# Anticlustering for Sample Allocation To Minimize Batch Effects

Supplementary Materials

Martin Papenberg

Generated on 2025-06-27

## Overview

This document is found at the accompanying online repository for the manuscript “Anticlustering for Sample Allocation To Minimize Batch Effects”. It can be retrieved from https://osf.io/eu5gd/. The repository also contains other information, data and files concerning the project.

This document is meant to (a) reproduce all analyses and Figures (in this case, this includes formulas) reported in the manuscript and (b) illustrate some additional analyses not included in the main text due to space constraints (in particular, Figures 3 and 4 and section “Optimal Anticlustering With the Must-Link Feature” contain results not reported in the paper).

This document is fully reproducible via R and R Markdown.

### Anticlustering algorithm

Anticlustering is an optimization method that is characterized by (a) an objective function that quantifies the balance among batches, and (b) an algorithm that conducts the batch assignment in such a way that balance among batches is maximized. Anticlustering owes its name to the fact that the objective functions it uses are the reversal of criteria used in cluster analysis. For example, Späth (1986) already recognized that by maximizing instead of minimizing the k-means criterion (the “variance”), he was able to create groups that are similar to each other, and presented it as an improvement over the more intuitive random assignment (Steinley 2006). Brusco, Cradit, and Steinley (2020) recognized that other objective functions known from cluster analysis can also be implemented in the context of anticlustering.

In our application, we optimized the diversity objective to maximize similarity among batches. While the diversity is technically a measure of within-batch heterogeneity, its maximization simultaneously leads to minimal difference between the distribution of the input variables among batches (Feo and Khellaf 1990; cf. Papenberg 2024). Papenberg and Klau (2021) referred to the maximization of the diversity as anticluster editing because the minimization of the diversity is also well-known from the area of cluster analysis—under the term “cluster editing” (Shamir, Sharan, and Tsur 2004; Böcker, Briesemeister, and Klau 2011). The diversity is computed on the basis of a measure of pairwise dissimilarity among samples. In particular, it is defined as the overall sum of all dissimilarities among samples that are assigned to the same batch (Brusco, Cradit, and Steinley 2020). Hence, the diversity is not directly computed on the basis of the samples’ features, but instead it relies on distance measure that is computed on the basis of the features, for each pair of samples. In the context of anticlustering, the Euclidean distance is the most common measure that translates features to pairwise dissimilarities (Gallego et al. 2013; Papenberg and Klau 2021). However, using other distance measures such as the squared Euclidean distance is also possible (Brusco, Cradit, and Steinley 2020). The Euclidean distance is defined as

$$d(x, y) = \sqrt{\sum_{i=1}^{M} (x_i - y_i)^2}$$

When samples are described by two features, the Euclidean distance corresponds to the geometric, "straight line" distance between two points in a two-dimensional space; more similar data points are closer to each other (see Figure 1). For categorical variables, we use "one-hot" binary coding before including them in the computation of the Euclidean distance (see Table 1). In our example application, we used binary coded categories and the squared Euclidean distance.

An anticlustering algorithm assigns samples to batches in such a way that the objective function—here, the diversity—is maximized. Anticlustering usually employs heuristic optimization algorithms (Yang et al. 2022). While heuristics generally provide satisfying results in the context of anticlustering (Papenberg and Klau 2021), they do not guarantee to find the globally best assignment among all possibilities. In principle, enumerating all possible assignments is a valid strategy to obtain an optimal assignment. However, this approach quickly becomes impossible due to an exponential growth of the way in which assignments can be conducted (Papenberg and Klau 2021). Moreover, because anticlustering problems (with the exception of some special cases) are also NP-hard (Feo and Khellaf 1990), there is probably no algorithm that identifies the globally best assignment without considering all possibilities (at least in the worst case). In practice, heuristics are therefore indispensable. In the following, we present an anticlustering heuristic that is generally applicable to large data sets, and outline how we included must-link constraints with this algorithm.

We used an heuristic exchange algorithm to maximize the diversity (Späth 1986; Papenberg and Klau 2021; Weitz and Lakshminarayanan 1998). It consists of two steps: an initialization step and an optimization step. As initialization, it randomly assigns samples to equal-sized batches. In principle, unequal-sized batches would also be possible, but equal-sized batches were required in the current application (and in general, this requirement is most common. After initialization, the algorithm selects the first sample and checks how the diversity would change if the sample were swapped with each sample that is currently assigned to a different batch. After simulating each exchange, it realizes the one exchange that increases the diversity the most. It does not conduct an exchange if no improvement in diversity is possible. This procedure is repeated for each sample and it terminates after the last sample was processed. The procedure might also restart at the first element and reiterate through all samples until no exchange leads to an improvement any more, i.e., until a local maximum is found. In anticlust, we also implemented this local maximum search, which corresponds to the algorithm LCW by Weitz and Lakshminarayanan (1998). For better results, it is also possible to restart the search algorithm multiple times using different (random) initializations (Späth 1986).

Table 1: Illustrates the 'one-hot' recoding of the categorical variable Stage using three binary variables.

| Stage | None | StageI_II | StageIII_IV |
|---|---|---|---|
| None | 1 | 0 | 0 |
| StageI_II | 0 | 1 | 0 |
| StageIII_IV | 0 | 0 | 1 |

## Example Application

UCSF-Stanford Endometriosis Center for Discovery, Innovation, Training and Community Engagement ("ENACT, https://enactcenter.org/) is concerned with researching endometriosis disease. One of the ENACT projects required to assign 320 cell samples of women either carrying the disease or not to 20 batches. To illustrate the application, we prepared a synthetic data set that resembles the actual data set, which is not disclosed for reasons of medical confidentiality. Women in varying disease stages provided a varying number of samples. All women who did not carry the disease ($n$ = 50) provided exactly one sample, while all women who carried the disease ($n$ = 89) provided multiple samples: 40 patients provided 2 samples, 25 patients provided 3 samples, 11 patients provided 4 samples, 10 patients provided 5 samples, and 6, 7 or 8 samples were provided by one patient, respectively. In total, the data set consists of 320 samples from 139 unique individuals.

For the assignment, we strived for balance with regard to four categorical variables: is disease present (yes or no); stage of disease ("none", "Stage I or II", "Stage III or IV"); clinical site (University of California SF, or

