

Supplemental material on “Improving the computational performance of TCLUSST through ensemble initialization”

Pedro C. Álvarez-Esteban, Luis A. García-Escudero,
Agustín Mayo-Iscar, Javier Crespo-Guerrero

1 R Code for `simula_tclust`

```
1
2 simula_tclust <- function(n, p = 4, G = 3, scenario = 2, balanced = 1, symmetry = 1)
3 {
4   if (G != 3 & G != 6)
5     stop("Invalid G")
6   if (p < 2)
7     stop("Invalid p")
8   if (symmetry != 1 & symmetry != 2)
9     stop("Invalid symmetry")
10  if (G == 6) {
11    nn <- n/2
12  }
13  else {
14    nn <- n
15  }
16  if (balanced == 1) {
17    nn1 = nn2 = nn3 = ceiling(nn * 0.3)
18  }
19  else {
20    nn1 = ceiling(nn * 0.25)
21    nn2 = ceiling(nn * 0.3)
22    nn3 = ceiling(nn * 0.35)
23  }
24  if (symmetry == 1){
25    cc1 <- -20
26    cc2 <- -50
27  } else {
28    cc1 <- 0
29    cc2 <- 0
30  }
31
32  nh <- nn1 + nn2 + nn3
33  Y <- matrix(NA, ncol = p, nrow = n)
34  if (scenario == 1) {
35    if (G == 3) {
36      Y[1:nn1, ] <- MASS::mvrnorm(nn1, c(4, 4), matrix(c(1, 0, 0, 1), nrow = 2))
37      Y[(nn1 + 1):(nn1 + nn2), ] <- MASS::mvrnorm(nn2, c(-4, 4), matrix(c(1, 0, 0, 1), nrow = 2))
38      Y[(nn1 + nn2 + 1):(nn1 + nn2 + nn3), ] <- MASS::mvrnorm(nn3, c(0, 0), matrix(c(1, 0, 0, 1),
39        nrow = 2))
40      Y[(nh + 1):n, ] <- matrix(runif(p * (n - nh), cc1, 20), ncol = p)
41      if (p > 2) {
42        Y[1:n, 3:p] <- MASS::mvrnorm(n, rep(0, p - 2), diag(p - 2))
43      }
44    }
45    else {
46      Y[1:nn1, ] <- MASS::mvrnorm(nn1, c(4, 4), matrix(c(1, 0, 0, 1), nrow = 2))
47      Y[(nn1 + 1):(nn1 + nn2), ] <- MASS::mvrnorm(nn2, c(-4, 4), matrix(c(1, 0, 0, 1), nrow = 2))
48      Y[(nn1 + nn2 + 1):(nn1 + nn2 + nn3), ] <- MASS::mvrnorm(nn3, c(0, 0), matrix(c(1, 0, 0, 1),
49        nrow = 2))
50      Y[(nh + 1):(nh + nn1), ] <- MASS::mvrnorm(nn1, c(24, 0), matrix(c(1, 0, 0, 1), nrow = 2))
51      Y[(nh + nn1 + 1):(nh + nn1 + nn2), ] <- MASS::mvrnorm(nn2, c(16, 0), matrix(c(1, 0, 0, 1),
52        nrow = 2))
53      Y[(nh + nn1 + nn2 + 1):(nh + nn1 + nn2 + nn3), ] <- MASS::mvrnorm(nn3, c(20, 4),
54        matrix(c(1, 0, 0, 1), nrow = 2))
55      Y[(2 * nh + 1):n, ] <- matrix(runif((n - 2 * nh) * p, cc1, 40), ncol = p)
56    }
57  }
```

```

52   if (p > 2) {
53     Y[1:n, 3:p] <- MASS::mvrnorm(n, rep(0, p - 2), diag(p - 2))
54   }
55 }
56 }
57 if (scenario == 2) {
58   alp1 <- runif(1, 0, 2 * pi)
59   u1 <- matrix(c(cos(alp1), sin(alp1), -sin(alp1), cos(alp1)), ncol = 2)
60   alp2 <- runif(1, 0, 2 * pi)
61   u2 <- matrix(c(cos(alp2), sin(alp2), -sin(alp2), cos(alp2)), ncol = 2)
62   if (G == 3) {
63     Y[1:nn1, ] <- MASS::mvrnorm(nn1, c(20, 20), t(u1) %*% diag(c(1, 9^2)) %*% u1)
64     Y[(nn1 + 1):(nn1 + nn2), ] <- MASS::mvrnorm(nn2, c(-20, -20), t(u1) %*% diag(c(9^2, 1)) %*%
65       u1)
