



Different algorithms, different models

Martyna Daria Swiatczak¹

Accepted: 23 June 2021
© The Author(s) 2021

Abstract

This study assesses the extent to which the two main Configurational Comparative Methods (CCMs), i.e. Qualitative Comparative Analysis (QCA) and Coincidence Analysis (CNA), produce different models. It further explains how this non-identity is due to the different algorithms upon which both methods are based, namely QCA's Quine–McCluskey algorithm and the CNA algorithm. I offer an overview of the fundamental differences between QCA and CNA and demonstrate both underlying algorithms on three data sets of ascending proximity to real-world data. Subsequent simulation studies in scenarios of varying sample sizes and degrees of noise in the data show high overall ratios of non-identity between the QCA parsimonious solution and the CNA atomic solution for varying analytical choices, i.e. different consistency and coverage threshold values and ways to derive QCA's parsimonious solution. Clarity on the contrasts between the two methods is supposed to enable scholars to make more informed decisions on their methodological approaches, enhance their understanding of what is happening behind the results generated by the software packages, and better navigate the interpretation of results. Clarity on the non-identity between the underlying algorithms and their consequences for the results is supposed to provide a basis for a methodological discussion about which method and which variants thereof are more successful in deriving which search target.

Keywords Configurational Comparative Methods (CCMs) · Set-theoretic methods · Qualitative Comparative Analysis (QCA) · Coincidence Analysis (CNA) · Quine–McCluskey optimization · CNA algorithm · Consistency Cubes (CCubes) algorithm

1 Introduction

Configurational Comparative Methods (CCMs) provide powerful methodological tools for researchers who are interested in inferring causal complexity in terms of equifinal conjunctural causation from Boolean models (Baumgartner and Ambühl 2020a; Furnari et al. 2020; Haesebrouck and Thomann 2021; Rihoux and Ragin 2009). Different CCMs propose several solution types and new solution types emerge continually (Schneider and Wagemann 2012, chap. 8.2). At the same time, there is an ongoing debate on which method and

✉ Martyna Daria Swiatczak
martyna.swiatczak@uib.no

¹ University of Bergen, Department of Philosophy, Postboks 7805, 5020 Bergen, Norway

solution type is best at finding which search target (Baumgartner 2021; Baumgartner and Thiem 2020; Baumgartner and Ambühl 2020a; Duşa 2019b; Haesebrouck and Thomann 2021). However, a prerequisite for discussing the explanatory power of different solution types in CCMs is to acknowledge and understand their actual differences, which, too often, are neglected.

The two most prominent CCMs are Qualitative Comparative Analysis (QCA, Cronqvist and Berg-Schlusser 2009; Ragin 1987, 2009; Rihoux and Ragin 2009) and the more recently introduced Coincidence Analysis (CNA, Baumgartner 2009; Baumgartner and Ambühl 2020a). Both methods emphasise a case-based approach to comparative research (Ragin 1987; Whitaker et al. 2020), they are both based on Boolean algebra and set theory (Baumgartner and Ambühl 2020b; Ragin 1987), both methods are designed to discover combinations of factor values that are associated with a certain outcome of interest (Baumgartner and Ambühl 2020b; Rihoux and Ragin 2009) and they both aim at analysing causal complexity (Baumgartner and Ambühl 2020b; Mahoney 2008; Misangyi et al. 2016; Ragin 1987). To draw causal inferences, they both require homogeneity as a background assumption (Álamos-Concha et al. 2021; Baumgartner and Ambühl 2020b; Berg-Schlusser and De Meur 2009), i.e. an unmeasured causal background in which every confounder is constant in all investigated cases (Baumgartner and Ambühl 2020b). Moreover, CNA and what Haesebrouck and Thomann (2021) refer to as the “RF-approach” to QCA (see also Thomann and Maggetti 2020) both seek for redundancy-free models and, thus, have causation as defined in the INUS-theoretic tradition (Baumgartner 2015; Graßhoff and May 2002; Hume 1999; Mackie 1974) as their exclusive search target. However, QCA and CNA are based upon different algorithms with inherently different rationales leading to different models. QCA in all its variants and advancements is built upon the Quine-McCluskey (QMC) minimisation algorithm that has a long tradition and stems from electrical engineering (McCluskey 1956, 1965; Quine 1955), whereas CNA is developed from the CNA algorithm custom-built for analyzing causal structures (Baumgartner 2015; Baumgartner and Ambühl 2020b, a).

While the differences between the various solution types within QCA, i.e. conservative, (enhanced) intermediate, (enhanced/more realistic) parsimonious solution are being extensively discussed (e.g. Baumgartner and Ambühl 2020b; Oana et al. 2021, chap. 4.4; Schneider and Wagemann 2012, 2013), the differences between QCA and CNA have not yet been sufficiently clarified leading to a partly incomplete, partly blurred, and partly false picture within the CCM literature. Until now, when contrasting QCA and CNA, much focus was put on CNA’s ability to design multi-outcome structures like causal chains or common cause structures (Baumgartner and Ambühl 2020a; Duşa 2019b, chap. 10.2, 2019c; Oana et al. 2021, chap. 5.4.2). This ability is inherent to CNA, which can treat any number of factors as endogenous within one analysis (Baumgartner and Ambühl 2020b). However, attributing the non-identity between outputs of CNA and QCA mainly to multi-outcome structures disregards the most fundamental difference between QCA and CNA, namely the different algorithms both methods are built upon—a fact that gained less attention so far (for an exception, see Haesebrouck and Thomann 2021). At the same time, it is precisely this difference that leads to different models, whether multi-outcome structures or not. Thus, when comparing QCA and CNA, I focus on single outcome structures in the remainder of the article and consequently refer to CNA’s single-outcome solution only, i.e. atomic solution.

Moreover, claims were made in the past that the QCA parsimonious solution equals the CNA solution. These claims rely on individual examples for which they hold (e.g. Duşa 2019c, chap. 10.2). However, based on a few such examples, no general claim regarding the

equality of the QCA parsimonious solution and the CNA solution can be made whatsoever. Nonetheless, these references are echoed in the QCA literature contributing to an incorrect picture of QCA and CNA producing “exactly the same solutions” (Duşa 2019c, p. 217). The erroneously implied general identity of the QCA parsimonious solution and the CNA solution is particularly detrimental as it not only obfuscates relevant decision criteria for CCM users designing their studies, but also blocks the important debate on which CCM, solution type, and algorithm is more suitable for which search target, purpose, or research question. To deskew the picture, this article shows that the QCA parsimonious solution and the CNA solution are not always identical, explains why their frequent non-identity stems from the different algorithms underlying both methods, and assesses the extent of their non-identity. As methods constantly evolve organically rather than linearly, along the way it may be useful to take a step back, assess the evolution of a family of methods, and understand the differences between original, emerged, and newly added methods to be able to review what the different methods’ consequences are.

I contribute to this overarching aim in three ways. First, I offer an overview of the fundamental differences between QCA and CNA. Second, I introduce the QMC algorithm and the CNA algorithm and provide data examples to demonstrate their differences. Third, using a series of simulations that benchmark the degree to which QCA and CNA produce identical models, I show overall low ratios of trials that produce identical models. In this context, I also show the non-identity of CNA solutions and QCA solutions produced by other algorithms incorporated in the R package **QCA** (Duşa 2019c) than QMC, i.e. eQMC (enhanced Quine McCluskey, Duşa and Thiem 2015) and CCubes (Consistency Cubes, Duşa 2019a), as well as QCA solutions derived via varying methods of solving QCA’s prime implicant chart. Overall, this article elucidates the implications of different algorithms underlying QCA and CNA for the models they produce, enables an informed understanding of the models’ non-identity, and seeks to enable a long-needed discussion about the consequences of this non-identity for the methods’ ability to uncover different search targets.

Supplementary material, including detailed R scripts, is available in the Online Appendix. It is meant to make replication accessible and enable scholars to challenge or further develop my calculations.

2 QMC vs CNA: top-down vs bottom-up

As CCMs, both QCA and CNA examine relations that hold between factors taking on specific values, i.e. 0 or 1 for crisp-set factors, any (continuous) values between 0 and 1 for fuzzy-set factors, or any of an open (but finite) number of non-negative integers in the case of multi-value factors (Baumgartner and Ambühl 2020b; Rihoux and Ragin 2009). *Factors* represent categorical properties that partition the observed cases into sets (Baumgartner and Ambühl 2020b). They comprise those factors that are hypothesised as causal for an outcome of interest (causal factors) and the outcome itself. This paper covers crisp- and fuzzy-set CCMs. For these two types of CCMs, *factor values* can be interpreted as membership scores in the set of cases exhibiting a certain factor. Causal factor values are usually referred to as *conditions* in QCA language (Haesebrouck and Thomann 2021; Oana et al. 2021; Rihoux and Ragin 2009; Schneider and Wagemann 2012).

In this paper, I use a shorthand notation conventional in Boolean algebra and express membership in a set by italicised upper case and non-membership by italicised lower case

Roman letters. Wherever I generally refer to a factor (but not a factor value) I use regular (non-italicised) Roman letters. Combinations of factor values that a particular case can instantiate are *configurations*. They are systematically compared by QCA and CNA in order to derive relations of necessity and sufficiency between causal factor values (e.g. the presence of a supportive supervisor, perceiving ones work as highly meaningful, or not being male) and the outcome factor value (e.g. high satisfaction at work or not belonging to a certain career group). However, the processes to do so and the offered solutions differ between QCA and CNA. More precisely, QCA and CNA are based on different algorithms to trace different search targets.

Search targets are properties of interest of the investigated system. As such, they are independent of the procedures (e.g. algorithms) that are used to achieve them. In the context of CCMs, I define search targets as properties of interest of the investigated system (e.g. robust sufficiency or INUS causality) that are represented by models in form of conjunctions and disjunctions in disjunctive normal form restricted to certain sufficiency and necessity relations contained in the investigated system (e.g. all sufficiency relations or sufficiency and necessity relations and only such relations). These restrictions define the features of the models (e.g. completeness or non-redundancy).

