

Article

Cognitively Diagnostic Analysis Using the G-DINA Model in R

Qingzhou Shi ^{1,*}, Wenchao Ma ¹, Alexander Robitzsch ^{2,3}, Miguel A. Sorrel ⁴ and Kaiwen Man ¹

¹ Department of Educational Studies in Psychology, Research Methodology, and Counseling, University of Alabama, Tuscaloosa, AL 35487, USA; wenchao.ma@ua.edu (W.M.); kman@ua.edu (K.M.)

² IPN—Leibniz Institute for Science and Mathematics Education, Olshausenstraße 62, 24118 Kiel, Germany; robitzsch@leibniz-ipn.de

³ Centre for International Student Assessment (ZIB), Olshausenstraße 62, 24118 Kiel, Germany

⁴ Department of Social Psychology and Methodology, Autonomous University of Madrid, 28049 Madrid, Spain; miguel.sorrel@uam.es

* Correspondence: qshi7@crimson.ua.edu

Abstract: Cognitive diagnosis models (CDMs) have increasingly been applied in education and other fields. This article provides an overview of a widely used CDM, namely, the G-DINA model, and demonstrates a hands-on example of using multiple R packages for a series of CDM analyses. This overview involves a step-by-step illustration and explanation of performing Q-matrix evaluation, CDM calibration, model fit evaluation, item diagnosticity investigation, classification reliability examination, and the result presentation and visualization. Some limitations of conducting CDM analysis in R are also discussed.

Keywords: cognitive diagnosis models; G-DINA; GDINA R package; CDM R package



Citation: Shi, Q.; Ma, W.; Robitzsch, A.; Sorrel, M.A.; Man, K. Cognitively Diagnostic Analysis Using the G-DINA Model in R. *Psych* **2021**, *3*, 812–835. <https://doi.org/10.3390/psych3040052>

Academic Editors: Gongjun Xu and André Kretzschmar

Received: 16 September 2021

Accepted: 4 December 2021

Published: 8 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Cognitive diagnosis models (CDMs), or diagnostic classification models (DCMs), are psychometric models for classifying individuals into latent classes with unique profiles of attributes. CDMs have increasingly attracted attention in education as they have shown the potential to identify students' strengths and weaknesses and thus aid classroom instruction and learning. In addition to the applications in education [1–4], CDMs have also been applied to other areas recently, such as industrial and organizational psychology [5] and psychiatry [6,7].

Despite the usefulness of CDMs in many fields, software programs for CDM analysis are still lacking. Programs such as Mplus [8,9], JAGS [10,11], and Stan [12,13] have been used for CDM analysis, but they are not without limitations. For example, CDM estimation using these programs often requires advanced coding skills, which may pose a formidable obstacle for CDMs practical application. Also, these general programs typically lack many essential functions, such as those for refining Q-matrix and assessing classification reliability. Recently, several R packages have been developed particularly for CDM analysis. Notably, George et al. [14] introduced the CDM package, and Ma and de la Torre [15] presented the GDINA package and how to apply it for a series of CDM analyses. However, different R packages have different functionality and features, and it remains unclear how these packages can be used in an integrated way for complete CDM analysis. This paper aims to fill this gap by illustrating a comprehensive CDM analysis with a particular emphasis on the use of multiple R packages under a widely used general CDM—the generalized deterministic input, noisy “and” gate (G-DINA) model [16] with a real dataset. As the first tutorial intended to introduce the state-of-the-art techniques for CDM analyses in the environment of R via multiple R packages, this paper will help researchers gain better insight into these packages and conduct CDM analyses in a more principled way.

2. The G-DINA Model

CDMs are latent variable models, where the latent variables may represent skills, abilities, misconceptions, or problem-solving strategies and are referred to as attributes. Attributes are often assumed to have only two statuses, mastery and nonmastery. To conduct a CDM analysis, the item response data and Q-matrix are required. Suppose a test measures K attributes and consists of J items. The $J \times K$ Q-matrix specifies the association between test items and the attributes measured, with q_{jk} being the element on the j^{th} row and k^{th} column. If the k^{th} attribute is assessed by the j^{th} item, the element q_{jk} equals to 1; otherwise, q_{jk} equals to 0. Let the response of examinee i to item j be denoted by Y_{ij} . For each examinee i , there is an attribute profile $a_i = \{a_{i1}, \dots, a_{ik}, \dots, a_{iK}\}$ containing K attributes to be inferred. In addition to the item responses and Q-matrix, one must specify the CDM to be used. A CDM consists of the measurement model and the structural model. The former establishes the relationship between the item response and attributes, and the latter specifies the relationship among attributes. To specify a measurement model, one needs to consider the nature of the response data (i.e., binary, ordinal, or nominal), the complexity, and the assumptions of different models. Many measurement models have been discussed in the literature [17]. DINA is a measurement model developed on specific assumptions regarding how attributes affect item responses and thus is often referred to as specific or reduced CDM. In contrast, several other models are referred to as general or saturated models because they have complex parametrizations and subsume many specific models. Examples of general CDMs include the generalized DINA model (G-DINA) [16], the log-linear CDM [18], and the general diagnostic model [19].

Although a simpler model is often preferred if its use could be justified, a saturated model, such as the G-DINA model, may be used to avoid potential model misspecifications. The item response function (IRF) of the G-DINA model [16] is expressed by

$$g\left[P\left(Y_{ij} = 1 \mid a_{ij}^*\right)\right] = \delta_{j0} + \sum_{k=1}^{K_j^*} \delta_{jk} a_{1k} + \sum_{k'=k+1}^{K_j^*} \sum_{k=1}^{K_j^*-1} \delta_{jkk'} a_{1k} a_{1kk'} \dots + \delta_{j12\dots K_j^*} \prod_{k=1}^{K_j^*} a_{1k}, \quad (1)$$

where $g[\cdot]$ represents an identity, logit, or log link function, δ_{j0} is the intercept of item j , δ_{jk} is the main effect of attribute k , $\delta_{jkk'}$ is the two-way interaction effect of attributes k and k' , and $\delta_{j12\dots K_j^*}$ is the K_j^* -way interaction effect of attributes 1 to K_j^* .

The G-DINA model is an unrestricted, saturated model that can be reduced to many other restricted models by imposing appropriate constraints. In particular, to obtain the deterministic-input, noisy-and-gate (DINA) model [20–22], all terms in Equation (1) except δ_{j0} and $\delta_{j12\dots K_j^*}$ are constrained to be 0. In this way, the IRF of the DINA model is expressed by

$$P\left(Y_{ij} = 1 \mid a_{ij}^*\right) = \delta_{j0} + \delta_{j12\dots K_j^*} \prod_{k=1}^{K_j^*} a_{1k}. \quad (2)$$

To obtain the deterministic input, noisy-or-gate (DINO) model [6], only the intercept and the main effect of attribute k are kept in the link function of Equation (1), and the IRF of DINO is expressed by

$$P\left(Y_{ij} = 1 \mid a_{ij}^*\right) = \delta_{j0} + \delta_{jk} a_{1k}, \quad (3)$$

where $\delta_{jk} = -\delta_{jkk'} = \dots = (-1)^{K_j^*+1} \delta_{j12\dots K_j^*}$, $k = 1, \dots, K_j^*$, $k' = 1, \dots, K_j^* - 1$, and $k'' > k', \dots, K_j^*$ [16]. In this regard, the number of item parameters for both DINA and DINO models was reduced to just two, regardless of the number of attributes measured.

To obtain the additive CDM (A-CDM) [16], only the intercept and the main effects in the identity link function of Equation (1) are kept. In this way, the IRF of A-CDM can be expressed by

$$P(Y_{ij} = 1 | a_{ij}^*) = \delta_{j0} + \sum_{k=1}^{K_j^*} \delta_{j0} a_{1k}. \quad (4)$$

The linear logistic model (LLM) [23] is the logit link function of A-CDM, and its IRF can be expressed by

$$\text{logit} \left[P(Y_{ij} = 1 | a_{ij}^*) \right] = \delta_{j0} + \sum_{k=1}^{K_j^*} \delta_{j0} a_{1k}, \quad (5)$$

and the reduced reparameterized unified model (R-RUM) [24] is the log-link function of A-CDM, and its IRF is given by

$$\log \left[P(Y_{ij} = 1 | a_{ij}^*) \right] = \delta_{j0} + \sum_{k=1}^{K_j^*} \delta_{j0} a_{1k}. \quad (6)$$

The reduced models presented here can be understood as particular cases of the G-DINA model that accommodate conjunctive or noncompensatory processes (DINA; mastery of all attributes is necessary to have a high probability of success), disjunctive processes (DINO; mastery of one attribute can compensate for the lack of the rest), or additive processes (A-CDM, LLM, and R-RUM; each attribute implies an independent increase in a function of the probability of success).

3. Overview of the CDM Analyses

This section will discuss the steps involved in cognitive diagnosis modeling using the G-DINA model, as shown in Figure 1. The development of diagnostic tests and specifications of Q-matrices will not be discussed here; detailed discussions of those can be found in Leighton and Gierl [25], Nichols et al. [26], and Tjoe and de la Torre [19], to name a few.

When the Q-matrix may not be entirely correct, the first step of CDM analysis should be the empirical Q-matrix evaluation, which involves validating the number of attributes and detecting the misidentified elements. To validate the number of attributes, Nájera and colleagues [27] adopted procedures for assessing the dimensionality, which were initially developed for exploratory analysis, often without a provisional Q-matrix. When the number of attributes has been validated, a host of methods have been developed for identifying misspecified elements [28–31]. De la Torre and Minchen [32] recommended employing a saturated CDM when conducting Q-matrix validation to avoid conflating Q-matrix misspecifications with model misspecifications. Also, although statistical procedures could provide some valuable insights into the Q-matrix, the appropriateness of the recommendations of these procedures should be carefully assessed by domain experts. In other words, the Q-matrix validation procedures should be used as a tool to facilitate domain experts developing the Q-matrix.

The second step of CDM analysis often involves model specification. The goal is to determine the measurement model—the model estimating the association between attributes and the observed data—for each item and specify the structural model—the model estimating the association among attributes—for joint attribute distribution. The measurement model can be specified on a priori grounds or determined by statistical procedures. For example, the Wald test and likelihood ratio test have been used to select the measurement model for each item [16,33–35]. Regularized CDMs have also been used to determine each item's most appropriate measurement model [36,37]. It should be noted that monotonicity constraints may need to be imposed because they are often theoretically reasonable and can stabilize the parameter estimation, especially when the sample size is small [38].

