A complete latent profile analysis workflow with the latents R package

Mohammed Saqr, University of Eastern Finland
Sonsoles López-Pernas, University of Eastern Finland

26 November 2026

Contents

Data 2
Selecting the covariance structure and the number of profiles 2
Estimation 5
Interpretation of the profiles 5
Classification quality 9
Posterior assignments 12
Stability across starting values 12
Covariance between chosen indicators 13
Further analyses 14
References 14

Latent profile analysis (LPA) is a person-centred method for identifying unobserved subgroups in data measured on continuous indicators. Whereas variable-centred methods, such as correlation and regression, describe how indicators are related on average across the whole sample, LPA assumes that the sample consists of a small number of unobserved subpopulations, referred to as profiles, each characterized by its own mean on every indicator. The model estimates the means, variances and, where specified, covariances of each profile, the proportion of the sample that belongs to each profile, and the posterior probability with which each observation belongs to each profile. Statistically, LPA is a finite Gaussian mixture model, and the two terms refer to the same model (Scrucca et al., 2024).

An LPA involves a sequence of decisions, each of which affects the interpretation of the final solution: the covariance structure of the profiles, the number of profiles, the estimation from multiple starting values, the interpretation of the profiles, and the evaluation of the resulting classification (Spurk et al., 2020). This vignette presents these steps in order, from the description of the data to the stability of the solution.

```
library(latents)
set.seed(1)
```

The estimation uses random starting values; the seed is therefore set once so that the results can be reproduced.

Data

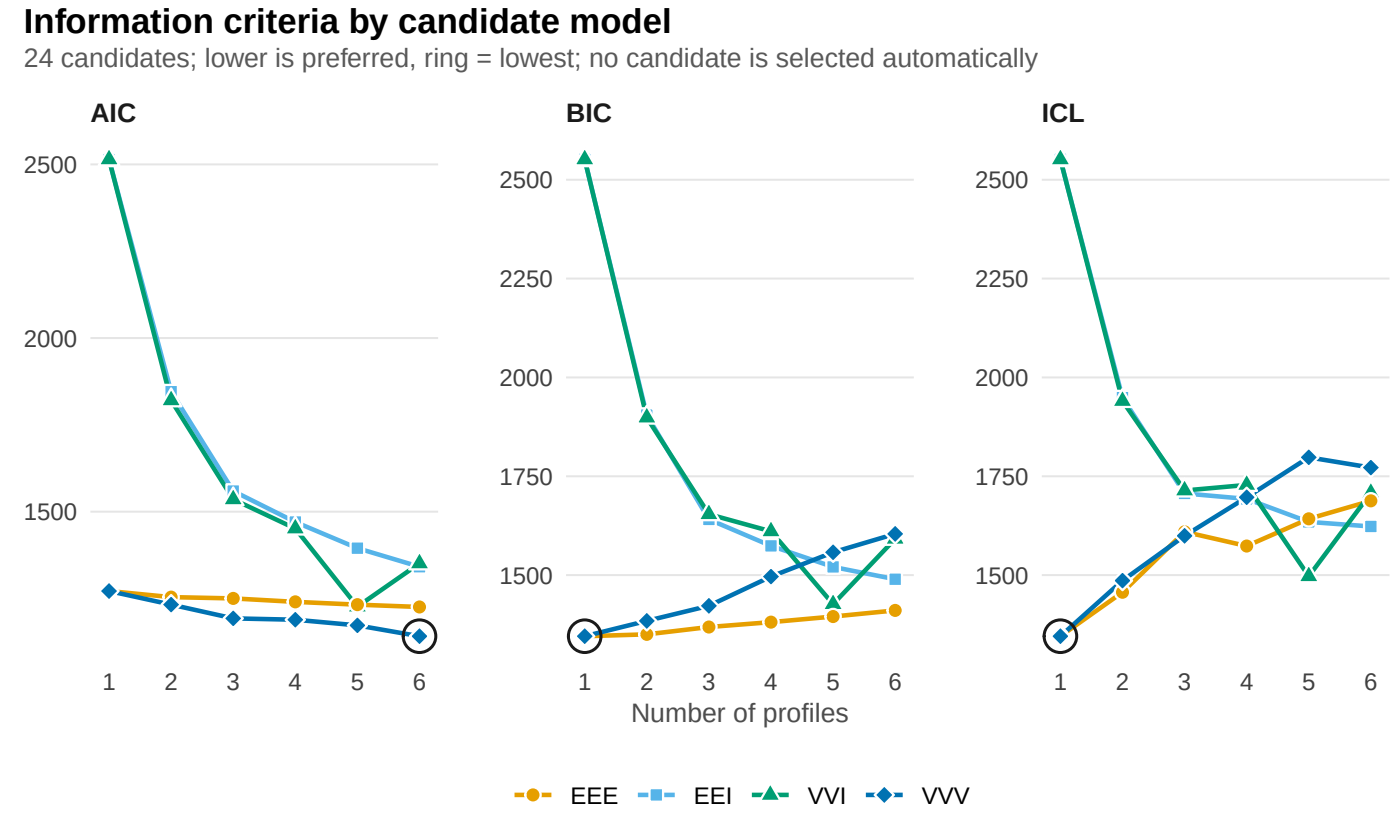
The `srl` data contain the scores of 300 respondents on five constructs of the Motivated Strategies for Learning Questionnaire: cognitive strategy use, intrinsic value, self-efficacy, self-regulation, and test anxiety. Each score is the mean of the construct's items on a scale from 1 to 7. The responses were generated by a large language model (GPT) in a study of the structure of self-regulated learning in machine-generated survey responses (Vogelsmeier et al., 2025). They serve here as a well-structured example and do not constitute evidence about students. Four of the constructs are strongly correlated (between approximately 0.78 and 0.86), whereas test anxiety is weakly and negatively related to them (`?srl`); this structure is central to the choice of the covariance structure below.

```
indicators <- c("cognitive_strategies", "intrinsic_value", "self_efficacy",
                "self_regulation", "test_anxiety")
descriptives(srl, indicators)
#>               variable   n n_missing mean    sd  min  max n_distinct
#> 1 cognitive_strategies 300         0 5.71 0.541 4.15 7.00         34
#> 2      intrinsic_value 300         0 6.12 0.523 4.22 7.00         22
#> 3        self_efficacy 300         0 5.93 0.587 4.00 7.00         25
#> 4      self_regulation 300         0 5.54 0.500 3.78 6.78         25
#> 5         test_anxiety 300         0 3.70 0.644 2.00 5.75         16
```

