show, the means and medians of the differences are negative for most research tasks and sample sizes. The only exception can be seen for the scaled GCD differences for the third research task; here, the means are all positive, indicating better results on the validation data on average. However, the corresponding standard deviations are large and the effect sizes are very small, indicating that the "improved" results on the validation data should probably not be over-interpreted. More detailed analyses show that the positive means are largely driven by a few outliers. Indeed, the *median* scaled differences are still negative, as are the mean and median unscaled differences.
Overall, the results indicate that some over-optimistic bias still exists even if fewer method combinations are tried. However, as expected, the absolute values of the mean/median

Research task 3: differential network analysis								
	$GCD_{valid} - GCD_{discov}$				$\frac{GCD_{valid} - GCD_{discov}}{GCD_{discov}}$			
n	mean	median	sd	mean/sd	mean	median	sd	mean/sd
100	-0.063	-0.130	0.649	-0.10	11.6%	-24.9%	101.2%	0.11
250	-0.213	-0.154	0.628	-0.34	3.3%	-21.2%	101.4%	0.03
500	-0.066	-0.025	0.289	-0.23	0.7%	-9.1%	71.6%	0.01
Research task 4: clustering of samples								
	$ASW_{valid} - ASW_{discov}$				$\frac{ASW_{valid} - ASW_{discov}}{ASW_{discov}}$			
n	mean	median	sd	mean/sd	mean	median	sd	mean/sd
100	-0.023	-0.017	0.068	-0.34	-9.8%	-12.4%	41.0%	-0.24
250	-0.011	-0.014	0.025	-0.45	-12.2%	-20.0%	37.8%	-0.32
500	-0.006	-0.005	0.017	-0.33	-6.3%	-9.6%	33.2%	-0.19
1000	-0.007	-0.005	0.013	-0.58	-12.3%	-10.0%	25.0%	-0.49
3500	-0.001	-0.002	0.010	-0.07	0.0%	-5.9%	25.4%	0.00

Table B. For research tasks 3 and 4: Mean, median, and standard deviation (over 50 samplings of discovery/validation data) of the difference (both unscaled and scaled) between the value of the evaluation criterion on the validation data and the corresponding value on the discovery data. Additionally, the effect size (mean divided by standard deviation) is reported. GCD_{discov} denotes the largest GCD on the discovery data and GCD_{valid} the GCD resulting from the corresponding method combination on the validation data. The quantities $ASW_{discov}, ASW_{valid}$ (average silhouette width) are defined analogously.

differences as well as the effect sizes tend to be smaller compared to Tables 1 and 2. Put differently, over-optimistic bias is less pronounced if fewer method combinations are tried. Of course, the exact amount of over-optimistic bias depends on the chosen (subsets of) method combinations, i.e., the results might be slightly different when choosing different subsets of methods.

Tables C and D show additional stability analyses for the first and second research task based on the reduced number of tried method combinations, analogously to Tables 3 and 4 in the main manuscript. Overall, the index values are similar compared to Tables 3 and 4, i.e., the extent of stability remains roughly the same when reducing the number of tried methods. For the second research task (hub detection), the Jaccard values are somewhat smaller for the reduced number of tried methods at sample sizes of $n = 100$ and $n = 4000$. This might be explained by the following observation: at these sample sizes, the SPRING method is more frequently selected in the setting with the reduced number of methods combinations compared to the setting with the full set of method combinations; at the same time, SPRING tends to yield lower stability values. However, based on this limited analysis, we cannot determine whether SPRING generally tends to produce more unstable results with respect to hub detection.

	ARI_{stab}		
n	mean	median	sd
100	0.408	0.403	0.138
250	0.491	0.415	0.175
500	0.599	0.558	0.180
1000	0.620	0.587	0.177
4000	0.807	0.886	0.164

Table C. Mean, median, and standard deviation of ARI_{stab}, i.e., the ARI between the clusterings on discovery and validation data, over 50 samplings of discovery/validation data.

	Jaccard			Cosine similarity		
n	mean	median	sd	mean	median	sd
100	0.127	0.083	0.106	0.834	0.878	0.130
250	0.339	0.333	0.135	0.906	0.955	0.109
500	0.465	0.458	0.144	0.950	0.964	0.047
1000	0.539	0.545	0.134	0.945	0.967	0.062
4000	0.548	0.569	0.186	0.944	0.965	0.054

Table D. Mean, median, and standard deviation (over 50 samplings of discovery/validation data) of a) the Jaccard index which compares the set of hubs obtained on the discovery data with the set of hubs on the validation data, and b) the cosine similarity which compares these sets of hubs, but on the level of families.

References

1. Ng A, Jordan M, Weiss Y. On spectral clustering: Analysis and an algorithm. Advances in Neural Information Processing Systems. 2001;14:849–856.

2. Clauset A, Newman ME, Moore C. Finding community structure in very large networks. Physical Review E. 2004;70(6):066111.

3. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment. 2008;2008(10):P10008.

4. Röttjers L, Faust K. Manta: A clustering algorithm for weighted ecological networks. Msystems. 2020;5(1):e00903–19.

5. Yoon G, Carroll RJ, Gaynanova I. Sparse semiparametric canonical correlation analysis for data of mixed types. Biometrika. 2020;107(3):609–625.

6. Yoon G, Müller CL, Gaynanova I. Fast computation of latent correlations. Journal of Computational and Graphical Statistics. 2021;30(4):1249–1256.

7. Yoon G, Gaynanova I, Müller CL. Microbial networks in SPRING - Semi-parametric rank-based correlation and partial correlation estimation for quantitative microbiome data. Frontiers in Genetics. 2019;10:516.

8. Meinshausen N, Bühlmann P. High-dimensional graphs and variable selection with the lasso. Annals of Statistics. 2006;34(3):1436–1462.

9. Lovell D, Pawlowsky-Glahn V, Egozcue JJ, Marguerat S, Bähler J. Proportionality: a valid alternative to correlation for relative data. PLoS Computational Biology. 2015;11(3):e1004075.

10. Quinn TP, Richardson MF, Lovell D, Crowley TM. propr: an R-package for identifying proportionally abundant features using compositional data analysis. Scientific Reports. 2017;7(1):1–9.

11. Holmes I, Harris K, Quince C. Dirichlet multinomial mixtures: generative models for microbial metagenomics. PloS One. 2012;7(2):e30126.

12. Aitchison J. On criteria for measures of compositional difference. Mathematical Geology. 1992;24(4):365–379.

13. Kaufman L, Rousseeuw PJ. Finding Groups in Data. John Wiley & Sons, Ltd; 1990.

C Contribution 3: "Over-optimistic evaluation and reporting of novel cluster algorithms: an illustrative study"

This chapter is a reprint of:

Ullmann, T., Beer, A., Hünemörder, M., Seidl, T., & Boulesteix, A.-L. (2022). Over-optimistic evaluation and reporting of novel cluster algorithms: An illustrative study. *Advances in Data Analysis and Classification*. https://doi.org/10.1007/s11634-022-00496-5

Copyright:

Author contributions:

T. Ullmann conceptualized the paper together with A. Beer, M. Hünemörder, and A.-L. Boulesteix (ideas, formulation of overarching aims, etc.). T. Ullmann was the lead for designing the methodology of the article, with A. Beer and M. Hünemörder providing support. The software for the analyses was written by M. Hünemörder, with support from T. Ullmann and A. Beer. T. Ullmann performed validation of the software to check the reproducibility of results. The majority of the original draft was written by T. Ullmann, with the exception of the subsections on "Information visualization", "Robustness", and "Adversarial Attacks", which were written by A. Beer. The other authors (M. Hünemörder, T. Seidl, and A.-L. Boulesteix) made several comments regarding the original draft that were included in the manuscript. The review and editing of the manuscript were led by T. Ullmann and supported by all other authors. All authors read and approved the final version of the article.

Advances in Data Analysis and Classification
https://doi.org/10.1007/s11634-022-00496-5

REGULAR ARTICLE

Over-optimistic evaluation and reporting of novel cluster algorithms: an illustrative study

Theresa Ullmann[1] · **Anna Beer**[2] · **Maximilian Hünemörder**[3] · **Thomas Seidl**[2] · **Anne-Laure Boulesteix**[1]

Received: 12 August 2021 / Revised: 18 December 2021 / Accepted: 14 February 2022

Abstract

When researchers publish new cluster algorithms, they usually demonstrate the strengths of their novel approaches by comparing the algorithms' performance with existing competitors. However, such studies are likely to be optimistically biased towards the new algorithms, as the authors have a vested interest in presenting their method as favorably as possible in order to increase their chances of getting published. Therefore, the superior performance of newly introduced cluster algorithms is over-optimistic and might not be confirmed in independent benchmark studies performed by neutral and unbiased authors. This problem is known among many researchers, but so far, the different mechanisms leading to over-optimism in cluster algorithm evaluation have never been systematically studied and discussed. Researchers are thus often not aware of the full extent of the problem. We present an illustrative study to illuminate the mechanisms by which authors—consciously or unconsciously—paint their cluster algorithm's performance in an over-optimistic light. Using the recently published cluster algorithm Rock as an example, we demonstrate how optimization of the used datasets or data characteristics, of the algorithm's parameters and of the choice of the competing cluster algorithms leads to Rock's performance appearing better than it actually is. Our study is thus a cautionary tale that illustrates how easy it can be for researchers to claim apparent "superiority" of a new cluster algorithm. This illuminates the vital importance of strategies for avoiding the problems of over-optimism (such as, e.g., neutral benchmark studies), which we also discuss in the article.

Theresa Ullmann
tullmann@ibe.med.uni-muenchen.de

1 Institute for Medical Information Processing, Biometry, and Epidemiology, LMU Munich, Marchioninistr. 15, 81377 München, Germany

2 Institute for Informatics, LMU Munich, München, Germany

3 Department of Computer Science, CAU Kiel, Kiel, Germany

Published online: 17 March 2022

Keywords Over-optimism · Clustering · Cluster analysis · Evaluation

Mathematics Subject Classification 62H30 · 68W40

1 Introduction

Cluster analysis refers to grouping similar objects in data, while separating dissimilar ones. While there already are a huge number of cluster algorithms (see e.g., Xu and Wunsch (2010) for an overview), researchers continue to propose novel algorithms every year. Researchers who introduce a new cluster algorithm typically publish it together with a demonstration of the strengths of their approach and its superiority over alternative methods.