QCA and CNA both subscribe to the INUS-theory of causation (e.g. for QCA Haesebrouck and Thomann 2021; Ragin 2000; Schneider and Wagemann 2012; Thomann and Maggetti 2020, and for CNA Baumgartner 2015). The acronym INUS stands for Insufficient but Necessary part of a conjunction which is Unnecessary but Sufficient for the outcome value (Mackie 1974, p. 62). Based on this notion, Mackie's INUS theory of causation then defines causation as follows: X is a cause of Y if, and only if, X is at least an INUS condition of Y . As INUS implies that a factor value is a *necessary* part of a sufficient conjunction, it follows that solutions must not contain parts that are unnecessary for the sufficiency of the conjunction and therefore need to be redundancy-free to be causally interpretable, a fact that previous literature has widely discussed (Baumgartner 2015; Baumgartner and Thiem 2017; Thomann and Maggetti 2020). QCA's most-parsimonious solution, or the "RF-approach" to QCA (Haesebrouck and Thomann 2021; Thomann and Maggetti 2020), and CNA both search for "redundancy-free" solutions that can be interpreted as causes in terms of the INUS theory. In contrast, the "SI-approach" to QCA (Haesebrouck and Thomann 2021; Thomann and Maggetti 2020) focuses on "substantive interpretability", emphasises the multi-method character of QCA, and advises to relate results to case studies and profound theoretical reasoning (Schneider and Wagemann 2010; Rihoux and Ragin 2009; Oana et al. 2021, chap. 6). However, this approach's search targets are not yet sufficiently defined in terms of the above given definition (see also Baumgartner 2021; Duşa 2019b).

There are two important differences between QCA and CNA: First, different combinations of analyses of necessity and sufficiency and, second, therein prevalent different top-down vs bottom-up search directions. I refer to *top-down* as an approach that starts with complex configurations and subsequently minimises these expressions, whereas a *bottom-up* approach starts with single factor values and then, step by step, adds factor values until the search target is reached (Baumgartner and Ambühl 2020a; Schneider and Wagemann 2012, p. 92). QCA generally separates the analysis of necessity from the analysis of sufficiency and puts the analysis of necessity first (Oana et al. 2021, chaps. 3, 4; Schneider and Wagemann 2010, 2012, chap. 3). Nonetheless, there exist multiple ways in which both analyses can be systematically combined (Haesebrouck 2019; Schneider 2019). While the analysis of necessity can be restricted to single factor values or can also include more complex necessary disjunctions of factor values (Thomann and Maggetti 2020), it

always proceeds bottom-up (Schneider and Wagemann 2012, p. 104; Duša 2019c, chap. 5.5), i.e. by first checking single factor values for their necessity and then, if desired, proceeding to more complex configurations. In contrast, in its analysis of sufficiency, QCA, as introduced by Ragin (1987), builds on the QMC algorithm and proceeds top-down. CNA, in contrast, proceeds bottom-up only and first performs an analysis of sufficiency and uses the solutions thereof for the subsequent analysis of necessity. As these steps are closely intertwined, it is sometimes wrongly assumed, that CNA performs both analyses simultaneously.

QCA's analysis of sufficiency in all its variants builds on the QMC algorithm. Meanwhile, the eQMC algorithm (Duša and Thiem 2015) and the CCubes algorithm (Duša 2019a) have been developed and implemented in the R package **QCA** to produce solutions in line with the QMC approach. Although these algorithms proceed differently than QMC, i.e. they do not follow QMC's pairwise minimisation process as it is computationally not very efficient, they are implemented in the R package in such a way as to produce solutions in line with the QMC algorithm, and only such solutions, while advancing the R package in terms of speed. eQMC and CCubes are implemented, so to say, to mimic the classical QMC top-down approach. Therefore, I will concentrate on the contrast between the QMC and CNA algorithm in the following. The QMC algorithm stems from electrical engineering where its original purpose is to obtain minimal combinations of inputs, i.e. closed and open gates, to produce a specific electrical output (McCluskey 1956, 1965; Quine 1955). It is mainly driven by cost minimisation and originally was not geared towards causal inference. Moreover, such a setting allows for purposefully realising all configurations of inputs.

In social science, however, data rarely contain all possible configurations leading to the much discussed problem of limited diversity (Cooper and Glaesser 2016b; Ragin 1987; Ragin and Sonnett 2005; Schneider and Wagemann 2006, 2012). Limitedly diverse data often do not allow to remove all redundancies following a top-down approach from observed configurations that display the outcome value of interest only. To approach this problem, Ragin and Sonnett (2005) introduce three strategies leading to three types of solutions: (1) the conservative (or most complex) solution assumes that unobserved configurations will not show the outcome value of interest, (2) the intermediate solution makes assumptions for certain of the unobserved configurations on the basis of theoretical expectations and (3) the parsimonious solution assumes that those unobserved configurations that allow for simplifying the solutions are associated with the outcome value of interest. Once all configurations that are observed or assumed to display the outcome value of interest are identified, the QMC algorithm proceeds by repeatedly minimising these configurations via pairwise comparisons in order to output the so called *prime implicants*, which are then selectively combined into disjunctions (Duša 2019c, chap. 8; Oana et al. 2021, chap. 4.3.1; Ragin 1987, chap. 6; Schneider and Wagemann 2012, chap. 4.3).

In contrast, the CNA algorithm uses a bottom-up search strategy for both the analysis of sufficiency and necessity. Moreover, it is custom-built to exclusively recover causality as defined by the INUS-theory of causation (Baumgartner 2015; Baumgartner and Ambühl 2020b). In a bottom-up manner, the CNA algorithm first checks single factor values for their sufficiency for the outcome and conjunctively adds additional factors only if no less complex conjunction is sufficient. The revealed minimally sufficient conditions are then tested for necessity and only combined disjunctively if no less complex disjunctions of minimally sufficient elements are necessary—a procedure that I will explain in more detail in the following chapter. The CNA algorithm is not restricted to configurations that display the outcome value of interest only, but draws causal inferences using both configurations that display the outcome value of interest

and not. Overall, the CNA approach automatically leads to redundancy-free solutions without the need for a minimisation procedure and without the need for making assumptions on unobserved configurations, that is without the need for counterfactual analyses. Another direct consequence of CNA's bottom-up approach is its ability to identify multi-outcome structures and treat any factor as exogenous (Baumgartner and Thiem 2015; Baumgartner and Ambühl 2020a). In contrast, QCA restricts itself to the QMC algorithm or, more precisely, to deriving models in line with QMC. If one applies QMC, one necessarily proceeds top-down which makes a minimisation process necessary.

The main reason for the top-down procedure given by QCA scholars is that in this way the integrity of all investigated causal factor values per case is ensured in the course of the whole analysis (Ragin 2014, p. xxi). It is argued that "causes are not viewed in isolation but always within the context of the presence and absence of other causally relevant conditions" (Ragin 1987, p. 93). For CNA, this statement could be reformulated as follows: Causes are not viewed in isolation but always within the context of the presence and absence of at least the outcome factor or any additional number of other investigated causal factors. As the bottom-up approach is still a rather novel add-on to the family of CCMs, it has not yet been subject to much explicit discussion by QCA scholars. Thus, it has not yet been explicitly excluded that QCA could also proceed bottom-up in the analysis of sufficiency for certain search targets in a similar way like CNA while leaving room for further expansion of the method.

3 QMC vs CNA: an empirical example

I will illustrate the differences between the CNA and QMC algorithm on an example using the R packages **QCA**, version 3.10 (Duşa 2019c), and **cna**, version 3.0.1 (Baumgartner and Ambühl 2020b). The main goal is to clarify the differences between the two algorithms so that the resulting differences in the produced solutions become more easily accessible and comprehensible. For the example, I assume a causal structure (ground truth) such that *A* alone or *B* in conjunction with *C* are sufficient and necessary for the outcome *Y* to occur.

$$\text{Ground Truth} : A + B * C \leftrightarrow Y \quad (1)$$

From this ground truth, I will generate crisp-set data sets of ascending proximity to real-world data and show how the two algorithms operate on these data sets, how the QCA and CNA approach deviate from each other, and which solutions they offer. Table 1 introduces the three data sets. Table 1a shows the first data set, dt1, with all eight possible configurations that are in line with the assumed ground truth (1). Every configuration is represented by one case. In other words, dt1 shows ideal-world data with no limited diversity and a frequency of 1 for every configuration. In the second data set, dt2 (Table 1b), I remove 25% of compatible configurations to introduce limited diversity. Two of eight logically possible configurations (c3, c6) are no longer represented. Finally, I introduce 11% of noise and varying frequencies in dt3 (Table 1c). Dt3 contains one configuration of *A*, *B*, *C*, and *Y* not in line with the assumed ground truth (c9) and all configurations are no longer represented by exactly one case but by a varying number of cases. Note that all three tables represent observed data. In order to still compactly represent the data, all tables list all observed configurations exactly once and contain a column *n* that indicates the number of observed

Table 1 (a) Ideal-world data (dt1), (b) data with limited diversity (dt2), and (c) data with limited diversity, noise and varying frequencies (dt3). Light grey rows in dt1 are missing from dt2 and dt3. The dark grey row in dt3 is incompatible with the ground truth (1). All tables represent observed data and summarise identical configurations of A, B, C, and Y. Column n indicates how often a configuration is observed

	A	B	C	Y	n
c_1	1	1	1	1	1
c_2	0	1	1	1	1
c_3	1	0	1	1	1
c_4	1	1	0	1	1
c_5	1	0	0	1	1
c_6	0	0	1	0	1
c_7	0	1	0	0	1
c_8	0	0	0	0	1

(a) dt1

	A	B	C	Y	n
c_1	1	1	1	1	1
c_2	0	1	1	1	1
c_4	1	1	0	1	1
c_5	1	0	0	1	1
c_7	0	1	0	0	1
c_8	0	0	0	0	1

(b) dt2

	A	B	C	Y	n
c_1	1	1	1	1	5
c_2	0	1	1	1	5
c_4	1	1	0	1	1
c_5	1	0	0	1	4
c_7	0	1	0	0	1
c_8	0	0	0	0	5
c_9	0	1	1	0	2

(c) dt3

cases per configuration. This way of depicting data is called *configuration table* in CNA jargon (Baumgartner and Ambühl 2020b).

3.1 CNA

To begin with, consider the ideal-world data set in Table 1a. The CNA algorithm proceeds in four stages to derive causally relevant models that are—to a certain degree specifiable by the researcher—sufficient and at the same time necessary for the outcome (Baumgartner and Ambühl 2020b, a).

