

QCA: A Package for Qualitative Comparative Analysis

by Alrik Thiem and Adrian Duşa

Abstract We present **QCA**, a package for performing Qualitative Comparative Analysis (QCA). QCA is becoming increasingly popular with social scientists, but none of the existing software alternatives covers the full range of core procedures. This gap is now filled by **QCA**. After a mapping of the method's diffusion, we introduce some of the package's main capabilities, including the calibration of crisp and fuzzy sets, the analysis of necessity relations, the construction of truth tables and the derivation of complex, parsimonious and intermediate solutions.

Introduction

Qualitative Comparative Analysis (QCA) - a research method popularized largely through the work of Charles Ragin (Ragin, 1987, 2000, 2008) - counts among the most influential recent innovations in social science methodology. In line with Ragin's own background, QCA has been initially employed only by a small number of (political) sociologists (e.g., Amenta et al., 1992; Griffin et al., 1991; Wickham-Crowley, 1991). Since then, however, the method has made inroads into political science and international relations (e.g., Thiem, 2011; Vis, 2009), business and economics (e.g., Evans and Aligica, 2008; Valliere et al., 2008), management and organization (e.g., Greckhamer, 2011), legal studies and criminology (Arvind and Stirton, 2010; Miethe and Drass, 1999), education (e.g., Glaesser and Cooper, 2011; Schneider and Sadowski, 2010), and health policy research (e.g., Harkreader and Imershein, 1999; Schensul et al., 2010). Figure 1 charts the trend in the total number of QCA applications that have appeared in peer-reviewed journal articles since 1984, broken down by its three variants *crisp-set QCA* (csQCA), *multi-value QCA* (mvQCA) and *fuzzy-set QCA* (fsQCA).¹

Allowing for a publication lag of about two years, four applications on average have been published throughout the first decade following the introduction of csQCA in Ragin (1987). But only his sequel "Fuzzy-Set Social Science" (Ragin, 2000) seems to have got the "Ragin Revolution" (Vaisey, 2009) eventually off ground. In the years from 2003 to 2007, the average number of applications had risen to 13 before the absolute number of applications almost tripled from 12 in 2007 to 35 in 2011. Despite the introduction of fsQCA, applications of csQCA have continued to increase from four in 2001 to 20 in 2011. In contrast to csQCA and fsQCA, mvQCA has remained underutilized. Of a total of 239 applications between 1984 and 2011, only seven have employed this variant. Even when accounting for the fact that it has been introduced in 2004, 17 years after csQCA and four years after fsQCA, this represents a disproportionately low number.²

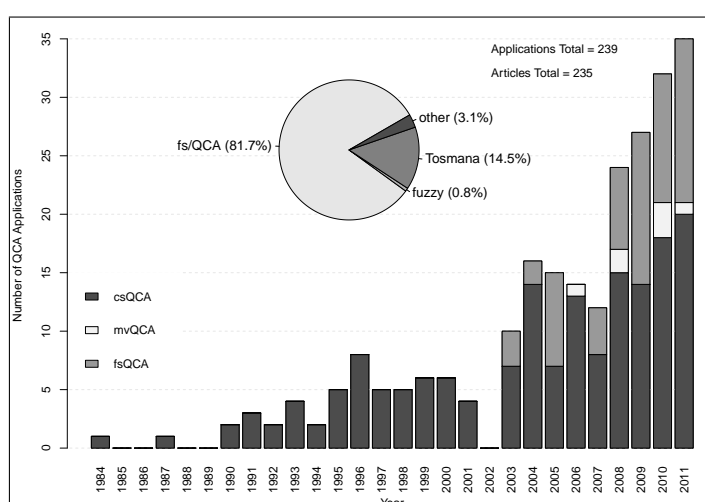


Figure 1: Trend in number of QCA applications by variants and year, and software market share.

¹The number of applications differs slightly from the number of articles as four articles have each presented two applications of QCA using two different variants. In order to be included in the data, applications had to focus primarily on a substantive research question, not QCA as a method. All entries have been recorded in the bibliography section on <http://www.compass.org>.

²See the debate in *Field Methods* for more details on mvQCA (Thiem, 2013; Vink and van Vliet, 2009).

QCA's methodological success story has created a growing demand for tailored software, which has been met almost exclusively by two programmes: Charles Ragin and Sean Davey's (2009) *fs/QCA* and Lasse Cronqvist's (2011) *Tosmana*. Until recently, however, users of non-Windows operating systems were limited as neither programme ran on other operating systems than Microsoft Windows. As of version 1.3.2.0, *Tosmana* has also supported other operating systems and in 2008, Kyle Longest and Stephen Vaisey's (2008) *fuzzy* package for STATA has been developed as another alternative. For the R environment, Adrian Duşa's *QCA* package has been first added in 2006 and in 2009, Ronggui Huang (2011) has released the *QCA3* package. The detailed market shares of these software solutions are shown in Figure 1.³ Holding a clear monopoly, *fs/QCA* is by far the most common software with over 80%, followed by *Tosmana* with 15% and *fuzzy* with 1%. Other solutions have claimed about 3%, but neither R package has managed to win any market shares thus far.