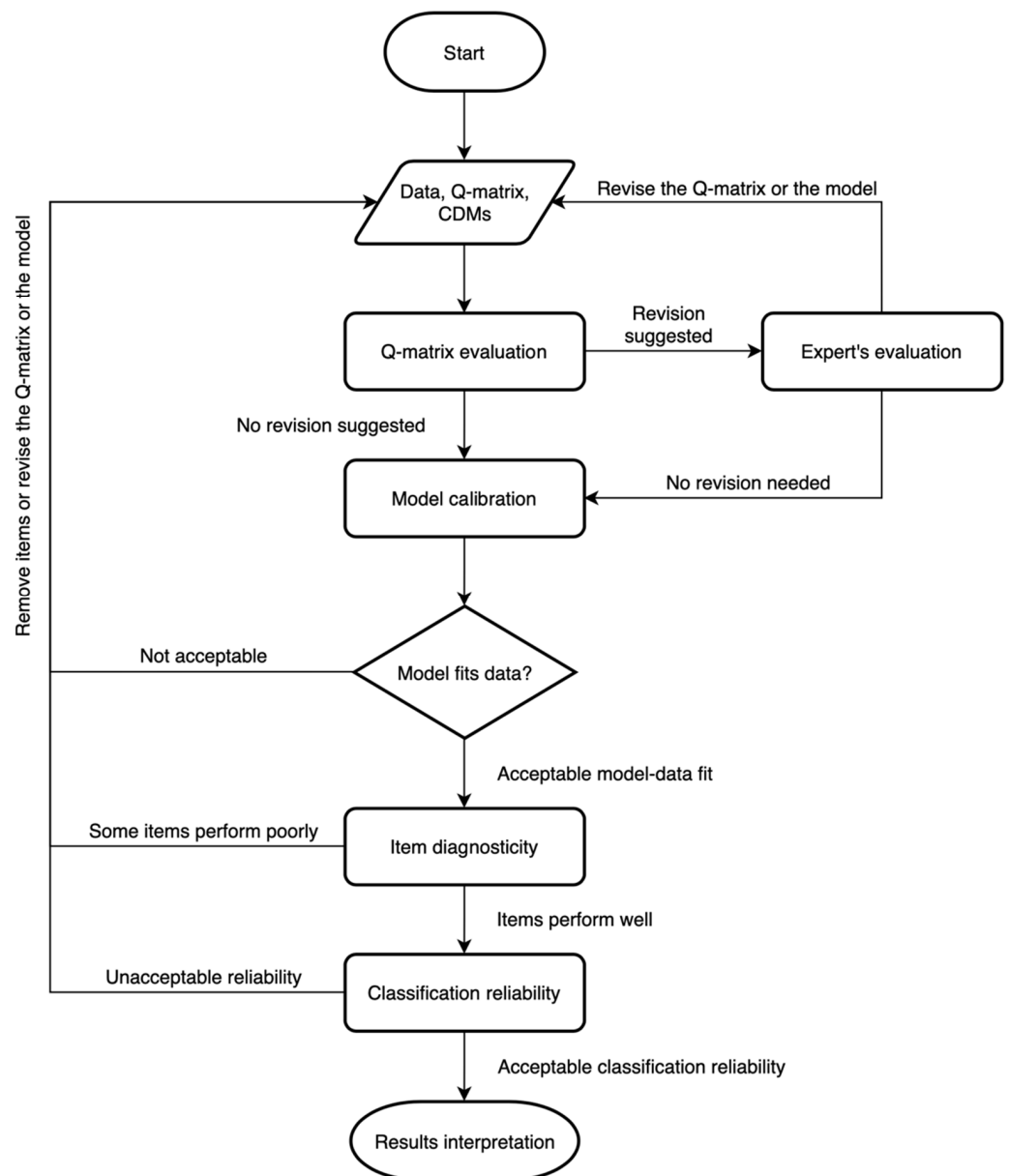


Figure 1. Data analysis diagram using the G-DINA model.

Similarly, the structural model can also be specified based on theories or statistical approaches. For example, when it is believed that attributes have a hierarchical relationship or are related to a common higher-order factor, the structural model should reflect such a belief. The likelihood ratio test can also be performed to compare the saturated structural model with a structural model it subsumes.

The next step of CDM analysis requires assessing model-data fit. Model-data fit can be gauged in an absolute sense, at either the test or item level. The test-level absolute fit evaluation provides information about to what extent the models can fit data well for the whole test, whereas the item-level absolute fit assesses whether or to what extent the model can fit data well for the item. Examples of absolute fit measures include the full information statistics such as Pearson χ^2 and limited information statistics such as M_2 statistic and $RMSEA_2$ [39,40]. Models can also be compared using relative fit measures at either test or item level. Examples of measures for relative fit evaluation include information criteria, such as AIC and BIC, and other inferential statistics, such as the Wald test and LR test [41].

When the goodness of fit is adequate, one can interpret model calibration results, including item diagnosticity and person classification reliability. In particular, the item

characteristic graph showing the probability of success for different latent groups can be displayed in a bar chart. Item discrimination index can also be calculated. Items with poor psychometric properties may need to be removed. In addition to item diagnosticity, test reliability should be investigated. The focus of CDM analysis is often classification, thus, classification accuracy and consistency should be assessed. With satisfactory classification reliability, the final step of CDM analysis is to report person classifications, which could be at the individual level or at an aggregated level for a group of students. It should be noted that CDM analysis may not necessarily be conducted sequentially. For example, the model or the Q-matrix may need to be revised, and some items may need to be removed if the model cannot fit data well.

4. An Illustration

This section will use a set of data to illustrate how to use different R packages for CDM analysis.

4.1. Data and Q-Matrix Preparation

The dataset we selected the grammar session of the Examination for the Certificate of Proficiency in English (ECPE) to illustrate an example of CDM analysis application, which has been used in several previous studies [9,36,42]. The dataset contains dichotomous responses to 28 items of 2922 students, reflecting their mastery of three grammar rules (attributes): morphosyntactic rules (A1), cohesive rules (A2), and lexical rules (A3). The Q-matrix is given in Table 1.

Table 1. Q-Matrix of the ECPE data.

Item	A1	A2	A3
1	1	1	0
2	0	1	0
3	1	0	1
4	0	0	1
5	0	0	1
6	0	0	1
7	1	0	1
8	0	1	0
9	0	0	1
10	1	0	0
11	1	0	1
12	1	0	1
13	1	0	0
14	1	0	0
15	0	0	1
16	1	0	1
17	0	1	1
18	0	0	1
19	0	0	1
20	1	0	1
21	1	0	1
22	0	0	1
23	0	1	0
24	0	1	0
25	1	0	0
26	0	0	1
27	1	0	0
28	0	0	1

Note. A1 = morphosyntactic rules; A2 = cohesive rules, A3 = lexical rules.

As shown in Table 1, the Q-matrix specified the attributes measured by each item. A cell with the value of 1 indicates that the corresponding item measures the corresponding

attribute, and a cell with the value of 0 indicates the opposite. For example, the attribute vector, or Q-vector, of item 1 is [1], indicating it measures attributes 1 and 2. One can find the ECPE data and corresponding Q-matrix from **CDM** [14], **GDINA** [15], or **edmdata** [43] R packages.

4.2. Empirical Q-Matrix Evaluation

Empirical Q-matrix evaluation involves validating the number of attributes or dimensionality evaluation and detecting misspecified elements in the provisional Q-matrix. Although it usually occurs during the Q-matrix development phase, dimensionality evaluation may provide valuable insight into the structure of the provisional Q-matrix. Dimensionality evaluation can be conducted by the **cdmTools** [44] package with `cdmTools::paK()` and `cdmTools::modelcompK()` functions. The `cdmTools::paK()` function adopts the parallel analysis method by comparing the eigenvalues generated from principal components, Pearson correlations, and mean criterion [27,45] of the randomly resampled correlation matrices and their sample correlation matrices. The argument `cor` specifies the type of correlations to be used, whose default value is “both”, implying using both Pearson and tetrachoric/polychoric correlations. In our code, we define `cor = “cor”`, indicating the Pearson correlations are employed. The number of suggested attributes is extracted by `$sug.K`. As presented in the output below, the suggested number of attributes is 3, which is equal to that of our provisional Q-matrix.

```
>R res.paK <- cdmTools::paK(dat, cor = “cor”)
>R res.paK$sug.K
[1] 3
```

The `cdmTools::modelcompK()` function compares several model fit indices of the CDMs fitted with different Q-matrices of a specified number of attributes that are developed through the discrete factor loading method (DFL) [46] and the Hull method [47]. Nájera and colleagues [27] suggested preferring the AIC over other indices. In `modelcompK()` function, `exploreK = 1:5` indicates that Q-matrices with one to five attributes were evaluated.

```
>R res.modelcompK <- cdmTools::modelcompK(dat, exploreK = 1:5)
Estimating and validating Q-matrix with K = 1 2 3 4 5
k = 1 explored | AIC = 86,059 | BIC = 86,400
k = 2 explored | AIC = 85,859 | BIC = 86,212
k = 3 explored | AIC = 85,367 | BIC = 85,995
k = 4 explored | AIC = 85,381 | BIC = 86,212
k = 5 explored | AIC = 85,359 | BIC = 86,645
```

The number of suggested attributes under each model fit index is extracted by `$sug.K` as well.

```
>R res.modelcompK$sug.K
      AIC      BIC      CAIC      SABIC      M2
5       3       3       3       3       5
M2.p    SRMSR    RMSEA2    RMSEA2.low  RMSEA2.high
5       5       5       5       5       5
sig.item.pairs
5
```

The AIC and BIC values can be plotted across the number of attributes using the `plot()` function to obtain a direct view of the comparison result. The plots in Figure 2 demonstrate that the tendency change of AIC values and the minimum BIC value are both at $K = 3$.

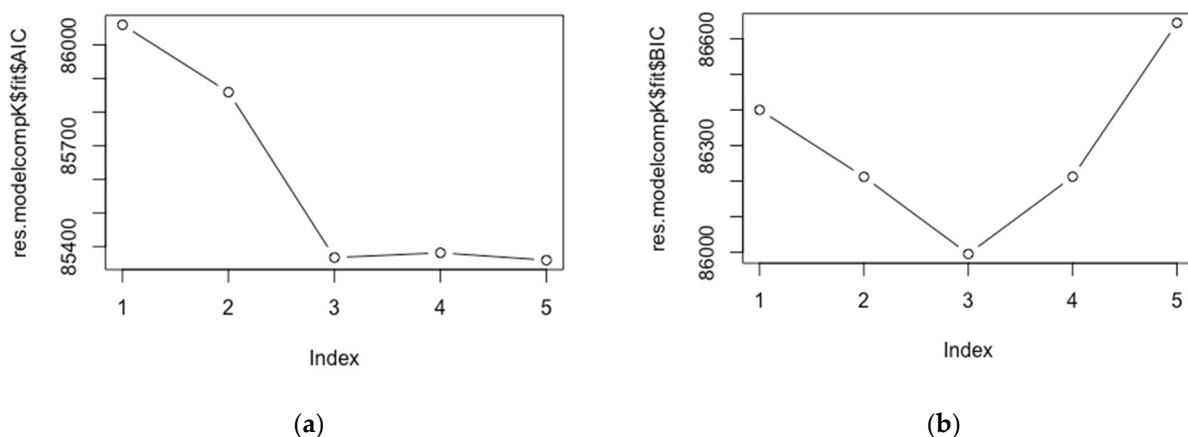


Figure 2. (a) The scatterplot of AIC; (b) The scatterplot of BIC.

```
>R plot(res.modelcompK$fit$AIC, type = "b")
>R plot(res.modelcompK$fit$BIC, type = "b")
```

After the number of attributes has been assessed, whether the Q-matrix consists of misspecified elements needs to be examined. Many R packages provide functions for this purpose. For example, the **CDM** package implements de la Torre's method [48], and the **NPCD** package [49] refines the Q-matrix based on Chiu's [50] nonparametric approach. Both methods may be used for mixed DINA and DINO models. The **GDINA** and **cdmTools** have functions for Q-matrix validation under the saturated CDMs.