Selecting the covariance structure and the number of profiles

In addition to the number of profiles, a latent profile model must specify how the indicators vary within a profile: whether the variances are equal across profiles or differ between them, and whether the indicators are allowed to covary within a profile. The four combinations are named by the volume, shape and orientation of each profile's covariance matrix, each equal (`E`) or varying (`V`) across profiles, with `I` for no covariances (Celeux & Govaert, 1995): `EEI` has equal variances and no covariances, `VVI` varying variances and no covariances, `EEE` one covariance matrix shared by all profiles, and `VVV` a covariance matrix for each profile. `enumerate_lpa()` estimates these four structures with every number of profiles requested.

```r
models <- enumerate_lpa(srl, indicators, n_profiles = 1:6)
#> Warning: A variance reached min_variance; this is a bound-active constrained fit.
#> Warning: Profile 4: 9.8 Gaussian membership mass for 10 free means and variances. This is
#> a conservative diagnostic, not an identification test; inspect the Gaussian covariance
#> diagnostics.
#> Warning: Profile 2: 12.4 Gaussian membership mass for 20 free means and variances. This
#> is a conservative diagnostic, not an identification test; inspect the Gaussian covariance
#> diagnostics.
#> 3 of 24 candidates reached their best log likelihood from one start only (profiles5_VVI, profiles4_VVV, prof
plot_enumeration(models)
```



The choice of structure is consequential for these data because four of the five constructs are strongly correlated. The structures without covariances (`EEI`, `VVI`) assume that the indicators are unrelated within a profile and can therefore reproduce these correlations only by introducing additional profiles, which then describe the correlations rather than distinct groups of respondents. BIC falls through six profiles for `EEI`. For `VVI`, it is lowest at five profiles, but that fit reaches the variance floor; BIC rises again at six profiles. The warnings printed during the enumeration report candidates with low Gaussian membership mass relative to the number of profile-specific means and variances (`latents_thin_profiles`). This is a conservative support diagnostic, not an identification test; inspect the covariance diagnostics as well. The `EEE` structure shares one covariance matrix and does not receive this profile-specific warning. The structures with covariances account for the correlations directly, and reach far lower BIC values with one or two profiles. The `EEE` structure attains the lowest values; its candidates are:

```r
summary(enumerate_lpa(srl, indicators, n_profiles = 1:6, model = "EEE"))
#> 1 of 6 candidates reached their best log likelihood from one start only (profiles4); refit those you consider
#> Class enumeration: 6 candidates, 6 converged, 0 failed to fit
#> 2 distinct candidate(s) are minimal under some criterion.
#> No candidate is selected automatically. Choose one convention and keep it.
#>
#> ! 1 of 6 candidates reached their best log likelihood from one start only
#>     (profiles4); refit those you consider with more starts.
#> ! 2 candidates have a class with average posterior below 0.8 (profiles3,
#>     profiles6).
#>
#> -- lowest value of each criterion ---------------------------------
#>  criterion  convention candidate value
#>        aic        <NA> profiles6  1226
#>        kic        <NA> profiles6  1279
#>        bic      groups profiles1  1345
#>        bic individuals profiles1  1345
#>      sabic      groups profiles6  1252
#>      sabic individuals profiles6  1252
#>       caic      groups profiles1  1365
#>       caic individuals profiles1  1365
#>        awe      groups profiles1  1520
#>        awe individuals profiles1  1520
#>        icl      groups profiles1  1345
#>        icl individuals profiles1  1345
#>        clc      groups profiles1  1231
#>        clc individuals profiles1  1231
#>
#> -- knees of bic_groups ----------------------------------------
#>   direction           line knee_candidate knee_without_largest stable
#>    profiles  1 group class           <NA>                 <NA>     NA
#>  parameters all candidates           <NA>                 <NA>     NA
#>
#> Tables: get_candidates(x), get_best(x), get_deltas(x), get_knees(x),
#>         get_tree(x), get_classification(x), get_entropy(x), get_checks(x).
#> Plots:  plot_criteria(x), plot_deltas(x), plot_avepp(x), plot_tree(x), plot_all(x).
```

The BIC (Schwarz, 1978) is lowest for one profile, and the two-profile model is only 4.7 points higher. The criteria that also penalize classification uncertainty (ICL, AWE and CLC) and the CAIC favour one profile as well, whereas the AIC, KIC and sample-size adjusted BIC, whose penalties are weaker, are lowest at six profiles, the largest number fitted. The information criteria therefore do not provide decisive evidence for one or two profiles. Because the one-profile model is the two-profile model with the probability of the second profile fixed at zero, the likelihood ratio statistic does not follow its usual chi-squared distribution; the parametric bootstrap likelihood ratio test approximates its distribution by simulation from the one-profile model (McLachlan, 1987) and was among the most accurate procedures in the simulations of Nylund et al. (2007).

```r
one <- lpa(srl, indicators, n_profiles = 1, model = "EEE")
fit <- lpa(srl, indicators, n_profiles = 2, model = "EEE")
lrt <- bootstrap_lrt(one, fit)
lrt
#> Parametric bootstrap likelihood-ratio comparison
#> Null: 1 profile; alternative: 2 profiles
```

```
#> Observed statistic: 29.518785
#> p-value: 0.005 (Monte Carlo SE 0.00499) from 199 of 199 valid replicates (0 invalid)
```

The test rejects the one-profile model ($p = 0.005$). The two-profile model with the `EEE` structure is therefore retained, with the caveat that the evidence for a second profile rests on the bootstrap test rather than on the information criteria.