Not as unequal as their market shares, but significantly different still, are the capabilities of these software solutions. Table 1 provides an overview of the functionality each programme offers. All alternatives to *QCA* have different capabilities, but none covers the entire range of basic procedures. Neither *fs/QCA* nor *fuzzy* can handle multi-value set data, whereas *Tosmana* cannot process fuzzy sets. The analysis of necessity relations - an important step in the procedural protocol of QCA - is not foreseen in *Tosmana*, either, and the other packages offer only limited user-driven routines. Complex and parsimonious solutions can be found by all packages, but only *fs/QCA* generates intermediate solutions on an automated basis.

Table 1: Overview Software Functionality ^a

Function	Tosmana ^b	fs/QCA ^c	fuzzy ^d	QCA3 ^e	QCA ^f
<i>variant</i>					
csQCA	●	●	●	●	●
mvQCA	●	○	○	●	●
fsQCA	○	●	●	●	●
(tQCA)	○	●	○	●	●
<i>solution type</i>					
complex	●	●	●	●	●
intermediate	○	●	○	◐	●
parsimonious	●	●	●	●	●
<i>procedures</i>					
necessity tests	○	◐	◐	◐	●
parameters of fit	○	●	◐	◐	●
calibration	◐	◐	◐	◐	●
factorization	○	○	○	○	●
identify (C)SAs	●	○	○	●	●
statistical tests	○	○	●	◐	○

^a ○/ ◐/ ● = no/partial/full functionality

^b version 1.3.2.0; ^c version 2.5; ^d version st0140_2

^e version 0.0-5; ^f version 1.0-4

The calibration of crisp sets is limited in *fs/QCA* and *QCA3*. *Tosmana* cannot handle fuzzy sets, but it provides more elaborate tools for the calibration of crisp sets. In addition to Ragin's (2008) "direct method" and "indirect method", *fuzzy* offers a set-normalizing linear membership function. Most importantly, it also includes various statistical procedures for coding truth tables, the appropriacy of which largely depends on the research design.⁴

QCA combines and enhances the individual strengths of other software solutions. It can process all QCA variants (including temporal QCA (tQCA)), generate all solution types, and offers a wide range of procedures.

³Only 126 out of 234 articles cite the software program that was used. It is most likely that the vast majority of those which did not employed *fs/QCA* or any of its predecessors.

⁴As QCA has been initially developed for medium-*n* (≈ 10 -30 cases) macro-level data analysis, often with universes of objects rather than random samples, statistical tests are not only often meaningless but also rarely produce statistically significant configurations. With large-*n* micro-level data, such tests may be more appropriate.

For example, **QCA** provides four classes of functions for almost all kinds of calibration requirements and has an automated routine for the analysis of necessity relations. **QCA** is also the only package that can factorize any Boolean expression. As in Tosmana and **QCA3**, *simplifying assumptions* can also be identified, whereby those remainders that have been used in deriving a parsimonious or intermediate solution are often referred to for convenience. Unlike Tosmana, however, which does not present any parameters of fit, **QCA** produces inclusion, coverage and PRI (Proportional Reduction in Inconsistency) scores for both necessity and sufficiency relations in mvQCA.

In summary, no comprehensive QCA software solution has existed so far. Researchers have often been limited in their analyses when using one software, or had to resort to different programmes for performing all analytical operations they required. This gap is now filled by the **QCA** package, which seeks to provide a user-friendly yet powerful command-line interface alternative to the two dominant graphical user interface software solutions fs/QCA and Tosmana. In the remainder of this article, we introduce some of the package's most important functions, including the calibration of sets, the analysis of necessity relations, the construction of truth tables and the derivation of complex, parsimonious and intermediate solution types.

Calibration of sets

The process of translating *base variables* (also referred to as *raw data*) into condition or outcome set membership scores is called *calibration*, in fsQCA also *fuzzification*. In contrast to csQCA, continuous base variables need not be categorized directly in fsQCA but can be transformed with the help of continuous functions, a procedure called *assignment by transformation* (Verkuilen, 2005, 465). Ragin (2008), for example, suggests a piecewise-defined logistic function. Sufficient for the vast majority of fuzzification needs, **QCA** offers the `calibrate()` function, one of whose flexible function classes for positive end-point concepts is given in Equation (1).

$$\mu_S(b) = \begin{cases} 0 & \text{if } \tau_{\text{ex}} \geq b, \\ \frac{1}{2} \left(\frac{\tau_{\text{ex}} - b}{\tau_{\text{ex}} - \tau_{\text{cr}}} \right)^p & \text{if } \tau_{\text{ex}} < b \leq \tau_{\text{cr}}, \\ 1 - \frac{1}{2} \left(\frac{\tau_{\text{in}} - b}{\tau_{\text{in}} - \tau_{\text{cr}}} \right)^q & \text{if } \tau_{\text{cr}} < b \leq \tau_{\text{in}}, \\ 1 & \text{if } \tau_{\text{in}} < b. \end{cases} \quad (1)$$