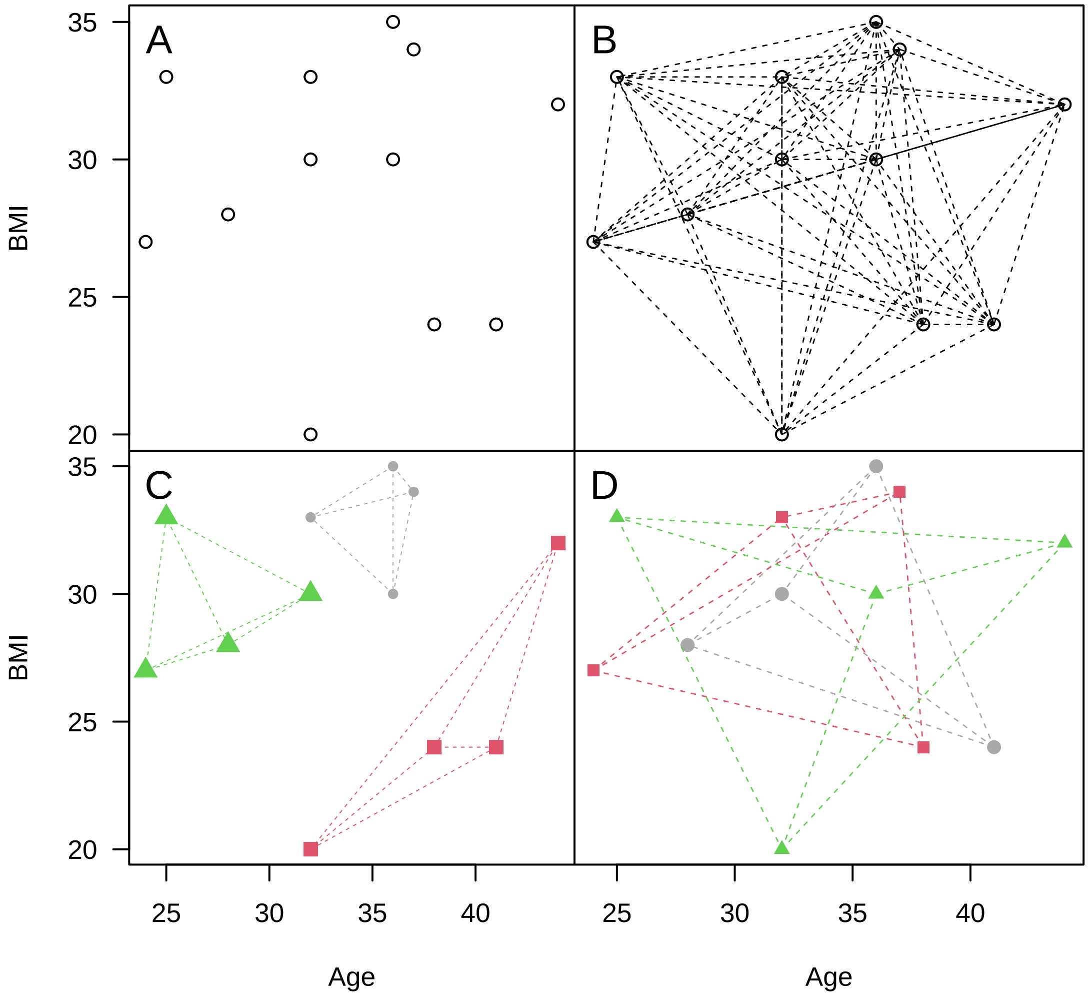

Figure 1: Illustrates the conversion from numeric features to Euclidean distance, and (anti)clustering assignments based on minimum and maximum diversity using the Euclidean distance. Panel A illustrates the BMI and age of twelve women in our synthetic data in a scatter plot. Panel B represents the Euclidean distances between features as a straight line in the two-dimensional space. The Euclidean distance is proportional to the length of the connecting lines in panel B. Panel C illustrates a clustering assignment of the 12 data points to $K = 3$ equal-sized groups via *minimum* diversity. Panel D illustrates an anticlustering assignment of the 12 data points to $K = 3$ equal-sized groups via *maximum* diversity. The diversity is computed as the sum of within-(anti)cluster distances, which are highlighted in Panel C and Panel D through connecting lines. Maximizing the diversity simultaneously leads to similar distribution of the input features among batches.

other); menstrual cycle phase (proliferative (“PE”) or secretory (“SE”). Other demographic variables such as age and BMI were also assessed but not deemed critical as covariates for the batch assignment. Post-hoc checks however revealed that no significant discrepancies occurred in these other variables, despite not being included.

For the purpose of illustration, we first applied an unconstrained anticlustering assignment and OSAT (see Table 2-5). Both methods successfully balanced all four variables among batches (all *p*s > .999 for anticlustering, all *p*s > .99 for OSAT). While the assignment was conducted using the 320 samples—as opposed to the 139 individual patients—as input, we also verified satisfactory balance on the level of individuals. Tables 6 and 7 illustrates the results of constrained anticlustering, which through the 2PML algorithm ensured that samples belonging to the same patient were assigned to the same batch. Note that the must-link constraints put rather severe restrictions to the assignments that were possible; for example, up to 50% of a batch had to be occupied with the samples of a single patient. Still, using 2PML, batches turned out to be highly balanced (all *p*s > .99).

Table 2: Application results of the unrestricted anticlustering assignment. Balance is assessed on individuals' level. Column 'Unique' encodes if a batch consists only of samples of individuals that are unique to this batch; '0' means there is at least one individual with a sample in this batch who also has at least one sample in another batch, '1' otherwise.

| | n | Unique | endo = yes (%) | site = UC (%) | phase = SE (%) | Stage = None | StageI_II | StageIII_IV |
|---|---|---|---|---|---|---|---|---|
| B1 | 16 | 0 | 14 (87.5) | 4 (25.0) | 7 (43.8) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B2 | 15 | 0 | 12 (80.0) | 3 (20.0) | 6 (40.0) | 3 (20.0) | 3 (20.0) | 9 (60.0) |
| B3 | 16 | 0 | 14 (87.5) | 3 (18.8) | 6 (37.5) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B4 | 14 | 0 | 12 (85.7) | 3 (21.4) | 5 (35.7) | 2 (14.3) | 4 (28.6) | 8 (57.1) |
| B5 | 16 | 0 | 13 (81.2) | 3 (18.8) | 7 (43.8) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| B6 | 15 | 0 | 13 (86.7) | 3 (20.0) | 7 (46.7) | 2 (13.3) | 4 (26.7) | 9 (60.0) |
| B7 | 14 | 0 | 12 (85.7) | 2 (14.3) | 6 (42.9) | 2 (14.3) | 3 (21.4) | 9 (64.3) |
| B8 | 16 | 0 | 14 (87.5) | 3 (18.8) | 7 (43.8) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B9 | 15 | 0 | 13 (86.7) | 3 (20.0) | 6 (40.0) | 2 (13.3) | 4 (26.7) | 9 (60.0) |
| B10 | 12 | 0 | 9 (75.0) | 3 (25.0) | 5 (41.7) | 3 (25.0) | 2 (16.7) | 7 (58.3) |
| B11 | 16 | 0 | 13 (81.2) | 4 (25.0) | 7 (43.8) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| B12 | 16 | 0 | 14 (87.5) | 3 (18.8) | 6 (37.5) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B13 | 16 | 0 | 13 (81.2) | 4 (25.0) | 7 (43.8) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| B14 | 15 | 0 | 12 (80.0) | 3 (20.0) | 7 (46.7) | 3 (20.0) | 2 (13.3) | 10 (66.7) |
| B15 | 16 | 0 | 13 (81.2) | 3 (18.8) | 6 (37.5) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| B16 | 14 | 0 | 12 (85.7) | 3 (21.4) | 6 (42.9) | 2 (14.3) | 3 (21.4) | 9 (64.3) |
| B17 | 14 | 0 | 11 (78.6) | 3 (21.4) | 5 (35.7) | 3 (21.4) | 3 (21.4) | 8 (57.1) |
| B18 | 16 | 0 | 13 (81.2) | 4 (25.0) | 7 (43.8) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| B19 | 16 | 0 | 14 (87.5) | 3 (18.8) | 6 (37.5) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B20 | 16 | 0 | 13 (81.2) | 4 (25.0) | 7 (43.8) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| p value | | | 1 | 1 | 1 | | 1 | |

Table 3: Application results of the unrestricted anticlustering assignment. Balance is assessed on samples' level. Column 'Unique' encodes if a batch consists only of samples of individuals that are unique to this batch; '0' means there is at least one individual with a sample in this batch who also has at least one sample in another batch, '1' otherwise.