Following the suggestion of de la Torre and Minchen [32], the G-DINA model was employed when conducting Q-matrix validation to avoid conflating Q-matrix misspecifications with model misspecifications. Specifically, the G-DINA model was fitted to the data using the code shown below. The argument *mono.constraint* is set to TRUE to impose monotonicity constraints to the model, ensuring that the probability of having a correct response of an item will not decrease as the student masters more required attributes. The argument *control = list(conv.crit = 0.000001)* indicates that convergence criterion was set to 0.000001 instead of the default value at 0.0001.

```
>R est <- GDINA::GDINA(dat, Q, model = "GDINA", mono.constraint = TRUE,
+                       control = list(conv.crit = 0.000001))
```

In this paper, the stepwise Wald test [29] was used by specifying *method = "wald"* in *GDINA::Qual()* function. Alternatively, the PVAF (i.e., the proportion of variance accounted for) method with fixed or predicted cutoffs can be applied [28] when using this function. In the **cdmTools** package, Q-matrix validation can be performed with *cdmTools::valQ()* function, which implements the Hull method with PVAF or McFadden's pseudo R-squared [47] with various iteration algorithms [31].

```
>R qv <- GDINA::Qual(GDINA.obj = est, method = "wald")
```

The suggested Q-matrix based on the stepwise Wald test is presented below. The cells marked with an asterisk are modified according to the validation results. In our case, the q-vector of items 9 and 13 was suggested to be modified.

```
>R print(qv)
```

Q-matrix validation based on Stepwise Wald test

Suggested Q-matrix:

	A1	A2	A3
Item 1	1	1	0
Item 2	0	1	0
Item 3	1	0	1
Item 4	0	0	1
Item 5	0	0	1
Item 6	0	0	1

Item 7	1	0	1
Item 8	0	1	0
Item 9	1*	0	1
Item 10	1	0	0
Item 11	1	0	1
Item 12	1	0	1
Item 13	1	0	1*
Item 14	1	0	0
Item 15	0	0	1
Item 16	1	0	1
Item 17	0	1	1
Item 18	0	0	1
Item 19	0	0	1
Item 20	1	0	1
Item 21	1	0	1
Item 22	0	0	1
Item 23	0	1	0
Item 24	0	1	0
Item 25	1	0	0
Item 26	0	0	1
Item 27	1	0	0
Item 28	0	0	1

Note: * denotes a modified element.

Additionally, the mesa plots were drawn [51] to visualize the PVAf of q-vectors using the code below for Items 9 and 13, as shown in Figure 3. In mesa plots, the x -axis represents the q-vectors and the y -axis their corresponding PVAf values. The default cutoff value for PVAf (eps) is set to 0.95 out of the range from 0 to 1. The cutoff can be adjusted according to the researcher's judgment. De la Torre and Ma (2016) recommended the q-vector on the edge of the "mesa" to be considered the correct q-vector for the item. In these mesa plots, the red dots are the original q-vectors. These plots indicate that attributes 3 and 1 each contributed to most of the variance of the item success probabilities of items 9 and 13, respectively, whereas the rest did not contribute much. According to these plots, the original q-vectors [001] and [100] are suggested to be the correct ones instead of the q-vectors [101] in the modified Q-matrix.

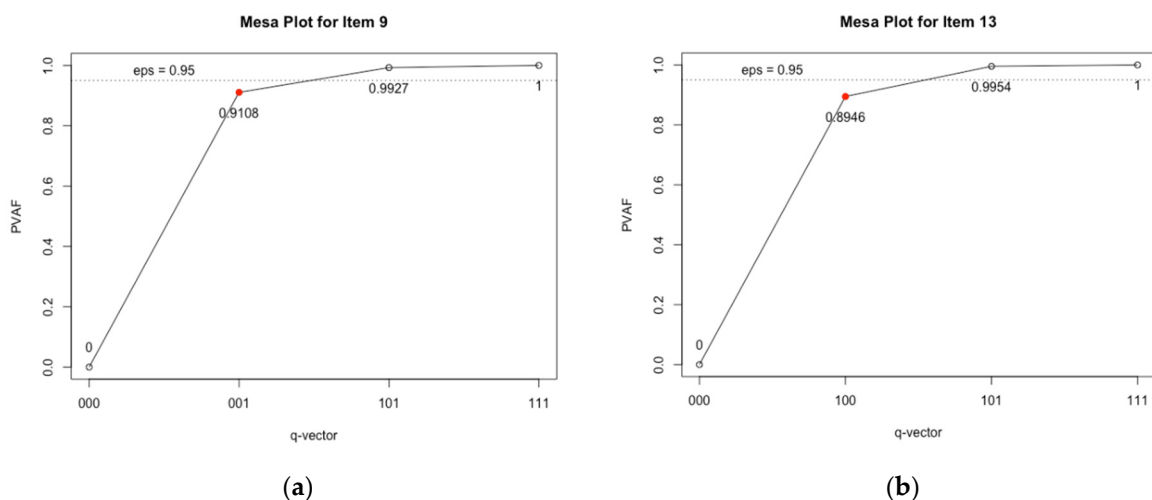


Figure 3. (a) Mesa plot for item 9; (b) Mesa plot for item 13.

```
R> plot(qv, c(9, 13), eps = 0.95, data.label = TRUE)
```


Please note that, in this section, only some Q-matrix validation methods were discussed when a provisional Q-matrix is available; however, some exploratory methods [52–55] have also been developed to estimate the Q-matrix based on the response data without the need for a provisional Q-matrix. Additionally, different Q-matrix validation methods may produce different recommendations because their performance may be affected by many factors. For example, it has been shown that the stepwise Wald method may prove difficult in converging when the number of attributes is large. As a result, although those recommendations could be valuable for refining the Q-matrix, whether to adopt the suggestions should be contingent on domain experts' judgment and interpretation.

4.3. CDM Calibration

After the Q-matrix is finalized, CDMs can fit into the data. Both **CDM** and **GDINA** packages can fit the G-DINA model to the data, but they have different default settings. This section will show how the model calibration using one package can be converted to the other package. First, the data is calibrated using the **GDINA** package with the code below:

```
>R tol <- 0.000001
>R GDINA.est <- GDINA::GDINA(dat, Q, model = "GDINA", mono.constraint = TRUE,
>+                               control = list(conv.crit = tol))
```

Based on the estimates from the **GDINA** package, the following code allows fixing the parameters at their estimated values and obtaining an object of "gdina" from the **CDM** package directly. In particular, item parameter estimates from the `GDINA::GDINA()` function of the **GDINA** package were extracted by `GDINA.est$delta.parm`, and the prior probabilities for each latent class for the last E-step of the EM cycle are obtained via `GDINA::extract(GDINA.est, what = "att.prior")`. Using the **CDM** package, the arguments in `CDM::gdina()` function can be defined with the elements we extracted from `GDINA.est`. In particular, the arguments `delta.fixed` and `attr.prob.fixed` make it possible to fix delta parameters and attribute probabilities, respectively. The argument `reduced.skillspace` is `FALSE`, indicating the attribute patterns were not reduced and all possible attribute patterns were included in the estimation [56]. It should be noted that the attribute space needs to be specified using argument `skillclasses` in that `CDM::gdina()` and `GDINA::GDINA()` functions use different attribute spaces by default. In the code below, the attribute space was defined as `att.pattern` and then specified in `CDM::gdina()` function. Because of those settings, `GDINA.est` and `CDM.est` contain equivalent estimation results.

```
>R delta.param <- extract(GDINA.est, "delta.parm")
>R mixing.proportions <- GDINA::extract(GDINA.est, what = "att.prior")
>R K <- ncol(Q)
>R att.pattern <- extract(GDINA.est, "attributepattern")
>R CDM.est <- CDM::gdina(dat, Q, skillclasses = att.pattern,
>+                               delta.fixed = delta.param,
>+                               attr.prob.fixed = mixing.proportions,
>+                               reduced.skillspace = FALSE)
```

The data can be fit first using the **CDM** package prior to fixing the parameter estimates in the **GDINA** package. As shown below, when fitting the data using `CDM::gdina()` function, monotonic constraints were imposed using argument `mono.constr` and set criteria for convergence using arguments `conv.crit` and `dev.crit`. It should be noted that when the monotonicity constraints are imposed, a logit link G-DINA model is adopted by default, which is mathematically equivalent to the identity link G-DINA model.

```
>R cdm.fit <- CDM::gdina(dat, Q, rule = "GDINA", conv.crit = tol,
>+                               dev.crit = tol,
>+                               mono.constr = TRUE)
```

The `GDINA::GDINA()` function does not allow fixing delta parameters directly; instead, the item success probabilities can be fixed. The code below extracts the probabilities of success for each reduced attribute profile on each item:

```
>R p <- list()
>R for(j in 1:ncol(dat)){
>+   p[[j]] <- unlist(subset(cdm.fit$probitem, itemno == j, select = prob))
>+ }
```

The code below calls `GDINA::GDINA()` with several arguments. In particular, the logit link G-DINA model was specified via the arguments `model` and `linkfunc`. The attribute space used in the **CDM** package was extracted via `cdm.fit$attribute.patt.splitted` and specified using `att.str` in `GDINA::GDINA()` function. The initial item success probabilities were specified via argument `catprob.parm` and the initial distribution of latent classes was specified using `att.prior`. By specifying `maxitr = 0` in argument `control`, the E-M cycle was disabled, and the initial item success probabilities and distribution of latent classes were used for the final E-step calculation.

```
>R gdina.fit <- GDINA::GDINA(dat, Q, model = "GDINA", linkfunc = "logit",
>+   att.str = cdm.fit$attribute.patt.splitted,
>+   catprob.parm = p, att.prior = cdm.fit$attr.prob,
>+   control = list(maxitr = 0))
```

After determining the CDMs and Q-matrix, the assessment is performed to determine whether the parameters of the model can be identified. The function `cdmTools::is.Qid()` from the **cdmTools** package checks model identifiability according to the criteria from Chen and colleagues [53] and Xu and Shang [57]. As shown below, all parameters of the G-DINA model can be identified in this example. Q-matrix in the `Q` argument, as well as the model that was estimated in the model, need to be provided. Available inputs for the model are "DINA", "DINO", or "others". Here "others" are indicated because of the use of the G-DINA model.

```
>R cdmTools::is.Qid(Q, model = "others")
```