Estimation

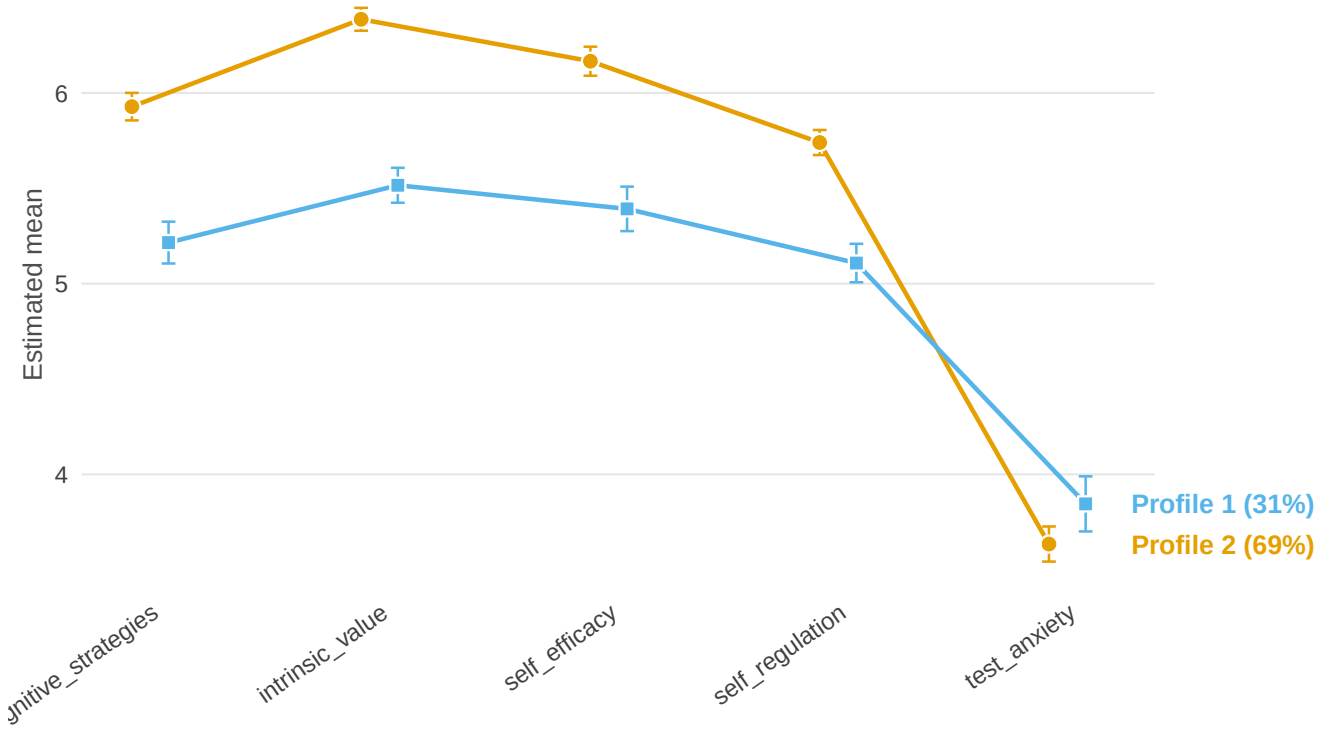
`lpa()` fits a single-level latent profile model, estimated by maximum likelihood with the expectation-maximization algorithm from multiple starting values. The fitted model:

```
fit
#> Latent profile analysis: 2 profiles
#> 300 observations; equal full residual covariance (EEE)
#> Log likelihood: -600.924999 | AIC: 1253.850 | BIC: 1350.148
#> Converged: TRUE | iterations: 18 | best log likelihood reached by 10 of 10 starts (10 converged; emEM: 50 scre
#> Smallest standardized Gaussian eigenvalue: 0.1025 (profiles, profile_1); Gaussian mass: 93.07; weight effec
#>
#> Class sizes   profile_1 31.0% (93)   profile_2 69.0% (207)
#>
#> Means
#>             indicator profile_1 profile_2
#>  cognitive_strategies      5.21      5.93
#>       intrinsic_value      5.52      6.39
#>         self_efficacy      5.39      6.17
#>       self_regulation      5.11      5.74
#>          test_anxiety      3.85      3.63
#>
#> Variances and standard errors: get_results(x, "profiles").
#> Every other table: get_results(x, what = ), or get_results(x, "all").
```

Interpretation of the profiles

The profile plot shows the estimated mean of each profile on each indicator with its 95% confidence interval.

```
plot_profiles(fit)
```