Stage 1: As the CNA algorithm is capable of treating values of more than one factor as outcomes, in the first stage it builds a set of potential outcomes \mathbf{O} and assigns a set of potential cause factors \mathbf{C}_O to each potential outcome. By default, the `cna()` function from the R package `cna` will treat the value 1 for every given factor as a potential outcome unless specified otherwise. Thus, the function does not need to be told which factors are the outcome factors. However, if prior knowledge exists about potential outcome factors or multi-outcome structures, a causal ordering can be provided. For example, one could assume a combined causal chain and common cause structure with two outcomes Y and C in which A and B are causes of C and $A, B,$ and C are causes of Y . This would lead to $\mathbf{O} = \{Y, C\}$, $\mathbf{C}_Y = \{A, B, C\}$ and $\mathbf{C}_C = \{A, B\}$. The ordering can be given to `cna()` via the argument `ordering`, for example `ordering = list(c("A", "B"), "C", "Y")`. But let us, for demonstration purposes, keep Y as the only assumed outcome and causally locate Y after all other factors using the argument `ordering = list("Y")` which leads to $\mathbf{O} = \{Y\}$ and $\mathbf{C}_Y = \{A, B, C\}$.

Stage 2: In the second stage, CNA looks for relations of *sufficiency*. To this end, it builds a set of so called *minimally sufficient conditions* (msc) for each potential outcome, thus, for the single outcome Y in our example. It first checks if single causal factor values, i.e. 0 or 1 for crisp set data, are sufficient for the outcome and then conjunctively adds factor values until sufficiency is reached. Sufficiency is assumed when the *consistency*, i.e. the degree to which the behaviour of an outcome obeys the conjunction of factor values (Baumgartner and Thiem 2015; Ragin 2006), equals or is above the defined consistency threshold. For the ideal-world data set dt1, I set the consistency threshold to `con = 1`. Put differently, I define that a factor value X (or a conjunction of multiple factor values) is sufficient for the outcome Y if, and only if, the statement “if X is present, then Y is present” (Baumgartner 2021) is strictly true.

Table 2 CNA - set of msc inferred from dt1

	Outcome	msc	Consistency	Coverage
1	Y	$A \rightarrow Y$	1	0.8
2	Y	$B * C \rightarrow Y$	1	0.4

As the CNA algorithm follows a bottom-up approach, it starts by checking single factor values for sufficiency. The first single factor value to be checked is A . In dt1 (Table 1a) there are cases, in which A is true and Y is true as well (c1, c3, c4, and c5) and at the same time there are no cases in which Y is not true but A is. Thus, the consistency with which A is sufficient for Y is 1 and A is part of the set of msc. For B , we observe three cases in which B is true and Y is true as well (c1, c2, and c4). However, in the seventh row, B is true, but Y is not true. Thus, the consistency with which B is sufficient for Y falls under 1, more precisely to 0.75, and therefore B is no part of the set of msc. The same holds true for C with a consistency of 0.75 as well as for a , b , and c with consistencies of 0.25, 0.5, and 0.5, respectively.

For every single factor value that is not sufficient (in this case B , C , a , b , and c), and only for these factor values, the CNA algorithm *conjunctively* adds one additional potential causal factor value and tests the so derived conjunctions for sufficiency. Again, sufficiency is assumed when the consistency equals or is above the defined consistency threshold. In the example, no conjunction is tested that includes A as A itself is already minimally sufficient. The next conjunction to be tested is $B * C$, which in fact is sufficient for Y with a consistency of 1. No other conjunction of two factor values meets the consistency threshold. This procedure is systematically repeated for all factor values in the data if, and only if, no less complex conjunction is already found to be sufficient. Thus, whenever a conjunction meets the consistency threshold, it is automatically minimally sufficient, that is redundancy-free. The set of msc can be called via the $msc()$ function and contains two msc for the example on dt1 (Table 2).

Stage 3: In the third stage, the CNA algorithm derives so called *atomic solution formulas* (asf) by checking all msc for their *necessity* for the outcome. In this way, the CNA algorithm accounts for both sufficiency and necessity in one solution formula—a major difference to QCA. Necessity is assumed when the *coverage*, i.e. the degree to which a model accounts for the behaviour of the corresponding outcome (Baumgartner and Thiem 2015; Ragin 2006), is met. For the ideal-world data set dt1, I set the coverage threshold, like the consistency threshold in the previous stage, to 1. Put differently, I define that a factor value X (or a disjunction of multiple factor values) is necessary for the outcome Y if, and only if, the statement “if Y is present, then X is present” (Baumgartner 2021) is strictly true.

Note that the notion of consistency and coverage in CNA is the same as consistency and coverage of sufficiency in QCA (Ragin 2006; Schneider and Wagemann 2012, chaps. 5.2, 5.3). Consequently, consistency in CNA is the same as coverage of necessity in QCA (Ragin 2006; Schneider and Wagemann 2012, chap. 5.5) and coverage in CNA is the same as consistency of necessity in QCA (Ragin 2006; Schneider and Wagemann 2012, chap. 5.4). However, the use of consistency and coverage thresholds differs fundamentally between the QCA and CNA approach, a fact that I will continue to illustrate throughout the example.

To come back to the CNA protocol, we see from the msc object in Table 2 that no msc is an asf on its own as their individual coverage values are both below 1 (more precisely 0.8 and 0.4). If, and only if, single msc do not meet the coverage threshold—like

Table 3 CNA - asf = csf inferred from dt1

Outcome	Solution	Consistency	Coverage
Y	$A + B * C \leftrightarrow Y$	1	1

in our example—the CNA algorithm combines msc *disjunctively* and systematically tests these disjunctions for their necessity for the outcome. In our example the disjunction of A and $B * C$ indeed meets the coverage threshold and, thus, is the only asf, which can be called via `asf()` (Table 3). From two msc, only one disjunction can be built and tested. If this disjunction had a coverage below 1, the set of asf would be empty and `cna()` could not derive any model given the set consistency and coverage thresholds.

Again, the CNA algorithm builds and tests more complex disjunctions only if no less complex disjunctions of msc are necessary. This means that disjunctions of three msc are only built and tested for those disjunctions of two msc that do not meet the coverage thresholds and so forth. Thus, whenever a disjunction meets the coverage threshold, it is automatically a minimally necessary condition, i.e. redundancy-free. In this way, the CNA algorithm derives solution formulas for *ex ante* defined consistency and coverage thresholds—a major difference to the sufficiency analysis in QCA. More precisely, every msc from stage 2 must meet the consistency threshold and every disjunction of several msc equals the maximum of each single msc’s consistencies contained in this disjunction. The coverage threshold, however, must be met by the asf only. In contrast, for the analysis of sufficiency in QCA, only the consistency for sufficient conditions is a defining threshold, while the coverage for sufficient conditions is a desirable value, but not a defining one and needs to be assessed *ex post*—a characteristic I will come back to later when illustrating the QMC algorithm.

Stage 4: To finish off the CNA protocol, in the fourth stage a set of complex solution formulas (csf) is built encompassing all atomic solution formulas for the different outcomes, if different outcomes are allowed. In this step all derived msc are not only conjunctively connected to one solution (like it is currently the case for causal chains that are mimicked by the `causalChain()` function in the R package **QCA** 3.10), but by default also structural redundancies, solutions with partial structural redundancies, and tautologous as well as contradictory solutions are eliminated. Overall, this step unveils multi-outcome structures like causal chains and common cause structures. As I specified only one outcome (Y) in the first stage, the csf will be the same as the asf (Table 3).

3.2 QCA

Contrary to CNA, QCA puts a separate analysis of necessity before the analysis of sufficiency (Oana et al. 2021, chaps. 3, 4; Schneider and Wagemann 2010, 2012, chap. 3). The analysis of necessity is based on calculating consistency values for the necessity of single factor values or disjunctions of factor values for the outcome of interest (Duşa 2019c, chap. 5.4; Oana et al. 2021, chap. 3.3; Thomann and Maggetti 2020). Contrary to QCA’s top-down analysis of sufficiency, QCA’s analysis of necessity proceeds bottom-up, similar to CNA, i.e. by checking simple expressions for their necessity first and subsequently adding more complex expressions. This can *inter alia* be performed by

Table 4 QCA analysis of necessity - solutions

	Consistency	Coverage	RoN
$A + B$	1	0.833	0.667
$A + C$	1	0.833	0.667

the `superSubset()` function in the R package **QCA** (Duşa 2019c, chap. 5.5; Schneider and Wagemann 2012, p. 104). Every factor value or disjunction of factor values is assessed as potentially necessary if its consistency of necessity is at least 0.9 (Ragin 2008). When performing the analysis of necessity on `dt1`, we see that there is no single factor value present in all cases in which Y is also present. In fact, with 0.8, the single factor value A reaches the highest consistency of necessity. However, the two disjuncts $A + B$ and $A + C$ both show a consistency of necessity of 1 (Table 4). For the analysis of necessity, the QCA protocol further advises to check for empirical relevance indicated by the coverage of necessity and the relevance of necessity (RoN) indicators (Duşa 2019c, chap. 5.3; Oana et al. 2021, chap. 3.2). In our example, both disjuncts pass this tests as the coverage is high (0.883) and the RoN exceeds 0.5 (0.667). To round up QCA's analysis of necessity, researchers should assess the conceptual meaningfulness of the necessary expressions (Oana et al. 2021, chap. 3.2). Due to its overall bottom-up procedure, QCA's analysis of necessity and CNA proceed rather similarly.

In contrast, QCA's analysis of sufficiency follows QMC's top-down approach in order to derive configurations that are sufficient for an outcome and at the same time minimised to a certain degree (Ragin 1987). While the main focus in the preceding analysis of necessity lies on the consistency of necessity, the main focus in the analysis of sufficiency lies on the consistency of sufficiency. In contrast, CNA is designed to derive solutions that meet both consistency and coverage thresholds at the same time. Further, the CNA algorithm operates bottom-up directly on the data set at all stages. In contrast, the QMC algorithm operates top-down and therefore requires a *truth table*, i.e. a matrix of all possible configurations of causal factors to which outcome values are assigned depending on the consistency to which a particular configuration is sufficient for the outcome. Table 6 represents `dt1`, `dt2` and `dt3` as truth tables.