66     Y[(nn1 + nn2 + 1):(nn1 + nn2 + nn3), ] <- MASS::mvrnorm(nn3, c(0, 0), t(u1) %*%
67       diag(c(3^2, 3^2)) %*% u1)
68     Y[(nn1 + nn2 + nn3 + 1):n, ] <- matrix(runif(p * (n - nh), cc2, 50), ncol = p)
69     if (p > 2) {
70       Y[1:n, 3:p] <- MASS::mvrnorm(n, rep(0, p - 2), diag(p - 2))
71     }
72   } else {
73     Y[1:nn1, ] <- MASS::mvrnorm(nn1, c(20, 20), t(u1) %*% diag(c(1, 9^2)) %*% u1)
74     Y[(nn1 + 1):(nn1 + nn2), ] <- MASS::mvrnorm(nn2, c(-20, -20), t(u1) %*% diag(c(9^2, 1)) %*%
75       u1)
76     Y[(nn1 + nn2 + 1):(nn1 + nn2 + nn3), ] <- MASS::mvrnorm(nn3, c(0, 0), t(u1) %*% diag(c(3^2,
77       3^2)) %*% u1)
78     Y[(nh + 1):(nh + nn1), ] <- MASS::mvrnorm(nn1, c(40, 20), t(u2) %*% diag(c(1, 9^2)) %*% u2)
79     Y[(nh + nn1 + 1):(nh + nn1 + nn2), ] <- MASS::mvrnorm(nn2, c(0, -20), t(u2) %*% diag(c(9^2,
80       1)) %*% u2)
81     Y[(nh + nn1 + nn2 + 1):(nh + nn1 + nn2 + nn3), ] <- MASS::mvrnorm(nn3, c(20, 0), t(u2) %*%
82       diag(c(3^2, 3^2)) %*% u2)
83     Y[(2 * nh + 1):n, ] <- matrix(runif((n - 2 * nh) * p, cc2-10, 80), ncol = p)
84     if (p > 2) {
85       Y[1:n, 3:p] <- MASS::mvrnorm(n, rep(0, p - 2), diag(p - 2))
86     }
87   }
88 }
89 }
90 if (scenario == 3) {
91   if (G == 3) {
92     Y[1:nn1, ] <- MASS::mvrnorm(nn1, c(5, 5), matrix(c(1, 0, 0, 1), nrow = 2))
93     Y[(nn1 + 1):(nn1 + nn2), ] <- MASS::mvrnorm(nn2, c(-15, -15), matrix(c(1, 0, 0, 1), nrow =
94       2))
95     Y[(nn1 + nn2 + 1):(nn1 + nn2 + nn3), ] <- MASS::mvrnorm(nn3, c(-15, 15), matrix(c(1, 0, 0,
96       1), nrow = 2))
97     Y[(nh + 1):n, ] <- matrix(runif(p * (n - nh), cc1, 20), ncol = p)
98     if (p > 2) {
99       Y[1:n, 3:p] <- MASS::mvrnorm(n, rep(0, p - 2), diag(p - 2))
100     }
101   } else {
102     Y[1:nn1, ] <- MASS::mvrnorm(nn1, c(5, 5), matrix(c(1, 0, 0, 1), nrow = 2))
103     Y[(nn1 + 1):(nn1 + nn2), ] <- MASS::mvrnorm(nn2, c(-15, -15), matrix(c(1, 0, 0, 1), nrow =
104       2))
105     Y[(nn1 + nn2 + 1):(nn1 + nn2 + nn3), ] <- MASS::mvrnorm(nn3, c(-15, 15), matrix(c(1, 0, 0,
106       1), nrow = 2))
107     Y[(nh + 1):(nh + nn1), ] <- MASS::mvrnorm(nn1, c(10, -10), matrix(c(1, 0, 0, 1), nrow = 2))
108     Y[(nh + nn1 + 1):(nh + nn1 + nn2), ] <- MASS::mvrnorm(nn2, c(6, -15), matrix(c(1, 0, 0,
109       1), nrow = 2))
110     Y[(nh + nn1 + nn2 + 1):(nh + nn1 + nn2 + nn3), ] <- MASS::mvrnorm(nn3, c(15, -15),
111       matrix(c(1, 0, 0, 1), nrow = 2))
112     Y[(2 * nh + 1):n, ] <- matrix(runif((n - 2 * nh) * p, cc1, 20), ncol = p)
113     if (p > 2) {
114       Y[1:n, 3:p] <- MASS::mvrnorm(n, rep(0, p - 2), diag(p - 2))
115     }
116   }
117 }
118 }
119 true <- rep(0, n)
120 if (G == 3) {
121   true[1:nh] <- rep(1:3, c(nn1, nn2, nn3))
122 }
123 else {
124   true[1:(2 * nh)] <- rep(1:6, c(nn1, nn2, nn3, nn1, nn2, nn3))
125 }
126 return(list(x = Y, true = true))
127 }