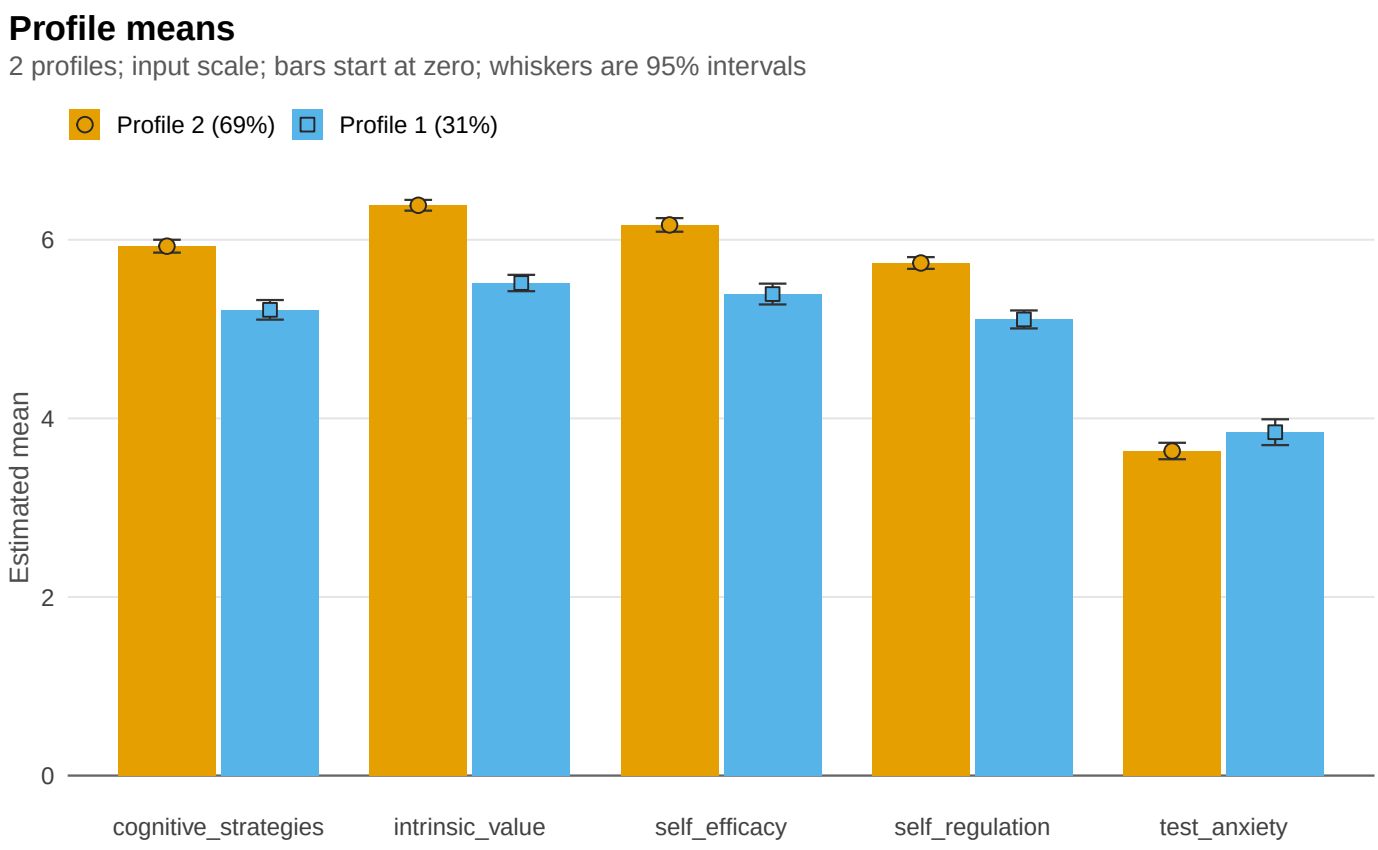
The larger profile, comprising 69% of the respondents, has higher means on cognitive strategy use, intrinsic value, self-efficacy and self-regulation than the smaller profile, comprising 31%. The two profiles differ little in test anxiety. The profiles can be described as high and moderate self-regulated learning. The estimates with their standard errors:

```
get_results(fit, "profiles")
#>    profile            indicator mean variance standard_deviation mean_standard_error
#> 1        1 cognitive_strategies 5.22    0.183              0.427               0.0559
#> 2        1      intrinsic_value 5.52    0.110              0.332               0.0467
#> 3        1        self_efficacy 5.39    0.215              0.464               0.0595
#> 4        1      self_regulation 5.11    0.164              0.405               0.0514
#> 5        1         test_anxiety 3.85    0.404              0.636               0.0737
#> 6        2 cognitive_strategies 5.93    0.183              0.427               0.0368
#> 7        2      intrinsic_value 6.39    0.110              0.332               0.0307
#> 8        2        self_efficacy 6.17    0.215              0.464               0.0389
#> 9        2      self_regulation 5.74    0.164              0.405               0.0336
#> 10       2         test_anxiety 3.63    0.404              0.636               0.0470
#>    variance_standard_error
#> 1                   0.0202
#> 2                   0.0130
#> 3                   0.0223
#> 4                   0.0166
#> 5                   0.0333
#> 6                   0.0202
#> 7                   0.0130
```

```
#> 8                  0.0223
#> 9                  0.0166
#> 10                 0.0333
```

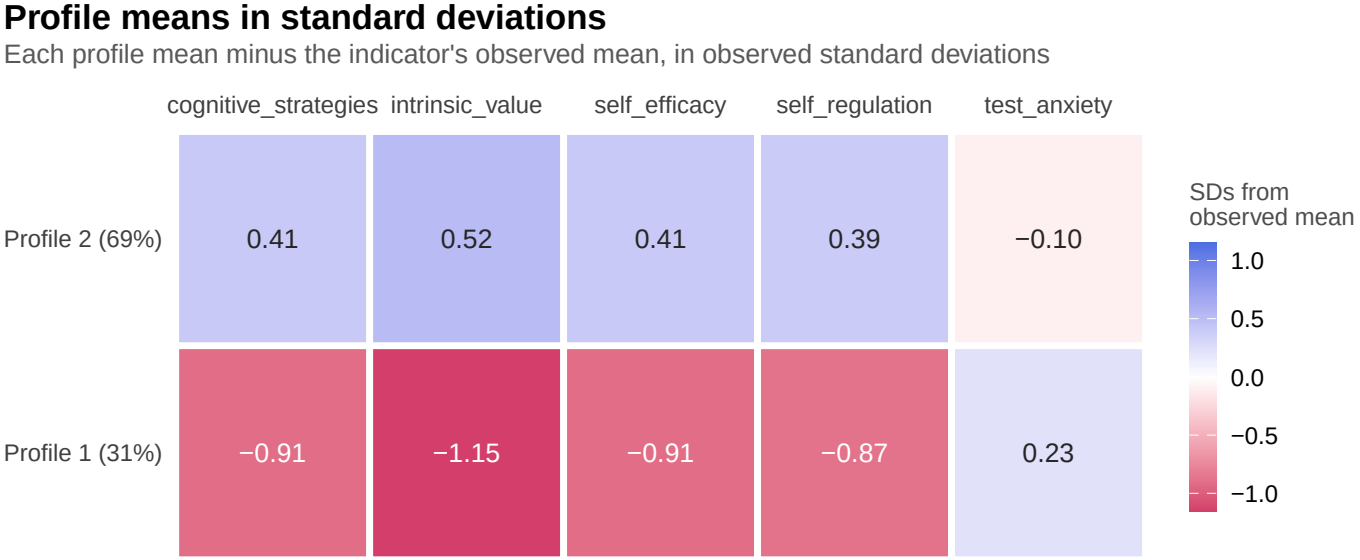
The same means as bars with their confidence intervals:

```
plot_profiles(fit, what = "bars")
```