So far, the discussion has been focused on how to estimate the G-DINA model using both packages, obtain the equivalent objects between two packages, and assess the identifiability of the G-DINA model globally. In practice, researchers may want to simplify the G-DINA model empirically because it has been shown that reduced models, when used appropriately, can provide better classification results than the G-DINA model [34]. The **GDINA** package offers a function called `GDINA::modelcomp()`, which implements the Wald test and likelihood ratio test for assessing whether the G-DINA model can be reduced to five commonly used reduced models, namely, the DINA model, the DINO model, A-CDM, LLM, and R-RUM, as shown below:

```
>R mc <- GDINA::modelcomp(GDINA.est)
>R mc
Item-level model selection:
test statistic: Wald
Decision rule: simpler model + largest p-value rule at 0.05 alpha level.
Adjusted p-values were based on holm correction.
```

	models	pvalues	adj.pvalues
Item 1	RRUM	0.4815	1
Item 2	GDINA		
Item 3	RRUM	0.7505	1
Item 4	GDINA		
Item 5	GDINA		
Item 6	GDINA		
Item 7	LLM	0.6565	1
Item 8	GDINA		
Item 9	GDINA		
Item 10	GDINA		

Item 11	ACDM	0.9209	1
Item 12	RRUM	0.4902	1
Item 13	GDINA		
Item 14	GDINA		
Item 15	GDINA		
Item 16	LLM	0.5678	1
Item 17	DINO	0.1332	1
Item 18	GDINA		
Item 19	GDINA		
Item 20	RRUM	0.3889	1
Item 21	LLM	0.9537	1
Item 22	GDINA		
Item 23	GDINA		
Item 24	GDINA		
Item 25	GDINA		
Item 26	GDINA		
Item 27	GDINA		
Item 28	GDINA		

Similarly, the **CDM** package implements the Wald test for comparing the G-DINA model with the DINA model, the DINO model, A-CDM in `CDM::gdina.wald()`. In addition, the **CDM** package allows researchers to fit the regularized G-DINA model using a variety of penalty terms. This is a flexible approach to simplifying the G-DINA model and interested readers may refer to Robitzsch [37] and Robitzsch and George [36] for more information. A caveat to the Wald and LR tests for model comparisons is that trivial discrepancy between two models may be detected when sample size is large and one should be aware that the logit link must be used when the regularized G-DINA model is specified in the **CDM** package.

4.4. Model Fit Evaluation

Both **CDM** and **GDINA** packages offer functions for assessing model–data fit. Table 2 shows the functions and the statistics calculated in each package. It is evident that both packages calculate various statistics for assessing both absolute and relative fit at test and item levels. This paper will not enumerate the outputs of all those statistics; instead, it will focus on absolute fit statistics, as only the G-DINA model was used, and present some results as an example.

4.4.1. Test-Level Fit Evaluation

Most test-level absolute fit measures gauge the discrepancy between observed quantities and model-implied counterparts. For example, the M_2 statistic [38,58,59] compares the univariate and bivariate distributions of observations and model predictions. Because it conforms to χ^2 distribution, hypothesis tests can be conducted to assess whether the model fits data. However, it is well-known that a hypothesis test is affected by sample size, and a large sample may capture trivial discrepancies between the model and the data. To address this issue, the root mean square error of approximation ($RMSEA_2$) [39,60] and the standardized root mean square residual (SRMSR) [39,61] can be used as effect–size measures. For both $RMSEA_2$ and SRMSR, a smaller value indicates a better absolute model data fit [62]. Simulation studies suggest that $RMSEA_2 < 0.03$ indicates excellent fit, $0.03 < RMSEA_2 < 0.045$ a good fit, and $RMSEA_2 > 0.045$ poor fit. SRMSR < 0.05 indicates good model fit [39,63]. It should be noted that when the number of parameters is large, the M_2 statistic, as well as $RMSEA_2$, may not be calculable.

Table 2. Model-data fit statistics.

	CDM		GDINA	
	Function	Statistics	Function	Statistics
Absolute fit	Test-level	<i>IRT.modelfit()</i>	<i>modelfit()</i> <i>itemfit()</i>	M ₂ RMSEA ₂ SRMSR MaxAD.r MaxAD.LOR
	Item-level	<i>IRT.RMSD()</i> <i>itemfit.sx2()</i>	<i>itemfit()</i>	MaxAD.r MaxAD.LOR
	Item-pair level	<i>IRT.modelfit()</i>	<i>itemfit()</i>	MaxAD.r MaxAD.LOR
Relative fit	Test-level	<i>IRT.modelfit()</i> <i>anova()</i>	<i>modelfit()</i> <i>anova()</i>	AIC BIC CAIC SABIC LR test
	Item-level	<i>gdina.wald()</i>	<i>modelcomp()</i>	Wald test LR test

Note. $\max(X^2)$ = the maximum chi-square statistic; MADcor = mean of absolute deviation of correlations; SRMSR = standardized mean square root of squared residuals; MADRESIDCOV = mean of absolute deviation of residual covariances; abs(fcor) = the absolute deviation of Fisher transformed correlations; RMSD = root mean square deviation; RMSD_bc = RMSD statistic with analytical bias correction; MAD = mean absolute deviation; MD = mean deviation; χ^2 = chi-square statistic; $S-\chi^2$ = S-chi-square statistic; RMSEA = the root mean square error of approximation; fcor = Fisher transformed correlations; AIC = Akaike's Information Criteria; BIC = Bayesian Information Criteria; CAIC = consistent AIC; AICc = the sample size adjusted AIC; M₂ = the second-order marginal statistic; RMSEA₂ = limited information RMSEA; MaxAD.r = maximum absolute deviation of transformed correlation; MaxAD.LOR = maximum absolute deviation of log odds ratio; SABIC = the sample size adjusted BIC; LR test = likelihood ratio test.

Aggregated item-level or item-pair level absolute fit measures have also been used to assess test-level fit. Examples include the mean absolute difference between the observed and expected correlations (MADcor) [64,65], the maximum absolute difference between observed and predicted Fisher transformed correlations (MaxAD.r) [64], the maximum absolute difference between observed and predicted log odds ratios (MaxAD.LOR) [64], the mean of absolute deviations of residual covariances (MADRESIDCOV) [66], and the maximum χ^2 value of all item pairs ($\max(X^2)$) [67]. The χ^2 statistic quantifies the deviance between the observed and predicted item-pair distributions, using individual posterior distributions of the specified model. For MaxAD.r, MaxAD.LOR and $\max(X^2)$, one can often report adjusted *p*-value to assess whether the model and the data fit well for the worst pair of items. For other measures, a small value indicates a good fit. Since the value of MADRESIDCOV is often small, the value of 100*MADRESIDCOV is usually adopted [68].

In the CDM package, function `CDM::IRT.modelfit()` can be used to calculate MADcor, MaxAD.r (labelled as `abs(fcor)`), MADRESIDCOV, $\max(X^2)$ and SRMSR. The `CDM::IRT.modelfit()` function also calculates information criteria, such as Akaike's Information Criteria (AIC) [69] and Bayesian Information Criteria (BIC) [70]. Both criteria are based on the maximum likelihood statistic, and BIC is additionally affected by sample size. Both AIC and BIC serve as a measure for comparing model fit, and a smaller value indicates better model fit. Nevertheless, please note that when parameters are fixed in model calibration, the calculation of information criteria is incorrect and must be manually corrected.

In the **GDINA** package, `GDINA::modelfit()` function calculates M_2 statistic, $RMSEA_2$, and SRMSR for absolute fit evaluation, and calculates log-likelihood, AIC, BIC, CAIC, and SABIC for relative fit evaluation. The `GDINA::itemfit()` function calculates MaxAD.r and MaxAD.LOR. The code below shows how to obtain these statistics from **CDM** and **GDINA** packages.

```
>R mf <- CDM::IRT.modelfit(CDM.est)
>R mf$modelfit.test
  type      value      p
1 max(X2)  39.5604293  1.202279e-07
2 abs(fcor) 0.1170423  4.831683e-08

>R mf$modelfit.stat
      est
MADcor      0.02516064
SRMSR       0.03174674
100*MADRESIDCOV 0.45668011
MADQ3       0.02267701
MADaQ3      0.02236228

>R GDINA::modelfit(GDINA.est)
Test-level Model Fit Evaluation

Absolute fit statistics:
M2 = 506.2694  df = 325  p = 0
RMSEA2 = 0.0138 with 90% CI: [0.0114, 0.0161]
SRMSR = 0.0317

>R GDINA::itemfit(GDINA.est)
Summary of Item Fit Analysis

Call:
GDINA::itemfit(GDINA.obj = GDINA.est)

      mean[stats] max[stats] max[z.stats] p-value adj.p-value
Proportion correct 0.0009  0.0025  0.3152  0.7526  1
Transformed correlation 0.0255  0.1173  6.3375  0.0000  0
Log odds ratio 0.1341  0.5335  6.5190  0.0000  0
Note: p-value and adj.p-value are associated with max[z.stats].
adj.p-values are based on the holm method.
```

4.4.2. Item-Level and Item-Pair Level Fit Evaluation

Item-level absolute fit can be assessed using $S\text{-}\chi^2$ item fit statistic [41,71], which can be calculated using `CDM::itemfit.sx2()` function. The $S\text{-}\chi^2$ item fit statistic compares observed and expected proportions for each item and each latent class and forms a chi-square distributed statistic. As a result, the items with p -values greater than 0.05 indicate good item fit at 0.05 nominal level. The output, for instance, indicates that item 13 has a significant misfit.

```
>R sx2 <- CDM::itemfit.sx2(CDM.est)
>R summary(sx2)
item  itemindex S-X2  df  p  S-X2_df  RMSEA  Nscgr  Npars  p.holm
1 Item 1  1  13.222  16  0.656  0.826  0.000  20  4  1.000
2 Item 2  2  22.492  18  0.211  1.250  0.009  20  2  1.000
3 Item 3  3  13.459  16  0.639  0.841  0.000  20  4  1.000
4 Item 4  4  20.444  18  0.308  1.136  0.007  20  2  1.000
5 Item 5  5  23.327  18  0.178  1.296  0.010  20  2  1.000
6 Item 6  6  16.232  18  0.576  0.902  0.000  20  2  1.000
7 Item 7  7  11.512  16  0.777  0.720  0.000  20  4  1.000
8 Item 8  8  10.404  18  0.918  0.578  0.000  20  2  1.000
9 Item 9  9  22.559  18  0.208  1.253  0.009  20  2  1.000
10 Item 10 10  39.520  18  0.002  2.196  0.020  20  2  0.065
```

```

11 Item 11 11 18.609 16 0.289 1.163 0.007 20 4 1.000
12 Item 12 12 20.022 16 0.219 1.251 0.009 20 4 1.000
13 Item 13 13 55.254 18 0.000 3.070 0.027 20 2 0.000
14 Item 14 14 18.053 18 0.452 1.003 0.001 20 2 1.000
15 Item 15 15 15.154 18 0.651 0.842 0.000 20 2 1.000
16 Item 16 16 32.079 16 0.010 2.005 0.019 20 4 0.254
17 Item 17 17 13.134 16 0.663 0.821 0.000 20 4 1.000
18 Item 18 18 17.355 18 0.499 0.964 0.000 20 2 1.000
19 Item 19 19 33.059 18 0.016 1.837 0.017 20 2 0.410
20 Item 20 20 14.929 16 0.530 0.933 0.000 20 4 1.000
21 Item 21 21 15.607 16 0.481 0.975 0.000 20 4 1.000
22 Item 22 22 27.171 18 0.076 1.509 0.013 20 2 1.000
23 Item 23 23 16.777 18 0.538 0.932 0.000 20 2 1.000
24 Item 24 24 18.182 18 0.444 1.010 0.002 20 2 1.000
25 Item 25 25 16.269 18 0.574 0.904 0.000 20 2 1.000
26 Item 26 26 25.935 18 0.101 1.441 0.012 20 2 1.000
27 Item 27 27 24.753 18 0.132 1.375 0.011 20 2 1.000
28 Item 28 28 13.008 18 0.791 0.723 0.000 20 2 1.000

```

```
--Average Item Fit Statistics--
```

```
S-X2 = 21.019 | S-X2_df = 1.206
```

Other item-level absolute fit measures can be requested through function `CDM::IRT.RMSD()` [63,72]. It computes item-wise and group-wise root mean square deviation (RMSD), bias corrected root mean square deviation (RMSD_bc), mean absolute deviation (MAD), mean deviation (MD), and chi square statistic [73,74].