Here, b is the base variable, τ_{ex} the threshold for full exclusion from set S , τ_{cr} the crossover threshold at the point of maximally ambiguous membership in S and τ_{in} the threshold for full inclusion in S . The parameters p and q control the degrees of concentration and dilation. The piecewise-defined logistic membership function suggested in Ragin (2008) is also available. Furthermore, `calibrate()` can generate set membership scores for sets based on negative or positive mid-point concepts (Thiem and Duşa, 2013, 55-62). If no suitable thresholds have been found even after all means of external and internal identification have been exhausted, **QCA**'s `findTh()` function can be employed for searching thresholds using hierarchical cluster analysis.

```
> library(QCA)
> # base variable and vector of thresholds
> b <- sort(rnorm(15))
> th <- quantile(b, c(0.1, 0.5, 0.9))

> # create bivalent crisp set
> calibrate(b, thresholds = th[2])
[1] 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1

> # create trivalent crisp set using thresholds derived from cluster analysis
> calibrate(b, thresholds = findTh(b, groups = 3))
[1] 0 0 0 1 1 1 1 1 1 1 1 2 2 2 2

> # fuzzification using Equation (1)
> round(calibrate(b, type = "fuzzy", thresholds = th), 2)
[1] 0.00 0.00 0.04 0.32 0.42 0.47 0.48 0.50 0.59 0.72 0.77 0.93 0.94 1.00 1.00

> # negation of previous result
> round(calibrate(b, type = "fuzzy", thresholds = rev(th)), 2)
[1] 1.00 1.00 0.96 0.68 0.58 0.53 0.52 0.50 0.41 0.28 0.23 0.07 0.06 0.00 0.00
```

```
> # fuzzification using piecewise logistic
> round(calibrate(b, type = "fuzzy", thresholds = th, logistic = TRUE), 2)
[1] 0.02 0.04 0.06 0.25 0.38 0.45 0.46 0.50 0.64 0.79 0.83 0.93 0.93 0.96 0.97

> # fuzzification using the ECDF of b
> round(calibrate(b, type = "fuzzy", thresholds = th, ecdf = TRUE), 2)
[1] 0.00 0.00 0.08 0.17 0.25 0.33 0.42 0.50 0.60 0.70 0.80 0.90 1.00 1.00 1.00
```

Analysis of necessity

Whenever the occurrence of an event B is accompanied by the occurrence of an event A , then B implies A and A is implied by B ($B \Rightarrow A = A \Leftarrow B$). Put differently, A is necessary for B and B is sufficient for A . Transposed to the set-theoretic terminology of QCA, analyses of necessity proceed from the observation of a *value* under the *outcome set* \mathbf{Y} - written $\mathbf{Y}\{v_l\}$ - to the observation of a value under the *condition set* \mathbf{X} - written $\mathbf{X}\{v_l\}$. For analyzing *necessity inclusion*, the decisive question is to which degree objects are members of $\mathbf{X}\{v_l\}$ and $\mathbf{Y}\{v_l\}$ in relation to their overall membership in $\mathbf{Y}\{v_l\}$.⁵ If necessity inclusion is high enough, the evidence is consistent with the hypothesis that $\mathbf{X}\{v_l\}$ is necessary for $\mathbf{Y}\{v_l\}$ ($\mathbf{X}\{v_l\} \supseteq \mathbf{Y}\{v_l\}$). The formula for necessity inclusion $\text{Incl}_N(\mathbf{X}\{v_l\})$ is presented in Equation (2).

$$\text{Incl}_N(\mathbf{X}\{v_l\}) = \frac{\sum_{i=1}^n \min(\{v_l\}x_i, \{v_l\}y_i)}{\sum_{i=1}^n \{v_l\}y_i} \quad (2)$$

Provided that $\mathbf{X}\{v_l\} \supseteq \mathbf{Y}\{v_l\}$ is sufficiently true, *necessity coverage* allows an assessment of the frequency with which B occurs relative to A . The formula for necessity coverage $\text{Cov}_N(\mathbf{X}\{v_l\})$ is given in Equation (3).

$$\text{Cov}_N(\mathbf{X}\{v_l\}) = \frac{\sum_{i=1}^n \min(\{v_l\}x_i, \{v_l\}y_i)}{\sum_{i=1}^n \{v_l\}x_i} \quad (3)$$

For analyzing necessity relations, QCA offers the `superSubset()` function. If p_j denotes the number of values of condition set j with $j = 1, 2, \dots, k$, the function returns necessity inclusion, PRI and coverage scores for those of the $d = \prod_{j=1}^k (p_j + 1) - 1$ combinations of condition set values that just meet the given inclusion and coverage cut-offs.⁶ Therefore, `superSubset()` does not require users to predefine the combinations to be tested, and so removes the risk of leaving potentially interesting results undiscovered. The initial set of combinations always consists of all $\prod_{j=1}^k p_j$ trivial intersections $\langle \mathbf{X}_1\{v_1\}, \mathbf{X}_1\{v_2\}, \dots, \mathbf{X}_1\{v_p\}, \dots, \mathbf{X}_k\{v_p\} \rangle$. The size of the intersection is incrementally increased from 1 to k until its inclusion score falls below the cut-off. If no trivial intersection passes the inclusion cut-off, `superSubset()` automatically switches to forming set unions until the least complex form has been found.