However, the results of such studies should be regarded with caution. *Publication bias* (Boulesteix et al. 2015) constitutes a considerable external incentive for researchers to demonstrate the superiority of their new approach: journals and conferences are much more likely to accept a paper about a novel computational method if this method shows good performance and is "better" than pre-existing approaches. This may tempt researchers to present their method's performance in an *over-optimistic* fashion, a mechanism that is also called the "self-assessment trap" (Norel et al. 2011). Such scenarios can not only appear in the research field of clustering but can also be found in all types of *methodological research*, i.e., the development and evaluation of data analytic techniques and algorithms (Boulesteix et al. 2020).

Over-optimization is not necessarily performed in a malicious or even intentional manner, but it is problematic because the new method may turn out to have a worse performance than initially claimed when it is later investigated in a neutral comparison study, i.e., a study whose authors do not have a vested interest in one of the competing methods, see Boulesteix et al. (2013). In other words, the good performance result is not replicable (Boulesteix et al. 2020). Anecdotal evidence for this lack of replicability is presented by Buchka et al. (2021) for a specific data analysis problem related to the pre-processing of a special type of high-throughput molecular data. The over-optimistic presentation of computational methods may lead to the usage of flawed methods in applications, which could ultimately hinder research progress or even lead to questionable results in applied research.

But how exactly may researchers present their new methods in an over-optimistic fashion? For *supervised* classification, an illustrative case has already been presented in the field of bioinformatics by Jelizarow et al. (2010). They considered a "promising" novel classification method, which in reality was not superior to other classifiers. Yet the authors were able to demonstrate that different mechanisms allow over-optimistic presentation of this new method's performance, namely choosing specific datasets, optimizing the method's settings and characteristics to these datasets while burying the other in the file drawer, and choosing suboptimal competing classifiers.

However, to the best of our knowledge, such a study has not yet been conducted for cluster analysis, i.e., the unsupervised scenario. While over-optimistic (selective) reporting is well understood in the context of statistical testing and supervised learning, where its impact can be easily measured, it is much less so in the field of cluster analysis,

which is characterized by the difficulty to properly evaluate methods. We thus aim at filling this gap by demonstrating how a novel cluster algorithm's performance can be presented in an (overly) favorable light.

The problem of over-optimism is in fact as important in unsupervised clustering as it is in supervised classification, and is probably even exacerbated because the performance evaluation of cluster algorithms has not been studied as systematically as the evaluation of supervised classifiers in the methodological literature. Guidance for proper benchmarking of cluster algorithms has only recently emerged (Van Mechelen et al. 2018). Even though the "true" cluster labels are unknown in clustering applications, researchers typically use datasets with known labels to evaluate their novel cluster algorithms. To some extent, the performance evaluation of cluster algorithms thus appears similar to the evaluation of classifiers. Yet for cluster analysis, the role of test data is not as clear-cut as in supervised classification (Ullmann et al. 2021), which entails that researchers are less aware that "overfitting" can not only happen in supervised classification, but also in cluster analysis. Moreover, optimizing hyperparameters such as the number of clusters based on the "ground truth", as is frequently done in cluster algorithm evaluation, does not take into account that other researchers who eventually want to use the algorithm in applications do not know the "true" cluster labels of their datasets, and will thus likely obtain worse results than the performances reported in the original evaluation of the novel algorithm. To evaluate their new method, researchers might also use performance evaluation measures which do not require a fixed "ground truth", such as internal validation indices which measure internal properties of the data (e.g., homogeneity and/or separateness of the clusters). However, over-optimism can still be an issue when using these indices.

In the present study, we use the "Rock" algorithm (Beer et al. 2019) as an illustrative example. Beer et al. (2019) agreed to the usage of their algorithm in our paper. Rock was originally introduced as a "promising" new algorithm and was presented as being able to outperform competitors. In subsequent studies, it turned out that Rock does not generally perform better than its competitors. In the present paper, we show that Rock outperforms competing algorithms in very specific scenarios and that these scenarios can be obtained by three different mechanisms: (1.) optimization of datasets and data characteristics, (2.) optimization of parameters of the Rock algorithm and (3.) the choice of the competing clustering approaches. We demonstrate that if the optimized scenarios are selectively reported and the settings in which Rock performs worse are omitted, the algorithm then appears to outperform its competitors—as a result of an over-optimistic presentation.

Rock is used only as an example—demonstrating the specific characteristics of the Rock algorithm is not the main interest of our work. Rather, we use Rock to illustrate more general mechanisms of over-optimization. We suspect that many studies which introduce new cluster algorithms are affected by these mechanisms. However, given that over-optimization can happen quite subtly and/or unintentionally, we do not cite any published papers here which probably presented their results in an over-optimistic fashion. Neither do we try to quantify the actual optimistic bias that currently exists in the literature on cluster algorithms. Rather, our study is intended as a cautionary tale to raise awareness of the over-optimism problem, and to illuminate the importance of using strategies to avoid over-optimism (e.g., avoiding selective reporting, using

independent test data and conducting neutral benchmark studies, as discussed in detail in Sect. 6).

We first give an overview of related work in Sect. 2. Section 3 explains how we performed optimization of Rock's performance. The corresponding results are presented in Sect. 4 and further discussed in Sect. 5. Possible solutions for the problem of over-optimism are outlined in Sect. 6. We conclude the paper in Sect. 7.

2 Related work

In this section we discuss studies that are related to our work. After presenting studies which directly look at the over-optimistic bias of new computational methods, we address aspects in the field of data mining that are connected to over-optimistic presentation of cluster algorithms.

2.1 Previous work about over-optimistic bias of new computational methods

There appears to be a lack of literature about over-optimism in the introduction of new cluster algorithms. For computational methods other than clustering, there exist some studies, to our knowledge mostly in the field of bioinformatics.

As mentioned above, a study similar to ours was previously reported by Jelizarow et al. (2010), but for supervised classification. Moreover, while this study illustrated over-optimism with a classification method for gene expression data and used real cancer gene expression datasets for this purpose, our example is not application specific. For performance evaluation we choose simulated and real datasets which are frequently used for the evaluation of cluster algorithms in computational research (e.g., the synthetic "Two Moons" dataset, the Iris dataset etc., see Sect. 3).

Broadly speaking, the three categories of optimization mechanisms that we analyze are similar to the categories previously considered in Jelizarow et al. (2010), i.e., optimization of the data, optimization of the algorithm's characteristics, and the choice of competing approaches. However, the use of simulated data allows us to systematically consider data characteristics such as noise or dimensionality, which was not done for the real datasets used in Jelizarow et al. (2010).

In a similar application context, Yousefi et al. (2010) also addressed over-optimism when reporting the performance of newly proposed classifiers. They focused on classification on high-dimensional data with low sample size, such as gene expression data. The authors specifically considered the optimization of the datasets, i.e., they analyzed the optimistic bias that results from reporting only the datasets with the best (or second best) performance of the new classifier. They estimated this bias in a simulation study, by repeatedly sampling sets of datasets, and recording the best (or second best) performing dataset of each set. The aim of their study thus was to *quantify* the optimistic bias with specific focus on the choice of datasets, whereas we model different over-optimization mechanisms of a (hypothetical) researcher in an illustrative way. The results of Yousefi et al. (2010) show that in the high-dimensional data setting, there is indeed a large optimistic bias when reporting only the best or second best performing dataset.

Finally, again in the context of bioinformatics, a recent study aimed to estimate the optimistic bias in the reported performance of new computational methods to preprocess a special type of raw high-throughput molecular data (Buchka et al. 2021). The approach was to perform a literature search and compare the reported performance of newly introduced methods against their performance in later neutral comparison studies. As expected, novel methods were ranked better than competitors in most of the papers introducing them, but outperformed competitors at a lesser rate in neutral studies. Yet the new methods still outperformed more than 50% of their paired competitors in neutral studies, showing that while there is optimistic bias, there is also some level of genuine scientific progress.

Outside of bioinformatics, Ferrari Dacrema et al. (2021) assessed optimistic bias in research about recommender systems. Recommender algorithms can be used, for example, to propose new movies to a media streaming user based on previously watched movies. Many new recommendation algorithms based on deep learning were published in recent years, which usually claimed superiority over previous approaches. Ferrari Dacrema et al. (2021) repeated the evaluations of the original authors, but with additional baseline algorithms. Their analysis showed that most of the new methods did not actually outperform simple and long-known baseline algorithms, provided strong-performing baselines were chosen and their hyperparameters were tuned as carefully as those of the new algorithms. This highlights that not including strong competitors or not treating the competing methods fairly might lead to optimistic bias.

2.2 Information visualization

Over-optimistic presentation of results can also be obtained by visualization methods, i.e., not only by a biased selection of *which* data to show, but also by *how* the selected data is shown. Studies on *information visualization* address the latter aspect. For example, visualization methods with a high lie factor (the ratio between "size of effect shown in graphic" and "size of effect in data", see Tufte (1983)), or misleading labeling and scaling of axes, could be used by a researcher to let their algorithm appear in a more favorable light.

We do not focus on such mechanisms in our study, and instead illustrate that over-optimistic reporting of results is also possible if all rules regarding "correct" information visualization are observed.

2.3 Robustness

Robust clustering algorithms yield a similar quality of results for similar input. Thus, it is unlikely that there are experimental setups which yield notably better results than similar experiments and could thus be selectively presented in an over-optimistic fashion. We do not systematically evaluate the robustness of any of the tested cluster algorithms in Sect. 4, but rather show how the lack of robustness can be exploited in order to over-optimistically present the results of the exemplary algorithm. Out of the diverse types of robustness, we focus on the lack of robustness regarding different properties of the data as well as hyperparameter settings. For example, we consider

robustness w.r.t. noise. "Noise" can mean either background noise, i.e., uniformly distributed points across the data space which do not belong to the original distribution, or jitter, i.e., small deviations or perturbations in the original distribution. We regard only the latter in our experiments.

That robustness is crucial for clustering algorithms was already stated by Davé and Krishnapuram (1997). In recent literature on cluster algorithms, the robustness regarding different properties of the data is often presented, e.g., the size of the dataset, number of clusters, dimensionality, and structure of the data. Usually there is a base case for which one property at a time is changed to regard the effects on the clustering result. However, it is often left unclear how and why this base case was obtained, and how the settings which are not regarded in the respective experiment are chosen.

Even though the robustness regarding the choice of hyperparameters seems similarly important, authors often refer to "expert knowledge" for finding the "best" setting, and omit a robustness analysis. This can lead to enormous disagreements in the evaluation of an algorithm, see, e.g., the controversy about DBSCAN (Ester et al. 1996; Gan and Tao 2015; Schubert et al. 2017). Even easily interpretable hyperparameters, such as the number of clusters k (e.g., for k-Means, Lloyd 1982), which at first sight do not seem to require a robustness analysis, might show better performance w.r.t. the evaluation measure when set at a value different from the "ground truth".