A truth table is constructed in the following three steps (Duşa 2019c, chap. 7; Oana 2021, chap. 4.3; Schneider and Wagemann 2012, chap. 7).

Step 1: For fuzzy as well as crisp sets, all logically possible combinations of the ideal values 0 and 1 for all investigated causal factors are listed. The first three columns of Table 5 a-c are the results of this step. Configurations comprising all causal factors, and only those, are also referred to as *minterms* in Boolean algebra.

Step 2: If not ambiguous, every case is assigned to exactly one ideal truth table row to which it belongs best. This is done by calculating the membership scores for all cases in all configurations. Each such case will have a membership of above 0.5 in only one configuration. If, in the case of fuzzy sets, a case has a membership value of exactly 0.5 in at least one causal factor, the case is ambiguous in the sense that it will have partial membership in two configurations. In this case, it cannot be determined to which row a case belongs best. In the current R package **QCA**, such cases will not appear in the truth table and the warning is produced that "Fuzzy causal conditions should not have values of 0.5 in the data." However, cases with ambiguous memberships in two configurations will nevertheless be used for calculating parameters of fit. As in `dt1` each configuration is covered by

Table 5 Truth tables based on the three data sets introduced in Table 1. Bold values are assigned based on consistency levels (incl) for particular configurations of causal factors

	A	B	C	OUT	incl	n	cases
(a) Truth table based on dt1							
c_1	1	1	1	1	1.000	1	1
c_2	0	1	1	1	1.000	1	2
c_3	1	0	1	1	1.000	1	3
c_4	1	1	0	1	1.000	1	4
c_5	1	0	0	1	1.000	1	5
c_6	0	0	1	0	0.000	1	6
c_7	0	1	0	0	0.000	1	7
c_8	0	0	0	0	0.000	1	8
(b) Truth table based on dt2							
c_1	1	1	1	1	1.000	1	1
c_2	0	1	1	1	1.000	1	2
c_3	1	0	1	?	-	0	-
c_4	1	1	0	1	1.000	1	4
c_5	1	0	0	1	1.000	1	5
c_6	0	0	1	?	-	0	-
c_7	0	1	0	0	0.000	1	7
c_8	0	0	0	0	0.000	1	8
(c) Truth table based on dt3							
c_1	1	1	1	1	1.000	5	1, 1.1, 1.2, 1.3, 1.4
c_2	0	1	1	0	0.714	7	2, 2.1, 2.2, 2.3, 2.4, 9, 9.1
c_3	1	0	1	?	-	0	-
c_4	1	1	0	1	1.000	1	4
c_5	1	0	0	1	1.000	4	5, 5.1, 5.2, 5.3
c_6	0	0	1	?	-	0	-
c_7	0	1	0	0	0.000	1	7
c_8	0	0	0	0	0.000	5	8, 8.1, 8.2, 8.3, 8.4

exactly one case and there are no deviant cases, the assignment of cases to truth table rows is straightforward. In dt2 cases 3 and 6 are missing. Thus, the corresponding configurations have no cases assigned.

Step 3: Based on the assigned cases and the computed consistency of sufficiency values, each row is classified as sufficient for the outcome, not sufficient for the outcome, or a logical remainder. Logical remainders are rows that are not represented by any case (c_3 and c_6 in Table 5b and c). For all other rows, their consistency as a sufficient row for the outcome is calculated (incl) and, based on this value, a row is declared as sufficient for the outcome (OUT=1) or not (OUT=0). These values are bold in Table 5. All configurations that display a consistency of necessity equal to or above a consistency threshold to be defined by the researcher are assigned an OUT value of 1, all configurations that display a consistency of necessity below this threshold, are assigned an OUT value of 0.

The construction of a truth table is only necessary as long as an algorithm proceeds top-down and becomes obsolete in the CNA procedure. In contrast, the QMC algorithm is necessarily based on a truth table (Duşa 2019a; Oana et al. 2021, chap. 4.3; Ragin 1987, chap. 6; Schneider and Wagemann 2012, chap. 4.3). All rows with an OUT value of 1 are then passed to the minimisation process. Logical remainders are treated differently depending

on the QCA solution type. The conservative solution assumes that all logical remainder rows have an OUT value of 0, the intermediate solution assigns some logical remainders OUT values of 0 and some OUT values of 1 based on theoretical knowledge, and the parsimonious solution assumes that all logical remainders that allow for simplifying the solution have an OUT value of 1. All these decisions are part of the analytic QCA process.

All minterms with an OUT value of 1 are then compared by pairs and if a pair differs by exactly one conjunct, this conjunct is eliminated. The results of this pairwise comparison process are called *implicants*. The implicants are then further minimised if they differ in one conjunct only, until nothing more can be minimised. The solutions from this step are the so called *prime implicants*, which build the prime implicant chart. For the truth table corresponding to dt1 (Table 5a), this means that e.g. c1-c5 are part of the minimisation process for the conservative solution. Further, c1 and c2 only differ in A and, thus, A is eliminated. This leads to $B * C$ as one implicant. Other implicants are $A * C$, $A * B$, $A * b$ and $A * c$, which are further minimised to A, so that the resulting two prime implicants are A and $B * C$.

To resolve the prime implicant chart, prime implicants are disjunctively combined in such a way as to cover each of the initial configurations that are part of the minimisation process. In our example, A covers the configurations 1, 3, 4, and 5, whereas $B * C$ covers the configurations 1 and 2. Only the disjunction of both prime implicants covers all configurations 1-5, and, thus, the conservative solution is $A + B * C$. The step of solving the prime implicant chart by disjunctively combining the prime implicants can be seen as an integrated analysis of necessity step in so far as the derived solution has to cover all initial configurations. Thus, one can summarise that while CNA proceeds from an analysis of sufficiency to an analysis of necessity both being intertwined, QCA puts a separate analysis of necessity first and then, in what is called the analysis of sufficiency also intertwines an analysis of sufficiency with a subsequent moment of analysis of necessity, namely when resolving the prime implicant chart. The process of resolving the prime implicant chart is performed by the `minimize()` function and the prime implicants are stored in the object `PIchart`. If more than one possibility exists to cover all initial configurations, `minimize()` follows the principle of row dominance, i.e. prime implicants that cover more configurations dominate those that cover less and are prioritised (Duša 2019a). By setting the argument `all.sols` from the `minimize` function to `TRUE`, this principle can be circumvented and all possible solutions outputted.

Figure 1 summarises the CNA and QCA protocol.

For ideal-world data like dt1 (Table 1a), the QCA conservative solution shown in Table 6 will be identical to the QCA parsimonious solution just like the CNA solution will be identical to all QCA solution types. Thus, all of the above elaborated different procedures will produce the same solutions for all ideal-world data sets.

However, when faced with noisy data or consistency and coverage thresholds below 1, the CNA solution may deviate from QCA's parsimonious solution just as QCA's conservative solution may deviate from QCA's parsimonious solution.¹ In fact, as widely agreed upon, real-world data tend to be less than perfect (e.g. Baumgartner 2021; Rohlfing 2016; Schneider 2018). They can be incomplete, i.e. limitedly divers, contain observations that are more frequent than others, or contain noise. By *noise* I refer to all configurations that

¹ As the intermediate solution is based on analytic decisions specific to research settings and due to the ongoing debate towards a sufficiently defined search target for the intermediate solution (Baumgartner 2021), this article considers QCA's conservative and parsimonious solutions only.

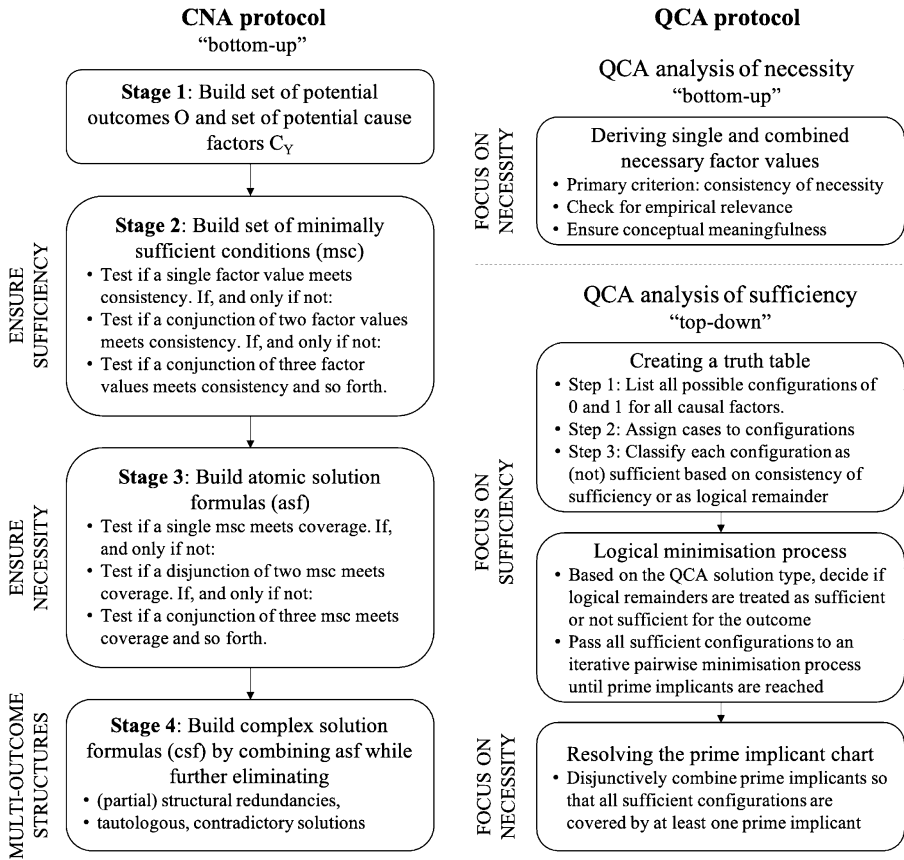


Fig. 1 CNA vs QCA protocol

are incompatible with the underlying causal structure (ground truth). Sources of noise may be measurement errors or deviant correctly measured configurations that e.g. may be due to variations of unmeasured causes. To account for real-world data, I will now gradually introduce limited diversity, noise, and varying frequencies (Table 1b and c) to the previous example to show how models produced by QMC and CNA can diverge. Later on, this one-shot example will be extended by simulation studies to contribute to a more general picture of the ratio of identical models produced by QCA and CNA.