| | n | Unique | endo = yes (%) | site = UC (%) | phase = SE (%) | Stage = None | StageI_II | StageIII_IV |
|---|---|---|---|---|---|---|---|---|
| B1 | 16 | 0 | 14 (87.5) | 4 (25.0) | 7 (43.8) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B2 | 16 | 0 | 13 (81.2) | 3 (18.8) | 6 (37.5) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| B3 | 16 | 0 | 14 (87.5) | 3 (18.8) | 6 (37.5) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B4 | 16 | 0 | 14 (87.5) | 3 (18.8) | 6 (37.5) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B5 | 16 | 0 | 13 (81.2) | 3 (18.8) | 7 (43.8) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| B6 | 16 | 0 | 14 (87.5) | 3 (18.8) | 7 (43.8) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B7 | 16 | 0 | 14 (87.5) | 3 (18.8) | 6 (37.5) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B8 | 16 | 0 | 14 (87.5) | 3 (18.8) | 7 (43.8) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B9 | 16 | 0 | 14 (87.5) | 3 (18.8) | 6 (37.5) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B10 | 16 | 0 | 13 (81.2) | 3 (18.8) | 6 (37.5) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| B11 | 16 | 0 | 13 (81.2) | 4 (25.0) | 7 (43.8) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| B12 | 16 | 0 | 14 (87.5) | 3 (18.8) | 6 (37.5) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B13 | 16 | 0 | 13 (81.2) | 4 (25.0) | 7 (43.8) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| B14 | 16 | 0 | 13 (81.2) | 3 (18.8) | 7 (43.8) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| B15 | 16 | 0 | 13 (81.2) | 3 (18.8) | 6 (37.5) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| B16 | 16 | 0 | 14 (87.5) | 3 (18.8) | 6 (37.5) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B17 | 16 | 0 | 13 (81.2) | 4 (25.0) | 6 (37.5) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| B18 | 16 | 0 | 13 (81.2) | 4 (25.0) | 7 (43.8) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| B19 | 16 | 0 | 14 (87.5) | 3 (18.8) | 6 (37.5) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B20 | 16 | 0 | 13 (81.2) | 4 (25.0) | 7 (43.8) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| p value | | | 1 | 1 | 1 | | 1 | |

Table 4: Application results of the OSAT assignment. Balance is assessed on individuals’ level. Column ’Unique’ encodes if a batch consists only of samples of individuals that are unique to this batch; ’0’ means there is at least one individual with a sample in this batch who also has at least one sample in another batch, ’1’ otherwise.

| | n | Unique | endo = yes (%) | site = UC (%) | phase = SE (%) | Stage = None | StageI_II | StageIII_IV |
|---|---|---|---|---|---|---|---|---|
| B1 | 15 | 0 | 11 (73.3) | 3 (20.0) | 6 (40.0) | 4 (26.7) | 3 (20.0) | 8 (53.3) |
| B2 | 16 | 0 | 14 (87.5) | 3 (18.8) | 7 (43.8) | 2 (12.5) | 3 (18.8) | 11 (68.8) |
| B3 | 16 | 0 | 14 (87.5) | 4 (25.0) | 8 (50.0) | 2 (12.5) | 3 (18.8) | 11 (68.8) |
| B4 | 16 | 0 | 14 (87.5) | 4 (25.0) | 6 (37.5) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B5 | 16 | 0 | 14 (87.5) | 4 (25.0) | 5 (31.2) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B6 | 16 | 0 | 14 (87.5) | 5 (31.2) | 7 (43.8) | 2 (12.5) | 3 (18.8) | 11 (68.8) |
| B7 | 15 | 0 | 11 (73.3) | 3 (20.0) | 7 (46.7) | 4 (26.7) | 2 (13.3) | 9 (60.0) |
| B8 | 16 | 0 | 14 (87.5) | 3 (18.8) | 7 (43.8) | 2 (12.5) | 5 (31.2) | 9 (56.2) |
| B9 | 16 | 0 | 15 (93.8) | 2 (12.5) | 7 (43.8) | 1 (6.2) | 4 (25.0) | 11 (68.8) |
| B10 | 16 | 0 | 12 (75.0) | 4 (25.0) | 6 (37.5) | 4 (25.0) | 2 (12.5) | 10 (62.5) |
| B11 | 15 | 0 | 12 (80.0) | 3 (20.0) | 6 (40.0) | 3 (20.0) | 3 (20.0) | 9 (60.0) |
| B12 | 16 | 0 | 14 (87.5) | 3 (18.8) | 6 (37.5) | 2 (12.5) | 3 (18.8) | 11 (68.8) |
| B13 | 16 | 0 | 14 (87.5) | 4 (25.0) | 7 (43.8) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B14 | 15 | 0 | 13 (86.7) | 3 (20.0) | 5 (33.3) | 2 (13.3) | 5 (33.3) | 8 (53.3) |
| B15 | 15 | 0 | 12 (80.0) | 1 (6.7) | 6 (40.0) | 3 (20.0) | 4 (26.7) | 8 (53.3) |
| B16 | 15 | 0 | 13 (86.7) | 3 (20.0) | 6 (40.0) | 2 (13.3) | 3 (20.0) | 10 (66.7) |
| B17 | 16 | 0 | 13 (81.2) | 5 (31.2) | 6 (37.5) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| B18 | 14 | 0 | 11 (78.6) | 2 (14.3) | 6 (42.9) | 3 (21.4) | 4 (28.6) | 7 (50.0) |
| B19 | 16 | 0 | 14 (87.5) | 4 (25.0) | 6 (37.5) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B20 | 16 | 0 | 13 (81.2) | 3 (18.8) | 6 (37.5) | 3 (18.8) | 4 (25.0) | 9 (56.2) |
| p value | | | 0.995 | 0.998 | 1.000 | | 1 | |

Table 5: Application results of the OSAT assignment. Balance is assessed on samples' level. Column 'Unique' encodes if a batch consists only of samples of individuals that are unique to this batch; '0' means there is at least one individual with a sample in this batch who also has at least one sample in another batch, '1' otherwise.

| | n | Unique | endo = yes (%) | site = UC (%) | phase = SE (%) | Stage = None | StageI_II | StageIII_IV |
|---|---|---|---|---|---|---|---|---|
| B1 | 16 | 0 | 12 (75.0) | 3 (18.8) | 7 (43.8) | 4 (25.0) | 3 (18.8) | 9 (56.2) |
| B2 | 16 | 0 | 14 (87.5) | 3 (18.8) | 7 (43.8) | 2 (12.5) | 3 (18.8) | 11 (68.8) |
| B3 | 16 | 0 | 14 (87.5) | 4 (25.0) | 8 (50.0) | 2 (12.5) | 3 (18.8) | 11 (68.8) |
| B4 | 16 | 0 | 14 (87.5) | 4 (25.0) | 6 (37.5) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B5 | 16 | 0 | 14 (87.5) | 4 (25.0) | 5 (31.2) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B6 | 16 | 0 | 14 (87.5) | 5 (31.2) | 7 (43.8) | 2 (12.5) | 3 (18.8) | 11 (68.8) |
| B7 | 16 | 0 | 12 (75.0) | 3 (18.8) | 7 (43.8) | 4 (25.0) | 2 (12.5) | 10 (62.5) |
| B8 | 16 | 0 | 14 (87.5) | 3 (18.8) | 7 (43.8) | 2 (12.5) | 5 (31.2) | 9 (56.2) |
| B9 | 16 | 0 | 15 (93.8) | 2 (12.5) | 7 (43.8) | 1 (6.2) | 4 (25.0) | 11 (68.8) |
| B10 | 16 | 0 | 12 (75.0) | 4 (25.0) | 6 (37.5) | 4 (25.0) | 2 (12.5) | 10 (62.5) |
| B11 | 16 | 0 | 13 (81.2) | 3 (18.8) | 6 (37.5) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| B12 | 16 | 0 | 14 (87.5) | 3 (18.8) | 6 (37.5) | 2 (12.5) | 3 (18.8) | 11 (68.8) |
| B13 | 16 | 0 | 14 (87.5) | 4 (25.0) | 7 (43.8) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B14 | 16 | 0 | 14 (87.5) | 3 (18.8) | 5 (31.2) | 2 (12.5) | 5 (31.2) | 9 (56.2) |
| B15 | 16 | 0 | 13 (81.2) | 1 (6.2) | 6 (37.5) | 3 (18.8) | 4 (25.0) | 9 (56.2) |
| B16 | 16 | 0 | 14 (87.5) | 3 (18.8) | 7 (43.8) | 2 (12.5) | 3 (18.8) | 11 (68.8) |
| B17 | 16 | 0 | 13 (81.2) | 5 (31.2) | 6 (37.5) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| B18 | 16 | 0 | 13 (81.2) | 2 (12.5) | 7 (43.8) | 3 (18.8) | 4 (25.0) | 9 (56.2) |
| B19 | 16 | 0 | 14 (87.5) | 4 (25.0) | 6 (37.5) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B20 | 16 | 0 | 13 (81.2) | 3 (18.8) | 6 (37.5) | 3 (18.8) | 4 (25.0) | 9 (56.2) |
| p value | | | 0.998 | 0.994 | 1.000 | | 1 | |