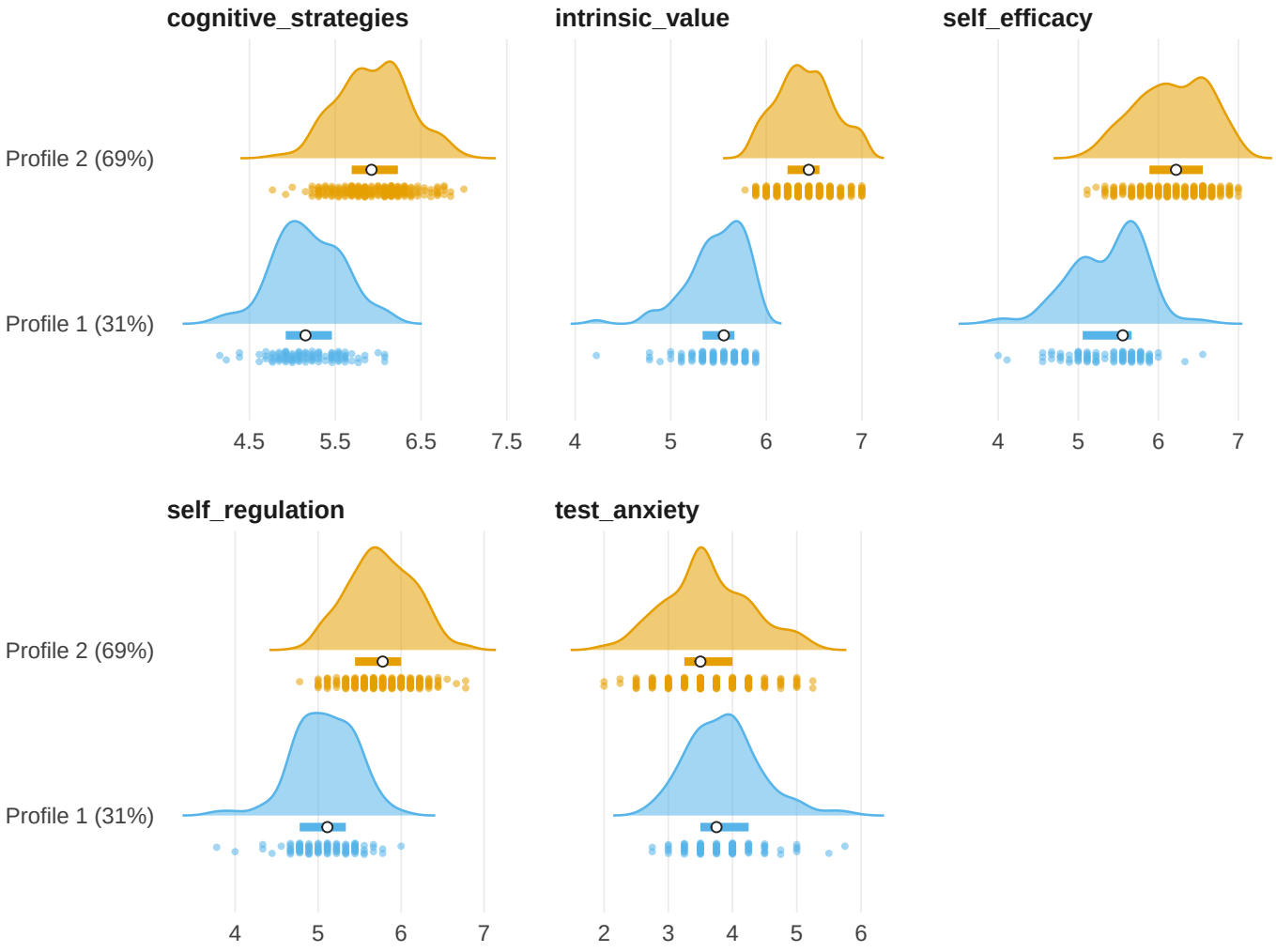
The heatmap expresses each mean as a distance from the indicator's grand mean in standard deviations, which makes the differences comparable across indicators:

```
plot_profiles(fit, what = "heatmap")
```



The means summarize distributions. The raincloud plot shows, for the respondents assigned to each profile, the density of each indicator, a bar from its first to its third quartile with the median marked, and the individual scores. It shows how much the profiles overlap on each indicator: they are best separated on intrinsic value, and overlap almost completely on test anxiety.

```
plot_profiles(fit, what = "raincloud")
```