Unlike the $S\text{-}\chi^2$ and RMSD statistics that focus on to what extent the model can fit data well for each item, the absolute difference between observed and predicted Fisher transformed correlations and the absolute difference between observed and predicted log odds ratios for all item pairs [64] are reported in the **GDINA** package. Both measures focus on to what extent the model can explain the association between each pair of items. A heatmap plot illustrating the adjusted p -values of transformed correlation between item pairs can be requested using `plot()`, as demonstrated in Figure 4. In the heatmap plot, items are presented on both x - and y -axes. The first item on the x -axis and the last on the y -axis were dropped for pairing items. The adjusted p -values of all item pairs are plotted in the lower right shading area, where those of adequately fitted item pairs are in grey ($p > 0.05$) and those of inadequately fitted item pairs are in different tones of red ($p < 0.05$), depending on the p -value [42]. In our case, some item pairs (e.g., items 9 and 10 and items 13 and 22) demonstrated significant misfit and thus are in demand for further exploration by domain experts.

```

>R itf <- GDINA::itemfit(GDINA.est)
>R plot(itf)

```

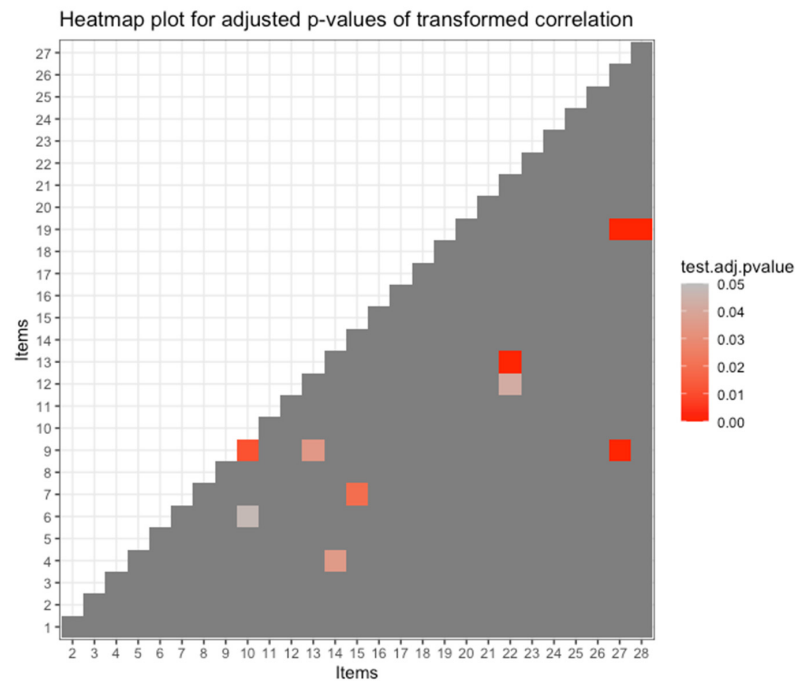


Figure 4. Heatmap plot for adjusted p -values for transformed correlation.

4.5. Item Diagnosticity Investigation

To assess item diagnosticity, the distribution of the probability of success across all latent groups in each item can be drawn, using `plot()` as the code presents below. The plots should show the distinctions between the bars representing each latent group. A good example is item 20 presented in Figure 5, where an increase is observed in the probability of success as a student masters more attributes measured by this item. A poor example is item 17, where the success probability of all four latent groups is over 0.75 and little difference is observed between the bars. This indicates that a student has a more than 75% chance to answer this item correctly whether they do not master any attributes required by this item, master only one attribute, or master both attributes. In this regard, item 17 does not have the ability to distinguish students in different latent groups. Similar plots can be drawn using the `plot()` function as well in the **CDM** package.

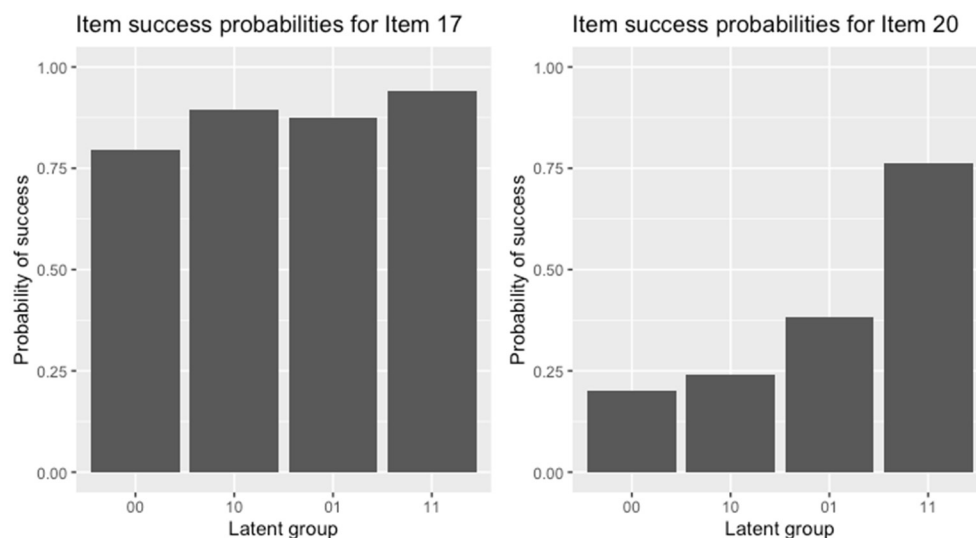


Figure 5. Plots of Success Probabilities of Items 17 and 20.

```
>R plot(GDINA.est, item = 1:28)
```

Another way to check item diagnosticity is to investigate the item discrimination indices. In the **GDINA** package, item discrimination is measured by two indices: P(1)-P(0) and G-DINA discrimination index (GDI). P(1)-P(0) measures the differences in success probabilities between those who master all required attributes and those who master none of them. GDI measures the variance of the item success probabilities based on the reduced attribute profile [28,75]. An item with a higher value of P(1)-P(0) or GDI has higher discrimination power. Currently, there is no agreement in the field regarding the value of good discrimination power. Although a higher value of P(1)-P(0) or GDI is desirable, it could be an indicator of overspecified q-vectors [28]. These two item discrimination indices can be requested using `GDINA::extract()` as in:

```
>R GDINA::extract(GDINA.est, what = "discrim")
```

	P(1)-P(0)	GDI
Item 1	0.2369939	0.011248186
Item 2	0.1693254	0.007113710
Item 3	0.3681559	0.025439442
Item 4	0.3598914	0.028732987
Item 5	0.2099038	0.009774136
Item 6	0.2243154	0.011162359
Item 7	0.4663292	0.037643097
Item 8	0.1529452	0.005803950
Item 9	0.2584621	0.014819434
Item 10	0.3776435	0.033710651
Item 11	0.4350662	0.031866830
Item 12	0.5924216	0.062055464
Item 13	0.2487938	0.014631276
Item 14	0.2802295	0.018562263
Item 15	0.2269881	0.011429941
Item 16	0.4319718	0.032239549
Item 17	0.1475092	0.004242115
Item 18	0.1938551	0.008336667
Item 19	0.3878091	0.033363663
Item 20	0.5640100	0.057509619
Item 21	0.3751756	0.024045446
Item 22	0.5030707	0.056143024
Item 23	0.2788246	0.019289193
Item 24	0.3625726	0.032616838
Item 25	0.2504526	0.014827037
Item 26	0.2420333	0.012995354
Item 27	0.4038545	0.038552543
Item 28	0.2717041	0.016376848

In the **CDM** package, P(1)-P(0) is referred to as item discrimination index or IDI [48,76]. The **CDM** package also calculates the discrimination index (DI) at the item-attribute level based on the mastery probability of including and excluding the measured attribute for a specific item and the DI at the test level by averaging the marginalized probability of DIs at the item-attribute level for each item. Using `CDM::discrim.index()`, the test, item, and item-attribute level DIs can be requested. Note that the IDI at item level is the same as the P(1)-P(0) values requested by `GDINA::extract()`. Although not presented here, the **CDM** package also calculates the cognitive diagnostic index (CDI) based on the Kullback-Leibler information (KLI) [76], which can be requested by using `CDM::cdi.kli()`.

```
>R summary(CDM::discrim.index(CDM.est))
```

```
-----  
CDM 7.5-15 (2020-03-10 14:19:21)  
-----
```


Test-level discrimination index

[1] 0.304

Item discrimination index (IDI)

Item1	Item2	Item3	Item4	Item5	Item6	Item7	Item8	Item9	Item10
0.237	0.169	0.368	0.360	0.210	0.224	0.466	0.153	0.258	0.378
Item11	Item12	Item13	Item14	Item15	Item16	Item17	Item18	Item19	Item20
0.435	0.592	0.249	0.280	0.227	0.432	0.148	0.194	0.388	0.564
Item21	Item22	Item 23	Item 24	Item 25	Item 26	Item 27	Item 28		
0.375	0.503	0.279	0.363	0.250	0.242	0.404	0.272		

Item-attribute discrimination index

item	A1	A2	A3
1 Item 1	0.127	0.237	0.000
2 Item 2	0.000	0.169	0.000
3 Item 3	0.282	0.000	0.183
4 Item 4	0.000	0.000	0.360
5 Item 5	0.000	0.000	0.210
6 Item 6	0.000	0.000	0.224
7 Item 7	0.466	0.000	0.226
8 Item 8	0.000	0.153	0.000
9 Item 9	0.000	0.000	0.258
10 Item 10	0.378	0.000	0.000
11 Item 11	0.210	0.000	0.240
12 Item 12	0.355	0.000	0.592
13 Item 13	0.249	0.000	0.000
14 Item 14	0.280	0.000	0.000
15 Item 15	0.000	0.000	0.227
16 Item 16	0.430	0.000	0.213
17 Item 17	0.000	0.098	0.081
18 Item 18	0.000	0.000	0.194
19 Item 19	0.000	0.000	0.388
20 Item 20	0.382	0.000	0.522
21 Item 21	0.231	0.000	0.244
22 Item 22	0.000	0.000	0.503
23 Item 23	0.000	0.279	0.000
24 Item 24	0.000	0.363	0.000
25 Item 25	0.250	0.000	0.000
26 Item 26	0.000	0.000	0.242
27 Item 27	0.404	0.000	0.000
28 Item 28	0.000	0.000	0.272

4.6. Classification Reliability

Classification reliability refers to whether the model can consistently and accurately classify test-takers into latent classes, usually measured by classification accuracy and consistency. Specifically, classification accuracy relates to the extent to which the estimated attribute classifications and the true classifications are the same, whereas classification consistency concerns the extent to which the estimated attribute classifications from two parallel test forms are consistent. Although different measures have been proposed in the literature [77–79], the **GDINA** package calculates classification accuracy at test, pattern, and attribute levels according to Iaconangelo [80] and Wang et al. [81]. In contrast, the **CDM** package calculates both classification accuracy and consistency at pattern and attribute levels using the estimator of Johnson and Sinharay [78].