Table 6: Application results of the must-link constrained anticlustering assignment. Balance is assessed on individuals' level. Column 'Unique' encodes if a batch consists only of samples of individuals that are unique to this batch; '0' means there is at least one individual with a sample in this batch who also has at least one sample in another batch, '1' otherwise.

|  | n | Unique | endo = yes (%) | site = UC (%) | phase = SE (%) | Stage = None | StageI_II | StageIII_IV |
|---|---|---|---|---|---|---|---|---|
| B1 | 7 | 1 | 4 (57.1) | 3 (42.9) | 3 (42.9) | 3 (42.9) | 1 (14.3) | 3 (42.9) |
| B2 | 5 | 1 | 3 (60.0) | 2 (40.0) | 1 (20.0) | 2 (40.0) | 1 (20.0) | 2 (40.0) |
| B3 | 7 | 1 | 5 (71.4) | 1 (14.3) | 2 (28.6) | 2 (28.6) | 1 (14.3) | 4 (57.1) |
| B4 | 9 | 1 | 6 (66.7) | 2 (22.2) | 3 (33.3) | 3 (33.3) | 1 (11.1) | 5 (55.6) |
| B5 | 6 | 1 | 4 (66.7) | 2 (33.3) | 2 (33.3) | 2 (33.3) | 1 (16.7) | 3 (50.0) |
| B6 | 8 | 1 | 5 (62.5) | 3 (37.5) | 4 (50.0) | 3 (37.5) | 1 (12.5) | 4 (50.0) |
| B7 | 6 | 1 | 4 (66.7) | 2 (33.3) | 3 (50.0) | 2 (33.3) | 1 (16.7) | 3 (50.0) |
| B8 | 7 | 1 | 5 (71.4) | 1 (14.3) | 3 (42.9) | 2 (28.6) | 1 (14.3) | 4 (57.1) |
| B9 | 7 | 1 | 4 (57.1) | 3 (42.9) | 2 (28.6) | 3 (42.9) | 1 (14.3) | 3 (42.9) |
| B10 | 5 | 1 | 3 (60.0) | 2 (40.0) | 2 (40.0) | 2 (40.0) | 1 (20.0) | 2 (40.0) |
| B11 | 8 | 1 | 5 (62.5) | 3 (37.5) | 4 (50.0) | 3 (37.5) | 1 (12.5) | 4 (50.0) |
| B12 | 5 | 1 | 3 (60.0) | 1 (20.0) | 2 (40.0) | 2 (40.0) | 1 (20.0) | 2 (40.0) |
| B13 | 8 | 1 | 5 (62.5) | 3 (37.5) | 3 (37.5) | 3 (37.5) | 1 (12.5) | 4 (50.0) |
| B14 | 9 | 1 | 6 (66.7) | 3 (33.3) | 4 (44.4) | 3 (33.3) | 1 (11.1) | 5 (55.6) |
| B15 | 8 | 1 | 5 (62.5) | 2 (25.0) | 5 (62.5) | 3 (37.5) | 1 (12.5) | 4 (50.0) |
| B16 | 7 | 1 | 5 (71.4) | 2 (28.6) | 2 (28.6) | 2 (28.6) | 1 (14.3) | 4 (57.1) |
| B17 | 6 | 1 | 4 (66.7) | 2 (33.3) | 2 (33.3) | 2 (33.3) | 1 (16.7) | 3 (50.0) |
| B18 | 7 | 1 | 4 (57.1) | 1 (14.3) | 2 (28.6) | 3 (42.9) | 1 (14.3) | 3 (42.9) |
| B19 | 6 | 1 | 4 (66.7) | 2 (33.3) | 2 (33.3) | 2 (33.3) | 1 (16.7) | 3 (50.0) |
| B20 | 8 | 1 | 5 (62.5) | 2 (25.0) | 2 (25.0) | 3 (37.5) | 2 (25.0) | 3 (37.5) |
| p value |  |  | 1.000 | 0.999 | 0.997 |  | 1 |  |

Table 7: Application results of the must-link constrained anticlustering assignment. Balance is assessed on samples' level. Column 'Unique' encodes if a batch consists only of samples of individuals that are unique to this batch; '0' means there is at least one individual with a sample in this batch who also has at least one sample in another batch, '1' otherwise.

| | n | Unique | endo = yes (%) | site = UC (%) | phase = SE (%) | Stage = None | StageI_II | StageIII_IV |
|---|---|---|---|---|---|---|---|---|
| B1 | 16 | 1 | 13 (81.2) | 3 (18.8) | 6 (37.5) | 3 (18.8) | 2 (12.5) | 11 (68.8) |
| B2 | 16 | 1 | 14 (87.5) | 2 (12.5) | 7 (43.8) | 2 (12.5) | 2 (12.5) | 12 (75.0) |
| B3 | 16 | 1 | 14 (87.5) | 3 (18.8) | 7 (43.8) | 2 (12.5) | 5 (31.2) | 9 (56.2) |
| B4 | 16 | 1 | 13 (81.2) | 3 (18.8) | 6 (37.5) | 3 (18.8) | 2 (12.5) | 11 (68.8) |
| B5 | 16 | 1 | 14 (87.5) | 2 (12.5) | 7 (43.8) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B6 | 16 | 1 | 13 (81.2) | 4 (25.0) | 6 (37.5) | 3 (18.8) | 2 (12.5) | 11 (68.8) |
| B7 | 16 | 1 | 14 (87.5) | 3 (18.8) | 5 (31.2) | 2 (12.5) | 5 (31.2) | 9 (56.2) |
| B8 | 16 | 1 | 14 (87.5) | 4 (25.0) | 7 (43.8) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B9 | 16 | 1 | 13 (81.2) | 3 (18.8) | 7 (43.8) | 3 (18.8) | 3 (18.8) | 10 (62.5) |
| B10 | 16 | 1 | 14 (87.5) | 2 (12.5) | 6 (37.5) | 2 (12.5) | 8 (50.0) | 6 (37.5) |
| B11 | 16 | 1 | 13 (81.2) | 3 (18.8) | 6 (37.5) | 3 (18.8) | 2 (12.5) | 11 (68.8) |
| B12 | 16 | 1 | 14 (87.5) | 4 (25.0) | 8 (50.0) | 2 (12.5) | 4 (25.0) | 10 (62.5) |
| B13 | 16 | 1 | 13 (81.2) | 3 (18.8) | 7 (43.8) | 3 (18.8) | 2 (12.5) | 11 (68.8) |
| B14 | 16 | 1 | 13 (81.2) | 4 (25.0) | 6 (37.5) | 3 (18.8) | 2 (12.5) | 11 (68.8) |
| B15 | 16 | 1 | 13 (81.2) | 4 (25.0) | 7 (43.8) | 3 (18.8) | 2 (12.5) | 11 (68.8) |
| B16 | 16 | 1 | 14 (87.5) | 3 (18.8) | 6 (37.5) | 2 (12.5) | 5 (31.2) | 9 (56.2) |
| B17 | 16 | 1 | 14 (87.5) | 4 (25.0) | 6 (37.5) | 2 (12.5) | 5 (31.2) | 9 (56.2) |
| B18 | 16 | 1 | 13 (81.2) | 5 (31.2) | 6 (37.5) | 3 (18.8) | 2 (12.5) | 11 (68.8) |
| B19 | 16 | 1 | 14 (87.5) | 4 (25.0) | 6 (37.5) | 2 (12.5) | 5 (31.2) | 9 (56.2) |
| B20 | 16 | 1 | 13 (81.2) | 3 (18.8) | 7 (43.8) | 3 (18.8) | 4 (25.0) | 9 (56.2) |
| p value | | | 1.000 | 1.000 | 1.000 | | 0.991 | |