For demonstration purposes, we reanalyze the data from Krook's (2010) csQCA on women's representation in 22 Western democratic parliaments. Countries with electoral systems of proportional representation (ES), parliamentary quotas (QU), social democratic welfare systems (WS), autonomous women's movements (WM), more than 7% left party seats (LP) and more than 30% seats held by women (WNP) are coded "1", all others "0". The first five sets are the conditions to be tested for necessity in relation to the outcome set WNP. For reasons of simplicity and space, we use lower case letters for denoting set negation in all remaining code examples.

```
> data(Krook)
> Krook
  ES QU WS WM LP WNP
SE  1  1  1  0  0   1
FI  1  0  1  0  0   1
NO  1  1  1  1  1   1
DK  1  0  1  1  1   1
NL  1  1  0  1  1   1
ES  1  1  0  1  0   1
..  .  .  .  .  .
<rest omitted>
```

⁵We prefer the term "inclusion" over the more common term "consistency" as introduced by Ragin (2006) because the measure represents the degree to which one set is included by another, producing evidence that is either consistent with the underlying hypothesis of necessity, inconsistent or mixed. The measure is also called Inclusion-1 index (cf. Smithson and Verkuilen, 2006, 65).

⁶The inclusion cut-off always enjoys priority over the coverage cut-off.

```
> superSubset(Krook, outcome = "WNP", cov.cut = 0.52)
```

		incl	PRI	cov.r
1	ES+LP	1.000	1.000	0.733
2	ES+WM	1.000	1.000	0.524
3	WS+WM+LP	1.000	1.000	0.611
4	QU+wm+LP	1.000	1.000	0.550
5	QU+WM+lp	1.000	1.000	0.524
6	QU+WS+LP	1.000	1.000	0.550
7	QU+WS+WM	1.000	1.000	0.524
8	es+QU+WS	1.000	1.000	0.524

When not specified otherwise, all sets in the data but the outcome are assumed to be conditions. By default, the function tests for necessity, but sufficiency relations can also be analyzed. No trivial intersection has passed the inclusion cut-off and `superSubset()` has thus formed unions of conditions. Substantively, the first combination **ES + LP** means that having proportional representation or strong left party representation is necessary for having more than 30% parliamentary seats held by women.

Analysis of sufficiency, step 1: Truth tables

Whenever the occurrence of an event A is accompanied by the occurrence of an event B , then A implies B and B is implied by A ($A \Rightarrow B = B \Leftarrow A$). Put differently, A is sufficient for B and B is necessary for A . Transposed to the set-theoretic terminology of QCA, analyses of sufficiency proceed from the observation of a value under \mathbf{X} to the observation of a value under \mathbf{Y} . For analyzing *sufficiency inclusion*, the decisive question is to which degree objects are members of $\mathbf{X}\{v_l\}$ and $\mathbf{Y}\{v_l\}$ in relation to their overall membership in $\mathbf{X}\{v_l\}$. If sufficiency inclusion is high enough, the evidence is consistent with the hypothesis that $\mathbf{X}\{v_l\}$ is sufficient for $\mathbf{Y}\{v_l\}$ ($\mathbf{X}\{v_l\} \subseteq \mathbf{Y}\{v_l\}$). The formula for sufficiency inclusion $\text{Incl}_S(\mathbf{X}\{v_l\})$ is presented in Equation (4).

$$\text{Incl}_S(\mathbf{X}\{v_l\}) = \frac{\sum_{i=1}^n \min(\{v_l\}x_i, \{v_l\}y_i)}{\sum_{i=1}^n \{v_l\}x_i} \quad (4)$$

The classical device for analyzing sufficiency relations is the *truth table*, which lists all $d = \prod_{j=1}^k p_j$ configurations and their corresponding outcome value.⁷ Configurations represent exhaustive combinations of set values characterizing the objects. For illustration, a simple hypothetical truth table with three bivalent condition sets \mathbf{X}_1 , \mathbf{X}_2 and \mathbf{X}_3 and the outcome value **OUT** is presented in Table 2. Three bivalent conditions yield the eight configurations listed under \mathcal{C}_i . The minimum number of cases n that is usually required for the respective outcome value is also appended.

Table 2: Hypothetical Truth Table

\mathcal{C}_i	\mathbf{X}_1	\mathbf{X}_2	\mathbf{X}_3	OUT	n
1	1	1	1	1	≥ 1
2	1	1	0	1	≥ 1
3	1	0	1	1	≥ 1
4	1	0	0	1	≥ 1
5	0	1	1	0	≥ 1
6	0	1	0	C	≥ 2
7	0	0	1	?	0
8	0	0	0	?	0

It is important to emphasize that the outcome value is not the same as the outcome set, the latter of which does not show up in QCA truth table. Instead, the *outcome value* is based on the sufficiency inclusion score, returning a truth value that indicates the degree to which the evidence is consistent with the hypothesis that a sufficiency relation between a configuration and the outcome set exists. Configurations \mathcal{C}_1 to \mathcal{C}_4 are *positive*

⁷This table is also often called “table of combinations” (McCluskey, 1965) or “function table” (Hohn, 1966).

because they support this hypothesis ($OUT = 1$), C_5 is *negative* because it does not ($OUT = 0$). Mixed evidence exists for C_6 ($OUT = C$). If at least two objects conform to one configuration, but the evidence neither sufficiently confirms nor falsifies the hypothesis of a subset relation between this configuration and the outcome set, *contradictions* arise. No empirical evidence at all exists for C_7 and C_8 . If a configuration has no or too few cases, it is called a *logical remainder* ($OUT = ?$).