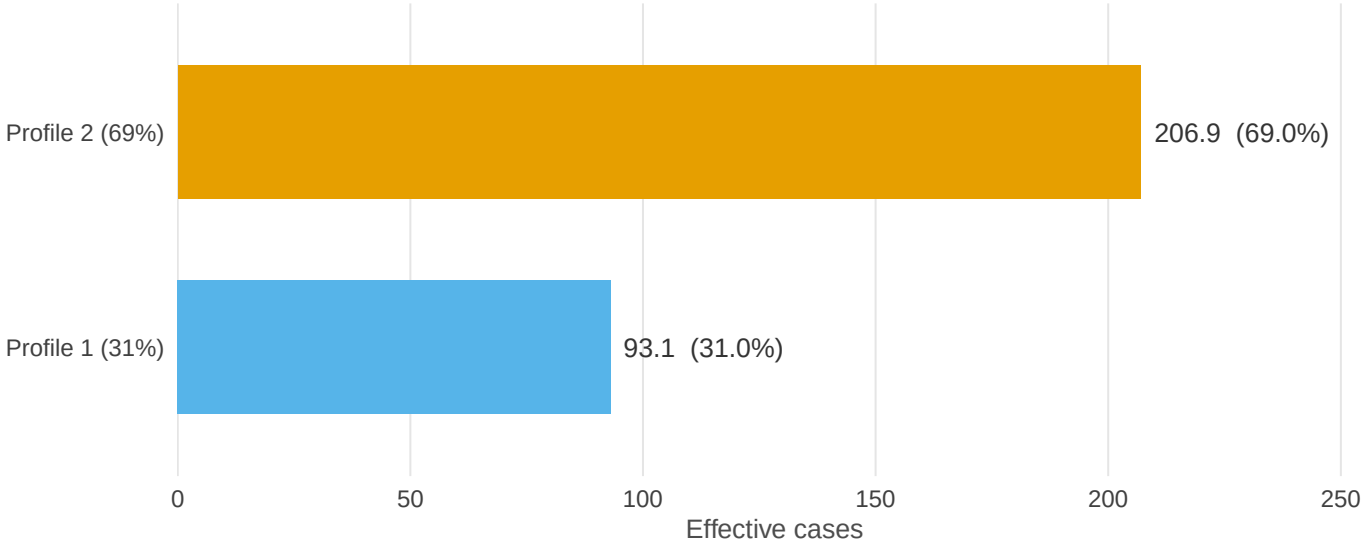
Classification quality

The usefulness of a profile solution depends on how accurately the observations can be assigned to the profiles. Relative entropy summarizes this accuracy across all observations on a scale from 0 to 1 (Ramaswamy et al., 1993), and the average posterior probability of each profile describes it for the observations assigned to that profile. `diagnostics()` reports the relative entropy, the size of the smallest profile, the average posterior probabilities, and the within-profile residual correlation with the smallest adjusted p-value, and draws the four classification plots: the effective size of each profile (the sum of the posterior probabilities of belonging to it), the distribution of the posterior probability of each respondent's assigned profile, each respondent's contribution to the classification entropy, and the average posterior probability matrix, in which each row contains the respondents assigned to one profile and each column the average posterior probability they place on each profile.

```
diagnostics(fit)
```

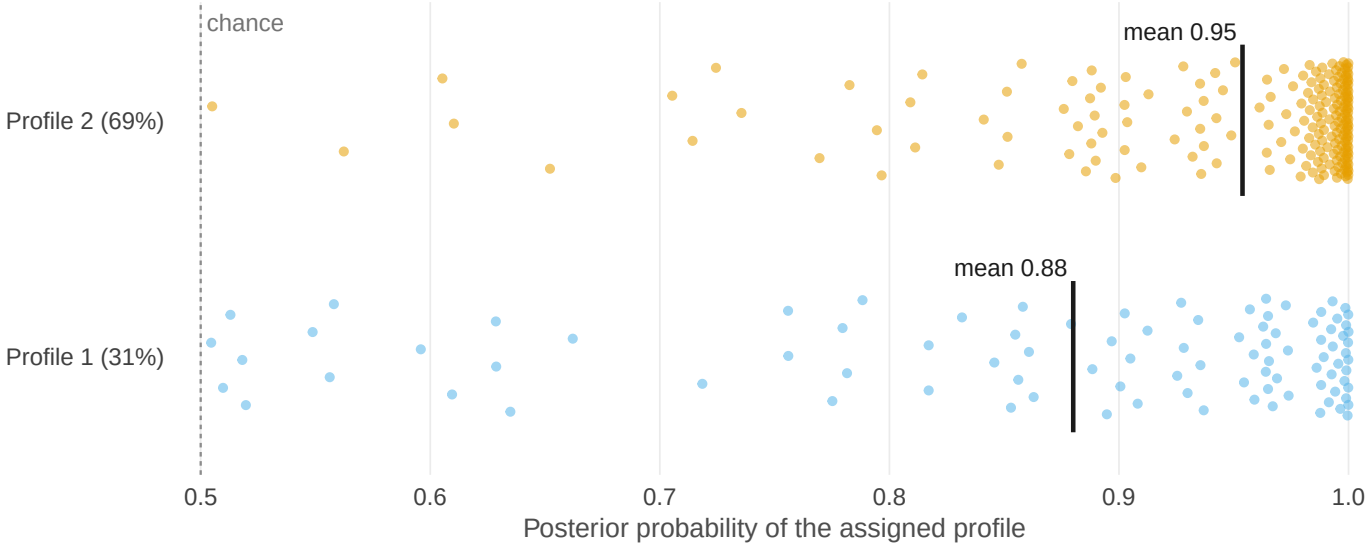
Profile sizes

Effective number of cases (sum of posterior probabilities); 300 in total



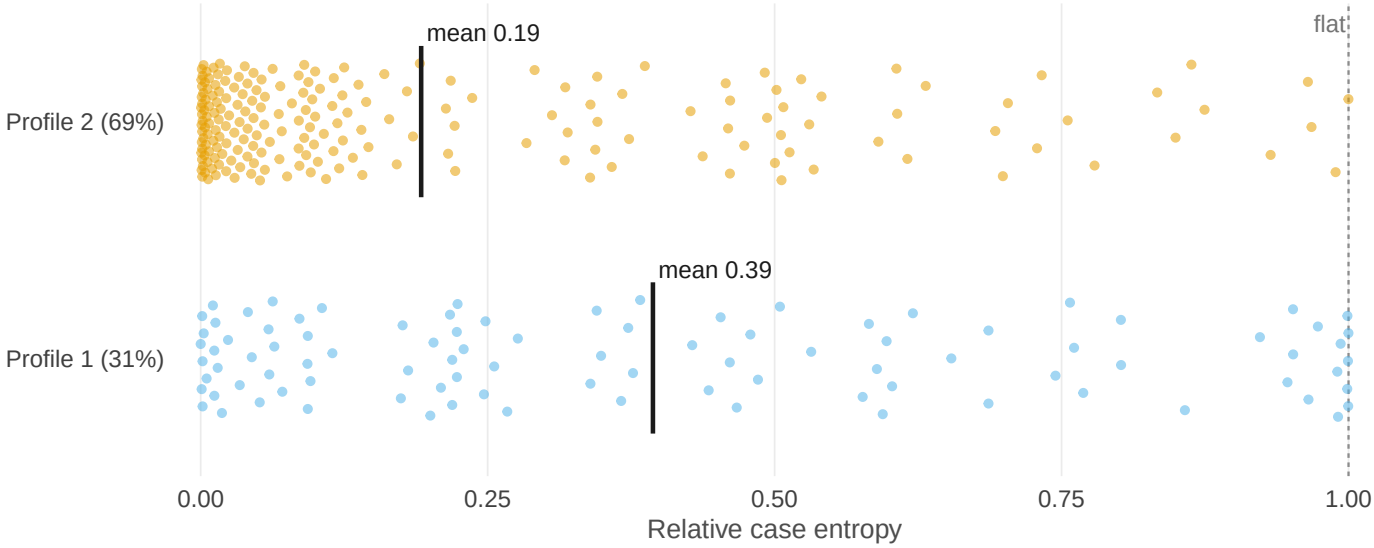
Classification certainty

Posterior probability of each case's assigned profile; the profile mean is its average posterior probability