## Simulation Study

For the simulation, we generated 10000 data sets. Data sets were processed via (a) OSAT, (b) PSBA, (c) unconstrained anticlustering and (d) anticlustering subject to must-link constraints. Because the OSAT method is only applicable to categorical variables, we used categorical variables in our simulation. For anticlustering and for PSBA, the categorical variables were binary coded before the methods were applied (see Table 2). For each data set, we randomly determined the number of categorical variables (2-5), the number of classes per variable (2-5; the distribution of classes was uniform), the total sample size $N$ (between 50 and 500) and the number of batches $K$ (2, 4, or 10 equal-sized batches). PSBA was only applied for $K = 2$ and $K = 4$ ($n = 6658$ data sets) because the authors' implementation only allows the assignment to a maximum of four batches. For each data set, we simulated the condition where dropout in samples due to technical failures occurs. To this end, 20% of samples were randomly discarded to test the methods' ability to preserve balance when samples cannot be used. Must-link constraints were generating a random integer between 1 and $N$ (with equal probability), and using the resulting numbers as must-link grouping variables. This rule resulted in a distribution of constraints that resembled our motivating application: 58% of all elements had no must-link partner; 29% had 1 must-link partner; 10% had 2 must-link partners, and 3% had 3 or more must-link partners.

Across the 10000 simulation runs, the average run time for the competing assignment methods was 0.09s for unconstrained anticlustering, 0.1s for anticlustering with the must-link feature, 2.68s for OSAT, and 27.22s for PSBA, making anticlustering about 30 and 302 times faster than OSAT and PSBA, respectively. For each simulation run, we computed $\chi^2$-tests to assess the imbalance among batches for each of the 2-5 variables, for each of the three competing methods. We stored the $p$-value associated with each test. A higher $p$-value indicates that there is less imbalance among batches, i.e., that the batches are more similar. The simulation revealed that in 84% of all variable comparisons, balance among batches was better when using anticlustering as compared to OSAT. Balance was equal in 15% of all comparisons, and only in 0.32% (= 110 of all 34880 pairwise comparisons) OSAT outperformed `anticlust`. In 78% of all variable comparisons, balance was better when using anticlustering as compared to PSBA. Balance was equal in 22% of all comparisons, and only in 0.15% (= 35 of all 23258 pairwise comparisons) PSBA outperformed anticlustering. There are fewer comparisons between anticlustering and PSBA than between anticlustering and OSAT because PSBA was not applied in the simulation runs where 10 batches were assigned.

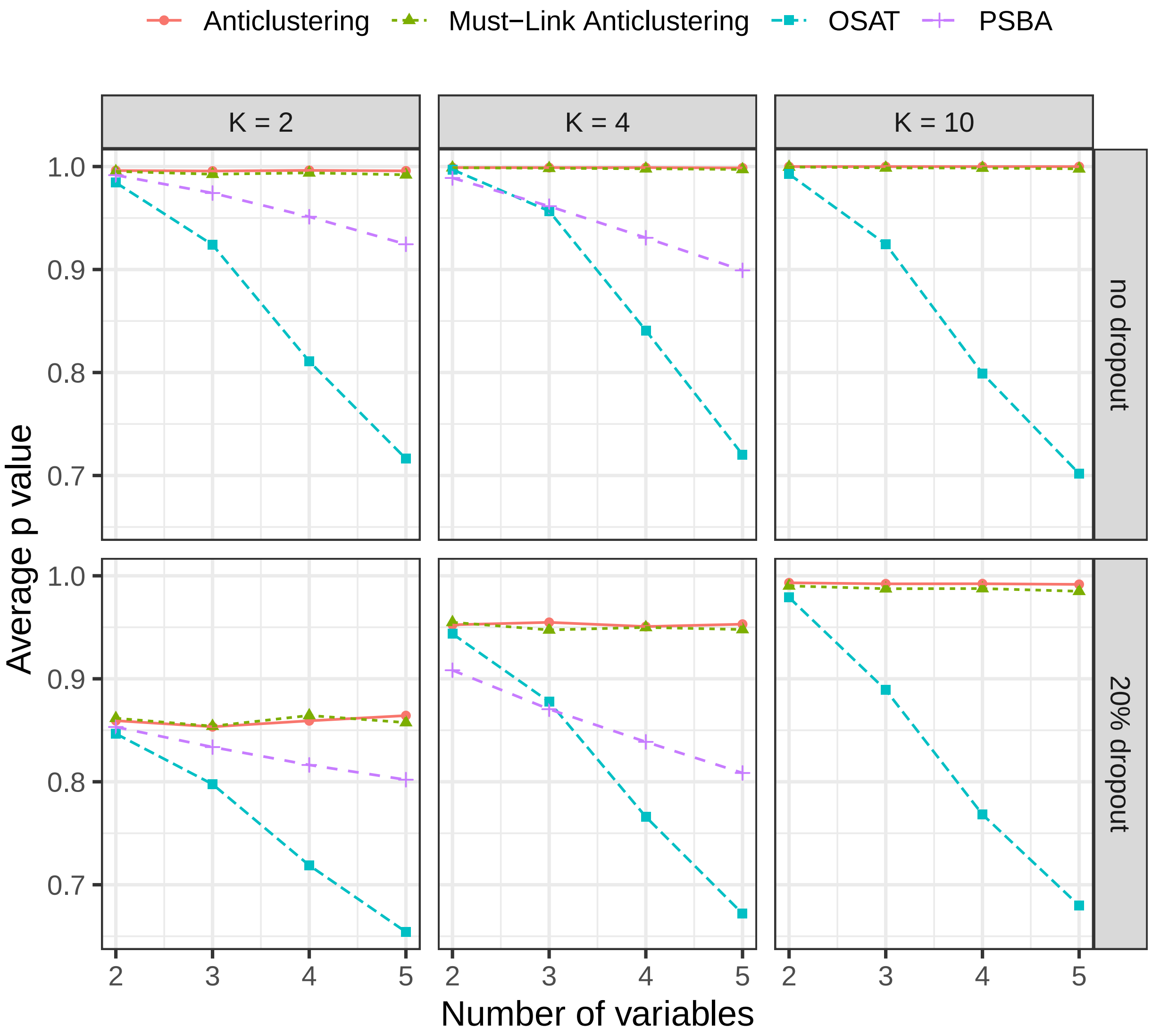

Figure 2: Average $p$ values in dependence of the number of batches and the number of variables. Higher $p$ values indicate better balance. Anticlustering maintained a comparable level of balance in all conditions. OSAT's performance decreased with increasing number of variables most strongly.

Figure 2 illustrates the average $p$-values in dependence of the number of variables ($M$) and the number of batches ($K$), when no dropouts occured or when 20% dropouts occured. Without dropouts, when increasing the number of variables from 2 to 5, the average $p$-value for OSAT declined from 0.99 to 0.71 whereas the average $p$-value for anticlustering remained greater than 0.99. PSBA also demonstrated a decrease in $p$-value when increasing the number of variables, but less so than OSAT (from 0.99 to 0.91). With 20% dropouts, the overall level of balance was decreased as compared to the condition where no dropouts occured, especially for K = 2 and K = 4. Still, anticlustering maintained the highest level of balance among the competing methods, with average $p$-values not dropping below 0.85. The worst case $p$-value observed for anticlustering under dropout conditions was 0.1043242. The other methods were more strongly affected by data loss, as evident by much smaller worst-case $p$-values ($1.4454321 \times 10^{-4}$ for OSAT and 0.0033143 for PSBA).