To summarize, robustness regarding different aspects is not only important to guarantee a predictable quality of clustering for users, but also reduces the potential for over-optimism.

2.4 Adversarial attacks

An adversarial attacker may corrupt the results of an algorithm by only performing small changes or additions in a dataset, leading to a wrong but more favorable outcome for the attacker (Goodfellow et al. 2018). Even though adversarial attacks are most often regarded in context of supervised machine learning, they can also influence results of unsupervised machine learning: recently, Chhabra et al. (2020) showed that adversarial attacks are also possible for clustering, even without knowing important details of the cluster algorithm. Algorithms which tend to return results of highly varying quality, also for only small perturbations in the data, are easy victims not only for adversarial attacks, but also for over-optimism. However, where adversarial attackers aim at changing only certain results, over-optimistic researchers would try to change the impression of an algorithm's overall quality. By knowing the details of their novel algorithm as well as deciding on all hyperparameters and competitive methods, the influence over-optimistic researchers can have on the presentation of their results is massive, especially compared to an adversarial attacker.

3 Over-optimization methods

In this section we outline the concept and the experimental design of our study. We first explain the three different categories of over-optimization mechanisms that we

illustrate in our study. We then detail our concrete implementation, e.g., the clustering algorithms, datasets, evaluation measure and optimization method.

3.1 Three categories of over-optimization

Imagine a researcher who wishes to present his/her cluster algorithm in a favorable light. We model the work process of this researcher as an "optimization task": the characteristics of the study in which the new algorithm is compared to existing ones are optimized such that the researcher's algorithm scores well, in particular better than the best performing competing algorithm. This optimization can refer to (1.) finding datasets or data characteristics for which the new algorithm works particularly well, (2.) finding optimal parameters of the algorithm (and vice versa, neglecting the search for optimal parameters for the competitors) or (3.) choosing specific competing algorithms.

Optimizing datasets or data characteristics. A new cluster algorithm might perform well for specific types of datasets, but not for other types. Researchers might decide to report only the best-performing types of datasets. Additionally, for synthetic datasets, there is potential for over-optimism when varying specific characteristics (e.g., the amount of noise, the sample size, or the number of dimensions), and reporting only the optimal settings. Moreover, simulated datasets depend on the random seed, such that in turn, the performance of the cluster algorithm might also vary over different random seeds. Researchers might actively look for a "good" random seed or simply stumble across a particular "good" random seed by chance, neglecting to try other random seeds to check for robustness.

Optimizing the algorithm's parameters or characteristics. Hyperparameters of the cluster algorithm, or characteristics of the algorithm designed during the development phase, could be varied by researchers to look for the best result. Hyperparameter optimization (HPO) is per se a legitimate procedure in performance evaluation. However, there is less awareness for proper evaluation of cluster algorithms combined with HPO, compared to the more extensive methodological literature on correct evaluation of supervised classifiers with HPO (Boulesteix et al. 2008; Bischl et al. 2021). In cluster analysis, over-optimism in relation to HPO may result from (1.) optimizing hyperparameters based on the "true" cluster labels known to the researchers, and (2.) not splitting the data into training and test sets. Both aspects will be discussed in more detail in Sects. 4 and 5. Moreover, over-optimism might also result when researchers neglect to set optimal parameters for the *competing* algorithms, e.g., when choosing suboptimal hyperparameter defaults for the competitors while finetuning their own algorithm.

Optimizing the choice of competing algorithms. Finally, researchers might pick specific competing clustering methods that let their own algorithm appear in a better light. They could neglect to look for the best state-of-the-art competitor, instead opting for less optimal comparison algorithms. Even if the researchers are aware of state-of-the-art competitors, they might not include them because the codes are not openly available, or implemented in a programming language which they are not familiar with. Researchers could also think of different groups of competing cluster algorithms, and

then pick the group that is most favorable for comparison with their own algorithm. A new density-based cluster algorithm could for example be compared either with a group of other density-based algorithms, or with a group of some well-known, not necessarily density-based cluster algorithms. While both choices could in principle be sensible, it is over-optimistic if researchers either deliberately exclude a class of competitors a priori because they expect their novel algorithm to perform worse than this class, or if they choose the competitor group a posteriori after having seen the results (Jelizarow et al. 2010).

Apart from these three categories of optimization, there are some further optimization possibilities (e.g., optimizing the evaluation measure) that we do not analyze here in detail, but briefly discuss in Sect. 5.

We assume that usually, researchers do not consciously perform the three classes of optimization tasks in a malicious and systematic manner. Nevertheless, in the course of a longer research process during which researchers try different datasets, algorithm parameters/configurations and competing algorithms, researchers might optimize the settings in an unsystematic and (probably) unintentional manner. Even if researchers start their analysis with the best intentions, they might post-hoc rationalize their (over-optimistic) choices as perfectly reasonable decisions, given that "[h]umans are remarkably good at self-deception" and scientists often "fool themselves" (Nuzzo 2015).

One might argue that the optimizations outlined above are not actually *over*-optimizations and that it is perfectly fine to look for scenarios in which a novel algorithm performs well. We would agree that it is not a priori wrong to search for and report such scenarios, as a new cluster algorithm can never be expected to outperform every other cluster algorithm in every situation. However, it should also be transparently reported how the presented "successful" scenarios were obtained, and how the algorithm performs in other settings. Over-optimism ultimately appears when performance results are *selectively* reported. We will illustrate this with our results in Section 4.

3.2 Experimental setup

We now present the exemplary cluster algorithm and its settings, the competing algorithms, the datasets and the evaluation measure. Our fully reproducible code is available at https://github.com/thullmann/overoptimism-clust-algo.

In accordance with the authors, we used the already published algorithm Rock (Beer et al. 2019) as a novel and promising algorithm. Rock is an iterative approach similar to Mean Shift (Fukunaga and Hostetler 1975), but based on the k nearest neighbors (kNN) instead of the bandwidth. In each step, points "roam" to the mean of their respective k nearest neighbors. Points with a similar final position are assigned to a common cluster. The algorithm involves the hyperparameter t_{max}, which gives the maximum number of iterations. As the maximum meaningful value for k is fixed ($k > \frac{n}{2}$ would lead to an assignment of all points to the same cluster), and the increase of k in every step is linear, t_{max} also determines the number k of nearest neighbors regarded in each iteration. The larger t_{max} is chosen, the closer values for k are in

consecutive steps. Lower values for t_{max} thus lead to larger gaps between consecutive values for k, which may cause volatile merges of different clusters. On the other hand, higher values for t_{max} lead to more iterations, which increases runtime.

As typical for short papers, only a limited number of experiments is presented in Beer et al. (2019), illustrating that the underlying idea is promising. The results for Rock looked good compared to k-Means (Lloyd 1982), DBSCAN (Ester et al. 1996) and Mean Shift, which are typical competitors in the field and representatives for algorithms finding different types of clusters. As examples for competing algorithms, we thus chose k-means, DBSCAN, Mean Shift and additionally Spectral Clustering (Ng et al. 2001).

As the clustering performance measure we use the Adjusted Mutual Information Score (AMI, Vinh et al. 2010), a version of the Mutual Information (MI) Score adjusted for chance agreement of random partitions. For each dataset and cluster algorithm, the known "true" clustering (as given either by the simulation design for the synthetic datasets or by additional label information for the real datasets) was compared via the AMI with the clustering found by the algorithm. The higher the AMI, the more similar the two clusterings are. The AMI attains its maximum value of 1 if the two clusterings are identical, and equals 0 if the MI between the two clusterings is equal to the MI value expected for two random partitions. We give the detailed mathematical definition of the AMI in the appendix A.

While we only use the AMI in our illustration for the sake of conciseness, a similar analysis could be performed for alternative indices which measure the agreement of the calculated clusterings with the "ground truth", or even for internal validation indices which evaluate clusterings based on internal properties of the data alone and do not require the "ground truth" (see also the discussion in Sect. 5.2).

The choice of exemplary datasets is linked to the three different optimization tasks outlined in Sect. 3.1. We thus give the datasets for each task in turn and explain how the optimization was performed. Note that we performed the three optimization tasks sequentially, building on the results of each previous task. Of course, in reality, a researcher will likely not perform the optimizations in such a perfectly sequential matter, and might jump between different tasks of optimization or try to optimize different aspects simultaneously. Again, our sequential procedure merely serves illustrative purposes.

For some specific details of the implementation, we refer to the appendix A.

Optimizing datasets and data characteristics. For this part of the analysis, we chose three commonly used different synthetic datasets from scikit-learn (Pedregosa et al. 2011), see Fig. 2: Two Moons[1], Blobs[2] (for details on this dataset, see the appendix A), and Rings[3].

[1] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_moons.html, visited: 05/31/2021.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_blobs.html, visited: 05/31/2021.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_circles.html, visited: 05/31/2021.

First, we performed optimization by varying the following data characteristics: a) for Two Moons, the sample size and the jitter values (where "jitter" denotes small random perturbations to the original data points in the clusters), b) for Blobs, the sample size, the number of dimensions and the number of generated clusters ("blobs"), and c) for Rings, the sample size and the jitter values. The goal of the optimization was to find the parameter configuration (e.g., for Two Moons, the configuration (n, j) of sample size and jitter value) that yields the largest performance difference between Rock and the best of the competitors – which is not necessarily the parameter configuration that yields the best *absolute* performance of Rock.

That is, for each of the three types of synthetic datasets in turn, we performed the following formal optimization task:

$$\text{argmax}_{D\in\mathcal{D}} \left\{ \frac{1}{10} \sum_{i=1}^{10} \left(AMI\left(Rock(D^i), y_{D^i}\right) - \max_{C\in\mathcal{C}} AMI\left(C(D^i), y_{D^i}\right)\right)\right\} \tag{1}$$

where $D \in \mathcal{D}$ denotes the different variants of the dataset. For example, for the Two Moons data, each dataset D is a version of Two Moons with a specific jitter value and sample size. Each D has a cluster label ground truth y_D. For each $D \in \mathcal{D}$, ten different versions of D, namely $D^i, i = 1, \ldots, 10$ resulting from ten different random seeds were generated. Put differently, we performed ten simulation iterations per setting, i.e., we sampled ten datasets from each data distribution with a specific data parameter setting. The AMI difference is then averaged over these ten versions. This is supposed to reduce the influence of the random seed. Only at a later point in the analysis did we look at the effect of picking specific random seeds (see below). $Rock(D^i)$ denotes the application of Rock to the data D^i, returning a partition of the objects. Analogously, the competing algorithms $C \in \mathcal{C}$ return a partition of D^i, with $\mathcal{C} = \{k\text{-means}, \text{DBSCAN}, \text{Mean Shift}, \text{Spectral Clustering}\}$.