The classification accuracy of maximum a posteriori or MAP method (by default, and maximum likelihood estimation or MLE can be requested) can be estimated in the **GDINA** package using `GDINA::CA()` as demonstrated below:

```
>R GDINA::CA(GDINA.est, what = "MAP")
Classification Accuracy
Test level accuracy = 0.747
Pattern level accuracy:
000 100 010 001 110 101 011 111
0.8942 0.1385 0.0000 0.4486 0.1891 0.0882 0.5780 0.9091
Attribute level accuracy:
      A1      A2      A3
0.8968 0.8538 0.9161
```

In the **CDM** package, one can calculate both classification accuracy and consistency using `CDM::cdm.est.class.accuracy()` as demonstrated below. The output gives classification accuracy and consistency statistics [77,81] at attribute and latent class level for both MLE and MAP estimators. `Pa_est` and `Pc_est` give classification accuracy and consistency by estimators of Johnson and Sinharay [78,82], respectively, and `Pa_sim` and `Pc_sim` give classification accuracy and consistency only for DINA, DINO, and mixed DINA and DINO models based on simulation, respectively. The classification accuracy values at latent class and attribute level for MAP estimators are the same as the values we obtained from the **GDINA** package.

```
>R summary(CDM::cdm.est.class.accuracy(CDM.est))
      Pa_est Pa_sim Pc_est Pc_sim
MLE_patt 0.594 0.621 0.409 0.437
MAP_patt 0.747 0.771 0.664 0.686
MLE_A1   0.859 0.886 0.760 0.798
MLE_A2   0.759 0.762 0.645 0.645
MLE_A3   0.893 0.911 0.811 0.838
MAP_A1   0.896 0.913 0.833 0.849
MAP_A2   0.854 0.860 0.807 0.817
MAP_A3   0.916 0.933 0.854 0.882
```

4.7. CDM Result Presentation

The primary goal of CDM analysis is to classify students into different latent classes or estimate students' attribute profiles. Both **CDM** and **GDINA** packages can estimate a person's parameters using expected a posteriori (EAP), MAP, or MLE methods [83]. In the **GDINA** package, the `GDINA::personparm()` function could be used, whereas in the **CDM** package, the `CDM::IRT.factor.scores()` function can be applied. Below are the estimated attribute profiles of the first six students using `GDINA::personparm()`.

```
>R head(personparm(GDINA.est))
      A1  A2  A3
[1,]  1   1   1
[2,]  1   1   1
[3,]  1   1   1
[4,]  1   1   1
[5,]  1   1   1
[6,]  1   1   1
```

Meanwhile, we can obtain the success probability of attribute mastery for each student by specifying the `what` argument to "mp" in `GDINA::personparm()` function.

```
>R head(personparm(GDINA.est, what = "mp"))
      A1      A2      A3
[1,] 0.9967 0.9615 0.9999
[2,] 0.9952 0.9150 0.9999
[3,] 0.9841 0.9898 1.0000
```

```
[4,] 0.9976 0.9913 1.0000
[5,] 0.9884 0.9845 0.9512
[6,] 0.9929 0.9908 1.0000
```

Using the *mpRadar()* function created from the *fmsb::radarchart()* function of the *fmsb* R package [84], a radar chart of the mastery probability for each student or for several students at the same time can be plotted. As presented in Figure 6, the student has a nearly 100% chance of mastering the lexical rules, a nearly 0% chance of mastering the morphosyntactic rules, and about a 45% chance of mastering the cohesive rules.

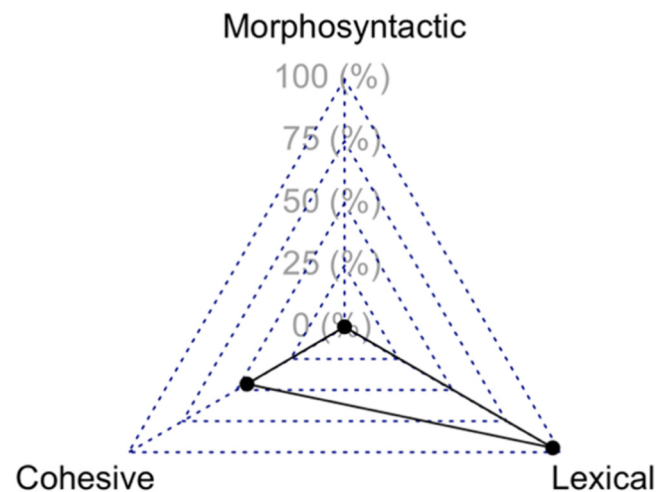


Figure 6. Radar Chart of the attribute mastery probabilities of student 8.

In addition to person classifications, the proportion of students who master or do not master each attribute referred to as attribute prevalence, and the proportion of students in each latent class referred to as latent class proportions can also be measured. They can be requested by calling *GDINA::extract()* function and specifying *what = "prevalence"* and *"posterior.prob"* in the *GDINA* package, respectively. In the *CDM* package, it can be obtained in the list named "Skill Pattern Probabilities" by calling the *summary()* function. Figure 7 presents a bar plot of the attribute prevalence, and Figure 8 presents a pie chart and a doughnut chart of the latent class proportions.

```
R> GDINA::extract(GDINA.est, what = "prevalence")
$all
      Level0  Level1
A1 0.6167223 0.3832777
A2 0.4565763 0.5434237
A3 0.3321875 0.6678125
R> GDINA::extract(GDINA.est, what = "posterior.prob")
      000      100      010      001      110
[1,] 0.3007218 0.008738675 0.01194014 0.1289744 0.01078687
      101      011      111
[1,] 0.01814145 0.1750859 0.3456107
```

Figure 9 presents a network plot showing both the tetrachoric correlations among attributes and the attribute prevalence. In particular, the tetrachoric correlations are displayed on the arrows between corresponding attributes, and the attribute prevalence is represented using pie charts for each attribute.

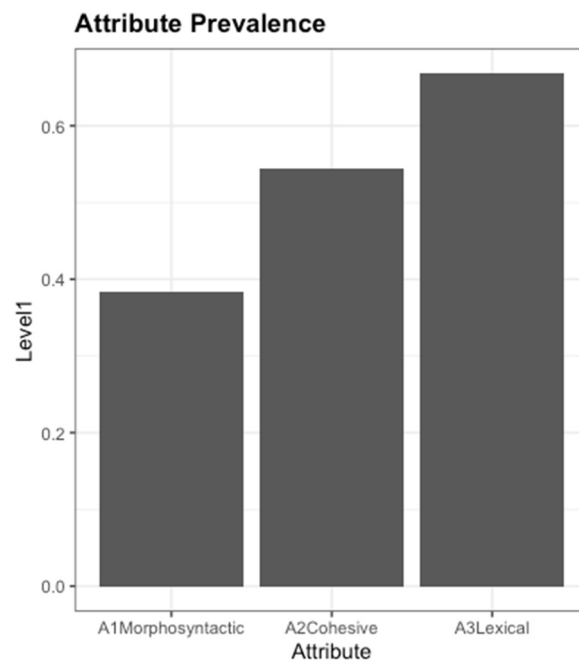


Figure 7. The bar plot of attribute prevalence.

Latent Class Proportions

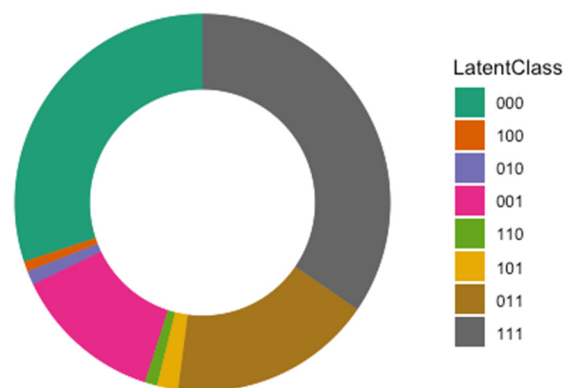


Figure 8. The doughnut chart of latent class proportions.

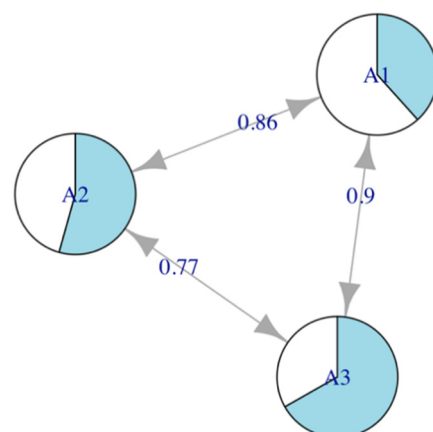


Figure 9. Network plot for attribute correlations and prevalence.

The attribute prevalence and latent class proportions can be plotted together as presented in Figure 10, similar to Bradshaw et al. [2]. The code for creating the plots in Figures 7–10 was written by the authors and can be requested from the first author of the article.

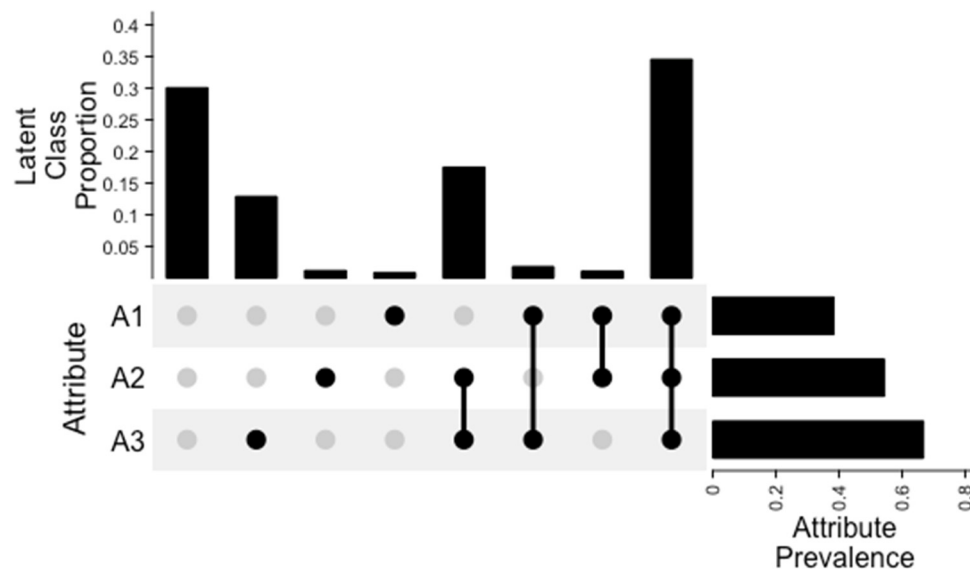


Figure 10. Attribute prevalence and latent class proportions.