Figures 3 and 4 display the results of the simulation study aggregating via the size of the data set ($N$, categorized for the purpose of illustration) and the number of categories per variable $P$, respectively. Increasing the number of categories per variable aggravated finding balance for OSAT and PSBA, while again, anticlustering was mostly unaffected. Increasing the number of samples led to improved balance for the OSAT method, while it was generally outperformed by anticlustering as well as PSBA.

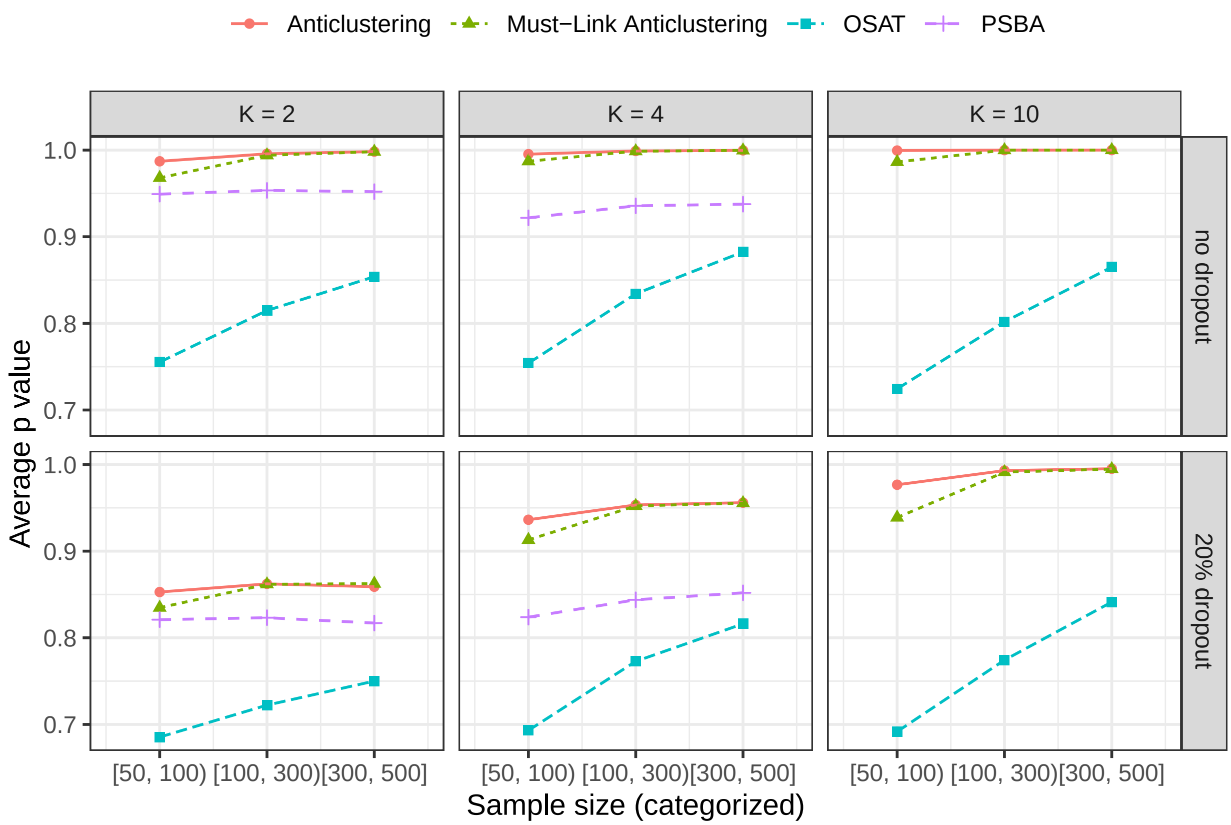

Figure 3: Average $p$ values in dependence of the number of batches and the number of simulated samples $N$ (categorized). Higher $p$ values indicate better balance. Anticlustering maintained a comparable level of balance in all conditions. OSAT's performance increased with an increasing number of samples, and was most affected when few samples were available.

The anticlustering assignment that was subjected to must-link constraints on average achieved 99.9% of the objective value of the unconstrained assignment. Hence, must-link constraints are not only desirable from a user's point of view, but they also do not decrease batch balance considerably; in 74% of all cases, balance was not at all reduced by the constraints. Remarkably, the constrained anticlustering assignment led to better balance than the OSAT and PSBA assignments that did not employ any constraints (see Figures 2-4).

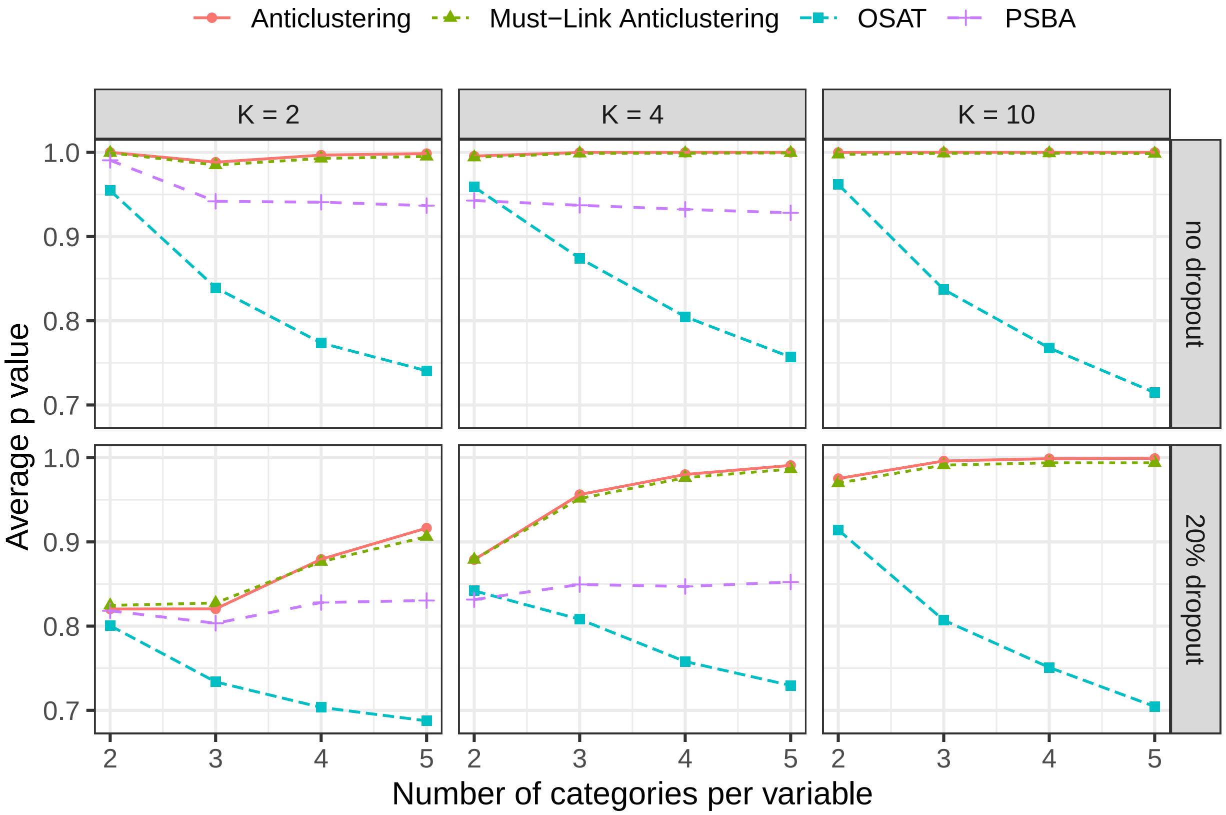

Figure 4: Average $p$ values in dependence of the number of batches and the number of categories per variable $P$. Higher $p$ values indicate better balance. Anticlustering maintained a comparable level of balance in all conditions. OSAT's and PSBA's performance decreased with an increasing number of categories per variable.