Dt2 from our example shows a data set with 25% limited diversity produced by deleting c3 and c6 from dt1 but no other deficiencies. Table 7 shows the CNA solution, the QCA conservative solution, and the QCA parsimonious solution inferred from dt2. As dt2 is no longer as informative as dt1, CNA will, in this case, produce the submodel $A + C$ of the ground truth (1). However, depending on which rows are deleted, the derived model could also equal the full ground truth or a different submodel of the ground truth. More precisely, if rows 1&3, 1&4, 1&5, 1&8, 3&4, 3&5, 3&8, 4&5, 4&8, or 5&8 were deleted, the produced model would equal the full ground truth. If rows 1&7, 3&7, 4&7, or 5&7 were deleted, the produced model would be $A + B$. If rows 1&6, 4&6, or 5&6 were deleted, the produced model would also be $A + C$. Finally, if any combination including row 2 was

Table 6 QCA analysis of sufficiency - conservative = parsimonious solution inferred from dt1

Solution	Consistency	Coverage
$A + B * C \leftrightarrow Y$	1	1

Table 7 CNA and QCA analysis of sufficiency - solutions inferred from dt2

Solution type	Solution	Consistency	Coverage	Threshold value*
CNA solution	$A + C \leftrightarrow Y$	1	1	1
QCA conservative solution	$A * c + B * C \leftrightarrow Y$	1	1	1
QCA parsimonious solution	$A + C \leftrightarrow Y$	1	1	1

* Note that thresholds are used differently in CNA and QCA

deleted, A would be a duplicate of Y leading to an error message in CNA due to insufficient variance in A. Following the CNA algorithm despite the insufficient variance would lead to the submodel A only. In any case, when only faced with limited diversity (and no noise), the CNA solution (just like the QCA parsimonious solution) does not include false positives. Depending on the data quality, the full ground truth or submodels of the full ground truth will be derived.

In contrast, the QMC top-down approach will produce the more complex model $A * c + B * C$ for the conservative solution as dt2 does not allow for maximum pairwise minimisation anymore. This model contains false positives but also the full ground truth (1) as it is a supermodel of the ground truth. If rows 1&4 or 1&8 were deleted, the produced solution would equal the ground truth. If any other combination of two rows was deleted, all solutions would produce a more complex model than the ground truth (1). In the most complex case, the solution $A * B * c + A * b * C + a * B * C$ would be outputted if rows 1&5 were deleted. The QCA parsimonious solution is identical to the CNA solution for our example with perfect consistency and no noise.

However, once data are generated that resemble real-world data even more, the non-identity of CNA and QCA solutions may also extend to the QCA parsimonious solution. The next data set dt3 shows different frequencies for the observations and 11% noise due to the incompatible configuration c9 that is observed twice. With different frequencies and deviant cases, full consistency cannot be achieved anymore. Thus, I set the consistency and coverage thresholds for CNA to 0.75, a usual threshold value in applied CNA (Baumgartner and Ambühl 2020b). The recently introduced robustness analysis by Parkkinen and Baumgartner (2021) offers a more sophisticated procedure to determine suitable threshold values. Table 8 summarises the CNA and QCA solutions inferred from dt3. For the given threshold values of 0.75, CNA produces two models that represent an ambiguity, i.e. models that fit the data equally well: $A + C$ and $A + B$.

It is important to stress that the CNA protocol imposes high standards on both consistency and coverage thresholds and uses them at different stages than the QMC algorithm. A consistency threshold of 0.75 means that not only all msc need to meet the specified consistency but also the asf and the csf. Moreover, the asf and the csf need to meet the coverage threshold. Overall, the CNA solutions reach a consistency of 0.882 and full coverage of 1.

Table 8 CNA and QCA analysis of sufficiency - solutions inferred from dt3

Solution type	Solution	Consistency	Coverage	Threshold value*
CNA solution	(1) $A+C \leftrightarrow Y$ (2) $A+B \leftrightarrow Y$	0.882	1	0.75
QCA conservative solution	$A * B + A * c \rightarrow Y$	1	0.667	0.75
QCA parsimonious solution	$A \rightarrow Y$	1	0.667	0.75

* Note that thresholds are used differently in CNA and QCA

In contrast, QCA's conservative solution shown in Table 8 only contains one disjunct of the ground truth (A). The other disjunct $B * C$ is not contained anymore. I specified the same value of 0.75 for the consistency threshold as I did for CNA, which is in line with the QCA protocol. Keep in mind that the consistency threshold in QCA's analysis of sufficiency is used differently than in CNA. It is used as the cut-off value above which truth table rows are declared as sufficient for the outcome. Indeed, any consistency threshold between 0.714 and 1 would lead to the same result. At the same time, a consistency threshold below 0.714 would not be in line with the current best practices for QCA applications (Schneider and Wagemann 2010). However, QCA puts a lot of emphasis on case knowledge throughout the analysis and case knowledge can also be used to determine consistency thresholds. More precisely, cases that show a different outcome value for a certain configuration than the assigned OUT value are so-called deviant consistency in kind cases (DCK). In our example, cases 9 and 9.1 in the truth table based on dt3 are such deviant cases. The more DCK occur in a truth table row, the more reason exists to not assume the associated configuration as sufficient for the outcome. However, if the study of DCK would uncover that there exist substantial reason to believe that these cases are due to noise in the data (e.g. due to response bias), the consistency threshold could be lowered to less than 0.714, which would reveal QCA solutions closer to the ground truth. This process would also be in line with QCA standards of good practice of engaging in an iterative process between theory and evidence. Overall, this example shows that while the QCA conservative solution strives for more complex models to integrate all causally relevant factors (while accepting false positives), it is not guaranteed to contain all disjuncts of the ground truth. Or put differently, the QCA conservative solution is not always a supermodel of the ground truth. Finally, the QCA parsimonious solution shown in Table 8 covers only one causally relevant disjunction of the ground truth (A) and is less complex than the CNA solution.

As the different ways of setting consistency and coverage thresholds are decisive features distinguishing the CNA and QCA approach, I introduce the concept of *ex ante* and *ex post* usage of thresholds to round up the comparison of the CNA and QCA protocol. QMC uses the consistency threshold to decide whether a configuration (truth table row) is treated as sufficient for the outcome or not. The overall consistency and coverage of the solution, however, are to be assessed *ex post*. While high coverage of sufficiency in the analysis of sufficiency is a desirable feature and the interpretation of results is bound to it, it is not a defining threshold. In our example, both QCA solutions have coverage values of 0.667, which is lower than what the CNA solution produces. In turn, both QCA solutions have full consistencies of 1. In other words, in the analysis of sufficiency, QCA strives for as high consistency as possible while accepting low coverage values *ex ante*. If QCA researchers do not allow for coverage values as low as 0.667, they would need to change the basic

research settings in a back and forth process. In contrast, CNA *ex ante* defines thresholds for both parameters of fit. For CNA, the consistency threshold is a decisive criterion for deriving both msc as well as asf, while coverage is a decisive criterion for deriving asf. Thus, CNA reveals a model with both solution consistency and coverage above 0.75. More precisely, the CNA solution has a lower solution consistency compared to the QCA solutions (0.833), but full coverage (1) and overall a higher model fit.

4 On the non-identity of models produced by QMC and CNA

The single example in the previous chapter shows that the solutions produced by QCA and CNA are not always equivalent. However, once the very different approaches of the algorithms underlying QCA and CNA, namely the QMC algorithm and the CNA algorithm, are understood, this observation comes as no surprise. More precisely, I have explained two fundamental differences between QCA and CNA: (1) the different arrangements of analyses of sufficiency and necessity and (2) their different use of top-down vs. bottom-up approaches including their different uses of parameters of fit therein. Building on this, the following simulation studies assess to which extent CNA and QCA output different solutions.

As the CNA algorithm does not need to assign crisp outcome values to configurations but instead calculates the parameters of fit for msc, asf and csf directly on the fuzzy or crisp outcome values, it also does not depend on a truth table. In contrast, the truth table is at the core of QCA. Once the QMC algorithm is confronted with limited diversity, it cannot assign outcome values to all configurations anymore and therefore depends on counterfactual assumptions. If a researcher decides to assume none of the non-observed configurations as sufficient for the investigated outcome, this will result in the conservative solution. If they decide to assume some of the non-observed configurations as sufficient for the outcome and some not based on theoretical knowledge, this will lead to the intermediate solution. If they decide to assume all of the non-observed configurations that allow for simplifying the solution as sufficient for the investigated outcome, this will result in the parsimonious solution.

In order to assess the extent to which the QCA parsimonious solution, i.e. the most similar QCA solution type to CNA, and the CNA solution differ, I conducted simulation studies. Simulation studies have a long history in the evaluation of new and the comparison of alternative methods. As such, they are an important tool for the constant improvement of methods (Hug 2014). Accordingly, CCM scholars increasingly call for simulation studies as an appropriate evaluation tool (Rohlfing 2016). However, their use is still surprisingly scarce compared to the wide spread of CCM applications (for exceptions, see e.g. Arel-Bundock 2019; Baumgartner and Ambühl 2020a; Baumgartner and Thiem 2020; Cooper and Glaesser 2016a; Duşa 2019b; Hug 2013; Kroglund et al. 2015).

I simulated 27 scenarios with varying (1) sample sizes ($n = 20, 60, 120$), (2) noise levels (noise = 0%, 10%, 20%) and (3) thresholds (incl.cut/con/cov = 0.75, 0.8, 1), for each of which I ran 1000 trials of QCA and CNA. To generate data structures for each of the trials, I first randomly selected 1000 Boolean causal structures (ground truths) out of five potential causal factors and one outcome factor. The drawn causal structures consist of two to nine (conjunctively and disjunctively grouped) causal factor values. Second, I built ideal-world data for these ground truths, i.e. data containing all configurations

in line with the ground truth and only such configurations. Third, I constantly introduced 30% limited diversity by randomly deleting 30% of the generated configurations. Fourth, from the 1000 limitedly diverse data sets, I randomly selected 20, 60, and 120 cases, which led to 3000 data sets. Fifth, I introduced noise ratios of 0%, 10%, and 20% into the 3000 limitedly diverse data sets with different sample sizes, which led to 9000 data sets. To do so, I randomly replaced 0%, 10%, and 20% of observations by observations not in line with the given ground truth (for a detailed description of the data generating process using the R package `cna`, see Baumgartner and Ambühl 2020b).