```

2 Additional simulation results

We have considered 100 random realizations of `simula_tclust()` for the parameter combinations discussed in the manuscript, but with configurations of `balance` and `symmetry` that were not included there. The results obtained in all cases are in perfect agreement with those reported in the manuscript. `s=1`, `s=2` and `s=3` refer to the 3 scenarios (`scenario=1, 2` and `3`).



Figure A.1: Target function values when `balance=1` and `symmetry=2`.



Figure A.2: Misclassification rates when `balance=1` and `symmetry=2`.

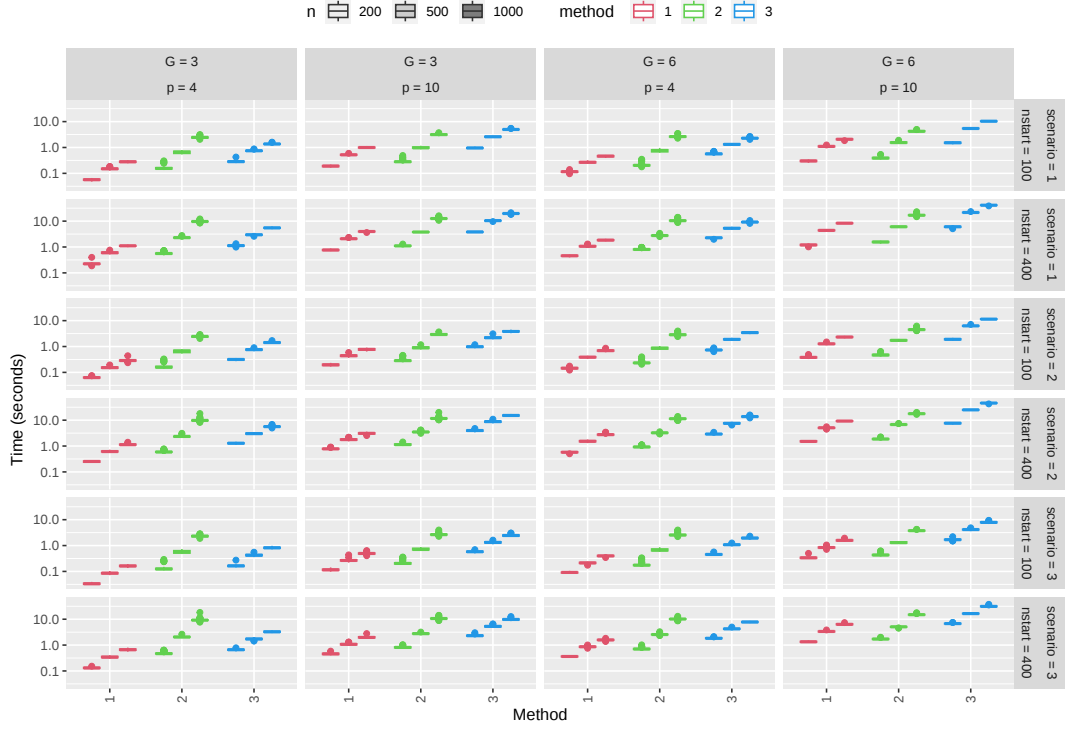


Figure A.3: Computing times in seconds (in logarithmical scale) when **balance=1** and **symmetry=2**.

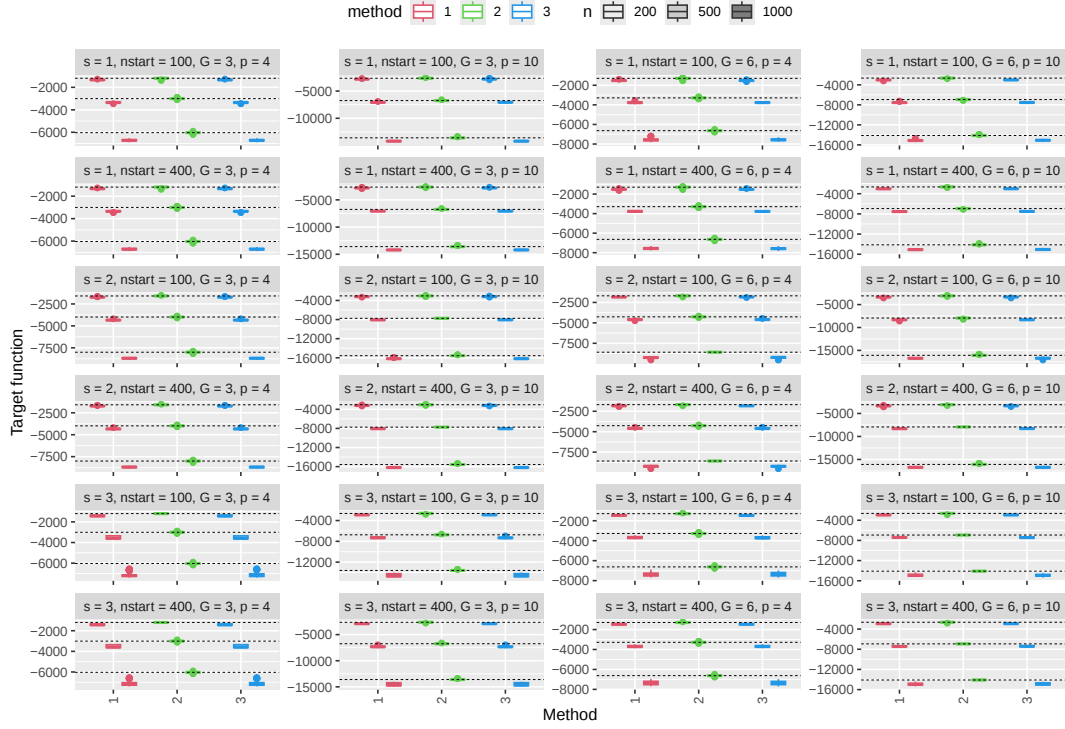


Figure A.4: Target function values when **balance=2** and **symmetry=1**.