### Quantification and Statistical Analysis

Our simulation study compared anticlustering, must-link constrained anticlustering, OSAT and PSBA regarding their ability to achieve balance between batches. All code to reproduce the simulation and its analysis are available from the accompanying Open Science Repository (https://doi.org/10.17605/OSF.IO/EU5GD). The simulation performed 10000 iterations. During each iteration, a synthetic data set was randomly generated that was processed by each each of the four methods, in two dropout conditions (no dropout vs. 20% dropout). We generated categorical variables having 2-5 levels. Each data set contained 2-5 of such categorical variables. The total sample size N (50-500), the number of variables (2-5), the number of categories per variable (2-5), and the number of batches (2, 4, 10) was randomly chosen from a uniform distribution for each iteration of the simulation.

To quantify balance among batches, a standard a $\chi^2$ test was performed. The $\chi^2$ test computes the discrepancy between the observed distribution of a categorical variable across batches, versus an assumed uniform distribution of the variable. The associated $p$-value quantifies the "extremeness" of the observed distribution. That is, a small $p$-value is associated with a small probability that the observed distribution of the variable (or a more extreme distribution) would be generated under true uniform sampling; a high $p$-value indicates that even when sampling from a true uniform distribution, we would usually expect an observed distribution that deviates more strongly from a uniform distribution. $P$-values were obtained for each for each of the 2-5 variables in each data set, for each of the four methods and for each of the two dropout conditions. As a consequence, we computed at least 16 $p$-values and a maximum of 40 $p$-values for each iteration of the simulation. Thus, the effective $n$ was not 10000, but depended on the number of variables that were generated per simulation run. As main comparison, we reported the relative proportion of variables where anticlustering outperformed the competing methods in the no dropout condition, was equal to them, or was inferior. The effective $n$ for this comparison was 34880 vs. OSAT and 23258 vs. PSBA. The $n$ is different for each set of comparisons because PSBA was not applied when 10 batches were assigned. In Figure 2 we depicted the average $p$-value by method, number of batches, number of variables, and dropout condition, while aggregating across the other variables and across all simulation runs.

We did not report indices of statistical significance pertaining to the comparison between methods. This was due to (a) the large sample size, rendering measures of significance largely meaningless, and the ease of reproducibility of the simulation. Similarly, we chose a fixed but large number of simulation iterations, instead of aiming for a specified level of statistical power. Instead, a large data basis was generated allowing for meaningful conclusions, which is common practice in simulation studies. The number of iterations we employed (10000) is a common choice in computational studies and was also practically feasible for our particular study: the entire simulation took a total time of about 3 days on a desktop computer. Note that for a revision of this article, we repeated the entire simulation (using a different random seed and adding the 20% dropout condition) and virtually obtained the same pattern of results. Our accompanying online repository contains the data set obtained for the initial dataset as well as the revised (and final) dataset, verifying the high level of reproducibility.

## Bin Packing to Initialize a Must-Link Assignment

In our application, samples belonging to the same patient were required to be assigned to the same batch. We refer to a set of samples that must be assigned to the same batch as a must-link clique. During initialization, we first assign all samples to a batch that are part of a clique. Each clique must be assigned completely to one of the batches and samples within a clique must not be split apart. At the same time, the maximum capacity of each batch must not be exceeded. Using this conceptualization, the initialization step corresponds to a bin packing problem, which is one the classical NP complete problem in computer science (Garey and Johnson 1979). That is, we assign a weight to each clique, corresponding to the number of samples it contains. When filling batches, the sum of the weights of the cliques in each batch must not exceed its capacity. Many optimal and heuristic algorithms have been developed to address such a bin packing problem. As the default method, we use a randomized first fit heuristic to fill batches: For each must-link clique, we iterate through all batches in random order and assign it to the first batch where it fits. The process is expected to evenly distribute the must-link cliques among batches. This random component is particularly useful if we use multiple restarts of the optimization algorithm. After assigning the must-link cliques to batches, the remaining samples can be assigned randomly to fill the remaining space. Note that our randomized first fit algorithm is a heuristic that may not find an assignment of must-link groups to batches even if one is theoretically available. If the heuristic indicates that the batches cannot hold the must-link groups, we therefore use an optimal algorithm based on integer linear programming as a fallback option, which allows us to verify if the constraints really cannot be fulfilled. To this end, we implemented an adaptation of the standard bin packing ILP model by Martello and Toth, (1990, 221). It is given as

$$
\begin{aligned}
\text{minimize} \quad & \sum_{1 \le i \le n} \sum_{1 \le j \le K} x_{ij} && && (1)\\
\text{subject to} \quad & \sum_{i=1}^{n} w_i x_{ij} \le c_j && j = (1, \ldots, K) && (2)\\
& \sum_{j=1}^{K} x_{ij} = 1 && i = (1, \ldots, n) && (3)\\
& x_{ij} \in \{0, 1\} && i = (1, \ldots, n), j = (1, \ldots, K) && (4)
\end{aligned}
$$

The number of must-link cliques is given by $n$. The model has decision variables $x_{ij}$ to encode whether clique $i$ ($i = 1, \ldots, n$) is assigned to batch $j$ ($j = 1, \ldots, K$). It uses $K$ values $c_j$ to represent the capacity of each batch; in the default case of equal-sized batches, we have $c_j = \frac{N}{K}$ for each batch. It uses $n$ values $w_i$ to encode the weight of each clique, i.e., the number of samples it represents that must be assigned to the same batch in order to fulfil the must-link constraints. Constraint (2) realizes that the weight of each batch is not exceeded; constraint (3) realizes that each clique is assigned to exactly one batch. Note that during the initialization step that assigns cliques to batches, we only need to test if the constraints (2) and (3) can be fulfilled at all; any feasible assignment is equally valid. For this reason, the objective function (1) is chosen to be constant for each assignment that satisfies the constraints ($n$). It does not actually contribute to solving the problem, and the model only test if the must-link constraints can be fulfilled.

## Optimal Anticlustering With the Must-Link Feature

We developed the new algorithm 2PML that tries to "cleverly" adapt the standard pairwise exchange method towards including must-link constraints. Actually, using an optimal/exact algorithm is easier than coming up with such a heuristic. However, optimal approaches do not scale to very large data sets. We used the ILP model by Papenberg and Klau to find globally optimal batch assignments for the diversity objective when must-link constraints are included. For unconstrained anticlustering, the model can be used to solve problem instances of up to about $N = 30$ in 1800s running time (Schulz 2022). For this paper, we extended the model to allow it to include must-link constraints. The commercial gurobi solver (Version 11.0.2, Gurobi Optimization LLC 2024) was used as backend to actually solve the model.

The extension of the original model towards including must-link constraints is actually quite straightforward: To induce must-link constraints in the context of an exact algorithm, it is sufficient to adjust the distance matrix used as input. The pairwise distance between two samples that must be linked is set to $\infty$ (Böcker, Briesemeister, and Klau 2011) (or, in practice, to a sufficiently large real value). If the set of must-link constraints can be satisfied, any globally optimal assignment will place these samples in the same batch, because the objective value associated with such an assignment is necessarily better than that of an assignment that places them in different batches.

To evaluate the optimal approach, we generated synthetic data sets with categorical variables, using the same parameters as in the simulation in the paper: $M = 2 - 5$ variables; $P = 2 - 5$ categories per variable; the distance matrix was generated via first binary coding the variables and then computing the pairwise Euclidean distances. The number of groups $K$ was varied between 2, 3, 4, 5, and 10. Must-link constraints were generated using the same rule as for the simulation study and were used to adjust the distance matrix.