For each of the three types of datasets in turn, we performed the optimization task (1) by using the Tree-structured Parzen Estimator (TPE, Bergstra et al. 2011), as implemented in the Optuna framework (Akiba et al. 2019) in Python[4]. TPE is a Bayesian optimization (BO) method. BO approaches sequentially propose new parameter configurations based on a library of previous evaluations of the objective function (for more details on BO methods and the TPE, see the appendix A). The TPE is often used for hyperparameter optimization of machine learning models, but in our case, we use it to optimize the *data* parameters. The TPE optimization can be considered as a very simplified model of the researcher's optimization procedure. Of course, a researcher's behavior does not exactly correspond to the mathematical procedure of the TPE. However, if researchers perform *intentional* (over-)optimization, then they might indeed use an optimization method such as the TPE to find the best data settings. The Bayesian optimization mimics the researcher's (unintentional) over-optimization in the following sense: as mentioned above, a researcher developing a new cluster algorithm might sequentially look for data settings in which the new algorithm performs well, taking

[4] https://optuna.readthedocs.io/en/stable/reference/generated/optuna.samplers.TPESampler.html, visited: 05/31/2021.

into account performance information from previously tried data parameters. This is the reason why we chose the TPE over a simple grid search or random search, because the latter do not use previously obtained performance information. To make the TPE process more "realistic", we supplied a grid of limited discrete values to the TPE, given that a researcher presumably would not try arbitrary real numbers. We performed this experiment with only 100 optimization steps for each of the three types of datasets, in order to fairly represent a researcher trying different data parameters by hand.

After determining the optimal values for the data parameters (which we will later report in Table 1 in Sect. 4.1), we analyzed the performance of Rock for non-optimal parameter values. That is, for each dataset and single data parameter in turn, the parameter was varied over a list of values, while the other data parameters were kept fixed at their optimal values. For example, for the Two Moons dataset we tried different jitter values and plotted the corresponding performance as measured by the mean AMI over ten random seeds against the jitter, keeping the sample size at the optimal value determined by the TPE. These analyses show the effects of selectively reporting only the best data parameters versus the performance of the algorithm over a broader range of each data parameter.

In the experiments given so far, we always considered the AMI averaged over ten random seeds. In the final step of the analysis for this section, we specifically study the influence of individual random seeds. We take the Two Moons dataset as an example, with a data parameter setting which is not optimal for Rock, but for which DBSCAN performs very well. We generate 100 datasets with these characteristics by setting 100 different random seeds, to check whether there exist particular seeds for which Rock does perform well, leading to over-optimization potential.

For all experiments described so far, we applied reasonable parameter choices (defaults or heuristics) for the cluster algorithms. For Rock we chose $t_{max} = 15$, as done for all experiments in the original paper (Beer et al. 2019), and for the competing algorithms see the appendix A.

Optimizing the algorithm's parameters or characteristics. For this example we varied Rock's hyperparameter t_{max} (maximum number of iterations). As t_{max} is discrete with a reasonable range of $\{1, \ldots, 30\}$, a researcher could easily try every value by hand. Thus we did not perform optimization with the TPE, but with a full grid search, i.e., we calculated the AMI performance of Rock for each value of t_{max} and for each dataset. For this illustration, we considered the *absolute* performance of Rock, given researchers would also strive to maximize the absolute performance of their novel algorithm.

As exemplary datasets, we again considered Two Moons, Blobs and Rings, and additionally four real datasets frequently used for performance evaluation: Digits, Wine, Iris and Breast Cancer as provided by scikit-learn[5] (see also the UCI Machine Learning Repository, Dua and Graff 2017). The data parameter settings for the three synthetic datasets (sample size, amount of jitter etc.) corresponded to the optimal settings from the TPE optimization of (1). We used a single random seed to generate the illustrative synthetic datasets.

[5] https://scikit-learn.org/stable/datasets/toy_dataset.html, visited: 05/31/2021.

In a next step, using the Two Moons dataset as an example, we compared the AMI performances of Rock and DBSCAN over ten random seeds, first without, then with hyperparameter optimization for Rock and DBSCAN. We used the TPE for HPO of DBSCAN. Here, the TPE was not intended to model a researcher's behavior, but was used as a classical HPO method. The comparison illustrates the effect of neglecting parameter optimization for competing algorithms.

Optimizing the choice of competing algorithms. We did not perform new experiments here. Rather, we looked at the results from the two previous optimization tasks to derive the potential for optimization of the choice of competing cluster algorithms.

4 Results

We present our results for the three optimization tasks outlined above, starting with the optimization of datasets and data characteristics.

4.1 Optimizing datasets and data characteristics

In this subsection we examine how strongly the choice of the "best" properties of a dataset, along with the type of dataset, can influence the performance estimation of Rock.

4.1.1 Optimization of the data parameters with TPE

Table 1 reports the optimal data parameters for the three synthetic datasets as determined by the TPE optimization. The search space for each parameter is given in parentheses and consists of discrete values. The column "AMI diff." shows the difference of the AMI obtained by Rock to the AMI obtained by the best competitor (averaged over ten random seeds). Recall that the AMI difference was used as the optimization criterion by the TPE to find the "optimal" parameter configuration. The column "Abs. AMI" denotes the absolute performance of Rock as measured by the AMI averaged over ten random seeds. The standard deviation over the seeds is also displayed.

Table 1 Optimal data parameters as determined by the TPE optimization

Dataset	Sample size	Jitter	# of dim.	# of clusters	AMI diff.	Abs. AMI
Two Moons	**1000**	**0.15**	2	2	+0.3581	0.7881
	([1, 16] · 100)	([1, 20] · 0.01)	(default)	(default)		±0.1583
Blobs	**300**	–	**3**	**2**	+0.0475	0.8881
	([1, 16] · 100)		([2,20])	([2,10])		±0.1573
Rings	**1600**	**0.02**	2	2	+0.1789	0.1789
	([1, 16] · 100)	([1, 20] · 0.01)	(default)	(default)		±0.0026

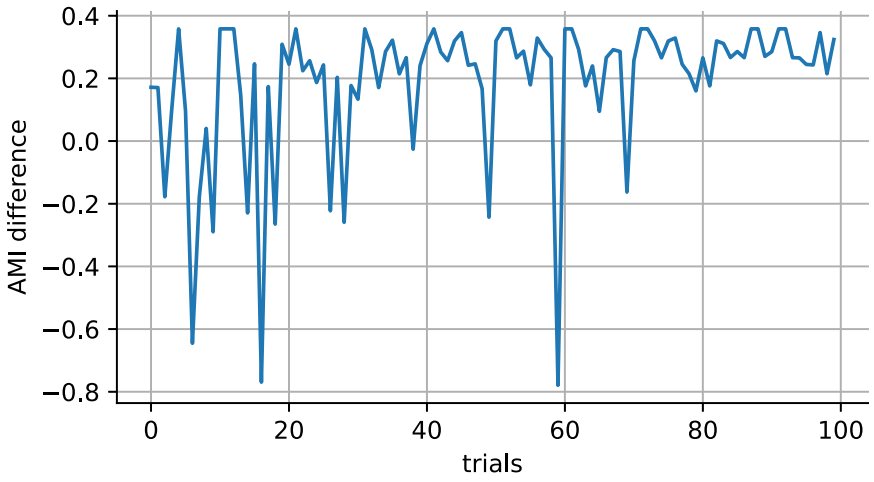


Fig. 1 Optimization progression for the Two Moons dataset, with the AMI difference averaged over ten random seeds

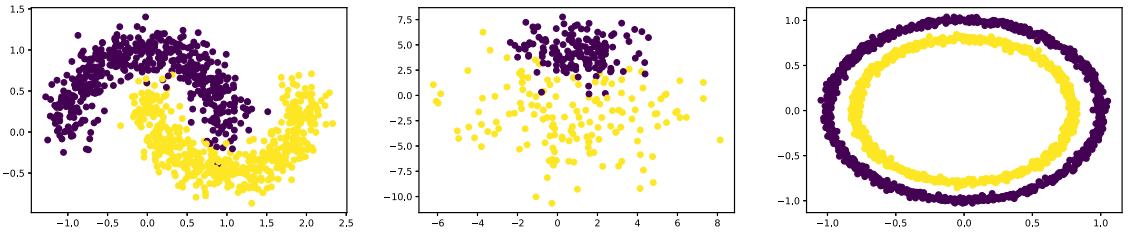


Fig. 2 Example datasets (Two Moons, Blobs, Rings) with the optimal data parameters. For the Blobs example we only show the first and second dimensions

For the example of the Two Moons dataset, Fig. 1 shows a graphical representation of the TPE process over 100 optimization steps. The final "optimal" result is given by the best trial out of the 100 trials. The datasets with the optimal settings are pictured in Fig. 2, using a single illustrative seed of 0.

Judging from the results in Table 1, Rock appears to show better performance than its competitors. A researcher could use the results to claim Rock's "superiority". However, the *absolute* performance of Rock for the Rings dataset is not very good with a mean AMI of only 0.1789. Rock is only the best algorithm here because the competing methods completely fail to detect the clustering. A researcher who tries to optimize the data types might thus decide to let the Rings dataset disappear in the "file drawer", particularly if he/she must omit some results due to page limits, and only present the Two Moons and Blobs datasets, for which Rock performs well, both in absolute and in relative (compared to competitors) terms. But would this presentation for Two Moons and Blobs be over-optimistic? To obtain a more realistic picture of Rock's abilities, we analyze the results when the data parameters are not set at the optimal values, but varied over a grid.