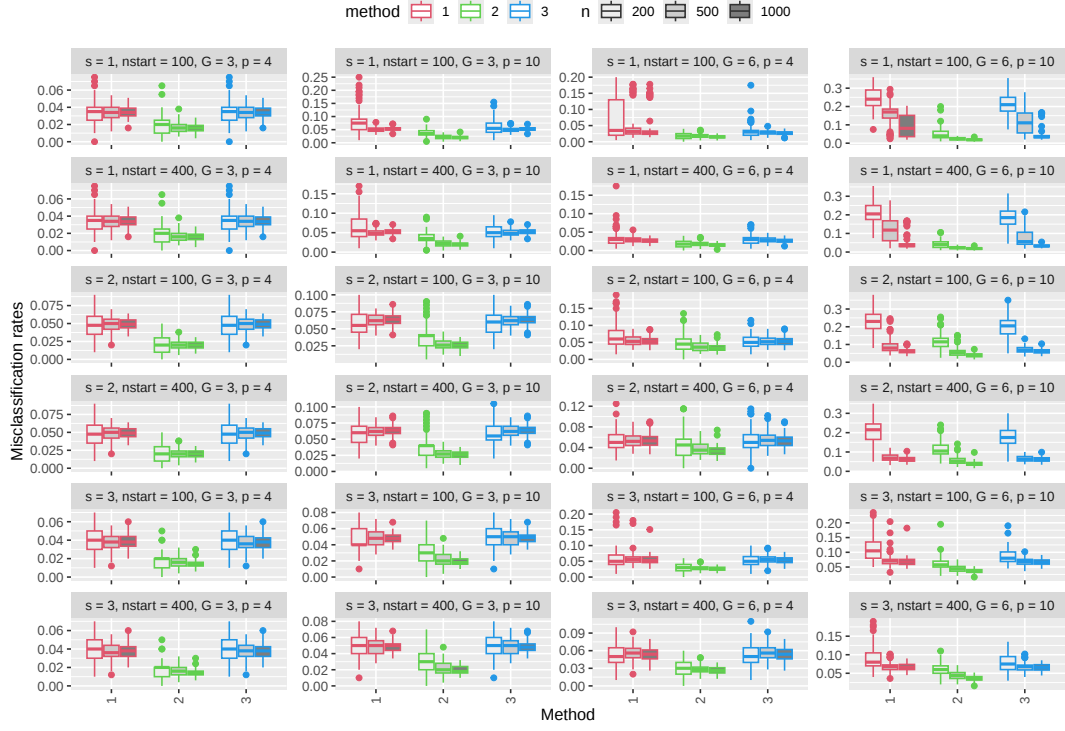


Figure A.5: Misclassification rates when balance=2 and symmetry=1.