The `truthTable()` function can generate truth tables for all three main QCA variants without users having to specify which variant they use. The structure of the data is automatically transposed into the correct format.

```
> KrookTT <- truthTable(Krook, outcome = "WNP")
> KrookTT
```

```
OUT: outcome value
n: number of cases in configuration
incl: sufficiency inclusion score
PRI: proportional reduction in inconsistency
```

	ES	QU	WS	WM	LP	OUT	n	incl	PRI
3	0	0	0	1	0	0	2	0.000	0.000
4	0	0	0	1	1	1	1	1.000	1.000
9	0	1	0	0	0	0	1	0.000	0.000
11	0	1	0	1	0	0	4	0.000	0.000
12	0	1	0	1	1	1	1	1.000	1.000
18	1	0	0	0	1	0	1	0.000	0.000
21	1	0	1	0	0	1	1	1.000	1.000
24	1	0	1	1	1	1	1	1.000	1.000
25	1	1	0	0	0	0	3	0.000	0.000
26	1	1	0	0	1	1	1	1.000	1.000
27	1	1	0	1	0	1	1	1.000	1.000
28	1	1	0	1	1	1	2	1.000	1.000
29	1	1	1	0	0	1	1	1.000	1.000
32	1	1	1	1	1	1	2	1.000	1.000

At a minimum, `truthTable()` requires an appropriate dataset and the `outcome` argument, which identifies the outcome set. If `conditions` is not provided as an argument, it is assumed that all other sets in the data but the outcome are the conditions. By default, logical remainders are not printed unless specified otherwise by the logical argument `complete`. The logical argument `show.cases` prints the names of the objects next to the configuration in which they have membership above 0.5.

The `truthTable()` function includes three cut-off arguments that influence how `OUT` is coded: `n.cut`, `incl.cut1` and `incl.cut0`. The first argument `n.cut` sets the minimum number of cases with membership above 0.5 needed in order to not code a configuration as a logical remainder. The second argument `incl.cut1` specifies the minimal sufficiency inclusion score for a non-remainder configuration to be coded as positive. The third argument `incl.cut0` offers the possibility of coding configurations as contradictions when their inclusion score is neither high enough to consider them as positive nor low enough to code them as negative. If the inclusion score of a non-remainder configuration falls below `incl.cut0`, this configuration is always considered negative. By means of the `sort.by` argument, the truth table can also be ordered along inclusion scores, numbers of cases or both, in increasing or decreasing order. If the original condition set labels are rather long, the logical letters argument can be used to replace the set labels with upper case letters in alphabetical order.

The leftmost column list the configuration row index values from the complete truth table. Sufficiency inclusion and PRI scores are also provided in the two rightmost columns. Once the truth table is fully coded, it can be minimized according to the theorems of Boolean algebra (McCluskey, 1965, 84-89).

Analysis of sufficiency, step 2: Boolean minimization

The *canonical union* resulting from the truth table presented in Table 2 is given by Equation (5). It consists of four *fundamental intersections* (FI), each of which corresponds to one positive configuration. Generally, all FIs also represent positive configurations, but not all positive configurations become FIs. The analyst may decide to exclude some of these configurations from the minimization process on theoretical or empirical grounds.

$$\overbrace{X_1 \cap X_2 \cap X_3}^{C_1} \cup \overbrace{X_1 \cap X_2 \cap x_3}^{C_2} \cup \overbrace{X_1 \cap x_2 \cap X_3}^{C_3} \cup \overbrace{X_1 \cap x_2 \cap x_3}^{C_4} \subseteq Y \quad (5)$$

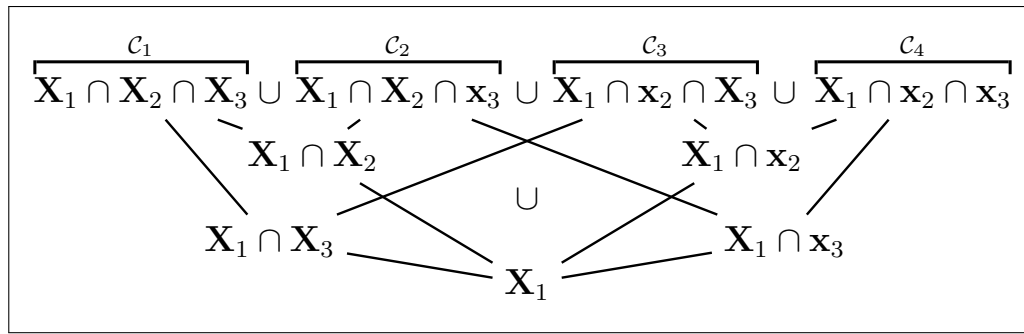


Figure 2: Boolean minimization of Equation (5).