4.1.2 Varying the data parameters

We consider the influence of the sample size, the number of dimensions and the amount of jitter. For each data parameter, we pick one data type for illustrative purposes (either Two Moons or Blobs). The data parameters that are not currently considered are set

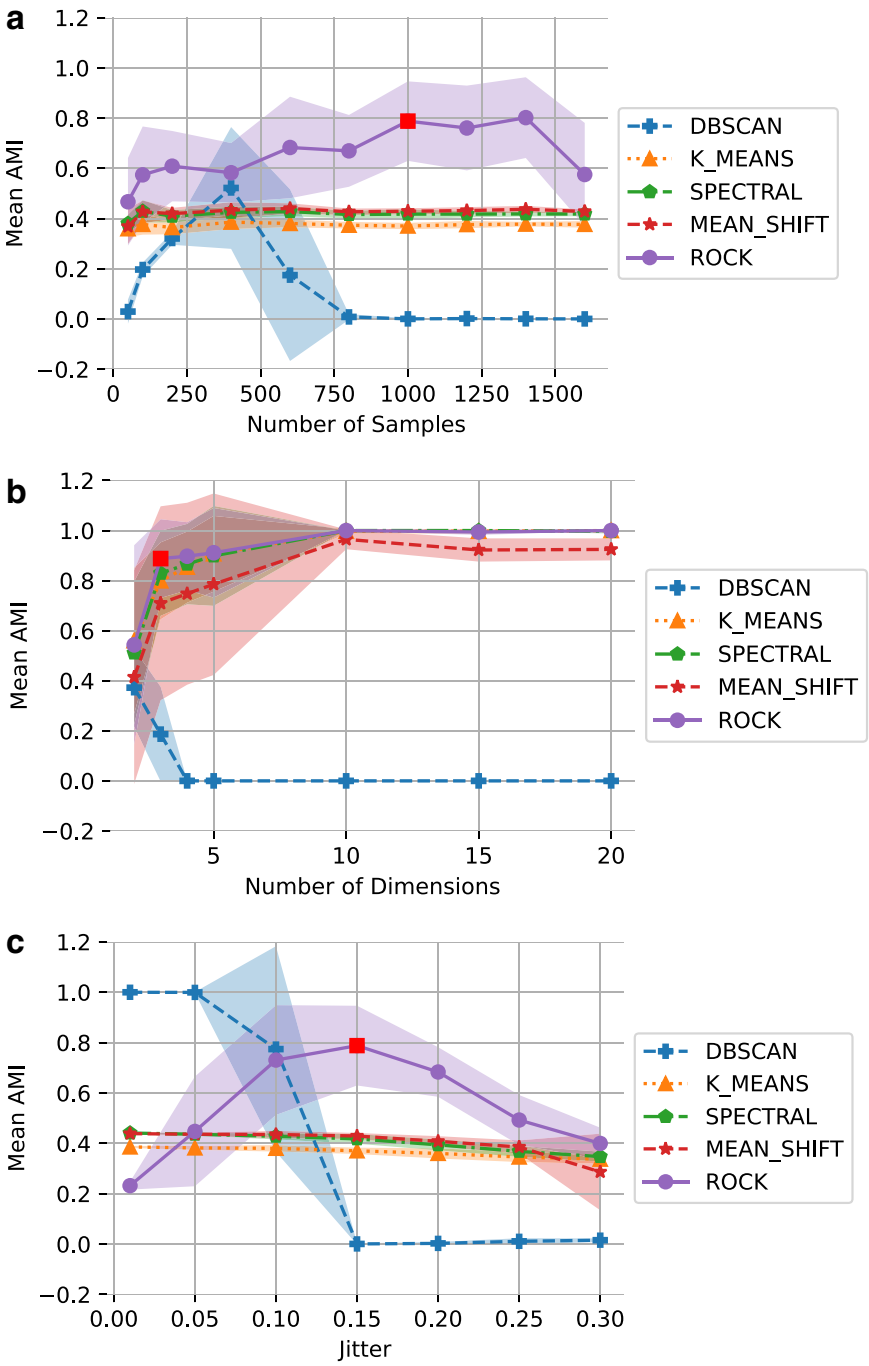


Fig. 3 **a** Varying the sample size for the Two Moons dataset (jitter = 0.15), **b** varying the number of dimensions for the Blobs dataset (sample size = 300, number of blobs = 2), **c** varying the jitter amount for the Two Moons dataset (sample size = 1000)

to their optimal values from Table 1. Figure 3a–c show the performance of Rock and its competitors measured by the AMI over ten random seeds, depending on the varied data parameters. The border around each line shows the standard deviation over the seeds. Red squares indicate the optimal setting from Table 1.

Sample size. Here we consider the Two Moons dataset in Fig. 3a. We tried the following sample sizes: 50, 100, 200, 400, 600, 800, 1000, 1200, 1400, 1600. The jitter value is set at its optimal value 0.15 from Table 1. Rock indeed appears to perform better

here than its competitors over a broader range of numbers of samples, not just for the optimal setting. However, at smaller sample sizes, the difference to k-means, spectral clustering and Mean Shift is less impressive than at Rock's optimal setting of $n = 1000$.

Dimensionality. The Blobs dataset is analyzed in Fig. 3b, varying the number of dimensions over {2, 3, 4, 5, 10, 15, 20}. The sample size is set at 300 and the number of generated blobs is 2, according to Table 1. Rock performs better than competitors mainly for small dimensions. Once the number of dimensions exceeds 5, Rock cannot outperform k-means and Spectral Clustering.

Jitter. The amount of jitter is varied for the Two Moons dataset, see Fig. 3c. We tried the following jitter amounts: 0.01, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30. The sample size is set to the optimal value of 1000 according to Table 1. Rock performs better than its competitors for the jitter set at 0.15 and above. However, for lower jitter values, Rock cannot outperform DBSCAN. Moreover, for jitter values of 0.25 and 0.30, the difference from Rock to k-means, spectral clustering and Mean Shift is quite low and not as impressive as at the optimal setting of 0.15.

To summarize, the performance of Rock is not robust with respect to variation of the data parameters, which leads to potential for over-optimization. While Rock is indeed better than its competitors for certain ranges of the data parameters, there are also settings for which Rock either does not perform better than the competitors, or the performance advantage is small. Thus the apparent "superiority" of Rock is generally less impressive than indicated by the results found from the TPE optimization in Table 1.

4.1.3 Influence of the random seed

For the analyses mentioned so far, the mean AMI over ten random seeds was considered. However, it is also possible that a researcher chooses a particular random seed for which Rock performs well. As seen in Fig. 3c, Rock is outperformed by DBSCAN on the Two Moons dataset for a jitter value of 0.05 and 1000 samples. This statement is based on the AMI averaged over 10 random seeds. But could there also be particular

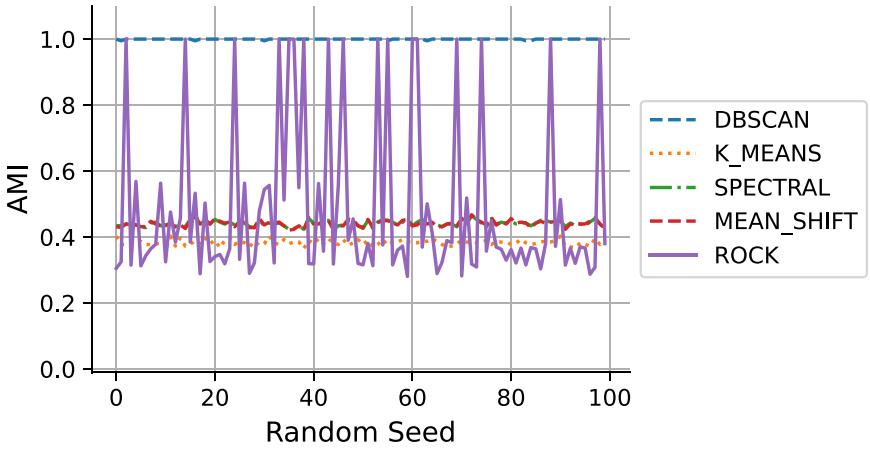


Fig. 4 Performance of the cluster algorithms on the Two Moons dataset (sample size = 1000, jitter = 0.05) over 100 random seeds

random seeds for which Rock does perform well? In Fig. 4, we display the behavior of Rock and its competitors over 100 different random seeds. Since Rock performs as well as DBSCAN for some particular seeds, there is potential for an over-optimizing researcher to pick such a seed. While deliberately trying multiple seeds and presenting only the best one can be considered as malicious behavior, it is also possible that the seed set by the researcher is by chance a "good one", and that the researcher does not consider a dependence of the performance on the random seed. To avoid such unintentional over-optimism, it is advisable to account for sampling variability and average over multiple random seeds, even when the cluster algorithm itself is deterministic. While the practice of sampling multiple datasets from a data distribution is well-known in statistics, this is sometimes neglected when evaluating data mining tasks like clustering.

4.2 Optimizing the algorithm's parameters

We analyze how the hyperparameter t_{max} of Rock can be optimized. In contrast to the previous sections, we now consider the absolute performance of Rock, given that a researcher would presumably not only try to outperform competitors, but also strive to obtain AMI values for Rock which are close to 1.

Additionally to Two Moons, Blobs and Rings, we consider the four real datasets mentioned in Sect. 3.2: Digits, Wine, Iris, Breast Cancer. For the Two Moons, Rings and Blobs datasets, we used the optimal data parameters from Table 1 and only generated a single illustrative dataset for each type by using 42 as a random seed. In accordance with typical evaluation of cluster algorithms, we do not split the datasets into training and test sets (see, however, the discussion in Sect. 6.2).

Figure 5 shows the performance of Rock as measured by the AMI, over t_{max} ranging from 1 to 30.

It can be seen that for different datasets, different t_{max} values are optimal. An optimistic researcher could report (only) the best t_{max} and the corresponding performance for each dataset. Optimizing hyperparameters of a cluster algorithm based on the "ground truth" of datasets (here via the AMI) is frequently seen in the literature.

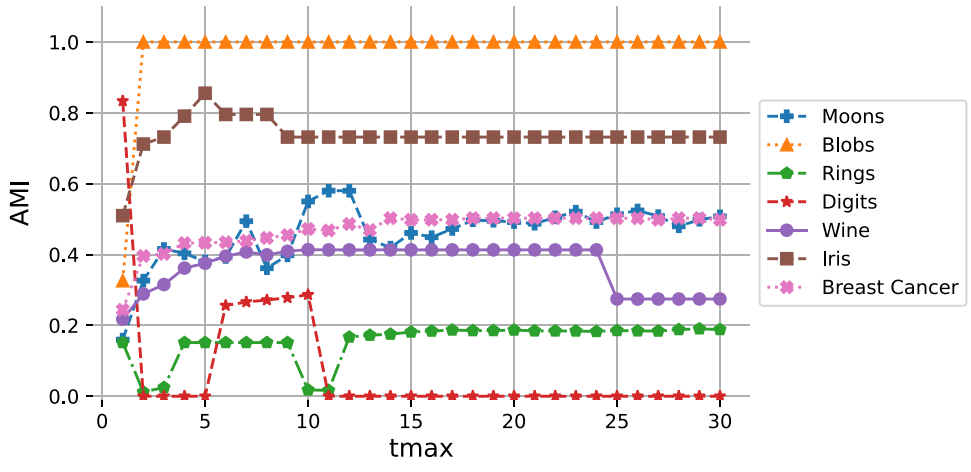


Fig. 5 Varying the hyperparameter t_{max} of Rock for different datasets

But as mentioned above, this could be over-optimistic with regards to the future performance of the algorithm: the evaluation of a novel algorithm is ultimately supposed to give hints about how well the algorithm will perform in future applications. But applied researchers usually do not know the "true" cluster labels of their datasets, as otherwise there would be no need for clustering. Thus the applied researchers cannot use a "ground truth" to determine a good t_{max} value for their specific datasets, and will thus obtain worse results for their datasets than the performances reported in the original paper which introduced the cluster algorithm. We will further discuss this issue in Sect. 5.1.