Starting with $N = 20$ (or $N = 21$ for $K = 3$) we generated five data sets for each $N$. We ensured that each data set was divisible by $K$ so equal-sized groups could be generated. Each combination of $K$ and $N$ was replicated 5 times to estimate the average run time. For each $K$, $N$ was sequentially increased by $K$ until

Table 8: Illustrates the run time of the exact ILP algorithm and the 2PML heuristic (using 1000 repetitions).

| K | Algorithm | max_mean_runtime | N_max |
|---|---|---|---|
| 2 | optimal | 562.8550 | 46 |
| 3 | optimal | 463.8440 | 39 |
| 4 | optimal | 579.0000 | 40 |
| 5 | optimal | 599.5360 | 40 |
| 10 | optimal | 6.4550 | 40 |
| 2 | heuristic | 2.5600 | 46 |
| 3 | heuristic | 2.1740 | 39 |
| 4 | heuristic | 2.3300 | 40 |
| 5 | heuristic | 2.3660 | 40 |
| 10 | heuristic | 2.7925 | 40 |

the time limit of 1800s was exceeded in a run (Schulz 2022). We also applied 2PML with 1000 repetitions (500x Phase 1 + 500x Phase 2) to investigate if our heuristic tends to produce optimal results.

Table 8 illustrate the maximum N achieved for the optimal algorithm, as well as the average run time for the maximum N that was still processed. Figure 5 shows the run time increase by N for both the optimal algorithm and the heuristic. In 94.74% of all data sets, the 2PML algorithm identified the optimal solution.

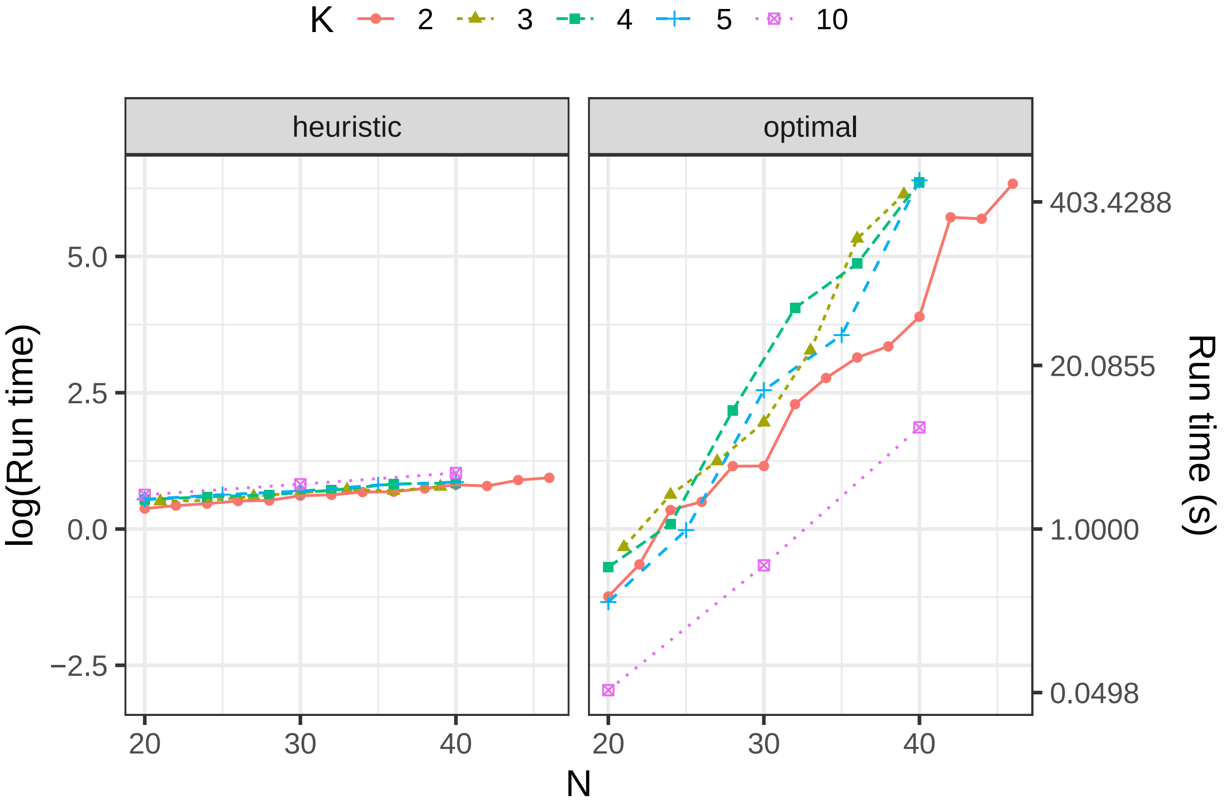

Figure 5: Illustrates the increase of the run time of the optimal and heuristic algorithms with increasing N. Note that the y-axis on a log scale, so a linear increase in the plot implies exponential increase of running time.

## References

Böcker, Sebastian, Sebastian Briesemeister, and Gunnar W Klau. 2011. “Exact Algorithms for Cluster Editing: Evaluation and Experiments.” *Algorithmica* 60: 316–34. https://doi.org/10.1007/s00453-009-9339-7.

Brusco, Michael J, J Dennis Cradit, and Douglas Steinley. 2020. “Combining Diversity and Dispersion Criteria for Anticlustering: A Bicriterion Approach.” *British Journal of Mathematical and Statistical Psychology* 73 (3): 375–96. https://doi.org/10.1111/bmsp.12186.

Feo, Thomas A, and Mallek Khellaf. 1990. “A Class of Bounded Approximation Algorithms for Graph Partitioning.” *Networks* 20 (2): 181–95.

Gallego, Micael, Manuel Laguna, Rafael Marti, and Abraham Duarte. 2013. “Tabu Search with Strategic Oscillation for the Maximally Diverse Grouping Problem.” *Journal of the Operational Research Society* 64 (5): 724–34. https://doi.org/10.1057/jors.2012.66.

Garey, Michael R, and David S Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-completeness*. New York: Freeman.

Gurobi Optimization LLC. 2024. “GUROBI Optimization.” https://www.gurobi.com/downloads/.

Martello, S, and P Toth. 1990. *Knapsack Problems: Algorithms and Computer Implementations*. Wiley.

Papenberg, Martin. 2024. “K-Plus Anticlustering: An Improved k-Means Criterion for Maximizing Between-Group Similarity.” *British Journal of Mathematical and Statistical Psychology* 77 (1): 80–102.

Papenberg, Martin, and Gunnar W Klau. 2021. “Using Anticlustering to Partition Data Sets into Equivalent Parts.” *Psychological Methods* 26 (2): 161–74. https://doi.org/10.1037/met0000301.

Schulz, Arne. 2022. “A New Mixed-Integer Programming Formulation for the Maximally Diverse Grouping Problem with Attribute Values.” *Annals of Operations Research* 318 (1): 501–30.

Shamir, Ron, Roded Sharan, and D Tsur. 2004. “Cluster Graph Modification Problems.” *Discrete Applied Mathematics* 144: 173–82. https://doi.org/10.1016/j.dam.2004.01.007.

Späth, H. 1986. “Anticlustering: Maximizing the Variance Criterion.” *Control and Cybernetics* 15 (2): 213–18.

Steinley, Douglas. 2006. “K-Means Clustering: A Half-Century Synthesis.” *British Journal of Mathematical and Statistical Psychology* 59 (1): 1–34. https://doi.org/10.1348/000711005X48266.

Weitz, RR, and S Lakshminarayanan. 1998. “An Empirical Comparison of Heuristic Methods for Creating Maximally Diverse Groups.” *Journal of the Operational Research Society* 49 (6): 635–46. https://doi.org/10.1057/palgrave.jors.2600510.

Yang, Xiao, Zonghui Cai, Ting Jin, Zheng Tang, and Shangce Gao. 2022. “A Three-Phase Search Approach with Dynamic Population Size for Solving the Maximally Diverse Grouping Problem.” *European Journal of Operational Research* 302 (3): 925–53.