For each of the so derived 9000 data sets, I computed the QCA parsimonious solution by means of the QMC algorithm by setting the method in the `minimize()` function to QMC and I computed the CNA solution by means of the CNA algorithm. For the analyses, I set the three threshold values 0.75, 0.8, and 1 as the consistency threshold above which configurations are assumed as sufficient for the outcome (`incl.cut`) for the QCA runs and as consistency (`con`) and coverage (`cov`) thresholds for the CNA runs. This led to 27000 pairs of analyses. I finally compared the solutions and measured the ratio of trials that produced identical outputs. As output, I defined the sum of all produced models for the QCA parsimonious solution and the CNA solution. If there was more than one model produced, i.e. ambiguity in the solutions, I counted the outputs only as identical in such trials where all of the produced models were identical. Accordingly, in trials in which some of the produced models were identical and some not, I counted the outputs as non-identical. However, if both methods did not reveal any model at all (empty solutions), I counted their outputs as identical.

Figure 2 shows the varying ratios of trials that produced identical models in 27 scenarios. I will call these ratios *identity ratios*. For example, an identity ratio of 0.359 means that out of 1000 trials, 359 trials produced an identical output. Thus, I use this simulation study for the pure assessment of the degree to which the QCA parsimonious solution and the CNA solution are identical, not for the evaluation of their quality as no comparison with the underlying ground truths is conducted. Moreover, although I fully share the understanding of QCA as both research approach and data analysis technique (Berg-Schlosser et al. 2009; Schneider and Wagemann 2010; Thomann and Maggetti 2020), which is also shared by proponents of CNA (Baumgartner and Ambühl 2020b), this study concentrates on CCMs as data analysis techniques in order to correct the existing picture according to which the QCA parsimonious solution and the CNA solution are identical or mostly identical. Overall, the ratios show that the QCA parsimonious solution and the CNA solution are in less-than-perfect scenarios, i.e. for thresholds below 1 or noisy data, not even frequently identical.

The average overall identity ratio is 15.7%. Only for thresholds set to 1 (`incl.cut/con/cov = 1`) and no noise whatsoever (`noise = 0%`) all of the solutions are identical for each sample size ($n = 20, 60, 120$). Averaged over all non-ideal scenarios, the identity ratios drop drastically to 5.16%. In light of the fundamentally different principles of both algorithms, this finding comes as no surprise. The higher the noise ratios, the lower the identity ratios. This is partly to be explained by the fact that for poor, non-informative data, CNA tends to not output any model at all because it poses high requirements on both parameters of fit (consistency and coverage) of the derived models. In contrast, QCA will more likely produce a solution given such data sets. But also the lower the thresholds, the lower the identity ratios, which can be explained by the inherently different ways, CNA and QMC treat those thresholds. In this case, the low identity ratios stem from differing models rather than from empty CNA solutions. Finally, the lowest sample size produces, *ceteris paribus*, the highest identity ratios. This is surprising as the lowest sample size of 20 is too

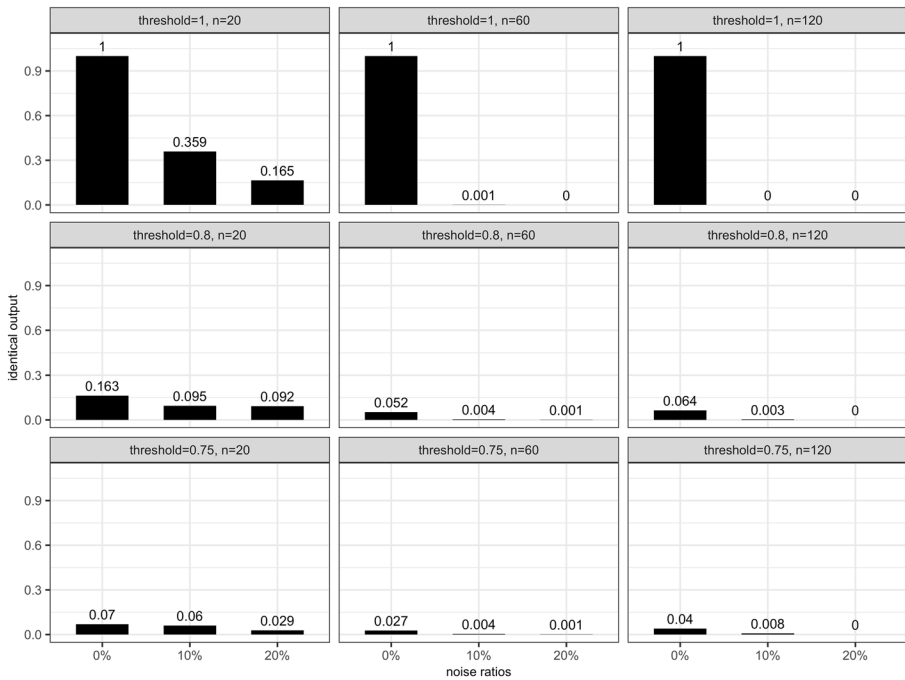


Fig. 2 QCA parsimonious solution (method=QMC) vs CNA atomic solution - ratios of trials that produce identical outputs

low as to include all possible configurations of five causal factors, thus leading to the highest limited diversity. Indeed, good practices of QCA recommend a sample size of at least $n = 32$ when analyzing five causal factors in one analysis (minimum of 2^k cases where k is the number of analysed causal factor values). One explanation for the high identity ratios is that higher limited diversity leads to more empty solutions for both methods. In addition, random noise has a higher impact on smaller sample sizes than on larger sample sizes leading also to more empty solutions for both methods.

5 CCubes - towards a better understanding of what is fuzzy or non-identical

In order to investigate if the previously derived identity ratios between the QCA parsimonious solution and the CNA solution can be further increased, I additionally analysed varying ways to derive the QCA parsimonious solution. More precisely, in addition to the QMC algorithm, Duşa and Thiem (2015) and Duşa (2019a) have meanwhile implemented two further algorithms in the R package **QCA** (Duşa 2019c), i.e. the eQMC (enhanced Quine-McCluskey) algorithm and the CCubes (consistency cubes) algorithm, to improve on memory and speed. Although eQMC and CCubes proceed differently than QMC, they are implemented in such a way as to produce solutions in line with the top-down approach of the QMC algorithm. More precisely, all algorithms are supposed to “return the same, exact solutions,” as stated in the current R Documentation of the `minimize()` function.

Nevertheless, a closer look at the different algorithms may be worthwhile as they inherently follow partly different logics compared to QMC.

As its name suggests, the eQMC, i.e. **enhanced** Quine McCluskey, algorithm complements the QMC algorithm instead of replacing it entirely (Duşa and Thiem 2015). More precisely, in addition to combinations where the outcome is present, eQMC also uses combinations where the outcome is not present to derive the same prime implicants for the parsimonious solution as the QMC algorithm. These prime implicants are derived by calculating the differences between all possible less complex logical expressions that cover all combinations where the outcome is present and all possible less complex logical expressions that cover all combinations where the outcome is absent, preserving those expressions where the outcome is present and subsequently further minimizing redundant expressions (for a more detailed description of eQMC, see Duşa and Thiem 2015). This procedure was found to lead to the same implicants for the parsimonious solution. Accordingly, for my previous simulation, the results did not change when I changed the argument `method = QMC` to `method = eQMC`.

The recently developed CCubes algorithm (Duşa 2019a), however, inherently follows a different logic and proceeds bottom-up, similar to CNA. Thus, besides its substantially increased computing power in deriving parsimonious solutions in line with QMC, CCubes also provides avenues for broadening the QCA parsimonious solution repertoire by “CNA-style” solution types. CCubes is founded on the observation that “[a] prime implicant is the most simple, non-redundant expression, a superset of at least one observed positive configuration, and not a superset of any of the observed negative ones” (Duşa 2019a, p. 4). Based on this finding, the CCubes algorithm follows a bottom-up logic in deriving the simplest possible (i.e. minimal) expressions that qualify as prime implicants. This is given if expressions are associated with the presence of the outcome and at the same time not with its absence (for a more detailed description of CCubes, see Duşa 2019a). Consequently, CCubes, just like CNA, is capable of running directly on the data set and does not necessarily need a truth table. Indeed, CCubes, just like CNA, does also not rely on counterfactuals to derive its solutions.

As it stands, CCubes can be called in the R package **QCA** in two ways. First, the argument `method` in the `minimize()` function can be set to `CCubes` (just like it can be set to `QMC` or `eQMC`). When called in this way, the `minimize()` function solves the derived prime implicant chart in line with the classical QCA procedure, i.e. by distributing the prime implicants on each of the observed positive configurations. In this way, CCubes is implemented so as to derive the same solutions as when `method` is set to `QMC`. Second, another possibility to call the CCubes algorithm is by giving non-default values to the arguments `sol.cons`, `sol.cov`, or `pi.cons`. When specified, all these arguments automatically change the minimisation method to `CCubes`. From the R package **QCA** 3.10 onward, these arguments are available in the function `causalChain()`.² However, when using the `causalChain()` function with only one outcome specified, it is comparable to the `minimize()` function.

When using the second method to call the CCubes algorithm, the argument `sol.cons` introduces another method of solving the PI chart. Instead of calculating parameters of fit *ex post* and combining the prime implicants so as to cover all observed positive

² In the R package **QCA** 3.10 these arguments were moved from the `minimize()` function to the `causalChain()` function to clearly separate both ways to call CCubes but they are currently also understood by the `minimize()` function in order to secure backwards compatibility

configurations, the prime implicants are combined so as to output only models that reach the consistency for sufficiency specified in the argument `sol.cons`. In doing so, the `causalChain()` function offers the possibility to produce solutions more in line with the CNA approach. Further, if the argument `sol.cov` is specified, the prime implicants are combined so as to output only models that reach the given value of coverage for sufficiency and `pi.cons` accepts only those prime implicants in the analysis of sufficiency that reach a certain consistency threshold. Thus, all these arguments can be used to produce solutions more similar to CNA.