If two FIs differ on the values of one condition only, then this condition can be eliminated so that a simpler term results. For example, Equation (5) can be reduced in two passes as shown in Figure 2. In the first pass, the four FIs can be reduced to four simpler terms. In the second pass, these four terms can then be reduced at once to a single term. No further reduction is possible, X_1 is the only term which is essential with respect to the outcome ($X_1 \subseteq Y$). All terms that survive the Boolean minimization process are called *prime implicants* (PI).

The central function of the **QCA** package that performs the minimization is `eqmcc()` (enhanced Quine-McCluskey) (Duşa, 2007, 2010). It can derive *complex*, *parsimonious* and *intermediate solutions* from a truth table object or a suitable dataset. In contrast to complex solutions, parsimonious solutions incorporate logical remainders into the minimization process without any prior assessment by the analyst as to whether a sufficiency relation is plausible or not. Intermediate solutions offer a middle way insofar as those logical remainders that have been used in the derivation of the parsimonious solution are filtered according to the analyst's *directional expectations* about the impact of each single condition set value on the overall sufficiency relation of the configuration of which it is part and the outcome set. By formulating such expectations, *difficult logical remainders* are excluded as FIs from the canonical union, whereas those logical remainders that enter the canonical union are *easy*. The complex solution, which is the default option, can be generated by `eqmcc()` with minimal typing effort.

```
> KrookSC <- eqmcc(KrookTT, details = TRUE)
> KrookSC

n OUT = 1/0/C: 11/11/0
Total      : 22

S1: ES*QU*ws*LP + ES*QU*ws*WM + es*ws*WM*LP + ES*WS*wm*lp + ES*WS*WM*LP

      incl  PRI  cov.r  cov.u
-----
ES*QU*ws*LP 1.000 1.000 0.273 0.091
ES*QU*ws*WM 1.000 1.000 0.273 0.091
es*ws*WM*LP 1.000 1.000 0.182 0.182
ES*WS*wm*lp 1.000 1.000 0.182 0.182
ES*WS*WM*LP 1.000 1.000 0.273 0.273
-----
S1          1.000 1.000 1.000
```

The truth table object `KrookTT` that was generated above is passed to `eqmcc()`. No further information is necessary in order to arrive at the complex solution. The logical argument `details` causes all parameters of fit to be printed together with the *minimal union* `S1`: inclusion (`incl`), PRI (`PRI`), raw coverage (`cov.r`) and unique coverage (`cov.u`) scores for each PI as well as the minimal union.⁸ If `details = TRUE`, the logical argument `show.cases` also prints the names of the objects that are covered by each PI.

If alternative minimal unions exist, all of them are printed if the row dominance principle for PIs is not applied as specified in the logical argument `rowdom`. One PI \mathcal{P}_1 dominates another \mathcal{P}_2 if all FIs covered by \mathcal{P}_2 are also covered by \mathcal{P}_1 and both are not interchangeable (cf. McCluskey, 1965, 150). Inessential PIs are listed in brackets in the solution output and at the end of the PI part in the parameters-of-fit table, together with

⁸Unique coverage scores do not apply to minimal unions.

their unique coverage scores under each individual minimal union. For example, the parsimonious solution without row dominance applied can be derived by making all logical remainders available for inclusion in the canonical union as FIs and by setting `rowdom` to `FALSE`.

```
> KrookSP <- eqmcc(KrookTT, include = "?", rowdom = FALSE, details = TRUE)
> KrookSP
```

```
n OUT = 1/0/C: 11/11/0
  Total      : 22
```

```
S1: WS + ES*WM + QU*LP + (es*LP)
S2: WS + ES*WM + QU*LP + (WM*LP)
```

	incl	PRI	cov.r	cov.u	(S1)	(S2)
WS	1.000	1.000	0.455	0.182	0.182	0.182
ES*WM	1.000	1.000	0.545	0.091	0.091	0.091
QU*LP	1.000	1.000	0.545	0.091	0.091	0.091
es*LP	1.000	1.000	0.182	0.000	0.091	
WM*LP	1.000	1.000	0.636	0.000		0.091
S1	1.000	1.000	1.000			
S2	1.000	1.000	1.000			

The intermediate solution for bivalent set data requires a vector of directional expectations in the `direxp` argument, where “0” denotes absence, “1” presence and “-1” neither. The intermediate solution with all conditions expected to contribute to a positive outcome value when present is generated as follows:

```
> KrookSI <- eqmcc(KrookTT, include = "?", direxp = c(1,1,1,1,1), details = TRUE)
> KrookSI
```

```
n OUT = 1/0/C: 11/11/0
  Total      : 22
```

```
p.sol: WS + ES*WM + QU*LP + WM*LP
```

```
S1: ES*WS + WM*LP + ES*QU*LP + ES*QU*WM
```

	incl	PRI	cov.r	cov.u
ES*WS	1.000	1.000	0.455	0.182
WM*LP	1.000	1.000	0.636	0.182
ES*QU*LP	1.000	1.000	0.455	0.091
ES*QU*WM	1.000	1.000	0.455	0.091
S1	1.000	1.000	1.000	

For intermediate solutions, `eqmcc()` also prints the parsimonious solution (`p.sol`) whose simplifying assumptions have been used in filtering logical remainders. The PI chart of this intermediate solution (`i.sol`) that has been derived from the (first and only) complex and the (first and only) parsimonious solution (`C1P1`) can then be inspected by accessing the corresponding component in the returned object.

```
> KrookSI$PIchart$i.sol$C1P1
```

	4	12	21	24	26	27	28	29	32
ES*WS	-	-	x	x	-	-	-	x	x
WM*LP	x	x	-	x	-	-	x	-	x
ES*QU*LP	-	-	-	-	x	-	x	-	x
ES*QU*WM	-	-	-	-	-	x	x	-	x

If several minimal sums exist under both the parsimonious and complex solution, the PI chart of the respective combination for the intermediate solution can be accessed by replacing the numbers in the `C1P1` component.