An alternative to reporting the best t_{max} for each dataset individually is to look for a t_{max} value that leads to good performance for multiple datasets. For example, $t_{max} = 12$ yields reasonable performance values for Blobs, Two Moons and Iris. Thus, optimistic researchers might only report these three datasets with $t_{max} = 12$ and claim that this choice of t_{max} will perform well for future datasets. However, such a statement would likely be over-optimistic as $t_{max} = 12$ was chosen on only a few datasets, and considering the varied behavior of the different datasets for different t_{max} in Fig. 5.

Over-optimism can not only result from optimizing the hyperparameters of the novel algorithm, but also from simultaneously neglecting to optimize the hyperparameters of the *competing* algorithms. As an example, we compare Rock with DBSCAN on the Two Moons dataset, with the data parameters optimized for Rock from Table 1. Recall that in Sect. 4.1, we did not perform hyperparameter optimization, and instead used hyperparameter defaults or heuristics for the algorithms which could be reasonably justified (see also the appendix A): for Rock, $t_{max} = 15$ as in the original paper of Beer et al. (2019), and for DBSCAN, $minPts = 2 \cdot$#of dimensions, leading to $minPts = 4$ for Two Moons, and $eps = 0.2$. The AMI for Rock for this case is 0.7881 ± 0.1583 (mean and standard deviation over ten random seeds), see also Table 1. This mean value is different from the AMI value in Fig. 5 at $t_{max} = 15$, because a single seed was used for the latter. The AMI performance of DBSCAN was only 0.0007 ± 0.0024.

We then performed hyperparameter optimization for both cluster algorithms (with regards to the absolute AMI performance over ten random seeds). For Rock, we performed a simple grid search over $t_{max} \in \{1, 2, \ldots, 30\}$. The optimal performance is at the previously used default $t_{max} = 15$, thus again yielding a mean AMI of 0.7881 ± 0.1583. This is not surprising, given that $t_{max} = 15$ was used in Sect. 4.1 to optimize the data parameters of Two Moons such that Rock obtains superior performance (although the performance *difference* was used as the optimization criterion in that section). For DBSCAN, we performed hyperparameter optimization with the TPE, and obtained optimal parameters of $minPts = 41$ and $eps = 0.4$, leading to a performance of 0.8300 ± 0.0244, which is a major improvement over the previous performance of DBSCAN. Thus DBSCAN outperforms Rock after hyperparameter optimization. This demonstrates that if researchers decide to perform hyperparameter optimization for the cluster algorithms to be compared, they should conduct the optimization not only for their own algorithm, but also equally carefully for all competing methods.

Returning to the topic of data type optimization (Sect. 4.1), Fig. 5 also shows the potential for picking specific datasets for which Rock performs reasonably well (e.g. Blobs, Iris, Two Moons) and discarding the ones with worse performance (Digits,

Rings). Again, *over*-optimization is marked by selective reporting: while no cluster algorithm can be expected to perform well on all types of data, it is still important to report data types for which a novel algorithm fails to detect clusters, to illuminate the limitations of the new method.

4.3 Optimizing the choice of competing algorithms

Here we revisit the results from Sect. 4.1 to analyze whether there is potential for picking specific competing cluster algorithms such that Rock appears better. For example, Fig. 3a–c show that Rock often performs better than DBSCAN, which was also due to neglecting hyperparameter optimization for DBSCAN, cf. Sect. 4.2. By picking suitable data parameter ranges, an over-optimistic researcher could praise the drastic performance improvement from Rock over DBSCAN. The same figures show that Rock is often better than Mean Shift. Thus, there is the potential for the following narrative: "Rock is an improvement of Mean Shift". As the figures show, this claim would sweep some caveats under the carpet. For example, the other competitors, k-means and spectral clustering, are (almost) as good as Rock for the Blobs dataset in Fig. 3b.

5 Discussion

We have illustrated that selective presentation of performance results can lead to over-optimistic assessment of a novel cluster algorithm. Neglecting to show limitations of a new algorithm can lead to users applying it in inappropriate settings for the algorithm, which leads to unusable results. In this section, we discuss potential further aspects of over-optimism that we did not focus on, but would be interesting to study in future work.

5.1 Hyperparameter tuning and development of the algorithm

As explained in Sect. 4.2, the current standard of reporting the performance of a novel algorithm with hyperparameters optimized to the clustering "ground truth" (e.g., with a grid search) is likely over-optimistic. Using the ground truth of datasets for performance evaluation of a novel algorithm has a further drawback: as the number of datasets labeled by experts is limited, researchers using these datasets optimize their algorithm's characteristics on these few labeled real world datasets, or alternatively use (unrealistic) synthetic datasets. Datasets such as Two Moons and Blobs are frequently used, but provide very limited information about how the cluster algorithm will perform in much more complex applied settings.

The optimization to a few datasets might not only concern the hyperparameters of the algorithm, but also the characteristics of the algorithm which are explored in the development phase. For example, Rock contains some "hidden hyperparameters" such as the growth rate of the number of neighbors considered in each iteration, or the weighting of the different nearest neighbours (Beer et al. 2019). These characteristics

are not intended to be changed by the user, but were decided on by the researchers during the development of the algorithm. However, if such characteristics are optimized according to the performance on just a few selected datasets, then this might result in an over-optimistic "overfitting" effect.

5.2 Evaluation measure

For all our experiments in this paper we used the Adjusted Mutual Information (AMI) as measure for the quality of clustering. Other partition similarity indices such as the Normalized Mutual Information (NMI, Strehl and Ghosh 2002), Adjusted Rand Index (ARI, Hubert and Arabie (1985)), Accuracy and F1-measure are often used in the field (see also Albatineh et al. (2006), for an overview). They all range in $[-1, 1]$ resp. $[0, 1]$ and describe how well the clustering results correspond to a ground truth, but have slightly different behaviors (Pfitzner et al. 2009). These indices are also called *external validation* indices, because they require an externally known partition (the ground truth) for evaluation. Yet evaluating a clustering based on the given "ground truth" might not always be the best choice. There could be interesting cluster structures in the data which differ from the given "true" labels, particularly because there is no unique definition of what a "good" clustering is (Hennig 2015). Moreover, as pointed out above, many real world datasets do not come with given labels. Thus researchers might also use internal validation indices (Halkidi et al. 2015) which do not require knowledge of the "true" labels, but evaluate a clustering based on internal properties of the data alone. Popular internal indices which measure within-cluster homogeneity and between-cluster heterogeneity/separateness include the Average Silhouette Width index (Kaufman and Rousseeuw 2009), the Caliński-Harabasz index (Caliński and Harabasz 1974), and the Davies-Bouldin index (Davies and Bouldin 1979). Such indices can also be used for performance evaluation of novel clustering algorithms, yet they might be susceptible to the over-optimism mechanisms outlined above. For example, researchers could optimize datasets and data characteristics with respect to an internal index, such that this index indicates a good performance for the new cluster algorithm, analogous to the optimization with the AMI discussed in Sect. 4.1.

The multitude of possible evaluation criteria—external or internal – gives rise to another potential source of over-optimism: Researchers could try different measures and pick the one that is most favorable to their novel algorithm. While researchers might be understandably uncertain about which evaluation measure to choose, they should not try different measures and then pick only the most favorable one *after* having seen the results. Researchers should carefully consider before starting the experimental evaluation which performance criterion is of particular interest in the considered context. If multiple measures are tried, then these should all be reported.

5.3 Preprocessing

Preprocessing the data can significantly influence the results of clustering. In our study, we scaled all the datasets. There are different normalizations that may be applied to the data, as well as methods to remove outliers or noise to improve the clustering results.

To avoid over-optimism, researchers should refrain from trying different preprocessing methods and reporting only the one most favorable to their new algorithm. Moreover, the same preprocessing steps should be applied to all datasets and *for all compared cluster algorithms*. Otherwise, if only the new algorithm is combined with suitable preprocessing, it might have an unfair advantage. A clear distinction should be made between preprocessing steps and steps belonging to the new cluster algorithm.

5.4 Theoretical evaluation

While we focus on the experimental evaluation of cluster algorithms with simulated or real-world datasets, it would also be interesting to study over-optimism in the context of theoretical analyses of algorithms. For example, researchers often make claims about their novel algorithms which they prove mathematically. But they could use very specific assumptions to yield the desired results. It might not always be easy for readers to judge how unrealistic these assumptions are, i.e., to which extent the assumptions restrict the use of the algorithm in real-world applications. Authors should thus always make their theoretical assumptions very clear, and thoroughly discuss how restrictive they are.

While theoretical analyses can, in principle, be affected by over-optimism, they are often a vital part of the evaluation of novel cluster algorithms. Theoretical results, if carefully deduced, can give a more complete picture of the algorithm's capabilities. Authors who thoroughly analyze their novel algorithm from a theoretical perspective might also use this background knowledge to choose a suitable and clearly defined experimental study design, such that unintentional over-optimization in the experimental part of the analysis could sometimes be partially avoided.

6 Possible solutions

As we have illustrated, there might be a strong over-optimistic bias when introducing a new cluster algorithm. How can such a bias be avoided or corrected? We discuss three options that all researchers can consider using in their research: (1.) avoiding selective reporting and analyzing robustness, (2.) evaluating the new method on independent data, and (3.) performing neutral benchmark studies. Moreover, we discuss (4.) how changing incentives in research culture and the publication system (that are beyond the control of individual researchers) might help to reduce over-optimism.

6.1 Avoiding selective reporting and analyzing the robustness of the algorithm

Our results have shown that over-optimistic presentation ultimately requires a certain amount of *selective reporting*, i.e., reporting only specific scenarios in which the new algorithm performs well. This might happen if many different scenarios are tried and only the "best" ones are reported, while the others are buried in the file drawer. Researchers might also omit the analysis of certain scenarios a priori, for example, when only considering data simulated according to a specific model. Such constraints

should be clearly explained, and the performance of the algorithm should not be oversold.