5. Discussion

The purpose of this study is to provide a hands-on example of conducting CDM analysis in the G-DINA framework using R packages and illustrating how different R packages can be used in an integrated manner, providing richer information for cognitive diagnosis. Utilizing an exemplary dataset, the study demonstrated a workflow of CDM analyses, from Q-matrix validation to classification visualization. Such an illustration will be helpful to researchers who plan to conduct CDM analysis in R. However, only a limited number of relevant procedures were discussed because of their availability in existing R packages. Other procedures that are equally if not more critical can often be found in the literature.

Despite the potential usefulness, the procedures discussed in this paper may not always work well. For example, the M_2 statistic and $RMSEA_2$ from the **GDINA** package may not be calculable if the number of parameters is too large. The $S-\chi^2$ item fit statistic from the **CDM** package would also not be calculated if there are missing data. In addition, although it was shown that the **CDM** and **GDINA** packages could complement each other in various aspects, researchers need to proceed with caution when using them together. Separate data calibrations may produce different parameter estimates due to the fact that (1) the EM algorithm may reach local maxima or (2) different default settings are specified for different packages. Therefore, this paper shows how to obtain equivalent calibration results by fixing parameter estimates obtained from one package in the other. Doing so, however, may lead to incorrect calculation of the number of free parameters and consequently affect the calculation of other statistics, such as information criteria.

Finally, it should be emphasized that this paper only focuses on the CDM analysis of dichotomous response data using the G-DINA model. However, researchers can do more than that in R. For example, the **CDM** package can also handle the general diagnostic model and regularized latent class model, while the **GDINA** package can handle several CDMs for multiple strategies. Both can also run CDMs for polytomous attributes and polytomous responses. Also, The **NPCD** and **ACTCD** [49,85] packages can conduct nonparametric cognitive diagnostic analysis.

Author Contributions: Conceptualization, Q.S. and W.M.; methodology, Q.S., W.M., A.R. and M.A.S.; formal analysis, Q.S., W.M. and M.A.S.; writing—original draft preparation, Q.S. and W.M.; writing—review and editing, Q.S., W.M., A.R., M.A.S. and K.M.; visualization, Q.S. and W.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: [<https://CRAN.R-project.org/package=GDINA>], accessed on 4 December 2021.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ma, W.; Minchen, N.; de la Torre, J. Choosing between CDM and unidimensional IRT: The proportional reasoning test case. *Measurement* **2020**, *18*, 87–96. [[CrossRef](#)]
2. Bradshaw, L.; Izsák, A.; Templin, J.; Jacobson, E. Diagnosing teachers' understandings of rational numbers: Building a multidimensional test within the diagnostic classification framework. *Educ. Meas.* **2014**, *33*, 2–14. [[CrossRef](#)]
3. Wang, S.; Yang, Y.; Culpepper, S.A.; Douglas, J.A. Tracking skill acquisition with cognitive diagnosis models: A higher-order, hidden Markov model with covariates. *J. Educ. Behav. Stat.* **2018**, *43*, 57–87. [[CrossRef](#)]
4. George, A.C.; Robitzsch, A. Validating theoretical assumptions about reading with cognitive diagnosis models. *Int. J. Test.* **2021**, *21*, 105–129. [[CrossRef](#)]
5. Sorrel, M.A.; Olea, J.; Abad, F.J.; de la Torre, J.; Aguado, D.; Lievens, F. Validity and reliability of Situational Judgement Test scores: A new approach based on cognitive diagnosis models. *Organ. Res. Methods* **2016**, *19*, 506–532. [[CrossRef](#)]
6. Templin, J.L.; Henson, R.A. Measurement of psychological disorders using cognitive diagnosis models. *Psychol. Methods* **2006**, *11*, 287–305. [[CrossRef](#)]
7. De la Torre, J.; van der Ark, L.A.; Rossi, G. Analysis of clinical data from a cognitive diagnosis modeling framework. *Meas. Eval. Couns. Dev.* **2018**, *51*, 281–296. [[CrossRef](#)]
8. Muthén, L.; Muthén, B. *Mplus. Version 8*; Muthén & Muthén: Los Angeles, CA, USA, 1998–2017.
9. Templin, J.; Hoffman, L. Obtaining diagnostic classification model estimates using Mplus. *Educ. Meas.* **2013**, *32*, 37–50. [[CrossRef](#)]
10. Plummer, M. JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. In Proceedings of the 3rd International Workshop on Distributed Statistical Computing, Vienna, Austria, 20–22 March 2003; p. 124.
11. Zhan, P.; Jiao, H.; Man, K.; Wang, L. Using JAGS for Bayesian cognitive diagnosis modeling: A tutorial. *J. Educ. Behav. Stat.* **2019**, *44*, 473–503. [[CrossRef](#)]
12. Carpenter, B.; Gelman, A.; Hoffman, M.D.; Lee, D.; Goodrich, B.; Betancourt, M.; Brubaker, M.A.; Guo, J.; Li, P.; Riddell, A. Stan: A probabilistic programming language. *J. Stat. Soft.* **2017**, *76*, 1–32. [[CrossRef](#)]
13. Jiang, Z.; Carter, R. Using Hamiltonian Monte Carlo to estimate the log-linear cognitive diagnosis model via Stan. *Behav. Res. Methods* **2019**, *51*, 651–662. [[CrossRef](#)] [[PubMed](#)]
14. George, A.C.; Robitzsch, A.; Kiefer, T.; Groß, J.; Ünlü, A. The R package CDM for cognitive diagnosis models. *J. Stat. Soft.* **2016**, *74*, 1–24. [[CrossRef](#)]
15. Ma, W.; de la Torre, J. GDINA: An R package for cognitive diagnosis modeling. *J. Stat. Soft.* **2020**, *93*, 1–26. [[CrossRef](#)]
16. De la Torre, J. The generalized DINA model framework. *Psychometrika* **2011**, *76*, 179–199. [[CrossRef](#)]
17. Nichols, P.D.; Chipman, S.F.; Brennan, R.L. (Eds.) *Cognitively Diagnostic Assessment*; Erlbaum: Hillsdale, NJ, USA, 1995.
18. Tjoe, H.; de la Torre, J. The identification and validation process of proportional reasoning attributes: An application of a cognitive diagnosis modeling framework. *Math. Ed. Res. J.* **2014**, *26*, 237–255. [[CrossRef](#)]
19. Von Davier, M. A general diagnostic model applied to language testing data. *Br. J. Math. Stat. Psychol.* **2008**, *61*, 287–307. [[CrossRef](#)] [[PubMed](#)]
20. Haertel, E.H. Using restricted latent class models to map the skill structure of achievement items. *J. Educ. Meas.* **1989**, *26*, 301–321. [[CrossRef](#)]
21. Junker, B.W.; Sijtsma, K. Cognitive assessment models with few assumptions, and connections with nonparametric Item Response Theory. *Appl. Psychol. Meas.* **2001**, *25*, 258–272. [[CrossRef](#)]
22. De la Torre, J.; Douglas, J.A. Higher-order latent trait models for cognitive diagnosis. *Psychometrika* **2004**, *69*, 333–353. [[CrossRef](#)]
23. Maris, E. Estimating multiple classification latent class models. *Psychometrika* **1999**, *64*, 187–212. [[CrossRef](#)]
24. Hartz, S.M. A Bayesian Framework for the Unified Model for Assessing Cognitive Abilities: Blending Theory with Practicality. *Diss. Abstr. Int. B Sci. Eng.* **2002**, *63*, 864.
25. Leighton, J.P.; Gierl, M.J. *Cognitive Diagnostic Assessment for Education: Theory and Applications*; Cambridge University Press: Cambridge, UK, 2007.
26. Henson, R.A.; Templin, J.L.; Willse, J.T. Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika* **2009**, *74*, 191–210. [[CrossRef](#)]