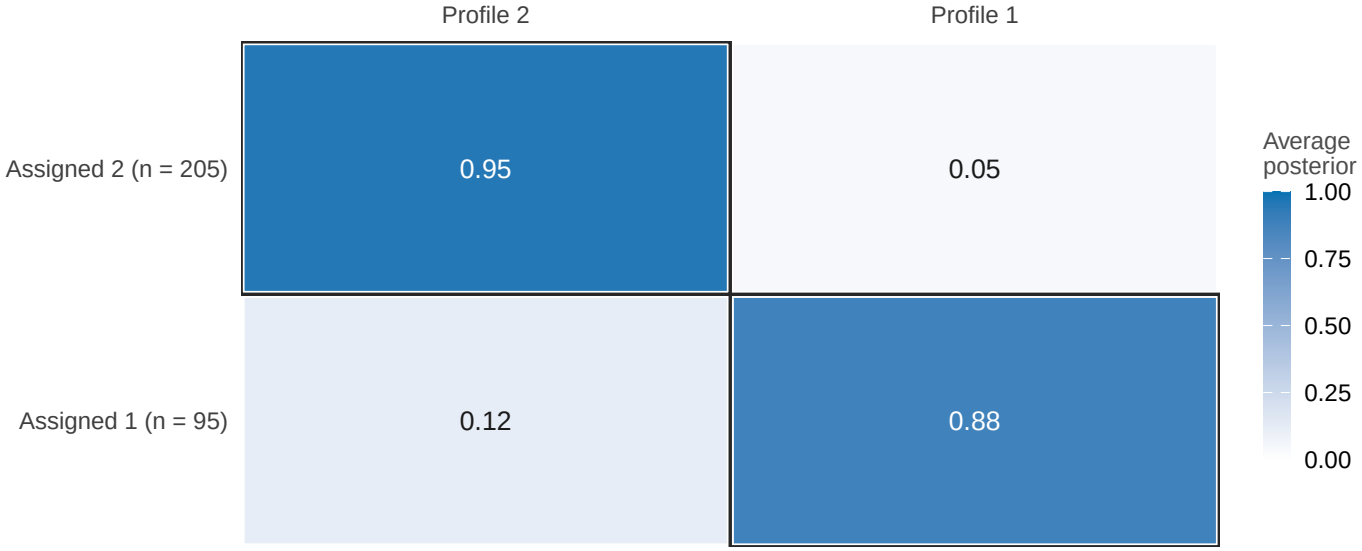
Classification uncertainty

Each case's entropy relative to a flat posterior (0 = certain, 1 = flat); relative entropy = 1 − mean = 0.744



Average posterior probability

Mean posterior each assigned group places on each profile; the outlined diagonal is classification accuracy




```
#> Diagnostics: Latent profile analysis, 2 profiles
#> No check needs attention.
#>
#> Estimation
#>   ok     best start converged                    yes
#>   info   best log likelihood reached by          10 of 10 starts
#>   ok     starts failed or unfinished             0 of 10 (40 screened)
```

```
#>   ok     estimates at a bound                                  no
#>
#> Class size
#>   info   smallest class                          individuals  31.0% (n = 93, class 1)
#>   info   modal vs estimated proportions          individuals  largest gap 0.006
#>   ok     classes with no assigned unit           individuals  0 of 2
#>
#> Classification
#>   info   relative entropy                        individuals  0.744
#>   info   lowest average posterior                individuals  0.880 (class 1)
#>   info   lowest odds of correct classification   individuals  9.3 (class 2)
#>
#> Local independence
#>   ok     pairs dependent within a class                       0 of 20 tested pairs     BH-
adjusted p < 0.05
#>       smallest adjusted p: self_efficacy x self_regulation in profile_1, residual 0.103, adjusted p 0.97
#>
#> Information
#>   info   log likelihood, parameters, BIC                      -600.92, 26, 1350.15
#>
#> Every check with its reference: get_results(x, "checks"). Other tables: get_results(x, what = ).
```

A relative entropy of 0.74 and average posterior probabilities of at least 0.88 indicate an acceptable classification. Relative entropy describes the quality of the classification for the chosen model; it is not a criterion for the number of profiles (Sorgente et al., 2025).

Posterior assignments

`get_results(fit, "assignments")` returns each respondent with the assigned profile, the posterior probability of every profile, and the classification uncertainty, in the row order of the data.

```
head(get_results(fit, "assignments"))
#>   cognitive_strategies intrinsic_value self_efficacy self_regulation test_anxiety profile
#> 1                 5.31            5.67          5.78            5.33         4.00       1
#> 2                 5.85            6.44          6.00            5.78         4.00       2
#> 3                 6.62            6.67          6.22            6.33         3.25       2
#> 4                 5.69            6.56          6.33            5.56         4.50       2
#> 5                 4.38            5.56          4.89            4.78         4.00       1
#> 6                 4.85            5.44          5.67            5.11         3.50       1
#>   uncertainty posterior_profile_1 posterior_profile_2
#> 1    0.154286            0.845714              0.154
#> 2    0.008309            0.008309              0.992
#> 3    0.005427            0.005427              0.995
#> 4    0.000982            0.000982              0.999
#> 5    0.281467            0.718533              0.281
#> 6    0.103158            0.896842              0.103
```

Stability across starting values

The likelihood of a mixture model can have several local maxima. `sensitivity()` re-estimates the model with different random seeds and reports the maximum reached by each run and the proportion of respondents assigned to the same profile as in the reported solution. A solution in which a profile holds about one

respondent with its variances at the lower bound is degenerate; `boundary = "exclude"` in `lpa()` keeps the best start without such a variance.

```
sensitivity(fit)
#>    seed log_likelihood converged iterations optimum best agreement
#> 1     1           -601      TRUE         18       1 TRUE         1
#> 2     2           -601      TRUE         18       1 TRUE         1
#> 3     3           -601      TRUE         18       1 TRUE         1
#> 4     4           -601      TRUE         18       1 TRUE         1
#> 5     5           -601      TRUE         18       1 TRUE         1
#> 6     6           -601      TRUE         18       1 TRUE         1
#> 7     7           -601      TRUE         18       1 TRUE         1
#> 8     8           -601      TRUE         18       1 TRUE         1
#> 9     9           -601      TRUE         18       1 TRUE         1
#> 10   10           -601      TRUE         18       1 TRUE         1
```

All ten refits reach the same maximum of the log likelihood and assign every respondent to the same profile, so the solution does not depend on the starting values.