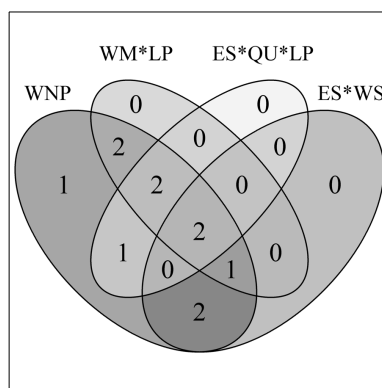


Figure 3: Venn diagram of three PIs from intermediate solution produced with **VennDiagram** package.

Besides the PI chart, the solution object returned by `eqmcc()` also contains a dataframe of PI set membership scores in the `pims` component. These scores can then be used to draw Venn diagrams of solutions, similar to the one shown in Figure 3, using suitable R packages such as **VennDiagram** (Chen and Boutros, 2011).

```
> KrookSI$pims$i.sol$C1P1
  ES*WS WM*LP ES*QU*LP ES*QU*WM
SE     1     0         0         0
FI     1     0         0         0
NO     1     1         1         1
DK     1     1         0         0
NL     0     1         1         1
ES     0     0         0         1
..     .     .         .         .
<rest omitted>
```

Summary

In recent years, Qualitative Comparative Analysis (QCA) has become the method of choice for testing configurational hypotheses. However, QCA is still very much a “method in the making”. Extensions, enhancements and alternative algorithms appear on a regular basis (Baumgartner, 2009; Eliason and Stryker, 2009; Schneider and Wagemann, 2012; Thiem, 2012). R provides an ideal environment within which established QCA procedures as well as more advanced techniques can be implemented in a manner as transparent and user-responsive as possible. The **QCA** package makes a significant contribution in this regard. It fills the individual gaps in other programmes’ coverage of basic functionality, and provides further improvements through complementary and advanced procedures.

The QCA software market remains dominated by the two graphical user interface programmes *fs/QCA* and *Tosmana*. **QCA** seeks to bridge the method of QCA with powerful command-line software while retaining a user-friendly code and command structure. In order to lower the barriers for social scientists to choosing R for QCA further, we have published an introductory textbook with extended replications of recent QCA studies from various research areas (Thiem and Duşa, 2013). Although most examples have been taken from political science, the book may also be of interest to researchers from related disciplines. At the same time, this textbook also serves as a comprehensive reference manual for the **QCA** package.

Bibliography

- E. Amenta, B. G. Carruthers, and Y. Zylan. A hero for the aged: The Townsend movement, the political mediation model, and United-States old-age policy, 1934-1950. *American Journal of Sociology*, 98(2):308–339, 1992. [p1]
- T. T. Arvind and L. Stirton. Explaining the reception of the Code Napoleon in Germany: A fuzzy-set qualitative comparative analysis. *Legal Studies*, 30(1):1–29, 2010. [p1]
- M. Baumgartner. Inferring causal complexity. *Sociological Methods & Research*, 38(1):71–101, 2009. [p9]