27. Nájera, P.; Abad, F.J.; Sorrel, M.A. Determining the number of attributes in cognitive diagnosis modeling. *Front. Psychol.* **2021**, *12*, 321. [[CrossRef](#)]
28. De la Torre, J.; Chiu, C.-Y. A general method of empirical Q-matrix validation. *Psychometrika* **2016**, *81*, 253–273. [[CrossRef](#)] [[PubMed](#)]
29. Ma, W.; de la Torre, J. An empirical Q-matrix validation method for the sequential generalized DINA model. *Br. J. Math. Stat. Psychol.* **2020**, *73*, 142–163. [[CrossRef](#)] [[PubMed](#)]
30. Nájera, P.; Sorrel, M.A.; Abad, F.J. Reconsidering cutoff points in the general method of empirical Q-Matrix validation. *Educ. Psychol. Meas.* **2019**, *79*, 727–753. [[CrossRef](#)]
31. Nájera, P.; Sorrel, M.A.; de la Torre, J.; Abad, F.J. Improving robustness in Q-Matrix validation using an iterative and dynamic procedure. *Appl. Psychol. Meas.* **2020**, *44*, 431–446. [[CrossRef](#)]
32. De la Torre, J.; Minchen, N.D. The G-DINA model framework. In *Handbook of Diagnostic Classification Models*; von Davier, M., Lee, Y.-S., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 155–169.
33. De la Torre, J.; Lee, Y.-S. Evaluating the Wald Test for item-level comparison of saturated and reduced models in cognitive diagnosis. *J. Educ. Meas.* **2013**, *50*, 355–373. [[CrossRef](#)]
34. Ma, W.; Iaconangelo, C.; de la Torre, J. Model similarity, model selection, and attribute classification. *Appl. Psychol. Meas.* **2016**, *40*, 200–217. [[CrossRef](#)]
35. Sorrel, M.A.; de la Torre, J.; Abad, F.J.; Olea, J. Two-step likelihood ratio test for item-level model comparison in cognitive diagnosis models. *Methodology* **2017**, *13*, 39–47. [[CrossRef](#)]
36. Robitzsch, A.; George, A.C. The R package CDM for diagnostic modeling. In *Handbook of Diagnostic Classification Models*; von Davier, M., Lee, Y.-S., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 549–572.
37. Robitzsch, A. Regularized latent class analysis for polytomous item responses: An application to SPM-LS data. *J. Intell.* **2020**, *8*, 30. [[CrossRef](#)]
38. Ma, W.; Jiang, Z. Estimating cognitive diagnosis models in small samples: Bayes modal estimation and monotonic constraints. *Appl. Psychol. Meas.* **2021**, *45*, 95–111. [[CrossRef](#)]
39. Maydeu-Olivares, A.; Joe, H. Assessing approximate fit in categorical data analysis. *Multivariate Behav. Res.* **2014**, *49*, 305–328. [[CrossRef](#)] [[PubMed](#)]
40. Liu, Y.; Tian, W.; Xin, T. An application of M_2 statistic to evaluate the fit of cognitive diagnostic models. *J. Educ. Behav. Stat.* **2016**, *41*, 3–26. [[CrossRef](#)]
41. Sorrel, M.A.; Abad, F.J.; Olea, J.; de la Torre, J.; Barrada, J.R. Inferential item-fit evaluation in cognitive diagnosis modeling. *Appl. Psychol. Meas.* **2017**, *41*, 614–631. [[CrossRef](#)]
42. Ma, W. Cognitive diagnosis modeling using the GDINA R package. In *Handbook of Diagnostic Classification Models*; von Davier, M., Lee, Y.-S., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 593–601.
43. Balamuta, J.J.; Culpepper, S.A.; Douglas, J.A. edmdata: Data Sets for Psychometric Modeling; R Package Version 1.2.0; 2021. Available online: <https://CRAN.R-project.org/package=edmdata> (accessed on 9 September 2021).
44. Nájera, P.; Sorrel, M.A.; Abad, F.J. cdmTools: Useful Tools for Cognitive Diagnosis Modeling; R Package Version 1.0.0. 2021. Available online: <https://CRAN.R-project.org/package=cdmTools> (accessed on 14 September 2021).
45. Garrido, L.E.; Abad, F.J.; Ponsoda, V. A new look at Horn's parallel analysis with ordinal variables. *Psychol. Methods* **2013**, *18*, 454–474. [[CrossRef](#)] [[PubMed](#)]
46. Wang, W.; Song, L.; Ding, S. An exploratory discrete factor loading method for Q-Matrix specification in cognitive diagnostic models. In *Springer Proceedings in Mathematics & Statistics, Quantitative Psychology, IMPS, 2017*; Wiberg, M., Culpepper, S., Janssen, R., González, J., Molenaar, D., Eds.; Springer: Cham, Switzerland, 2018; Volume 233. [[CrossRef](#)]
47. Nájera, P.; Sorrel, M.A.; de la Torre, J.; Abad, F.J. Balancing fit and parsimony to improve Q-matrix validation. *Br. J. Math. Stat. Psychol.* **2021**, *74*, 110–130. [[CrossRef](#)]
48. De la Torre, J. An empirically based method of Q-matrix validation for the DINA model: Development and applications. *J. Educ. Meas.* **2008**, *45*, 343–362. [[CrossRef](#)]
49. Zheng, Y.; Chiu, C.-Y. NPCD: Nonparametric Methods for Cognitive Diagnosis; R Package Version 1.0-11. 2019. Available online: <https://CRAN.R-project.org/package=NPCD> (accessed on 9 September 2021).
50. Chiu, C.-Y. Statistical refinement of the Q-matrix in cognitive diagnosis. *Appl. Psychol. Meas.* **2013**, *37*, 598–618. [[CrossRef](#)]
51. De la Torre, J.; Ma, W. Cognitive diagnosis modeling: A general framework approach and its implementation in R. In *Proceedings of the Fourth Conference on Statistical Methods in Psychometrics*, Columbia University, New York, NY, USA, 30 August 2016.
52. Liu, J.; Xu, G.; Ying, Z. Data-driven learning of Q-matrix. *Appl. Psychol. Meas.* **2012**, *36*, 548–564. [[CrossRef](#)]
53. Chen, Y.; Liu, J.; Xu, G.; Ying, Z. Statistical analysis of Q-matrix based diagnostic classification models. *J. Am. Stat. Assoc.* **2015**, *110*, 850–866. [[CrossRef](#)] [[PubMed](#)]
54. Chen, Y.; Culpepper, S.A.; Chen, Y.; Douglas, J. Bayesian estimation of the DINA Q. *Psychometrika* **2018**, *83*, 89–108. [[CrossRef](#)]
55. Xu, G.; Shang, Z. Identifying latent structures in restricted latent class models. *J. Am. Stat. Assoc.* **2018**, *113*, 1284–1295. [[CrossRef](#)]
56. Xu, X.; von Davier, M. *Fitting the Structured General Diagnostic Model to NAEP Data (RR-08-27)*; ETS Research Report Series; ETS: Princeton, NJ, USA, 2008.
57. Zhang, C.-H. Nearly unbiased variable selection under minimax concave penalty. *Ann. Stat.* **2010**, *38*, 894–942. [[CrossRef](#)]

58. Maydeu-Olivares, A.; Joe, H. Limited information goodness-of-fit testing in multidimensional contingency tables. *Psychometrika* **2006**, *71*, 713–732. [[CrossRef](#)]
59. Hansen, M.; Cai, L.; Monroe, S.; Li, Z. Limited-information goodness-of-fit testing of diagnostic classification item response models. *Br. J. Math. Stat. Psychol.* **2016**, *69*, 225–252. [[CrossRef](#)] [[PubMed](#)]
60. Hu, J.; Miller, M.D.; Huggins-Manley, A.C.; Chen, Y.-H. Evaluation of model fit in cognitive diagnosis models. *Int. J. Test.* **2016**, *16*, 119–141. [[CrossRef](#)]
61. Maydeu-Olivares, A. Goodness-of-fit assessment of item response theory models. *Measurement* **2013**, *11*, 71–137. [[CrossRef](#)]
62. Ma, W. Evaluating the fit of sequential G-DINA model using limited-information measures. *Appl. Psychol. Meas.* **2020**, *44*, 167–181. [[CrossRef](#)]
63. Liu, R.; Huggins-Manley, A.C.; Bulut, O. Retrofitting diagnostic classification models to responses from IRT-based assessment forms. *Educ. Psychol. Meas.* **2018**, *78*, 357–383. [[CrossRef](#)]
64. Chen, J.; de la Torre, J.; Zhang, Z. Relative and absolute fit evaluation in cognitive diagnosis modeling. *J. Educ. Meas.* **2013**, *50*, 123–140. [[CrossRef](#)]
65. DiBello, L.V.; Roussos, L.A.; Stout, W.F. Review of cognitively diagnostic assessment and a summary of psychometric models. In *Handbook of Statistics*; Rao, C.R., Sinharay, S., Eds.; Elsevier: Amsterdam, The Netherlands, 2007; pp. 979–1030.
66. McDonald, R.P.; Mok, M.M.-C. Goodness of fit in item response models. *Multivariate Behav. Res.* **1995**, *30*, 23–40. [[CrossRef](#)] [[PubMed](#)]
67. Chen, W.; Thissen, D. Local dependence indexes for item pairs using item response theory. *J. Educ. Behav. Stat.* **1997**, *22*, 265–289. [[CrossRef](#)]
68. Xue, Z.; Juntao, W. On the sequential hierarchical cognitive diagnostic model. *Front. Psychol.* **2020**, *11*, 2562.
69. Akaike, H. A new look at the statistical identification model. *IEEE Trans. Automat. Contr.* **1974**, *19*, 716–723. [[CrossRef](#)]
70. Schwarz, G. Estimating the dimension of a model. *Ann. Stat.* **1978**, *6*, 461–464. [[CrossRef](#)]
71. Orlando, M.; Thissen, D. Likelihood-based item-fit indices for dichotomous item response theory models. *Appl. Psychol. Meas.* **2000**, *24*, 50–64. [[CrossRef](#)]
72. Kunina-Habenicht, O.; Rupp, A.A.; Wilhelm, O. A practical illustration of multidimensional diagnostic skills profiling: Comparing results from confirmatory factor analysis and diagnostic classification models. *Stud. Educ. Eval.* **2009**, *35*, 64–70. [[CrossRef](#)]
73. Oliveri, M.E.; von Davier, M. Investigation of model fit and score scale comparability in international assessments. *Psychol. Test Assess. Model.* **2011**, *53*, 315–333.
74. Yamamoto, K.; Khorramdel, L.; von Davier, M. Scaling PIAAC cognitive data. In *Technical Report of the Survey of Adults Skills (PIAAC)*; Organisation for Economic Co-operation and Development(OECD), Ed.; OECD: Paris, France, 2013.
75. Kaplan, M.; de la Torre, J.; Barrada, J.R. New item selection methods for cognitive diagnosis computerized adaptive testing. *Appl. Psychol. Meas.* **2015**, *39*, 167–188. [[CrossRef](#)]
76. Henson, R.; DiBello, L.; Stout, B. A generalized approach to defining item discrimination for DCMs. *Measurement* **2018**, *16*, 18–29. [[CrossRef](#)]
77. Cui, Y.; Gierl, M.J.; Chang, H.-H. Estimating classification consistency and accuracy for cognitive diagnostic assessment. *J. Educ. Meas.* **2012**, *49*, 19–38. [[CrossRef](#)]
78. Johnson, M.S.; Sinharay, S. Measures of agreement to assess attribute-level classification accuracy and consistency for cognitive diagnostic assessments. *J. Educ. Meas.* **2018**, *45*, 635–664. [[CrossRef](#)]
79. Chen, Y.; Liu, Y.; Xu, S. Mutual information reliability for latent class analysis. *Appl. Psychol. Meas.* **2018**, *42*, 460–477. [[CrossRef](#)] [[PubMed](#)]
80. Iaconangelo, C. Uses of Classification Error Probabilities in the Three-Step Approach to Estimating Cognitive Diagnosis Models. Doctoral Dissertation, Rutgers University, New Brunswick, NJ, USA, 2017, unpublished. [[CrossRef](#)]
81. Wang, W.; Song, L.; Chen, P.; Meng, Y.; Ding, S. Attribute-level and pattern-level classification consistency and accuracy indices for cognitive diagnostic assessment. *J. Educ. Meas.* **2015**, *52*, 457–476. [[CrossRef](#)]
82. Sinharay, S.; Johnson, M.S. Measures of agreement: Reliability, classification accuracy, and classification consistency. In *Handbook of Diagnostic Classification Models*; von Davier, M., Lee, Y.-S., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 359–377. [[CrossRef](#)]
83. Huebner, A.; Wang, C. A note on comparing examinee classification methods for cognitive diagnosis models. *Educ. Psychol. Meas.* **2011**, *71*, 407–419. [[CrossRef](#)]
84. Nakazawa, M. fmsb: Functions for Medical Statistics Book with Some Demographic Data; R Package Version 0.7.1. 2021. Available online: <https://CRAN.R-project.org/package=fmsb> (accessed on 14 September 2021).
85. Chiu, C.-Y.; Ma, W. ACTCD: Asymptotic Classification Theory for Cognitive Diagnosis; R Package Version 1.2-0. 2018. Available online: <https://CRAN.R-project.org/package=ACTCD> (accessed on 9 September 2021).