To assess the degree to which the QCA parsimonious solution and the CNA solution are equivalent when using CCubes in several different ways, I ran another round of simulations. I used the same data generation processes as specified in the previous section. However, I produced two different QCA parsimonious solutions and one different CNA solution to account for various circumstances that might reach higher identity ratios.

First, for the comparison that I call *standard_CCubes*, for the QCA solution, I set `method` to CCubes in the `minimize()` function and left all other settings unchanged compared to the simulation in the previous section. Thus, I also derived the CNA solution in the very same way as before.

Second, I produced a CNA solution that I forced to imitate the QCA solution produced in the *standard_CCubes* runs. To do so, I first produced the QCA solution from *standard_CCubes* for each run and then set the `con` and `cov` argument for CNA to the minimal solution coverage and consistency of the produced QCA models. This procedure takes into account the inherently different principles of QCA and CNA when dealing with thresholds. It is important to stress, however, that CNA solutions produced in this way are against the CNA protocol and are only produced to determine the identity ratios under such circumstances, in which the different approaches regarding parameters of fit are mitigated. I call this comparison *imitate*.

Third, I compared the standard CNA solution to what CCubes produces when the `sol.cons` argument is specified, a comparison I call *CCubes*. To do so, I set `sol.cons` to the same value as the arguments `incl.cut` (QCA), `con` (CNA), and `cov` (CNA), i.e. to the threshold specified for the respective run (0.75, 0.8, or 1). Likewise, it is important to stress that this way of producing a QCA solution is against the current QCA protocol.

Figure 3 shows the identity ratios for the three outlined comparisons. The first important insight is that *standard_CCubes* produces the same solutions as shown in Figure 2. Thus, in my simulations the `minimize()` function produces the same solutions in line with the QMC procedure independently of whether `method` is set to QMC, eQMC, or CCubes. For the *standard_CCubes* comparison the identity ratio is 15.7% on average. The results for the *imitate* comparison show differences between both methods even when forcing CNA to use the same, *ex post* derived, consistency and coverage thresholds as QCA produces. The identity is the lower, the higher the noise levels. For small n samples, identity levels are high for all noise levels, which can be explained due to the fact that random noise affects smaller sample sizes more than larger sample sizes. On average, even for the *imitate* comparison, identity ratios are not perfect, more precisely at 78.5% on average. Finally, the *CCubes* comparison shows that when using CCubes with a different method of solving the prime implicant chart (when `sol.cons` is specified), the results equal the CNA results

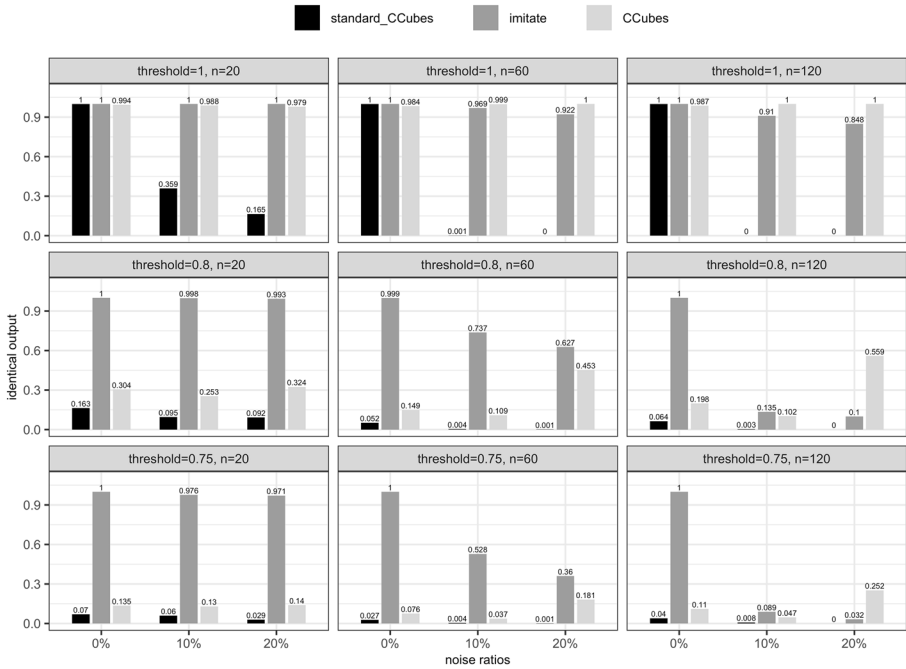


Fig. 3 Three comparisons of QCA parsimonious solutions (method=CCubes) vs CNA solutions - ratios of trials that procure identical output

to a higher extent than the standard procedure. However, the identity ratios are overall still below half of the conducted trials, more precisely at 46.2%.

Overall, the second run of simulations compared a wide variety of QCA parsimonious solutions that can be produced using the CCubes algorithm (Duşa 2019a). The way, however, this powerful algorithm is currently implemented in the R package **QCA** is prone to some confusion as the basic logic of the CCubes algorithm differs substantially from the basic logic of the classical QMC algorithm, which QCA as introduced by Ragin (1987) is based on. An additional confusion might be due to the fact that CCubes (as well as eQMC) is only implemented to produce the QCA *parsimonious* solution. Any *conservative* solution is produced by the classical QMC algorithm, no matter what is specified in the argument method from the `minimize()` function.

6 Discussion and conclusion

My study provides broad evidence that, in real-world scenarios, the QCA parsimonious solution and the CNA solution are only rarely identical. In fact, simulation studies for 27 scenarios with varying consistency and coverage thresholds, sample sizes, and noise in the data show that only in 15.7% of all trials the QCA parsimonious solution is identical to the CNA solution. Even when forcing CNA to produce solutions that equal the overall consistency and coverage of the QCA parsimonious solution, the produced models are not always identical. More precisely, the identity ratio is 78.5%. When using a new form of solving the

prime implicant chart provided by CCubes in the R package **QCA**, 46.2% of all produced parsimonious solutions are identical to the CNA solutions.

Overall, I show that the previously postulated identity between CNA and QCA does not hold true, not even under very identity-favourable settings. By further outlining the substantial degree to which solutions produced by CNA and QCA are not identical, a fact that has been widely disregarded to date, I pave the way for subsequent research on the question which algorithm serves which search target better. My study also points to the need for more simulation studies to validate CCMs given their high-speed developments. This holds especially true for QCA where simulation studies are still scarce while at the same time the number of QCA applications has been accelerating for decades, the quality of which is being criticised especially in business and management research, where much focus is put on QCA as a data analysis technique (Mello 2013; Wagemann et al. 2016). More simulation studies are needed to back up the progressive developments in QCA as a data analysis technique in order to not further aggravate its “teething problem[s]” (Schneider and Wagemann 2010, p. 18), but to reinforce its strengths.

By bringing to light the different possible ways to call CCubes in the R package **QCA** and comparing these possibilities to CNA, my intention is to initiate a long needed discussion on several questions, such as “Is CCubes as implemented in the `minimize()` function really still QCA as defined by Ragin?” “Which algorithm (and how implemented) is most suitable for which search target?” or “Could an additional solution type be added to the QCA protocol, namely a solution type that specifies overall solution consistencies, coverages and PI-consistencies (i.e. solution type *CCubes* from Figure 3)?”. By raising these questions, I propose to open avenues for future integration of CNA’s rationale into specific QCA solution types. I also encourage future research to discuss the pros and cons of this proposition. Moreover, more transparency could be provided on how CCubes, as an inherently bottom-up procedure, is implemented in the R package **QCA** so as to arrive at prime implicants and solutions in line with the top-down QMC approach.

My simulation studies have two main limitations. First, as in every simulation study, more scenarios could be analysed. I analysed three different QCA parsimonious solution types and compared them to CNA solutions in 27 scenarios running the simulations 1000 times each, which led to 108.000 pairs of analyses. However, more settings could be varied, for example the number of causal factors, which I set to a maximum of 5, or the level of limited diversity, which I constantly set to 30%. I also covered only three QCA parsimonious solution types, namely parsimonious solution using `method = QMC` in the function `minimize()`, `method = CCubes` in the function `minimize()`, and `method = CCubes` with `sol.cons = threshold` in the function `causalChain()`. Moreover, I run the simulation in section 4, in addition to `method = QMC` also for `method = eQMC` and both runs produced identical outputs. Nevertheless, there exists a broad range of further solutions that could be produced using the powerful CCubes algorithm. For example, different combinations of setting `sol.cons`, `sol.cov`, or `pi.cons` to non-default values are possible and could be further examined. Finally, noise as well as frequencies were randomly introduced into the analyzed data sets. However, in real-world data some configurations might be more likely to be observed than others and some patterns of noise more likely to occur than others. It might be worthwhile considering how such preferences could be modeled.

Second, the thresholds set in my simulations are prone to criticism. One major argument is that a good QCA application would always chose the consistency value based on the particular research setting and on specific case knowledge. To leverage this shortcoming, contrary to some previous simulation studies that chose extreme threshold

values (e.g. Arel-Bundock 2019), I followed best practices as to the chosen range of threshold values and varied three different threshold values while setting all of them to common practice values in both QCA and CNA (Baumgartner and Ambühl 2020b; Schneider and Wagemann 2010). Moreover, as the main purpose of this paper is to show the degree of non-identity between models produced by different algorithms, the question of setting the right threshold values is not the main focus of this study. Nonetheless, I encourage more research on the choice of consistency thresholds, especially on rethinking the lower bounds of consistency thresholds in current good practices for both methods.

This research set out to show that the QMC and CNA algorithms to a substantial degree produce different models, and, on its way, it also showed that the CCubes algorithm lately implemented in the R package **QCA** is capable of producing a wide range of QCA parsimonious solutions. This powerful algorithm is, inter alia, capable of producing CNA-style solutions due to its inherent bottom-up procedure. More transparency on how CCubes is and could be implemented in QCA could reinforce the strength of QCA by broadening the repertoire of QCA's solution types. More studies are needed to absolutely or comparatively evaluate the different CCMs and solution types. However, to perform such evaluations, clear and agreed-upon search targets are needed in the first place.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s11135-021-01193-9>.

Acknowledgements I thank Michael Baumgartner for his guidance with simulation studies and his much appreciated comments on previous versions of the paper. I also thank both Michael Baumgartner and Adrian Dusa for many fruitful discussions around the topic of this article.