Covariance between chosen indicators

Between local independence within profiles (EEI) and a full covariance (EEE) lies a covariance for chosen pairs of indicators only. `local_dependence` names the pairs, and every other covariance within a profile stays zero. The bivariate residuals of the model without covariances indicate which pairs carry the dependence.

```
independent <- lpa(srl, indicators, n_profiles = 2, model = "EEI")
head(subset(get_results(independent, "residuals"), statistic > 5))
#>     profile          indicator_1     indicator_2     kind observed expected residual
#> 1 profile_1 cognitive_strategies intrinsic_value gaussian    0.692        0    0.692
#> 2 profile_1 cognitive_strategies self_regulation gaussian    0.687        0    0.687
#> 3 profile_1 cognitive_strategies   self_efficacy gaussian    0.674        0    0.674
#> 4 profile_1      intrinsic_value self_regulation gaussian    0.664        0    0.664
#> 5 profile_1        self_efficacy self_regulation gaussian    0.644        0    0.644
#> 6 profile_2 cognitive_strategies self_regulation gaussian    0.604        0    0.604
#>   effective_n statistic df  p_value p_adjusted
#> 1         130      9.62 NA 6.59e-22   1.32e-20
#> 2         130      9.50 NA 2.03e-21   2.03e-20
#> 3         130      9.23 NA 2.63e-20   1.75e-19
#> 4         130      9.02 NA 1.80e-19   6.60e-19
#> 5         130      8.63 NA 6.22e-18   1.78e-17
#> 6         170      9.03 NA 1.77e-19   6.60e-19
```

```
paired <- lpa(srl, indicators, n_profiles = 2, variance_model = "equal",
              local_dependence = "cognitive_strategies:intrinsic_value")
compare_models(independent = independent, paired = paired, full = fit)
#> Model comparison
#>
#> Model       Profiles   k   LogLik     BIC    ΔBIC  Weight     ICL  Entropy  Min AvePP  < 0.8
#> ----------- -------- --- ------- ------- ------ ------ ------- -------- --------- -----
#> independent        2  16 -906.73 1904.71 554.56  0.000 1948.82     0.89      0.97     0
#> paired             2  17 -823.54 1744.04 393.90  0.000 1805.78     0.85      0.95     0
#> full               2  26 -600.92 1350.15   0.00  1.000 1456.62     0.74      0.88     0
```

12 of the 20 profile-specific pair residuals have a z statistic above 5 in the model without covariances, so the dependence is not confined to one pair. The covariance of one pair lowers BIC from 1904.7 to 1744.0, and the full covariance lowers it to 1350.1. A covariance for chosen pairs suits a dependence between one or two pairs of indicators; when every pair is dependent, the full covariance is the model.

Further analyses

The profiles can be related to covariates with `r3step()` and to distal outcomes with `three_step()`, both of which correct for classification error (Bolck et al., 2004; Vermunt, 2010). When the observations are nested in groups, `multilpa()` estimates the two-level model, in which the groups are in addition grouped into latent classes, and `enumerate_classes()` compares its numbers of profiles and group classes; both are presented in `vignette("lpa")`. The ten other covariance structures, which constrain the volume, shape and orientation separately, are estimated with `enumerate_lpa(model = "all")`.

References

Bolck, A., Croon, M., & Hagenaars, J. (2004). Estimating latent structure models with categorical variables: One-step versus three-step estimators. *Political Analysis*, 12(1), 3–27.

Celeux, G., & Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5), 781–793.

McLachlan, G. J. (1987). On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture. *Applied Statistics*, 36, 318–324.

Nylund, K. L., Asparouhov, T., & Muthén, B. O. (2007). Deciding on the number of classes in latent class analysis and growth mixture modeling: A Monte Carlo simulation study. *Structural Equation Modeling*, 14, 535–569.

Ramaswamy, V., DeSarbo, W. S., Reibstein, D. J., & Robinson, W. T. (1993). An empirical pooling approach for estimating marketing mix elasticities with PIMS data. *Marketing Science*, 12, 103–124.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6, 461–464.

Scrucca, L., Saqr, M., López-Pernas, S., & Murphy, K. (2024). An introduction and R tutorial to model-based clustering in education via latent profile analysis. In M. Saqr & S. López-Pernas (Eds.), *Learning analytics methods and tutorials: A practical guide using R* (pp. 285–317). Springer. https://doi.org/10.1007/978-3-031-54464-4_9

Sorgente, A., Caliciuri, R., Robba, M., Lanz, M., & Zumbo, B. D. (2025). A systematic review of latent class analysis in psychology: Examining the gap between guidelines and research practice. *Behavior Research Methods*, 57, 301. https://doi.org/10.3758/s13428-025-02812-1

Spurk, D., Hirschi, A., Wang, M., Valero, D., & Kauffeld, S. (2020). Latent profile analysis: A review and "how to" guide of its application within vocational behavior research. *Journal of Vocational Behavior*, 120, 103445. https://doi.org/10.1016/j.jvb.2020.103445

Vermunt, J. K. (2010). Latent class modeling with covariates: Two improved three-step approaches. *Political Analysis*, 18, 450–469.

Vogelsmeier, L. V. D. E., Oliveira, E., Misiejuk, K., López-Pernas, S., & Saqr, M. (2025). Delving into the psychology of machines: Exploring the structure of self-regulated learning via LLM-generated survey responses. *Computers in Human Behavior*, 173, 108769. https://doi.org/10.1016/j.chb.2025.108769