In the context of *model-based* cluster algorithms (see McLachlan et al. (2019) for an overview), selective reporting might be easier to detect. For example, if mainly datasets generated by the model of the newly developed algorithm are chosen, and/or the novel algorithm is compared with competing methods that were developed for the detection of clusters generated by other models, then the novel algorithm immediately has an advantage, which can be easily spotted. Nevertheless, there is still potential for an over-optimistic selection of datasets and comparative methods among all "reasonable" possibilities. Moreover, other potential sources of over-optimism discussed above, such as (hyper)parameter optimization, are also existent for model-based clustering. Readers and reviewers of articles about novel model-based cluster algorithms should keep this in mind, and the authors themselves must be careful to avoid over-optimistic choices.

Ideally, researchers should report scenarios in which their algorithm performed worse, to give a more realistic picture of the limitations of the novel approach. This may also require researchers to check the robustness of their algorithm (cf. Sect. 2.3): if the cluster algorithm is not robust with respect to certain data parameters, this should be honestly reported. Discussing the evaluation results for various parameter choices could also be beneficial as there is often not a single "best" choice and different parameters could be useful in different applications (Cerioli et al. 2018).

6.2 Validation on independent data

It is advisable to evaluate a new algorithm's performance on fresh data that was *not* used for developing the algorithm and assessing its performance (Jelizarow et al. 2010). As we have demonstrated in Sects. 4.1 and 4.2, looking for specific data parameters or tweaking the algorithm's hyperparameters might cause unintentional overfitting to the datasets used during the research process. As discussed in Sect. 5.1, overfitting to the used datasets could also concern the algorithm's characteristics that were engineered in the development phase. The algorithm might not perform quite as well on new data, which would constitute a more realistic assessment of its performance.

More realistic performance values might also be obtained by taking inspiration from supervised classification and splitting the used datasets into "training" and "test" sets (Ullmann et al. 2021). Then hyperparameters such as t_{max} are optimized on the training set, and the chosen t_{max} is evaluated on the test set to assess performance. This could partially avoid "overfitting" of the hyperparameters to the data. However, a) this splitting procedure does not say anything about the performance on genuinely new data/data from different distributions, and b) when using the ground truth for optimization on the training set, this does not solve the problem that applied researchers who wish to use the new cluster algorithm in practice usually do not know the ground truth of their datasets, and thus cannot use the hyperparameter optimization procedure of the original authors. Therefore, it is advisable for authors who introduce a new algorithm to discuss and evaluate criteria for hyperparameter choice that do not require the ground truth, for example internal validation indices. Such indices could be used to

choose hyperparameters on the training set, and to evaluate the chosen hyperparameters on the test set to ensure that potential overfitting effects are detected.

6.3 Neutral benchmark studies

Awareness about the dangers of selective reporting and the importance of evaluation on fresh data might help to alleviate the problem of over-optimism. Academic teaching/training and illustrative studies such as ours can contribute to creating such awareness. Moreover, following guidelines for methodological computational research can help researchers avoid over-optimism (Boulesteix 2015). Ultimately, this will probably not solve the problem completely. Researchers are incentivized by the publication system to present their new algorithm favorably, which is unlikely to change in the short term (see 6.4). They are also more competent with respect to their own methods—and thus more likely to use them optimally than competing methods when conducting the evaluation. Thus, neutral benchmark studies are additionally required.

A neutral benchmark study is characterized by the comparison of existing algorithms (instead of the introduction of a new method), and neutrality of the authors, i.e., the authors do not have a vested interest in a particular method showing better performance than the others and are as a group approximately equally familiar with all considered methods, see Boulesteix et al. (2013, 2017) for an extensive discussion of these concepts. As mentioned in the introduction, neutral benchmark studies are less likely to suffer from over-optimism and usually offer a more realistic performance evaluation than studies presenting new methods.

In the field of clustering methodology, neutral benchmark studies are rarer than for supervised classification. Lately, however, there have been some advances: guidelines for performing benchmark studies for cluster algorithms were published in Van Mechelen et al. (2018). Following these guidelines, Hennig (2021) compared nine popular cluster algorithms, mainly with respect to various internal validation indices, but also regarding the recovery of the "true" clusterings. For an overview of previous cluster benchmark studies, see Van Mechelen et al. (2018) and Hennig (2021). In principle, the guidelines of Van Mechelen et al. (2018) could and should also be followed by non-neutral researchers who evaluate their new algorithm.

6.4 Changing incentives in the culture of research and the publication system

The three possible solutions presented so far are in principle accessible to individual researchers or teams of researchers. Ultimately, however, each researcher is subject to the constraints of the research and publication system. For example, researchers might hesitate to report limitations of their novel algorithm, because this could reduce their chances of getting published. Moreover, it can still be difficult to publish a neutral comparison study as many journals and conferences—stressing the importance of "novelty"—prefer studies introducing new methods (Boulesteix et al. 2018). In our view, changes in this attitude are necessary to further reduce over-optimism. Accepting neutral benchmark studies for publication should become more widespread. Furthermore, reporting limitations of novel algorithms should not be considered a "failure"

and instead an integral part of a healthy research culture. Journals and conferences should actively encourage authors to report scenarios in which their new algorithm does not perform optimally, or at least should not consider such reporting to be a cause for rejection. At the same time, editors and reviewers play an important role in filtering manuscripts in which authors do not carefully justify their experimental choices and only present very specific settings, which may be a hint that the results could potentially be over-optimistic. It should be taken into account, however, that even when a persuasive justification is given, the authors might still have arrived at these choices by (intentional or unintentional) over-optimization.

7 Conclusion

We have shown that studies which introduce new cluster algorithms might be affected by over-optimistic presentation of the results. For illustrative purposes, we have demonstrated different over-optimism mechanisms using the recently developed Rock algorithm as an example. While this is a specific example, we believe that these mechanisms might similarly apply to other novel clustering algorithms. We have also given some recommendations for avoiding over-optimism. It is our hope that going forwards, these guidelines will be taken into account. After all, overselling of novel methods does not contribute to genuine scientific progress.

Acknowledgements We thank Oliver Langselius and Anna Jacob for making valuable language corrections.

Author Contributions Conceptualization: TU (Lead), AB (Supporting), MH (Supporting), A-LB (Supporting)
Methodology: TU (Lead), AB (Supporting), MH (Supporting)
Software: MH (Lead), TU (Supporting), AB (Supporting) Validation: TU (Lead)
Writing—original draft preparation: TU (Lead), AB (Supporting), MH (Supporting), A-LB (Supporting), TS (Supporting)
Writing—review and editing: TU (Lead), AB (Supporting), MH (Supporting), A-LB (Supporting), TS (Supporting)
Funding acquisition: TS (Lead), A-LB (Supporting) Supervision: A-LB (Lead), TS (Supporting).

Funding Open Access funding enabled and organized by Projekt DEAL. This work has been funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibilities for its content.

Availability of data and material All used datasets are publicly available in scikit-learn for Python (Pedregosa et al. 2011).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Code availability Our fully reproducible code is available at: https://github.com/thullmann/overoptimism-clust-algo.

A Appendix

In this appendix we give some details about the implementation outlined in Sect. 3.2. More information can be found in our fully reproducible code which is available at https://github.com/anonresearcher461/over-optimism. All experiment were performed with Python[6], version 3.9.5.

A.1 Adjusted mutual information (AMI)

Here we give the mathematical definition of the Adjusted Mutual Information Score (AMI, Vinh et al. 2010) which we use to compare the calculated clusterings with the "true" cluster labels. To define the AMI, we first discuss the entropy H of a single clustering and the Mutual Information (MI) of two clusterings. See Vinh et al. (2010) and Meila (2015) for more detailed explanations.

Let C and C' be two clusterings with k respectively l clusters. Let $n_{ij}, i = 1, \ldots, k, j = 1, \ldots, l$ the number of data points which are in cluster i of C and cluster j of C'. Let $n_{i\bullet}$ and $n_{\bullet j}$ be the respective marginal sums, and n the overall number of data points.

The entropy H of clustering C is defined as

$$H(C) = -\sum_{i=1}^{k} \frac{n_{i\bullet}}{n} log\left(\frac{n_{i\bullet}}{n}\right).$$

The entropy can be interpreted as the level of uncertainty associated with the clustering C. The Mutual Information (MI) of the clusterings C, C' is defined as

$$MI(C, C') = \sum_{i=1}^{k}\sum_{j=1}^{l} \frac{n_{ij}}{n} \log\left(\frac{n_{ij}/n}{n_{i\bullet}n_{\bullet j}/n^2}\right).$$

The MI measures to which extent knowledge of the clustering C reduces uncertainty about the clustering C'. The MI is a symmetric measure, and it holds that

$$0 \leq MI(C, C') = MI(C', C) \leq min(H(C), H(C')).$$

[6] https://www.python.org, visited: 05/31/21.

The MI can be normalized to ensure the measure ranges in [0, 1], yielding the Normalized Mutual Information (NMI):

$$NMI(C, C') = \frac{MI(C, C')}{avg(H(C), H(C'))}.$$

Different choices for the "average" avg are possible, e.g., the arithmetic mean, the geometric mean, the minimum or maximum. We use the arithmetic mean (Kvalseth 1987), which is the scikit-learn default.[7]

Both the MI and NMI tend to increase with an increasing number of clusters, even if the information shared mutually between the clusterings does not actually increase. To account for this effect, the MI can be adjusted for chance: the MI of C, C' is compared with the expected MI for two random clusterings drawn from a permutation model (see Vinh et al. (2010) for details). The Adjusted Mutual Information Score (AMI) is thus calculated as follows:

$$AMI(C, C') = \frac{MI(C, C') - E[MI(C, C')]}{avg(H(C), H(C')) - E[MI(C, C')]}. \tag{2}$$

The AMI attains its maximum value of 1 if the two clusterings are identical, and equals 0 if the MI between the two clusterings is equal to the MI value expected for two random partitions. Negative values occur if the agreement between C and C' is "worse" than chance.