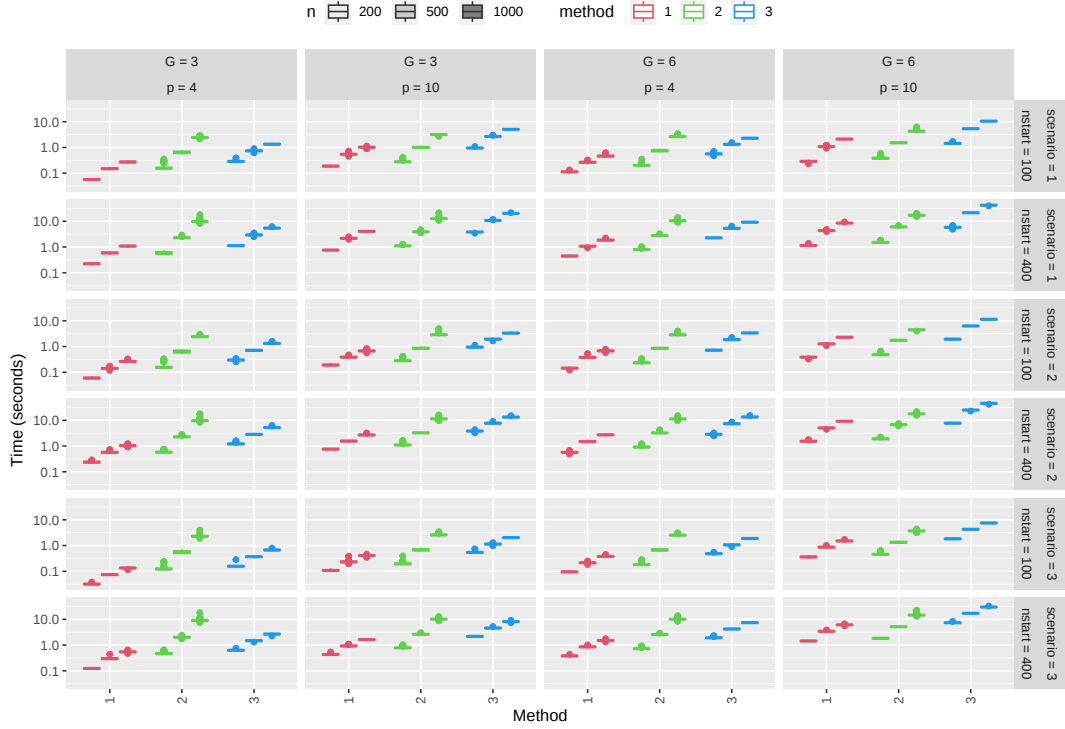


Figure A.6: Computing times in seconds (in logarithmical scale) when balance=2 and symmetry=1.



Figure A.7: Target function values when $\text{balance}=2$ and $\text{symmetry}=1$.