Funding Open access funding was provided by the University of Bergen. Moreover, I gratefully acknowledge the support by the Toppforsk program of the University of Bergen, co-financed by the Trond Mohn Foundation (grant nr. 811886).

Declarations

Conflicts of interest The author has no conflicts of interest to declare.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Álamos-Concha, P., Pattyn, V., Rihoux, B., Schalembier, B., Beach, D., Cambré, B.: Conservative solutions for progress: On solution types when combining QCA with in-depth Process-Tracing. *Quality and Quantity* (2021)
- Arel-Bundock, V.: The double bind of qualitative comparative analysis. *Soc. Methods Res.* 1–20 (2019). <https://doi.org/10.1177/0049124119882460>
- Baumgartner, M.: Inferring causal complexity. *Soc. Methods Res.* **38**(1), 71–101 (2009)

- Baumgartner, M.: Parsimony and causality. *Qual. Quant.* **49**(2), 839–56 (2015)
- Baumgartner, M.: Qualitative comparative analysis and robust sufficiency. *Qual. Quant.* (2021)
- Baumgartner, M., Ambühl, M.: Causal modeling with multi-value and fuzzy-set coincidence analysis. *Polit. Sci. Res. Methods* **8**(3), 526–42 (2020a)
- Baumgartner, M., Ambühl, M.: cna: An R package for configurational causal inference and modeling. (2020b). <https://cran.r-project.org/web/packages/cna/vignettes/cna.pdf>
- Baumgartner, M., Thiem, A.: Identifying complex causal dependencies in configurational data with coincidence analysis. *R J.* **7**(1), 176–84 (2015)
- Baumgartner, M., Thiem, A.: Model ambiguities in configurational comparative research. *Soc. Methods Res.* **46**(4), 954–987 (2017)
- Baumgartner, M., Thiem, A.: Often trusted but never (properly) tested: evaluating qualitative comparative analysis. *Soc. Methods Res.* **49**(2), 279–311 (2020)
- Berg-Schlosser, D., De Meur, G.: Comparative research design: case and variable selection. In: Rihoux, B., Ragin, C.C. (eds). *Configurational comparative methods: Qualitative Comparative Analysis (QCA) and related techniques*, pp. 19–32. Sage Publications, London (2009)
- Berg-Schlosser, D., De Meur, G., Rihoux, B., Ragin, C.C.: Qualitative comparative analysis (QCA) as an approach. In: Rihoux, B., Ragin, C.C. (eds). *Configurational comparative methods: Qualitative Comparative Analysis (QCA) and related techniques*, pp. 1–32. Sage Publications, London (2009)
- Cooper, B., Glaesser, J.: Analysing necessity and sufficiency with qualitative comparative analysis: how do results vary as case weights change? *Qual. Quant.* **50**(1), 327–46 (2016a)
- Cooper, B., Glaesser, J.: Qualitative Comparative Analysis, Necessary Conditions, and Limited Diversity: Some Problematic Consequences of Schneider and Wagemann's Enhanced Standard Analysis. *Field Methods* **28**(3), 300–315 (2016b)
- Cronqvist, L., Berg-Schlosser, D.: Multi-value QCA (mvQCA). In: Rihoux, B., Ragin, C.C. (eds). *Configurational comparative methods: Qualitative Comparative Analysis (QCA) and related techniques*, pp. 69–86. Sage Publications, London (2009)
- Duša, A.: Consistency cubes: a fast, efficient method for exact Boolean minimization. *R J.* **10**(2), 357–70 (2019a)
- Duša, A.: Critical tension: sufficiency and parsimony in QCA. *Soc. Methods Res.* 1–25 (2019b). <https://doi.org/10.1177/0049124119882456>
- Duša, A.: QCA with R: A Comprehensive Resource. Springer International Publishing, Berlin (2019c)
- Duša, A., Thiem, A.: Enhancing the minimization of boolean and multivalued output functions with eQMC. *J. Math. Sociol.* **39**(2), 92–108 (2015)
- Furnari, S., Crilly, D., Misangyi, V.F., Greckhamer, T., Aguilera, R., Fiss, P.C.: Capturing causal complexity: a configurational theorizing process. *Manag. Rev. Acad* (2020). <https://doi.org/10.5465/amr.2019.0298>
- Graßhoff, G., May, M.: Causal regularities. In: Spohn, W., Ledwig, M., Esfeld, M. (eds.) *Current issues in causation*, pp. 85–114. Mentis Verlag, Münster (2001)
- Haesebrouck, T.: An alternative update of the two-step QCA procedure. *Qual. Quant.* **53**(6), 2765–80 (2019)
- Haesebrouck, T., Thomann, E.: Introduction: Causation, inferences, and solution types in configurational comparative methods. *Qual. Quant.* (2021)
- Hug, S.: Qualitative comparative analysis: how inductive use and measurement error lead to problematic inference. *Polit. Anal.* **21**(2), 252–65 (2013)
- Hug, S.: We need an open discussion of QCA's limitations: a comment on Thiem. *Qual. Multi-Method Res.* **2**, 24–27 (2014). <https://doi.org/10.5281/zenodo.894319>
- Hume, D.: (1748) *An Enquiry Concerning Human Understanding*. Oxford University Press, Oxford (1999)
- Krogslund, C., Choi, D.D., Poertner, M.: Fuzzy sets on shaky ground: parameter sensitivity and confirmation bias in fsQCA. *Polit. Anal.* **23**(1), 21–41 (2015)
- Mackie, J.L.: *The Cement of the Universe: A Study of Causation*. Clarendon Press, Oxford (1974)
- Mahoney, J.: Toward a unified theory of causality. *Comp. Polit. Stud.* **41**(4–5), 412–36 (2008)
- McCluskey, E.J.: Minimization of Boolean functions. *Bell Syst. Tech. J.* **35**(6), 1417–44 (1956)
- McCluskey, E.J.: *Introduction to the Theory of Switching Circuits*. Princeton University Press, Princeton (1965)
- Mello, P.A.: From prospect to practice: a critical review of applications in fuzzy-set qualitative comparative analysis. *SSRN Electron. J* (2013). https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2349513
- Misangyi, V.F., Greckhamer, T., Furnari, S., Fiss, P.C., Crilly, D., Aguilera, R.: Embracing causal complexity: the emergence of a neo-configurational perspective. *J. Manage.* **43**(1), 255–82 (2016)
- Oana, I.-E., Schneider, C.Q., Eva, T.: *Qualitative Comparative Analysis (QCA) Using R: A Beginner's Guide*. University Press, Cambridge (2021)

- Parkkinen, V.-P., Baumgartner, M.: Robustness and model selection in configurational causal modeling. *Soc. Methods Res.* (2021). <https://doi.org/10.1177/0049124120986200>
- Quine, W.V.: A way to simplify truth functions. *Am. Math. Mon.* **62**(9), 627–31 (1955)
- Ragin, C.C.: *The Comparative Method: Moving Beyond Qualitative and Quantitative Strategies*. University of California Press, Berkeley, Los Angeles, London (1987)
- Ragin, C.C.: *Fuzzy-Set Social Science*. University of Chicago Press, Chicago (2000)
- Ragin, C.C.: Set relations in social research: evaluating their consistency and coverage. *Polit. Anal.* **14**(3), 291–310 (2006)
- Ragin, C.C.: *Redesigning social inquiry: Fuzzy sets and beyond*. University of Chicago Press, Chicago (2008)
- Ragin, C.C.: Qualitative comparative analysis using fuzzy sets (fsQCA). In: Rihoux, B., Ragin, C.C. (eds.) *Configurational comparative methods: Qualitative Comparative Analysis (QCA) and related techniques*, pp. 87–121. Sage Publications, London (2009)
- Ragin, C.C.: Comment: Lucas and Szatrowski in critical perspective. *Soc. Methodol.* **44**(1), 80–158 (2014)
- Ragin, C.C., Sonnett, J.: Between complexity and parsimony: limited diversity, counterfactual cases, and comparative analysis." In *Vergleichen Der Polit.*, edited by Sabine Kropp and Michael Minkenberg, 180–97 (2005). Wiesbaden: VS Verlag für Sozialwissenschaften. <http://www.elsevier.com/locate/scp>
- Rihoux, B., Ragin, C.C. (eds.): *Configurational comparative methods: qualitative comparative analysis (QCA) and related techniques*, edited by Benoît Rihoux and Charles C. Ragin. Sage Publications, London (2009)
- Rohlfing, I.: Why simulations are appropriate for evaluating qualitative comparative analysis'. *Qual. Quant.* **50**(5), 2073–84 (2016)
- Schneider, C.Q.: Realists and idealists in QCA. *Polit. Anal.* **26**(2), 246–54 (2018)
- Schneider, C.Q.: Two-step QCA revisited: the necessity of context conditions. *Qual. Quant.* **53**(3), 1109–1126 (2019)
- Schneider, C.Q., Wagemann, C.: Reducing complexity in qualitative comparative analysis (QCA): remote and proximate factors and the consolidation of democracy. *Eur. J. Polit. Res.* **45**(5), 751–786 (2006)
- Schneider, C.Q., Wagemann, C.: Standards of good practice in qualitative comparative analysis (QCA) and fuzzy-sets. *Comp. Soc.* **9**(3), 397–418 (2010)
- Schneider, C.Q., Wagemann, C.: *Set-theoretic methods for the social sciences: a guide to Qualitative Comparative Analysis*. Cambridge University Press, Cambridge (2012)
- Schneider, C.Q., Wagemann, C.: Doing justice to logical remainders in QCA: moving beyond the standard analysis. *Polit. Res. Q.* **66**(1), 211 (2013)
- Thomann, E., Maggetti, M.: Designing research with qualitative comparative analysis (QCA): approaches, challenges, and tools. *Soc. Methods Res.* **49**(2), 356–386 (2020)
- Wagemann, C., Buche, J., Siewert, M.B.: QCA and business research: work in progress or a consolidated agenda? *J. Bus. Res.* **69**(7), 2531–2540 (2016)
- Whitaker, R.G., Sperber, N., Baumgartner, M., Thiem, A., Cragun, D., Damschroder, L., Miech, E., Slade, A., Birken, S.: Coincidence analysis: a new method for causal inference in implementation science. *Implement. Sci.* **15**(108), 1–10 (2020)