A.2 Scaling of the datasets

All datasets used in our study were scaled with the scikit-learn standard scaler[8], by subtracting the mean and dividing by the standard deviation of each variable. That is, for each dataset $D = (x_{ij})_{i=1,\ldots,n, j=1,\ldots,d}$, with n samples and d dimensions, each entry x_{ij} is scaled according to

$$\frac{x_{ij} - \frac{1}{n}\sum_{i=1}^{n} x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_{ij} - \frac{1}{n}\sum_{i=1}^{n} x_{ij}\right)^2}}$$

A.3 Details about the blobs dataset

The Blobs dataset[9] consists of isotropic Gaussian clusters, i.e., each cluster $k \in \{1, \ldots, K\}$ (with K the number of generated clusters) corresponds to a Gaussian distribution with covariance matrix $\sigma_k^2 I_d$, where $\sigma_k^2 \geq 0$ and I_d is the d-dimensional

[7] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_mutual_info_score.html, visited: 05/31/2021.

[8] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html, visited: 05/31/2021.

[9] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_blobs.html, visited: 05/31/2021.

identity matrix. We chose a standard deviation of $\sigma_k = 3\frac{k}{K}$ for each cluster k. This generates different variances for the clusters, making some clusters more compact and thus easier to detect, and others more scattered and harder to find.

A.4 Bayesian optimization (BO) and the tree-structured parzen estimator (TPE)

BO approaches (see Shahriari et al. (2016) for an introduction) are popular for optimization problems of the type $\text{argmax}_{x \in \mathcal{X}} f(x)$, where $f : \mathcal{X} \mapsto \mathbb{R}$ is expensive to evaluate. In each step of a BO procedure, f is modelled with a *surrogate model*, based on a library of evaluations of f from previous steps: $((x^{(1)}, f(x^{(1)}), \ldots, (x^{(k-1)}, f(x^{(k-1)}))$. The surrogate model is used to construct an acquisition function, which is cheaper to evaluate and easier to optimize than f, yielding the optimal argument $x^{(k)}$. Then $(x^{(k)}, f(x^k))$ is added to the library, and the process is repeated by updating the surrogate model. The concrete surrogate model and the acquisition function of the TPE were chosen by Bergstra et al. (2011) such that optimization of the acquisition function ultimately leads to optimization of $x \mapsto l(x)/g(x)$, where $l(x), g(x)$ are two Gaussian Mixture Models. $l(x)$ is fitted to the observations $(x^{(i)})_i$ that performed well so far, i.e., for which $f(x^{(i)}) > y^*$ for some threshold value y^*. $g(x)$ is fitted to the remaining observations. The threshold y^* is chosen as a quantile of the observed $y^{(i)} = f(x^{(i)})$ values, such that $p(y > y^*) = \gamma$ for a suitable $\gamma \in (0, 1)$. For more details on the TPE, see the original paper of Bergstra et al. (2011), the Optuna documentation[10], and our reproducible code.

A.5 Default settings for the hyperparameters of the cluster algorithms

For the analysis in Sect. 4.1 (optimizing datasets and data characteristics), we used defaults or heuristics for the hyperparameters of the cluster algorithms which a researcher could justify as "reasonable choices". For Rock, we chose $t_{max} = 15$, as in the original paper of Beer et al. (2019). For k-Means and Spectral Clustering we used the number of ground truth clusters for the parameter k and the default settings from scikit-learn. For DBSCAN, we followed Schubert et al. (2017) to set $minPts = 2d$ with d being the number of dimensions. Moreover, we set $eps = 0.2$, which can be seen as a sensible value, given that the samples were scaled to unit variance. For estimation of the bandwidth for Mean Shift we use the scikit-learn function estimate_bandwidth[11].

References

Akiba T, Sano S, Yanase T, Ohta T, Koyama M (2019) Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 2623–2631

[10] https://optuna.readthedocs.io/en/stable/reference/generated/optuna.samplers.TPESampler.html, visited: 05/31/2021.

[11] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.estimate_bandwidth.html, visited: 05/31/2021.

Albatineh AN, Niewiadomska-Bugaj M, Mihalko D (2006) On similarity indices and correction for chance agreement. J Classif 23(2):301–313
Beer A, Kazempour D, Seidl T (2019) Rock-let the points roam to their clusters themselves. In: Proceedings of the 22nd International Conference on Extending Database Technology (EDBT), pp 630–633
Bergstra J, Bardenet R, Bengio Y, Kégl B (2011) Algorithms for hyper-parameter optimization. Adv Neural Inf Process Syst NIPS 24:2546–2554
Bischl B, Binder M, Lang M, Pielok T, Richter J, Coors S, Thomas J, Ullmann T, Becker M, Boulesteix AL, Deng D, Lindauer M (2021) Hyperparameter optimization: Foundations, algorithms, best practices and open challenges. arXiv preprint arXiv:2107.05847
Boulesteix AL (2015) Ten simple rules for reducing overoptimistic reporting in methodological computational research. PLoS Comput Biol 11(4):e1004191
Boulesteix AL, Strobl C, Augustin T, Daumer M (2008) Evaluating microarray-based classifiers: an overview. Cancer Inf 6:77–97
Boulesteix AL, Lauer S, Eugster MJ (2013) A plea for neutral comparison studies in computational sciences. PLoS ONE 8(4):e61562
Boulesteix AL, Stierle V, Hapfelmeier A (2015) Publication bias in methodological computational research. Cancer Informatics 14(S5):11–19
Boulesteix AL, Wilson R, Hapfelmeier A (2017) Towards evidence-based computational statistics: lessons from clinical research on the role and design of real-data benchmark studies. BMC Med Res Methodol 17:138
Boulesteix AL, Binder H, Abrahamowicz M, Sauerbrei W (2018) On the necessity and design of studies comparing statistical methods. Biometr J 60(1):216–218
Boulesteix AL, Hoffmann S, Charlton A, Seibold H (2020) A replication crisis in methodological research? Significance 17(5):18–21
Buchka S, Hapfelmeier A, Gardner PP, Wilson R, Boulesteix AL (2021) On the optimistic performance evaluation of newly introduced bioinformatic methods. Genome Biol 22:152
Caliński T, Harabasz J (1974) A dendrite method for cluster analysis. Commun Stat 3(1):1–27
Cerioli A, García-Escudero LA, Mayo-Iscar A, Riani M (2018) Finding the number of normal groups in model-based clustering via constrained likelihoods. J Comput Graph Stat 27(2):404–416
Chhabra A, Roy A, Mohapatra P (2020) Suspicion-free adversarial attacks on clustering algorithms. Proc AAAI Conf Artif Intell 34:3625–3632
Davé RN, Krishnapuram R (1997) Robust clustering methods: a unified view. IEEE Trans Fuzzy Syst 5(2):270–293
Davies DL, Bouldin DW (1979) A cluster separation measure. IEEE Trans Pattern Anal Mach Intell. PAMI-1(2):224–227
Dua D, Graff C (2017) UCI machine learning repository. http://archive.ics.uci.edu/ml
Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pp 226–231
Ferrari Dacrema M, Boglio S, Cremonesi P, Jannach D (2021) A troubling analysis of reproducibility and progress in recommender systems research. ACM Trans Inf Syst 39(2):1–49
Fukunaga K, Hostetler L (1975) The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Trans Inf Theory 21(1):32–40
Gan J, Tao Y (2015) DBSCAN revisited: mis-claim, un-fixability, and approximation. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pp 519–530
Goodfellow I, McDaniel P, Papernot N (2018) Making machine learning robust against adversarial inputs. Commun ACM 61(7):56–66
Halkidi M, Vazirgiannis M, Hennig C (2015) Method-independent indices for cluster validation and estimating the number of clusters. In: Hennig C, Meila M, Murtagh F, Rocci R (eds) Handbook of cluster analysis. Chapman and Hall/CRC, Boca Raton, pp 616–639
Hennig C (2015) What are the true clusters? Pattern Recogn Lett 64:53–62
Hennig C (2021) An empirical comparison and characterisation of nine popular clustering methods. Adv Data Anal Classif. https://doi.org/10.1007/s11634-021-00478-z
Hubert L, Arabie P (1985) Comparing partitions. J Classif 2(1):193–218
Jelizarow M, Guillemot V, Tenenhaus A, Strimmer K, Boulesteix AL (2010) Over-optimism in bioinformatics: an illustration. Bioinformatics 26(16):1990–1998

Kaufman L, Rousseeuw PJ (2009) Finding groups in data: an introduction to cluster analysis. John Wiley & Sons, Hoboken, NJ
Kvalseth TO (1987) Entropy and correlation: some comments. IEEE Trans Syst Man Cybern 17(3):517–519
Lloyd S (1982) Least squares quantization in PCM. IEEE Trans Inf Theory 28(2):129–137
McLachlan GJ, Lee SX, Rathnayake SI (2019) Finite mixture models. Ann Rev Stat Appl 6:355–378
Meila M (2015) Criteria for comparing clusterings. In: Hennig C, Meila M, Murtagh F, Rocci R (eds) Handbook of cluster analysis. Chapman and Hall/CRC, London, pp 640–657
Ng AY, Jordan MI, Weiss Y (2001) On spectral clustering: Analysis and an algorithm. In: Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, pp 849–856
Norel R, Rice JJ, Stolovitzky G (2011) The self-assessment trap: can we all be better than average? Mol Syst Biol 7(1):537
Nuzzo R (2015) How scientists fool themselves-and how they can stop. Nat News 526:182–185
Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830
Pfitzner D, Leibbrandt R, Powers D (2009) Characterization and evaluation of similarity measures for pairs of clusterings. Knowl Inf Syst 19(3):361–394
Schubert E, Sander J, Ester M, Kriegel HP, Xu X (2017) DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. ACM Trans Database Syst 42(3):1–21
Shahriari B, Swersky K, Wang Z, Adams RP, de Freitas N (2016) Taking the human out of the loop: a review of Bayesian optimization. Proc IEEE 104(1):148–175
Strehl A, Ghosh J (2002) Cluster ensembles–a knowledge reuse framework for combining multiple partitions. J Mach Learn Res 3:583–617
Tufte E (1983) The visual display of quantitative information. Graphics Press, Cheshire, CT
Ullmann T, Hennig C, Boulesteix AL (2021) Validation of cluster analysis results on validation data: a systematic framework. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery e1444
Van Mechelen I, Boulesteix AL, Dangl R, Dean N, Guyon I, Hennig C, Leisch F, Steinley D (2018) Benchmarking in cluster analysis: a white paper. arXiv preprint arXiv:1809.10496
Vinh NX, Epps J, Bailey J (2010) Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J Mach Learn Res 11:2837–2854
Xu R, Wunsch DC (2010) Clustering algorithms in biomedical research: a review. IEEE Rev Biomed Eng 3:120–154
Yousefi MR, Hua J, Sima C, Dougherty ER (2010) Reporting bias when using real data sets to analyze classification performance. Bioinformatics 26(1):68–76

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

München, den 25.01.2023 ----------

Theresa Ullmann