- H. Chen and P. Boutros. VennDiagram: A package for the generation of highly-customizable Venn and Euler diagrams in R. *BMC Bioinformatics*, 12(1):35–41, 2011. [p9]
- L. Cronqvist. *Tosmana - tool for small-n analysis, version 1.3.2*. University of Trier, Trier, 2011. [p2]
- A. Duşa. *Enhancing Quine-McCluskey*. WP 2007-49, COMPASSS, 2007. URL <http://www.compassss.org/files/WPfiles/Dusa2007a.pdf>. [p7]
- A. Duşa. A mathematical approach to the Boolean minimization problem. *Quality & Quantity*, 44(1):99–113, 2010. [p7]
- S. R. Eliason and R. Stryker. Goodness-of-fit tests and descriptive measures in fuzzy-set analysis. *Sociological Methods & Research*, 38(1):102–146, 2009. [p9]
- A. J. Evans and P. D. Aligica. The spread of the flat tax in Eastern Europe. *Eastern European Economics*, 46(3): 49–67, 2008. [p1]
- J. Glaesser and B. Cooper. Selectivity and flexibility in the German secondary school system: A configurational analysis of recent data from the German socio-economic panel. *European Sociological Review*, 27(5):570–585, 2011. [p1]
- T. Greckhamer. Cross-cultural differences in compensation level and inequality across occupations: A set-theoretic analysis. *Organization Studies*, 32(1):85–115, 2011. [p1]
- L. J. Griffin, C. Botsko, A.-M. Wahl, and L. W. Isaac. Theoretical generality, case particularity: Qualitative comparative analysis of trade-union growth and decline. *International Journal of Comparative Sociology*, 32 (1-2):110–136, 1991. [p1]
- S. Harkreader and A. W. Imershein. The conditions for state action in Florida’s health-care market. *Journal of Health and Social Behavior*, 40(2):159–174, 1999. [p1]
- F. E. Hohn. *Applied Boolean algebra: An elementary introduction*. Macmillan, New York, 2nd edition, 1966. [p5]
- R. Huang. *QCA3: Yet another package for qualitative comparative analysis*, 2011. R package version 0.0-4. [p2]
- M. L. Krook. Women’s representation in parliament: A qualitative comparative analysis. *Political Studies*, 58 (5):886–908, 2010. [p4]
- K. C. Longest and S. Vaisey. fuzzy: A program for performing qualitative comparative analyses (QCA) in Stata. *Stata Journal*, 8(1):79–104, 2008. [p2]
- E. J. McCluskey. *Introduction to the theory of switching circuits*. Princeton University Press, Princeton, 1965. [p5, 6, 7]
- T. D. Miethe and K. A. Drass. Exploring the social context of instrumental and expressive homicides: An application of qualitative comparative analysis. *Journal of Quantitative Criminology*, 15(1):1–21, 1999. [p1]
- C. C. Ragin. *The Comparative method: Moving beyond qualitative and quantitative strategies*. University of California Press, Berkeley, 1987. [p1]
- C. C. Ragin. *Fuzzy-set social science*. University of Chicago Press, Chicago, 2000. [p1]
- C. C. Ragin. Set relations in social research: Evaluating their consistency and coverage. *Political Analysis*, 14(3): 291–310, 2006. [p4]
- C. C. Ragin. *Redesigning social inquiry: Fuzzy sets and beyond*. University of Chicago Press, Chicago, 2008. [p1, 2, 3]
- C. C. Ragin and S. Davey. *Fuzzy-set/Qualitative Comparative Analysis fs/QCA 2.5*. Department of Sociology, University of Arizona, Tucson, 2009. [p2]
- J. Schensul, D. Chandran, S. Singh, M. Berg, S. Singh, and K. Gupta. The use of qualitative comparative analysis for critical event research in alcohol and HIV in Mumbai, India. *AIDS and Behavior*, 14(S1):113–125, 2010. [p1]
- C. Q. Schneider and C. Wagemann. *Set-Theoretic Methods for the Social Sciences: A Guide to Qualitative Comparative Analysis (QCA)*. Cambridge University Press, Cambridge, 2012. [p9]

- P. Schneider and D. Sadowski. The impact of new public management instruments on PhD education. *Higher Education*, 59(5):543–565, 2010. [p1]
- M. Smithson and J. Verkuilen. *Fuzzy set theory: Applications in the social sciences*. SAGE, London, 2006. [p4]
- A. Thiem. Conditions of intergovernmental armaments cooperation in Western Europe, 1996-2006. *European Political Science Review*, 3(1):1–33, 2011. [p1]
- A. Thiem. *Unifying Configurational Comparative Methodology: Generalized-Set Qualitative Comparative Analysis*. Working Paper Series, 2012-34, IPSA Committee on Concepts & Methods, 2012. URL <http://www.concepts-methods.org>. [p9]
- A. Thiem. Clearly crisp, and not fuzzy: A comment on Vink and van Vliet (2009). *Field Methods*, 25(2):197–207, 2013. [p1]
- A. Thiem and A. Duşa. *Qualitative comparative analysis with R: A user's guide*. Springer, New York, 2013. [p3, 9]
- S. Vaisey. QCA 3.0: The “Ragin revolution” continues. *Contemporary Sociology: A Journal of Reviews*, 38(4): 308–312, 2009. [p1]
- D. Valliere, N. Na, and S. Wise. Prior relationships and M&A exit valuations: A set-theoretic approach. *Journal of Private Equity*, 11(2):60–72, 2008. [p1]
- J. Verkuilen. Assigning membership in a fuzzy set analysis. *Sociological Methods & Research*, 33(4):462–496, 2005. [p3]
- M. P. Vink and O. van Vliet. Not quite crisp, not yet fuzzy? Assessing the potentials and pitfalls of multi-value QCA. *Field Methods*, 21(3):265–289, 2009. [p1]
- B. Vis. Governments and unpopular social policy reform: Biting the bullet or steering clear? *European Journal of Political Research*, 48(1):31–57, 2009. [p1]
- T. P. Wickham-Crowley. A qualitative comparative approach to Latin American revolutions. *International Journal of Comparative Sociology*, 32(1-2):82–109, 1991. [p1]

Alrik Thiem

Department of Humanities, Social and Political Sciences

Swiss Federal Institute of Technology Zurich (ETH)

IFW C29.2, Haldeneggsteig 4

8092 Zurich

Switzerland

thiem@sipo.gess.ethz.ch

Adrian Duşa

Department of Sociology

University of Bucharest

1 Schitu Măgureanu

050025 Bucharest

Romania

dusadrian@unibuc.ro