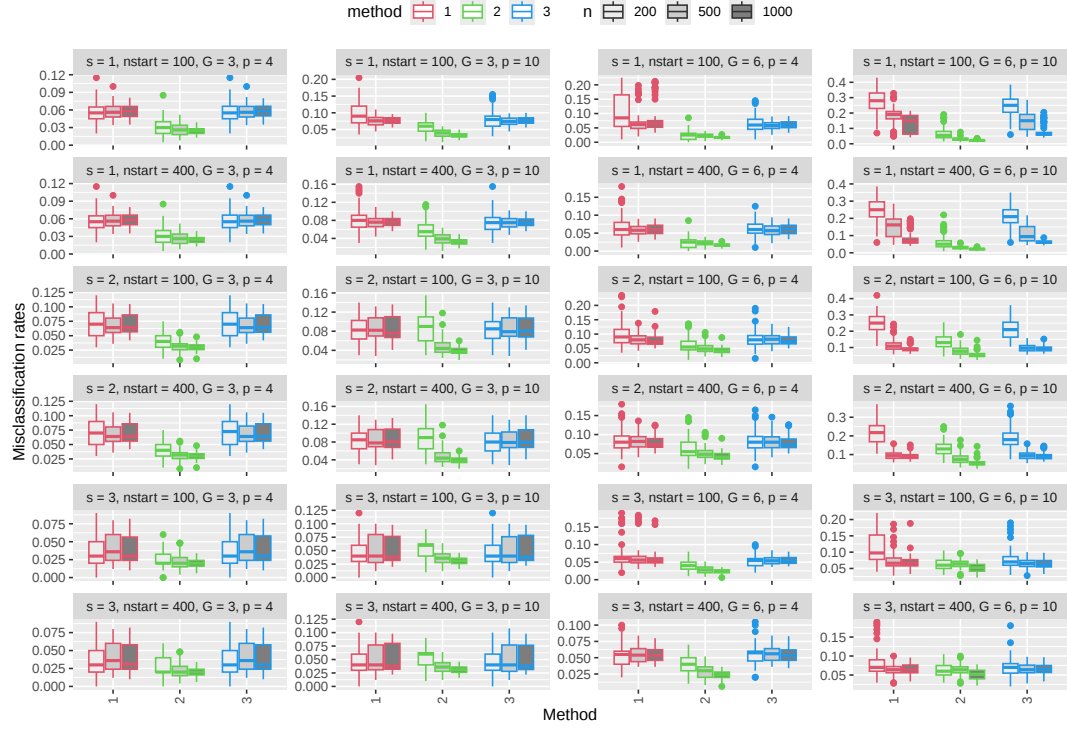


Figure A.8: Misclassification rates when $\text{balance}=2$ and $\text{symmetry}=2$.

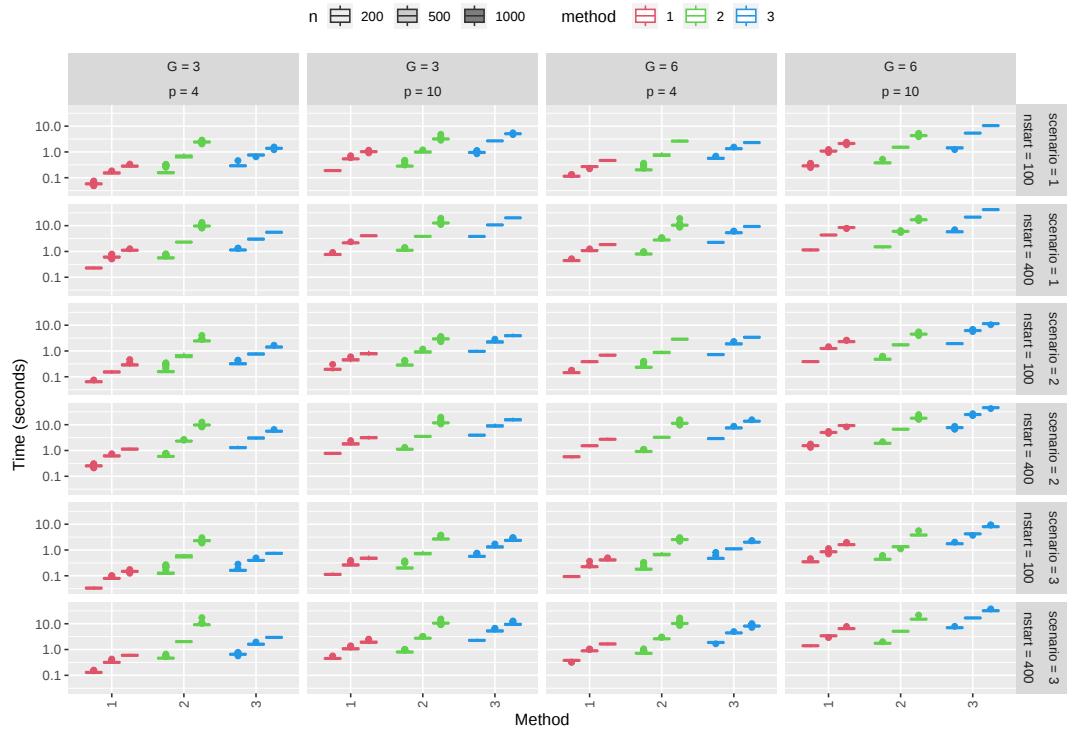


Figure A.9: Computing times in seconds (in logarithmical scale) when `balance=2` and `symmetry=2`.

3 Subsampling and ensemble initialization with $n = 200000$

Results of *Method 1* and *Method 2* for 100 datasets generated by `simula.tclust(n=200000,p,G,scenario,balanced=1,symmetry=1)`. *Method 2* refers to the proposed “Combining subsampling and ensemble initialization” approach with $n_0 = 800$ (notably smaller than $n = 200000$). Results for *Method 3* with $5 \times \text{nstart}$ are not provided due to the substantial computing times required.

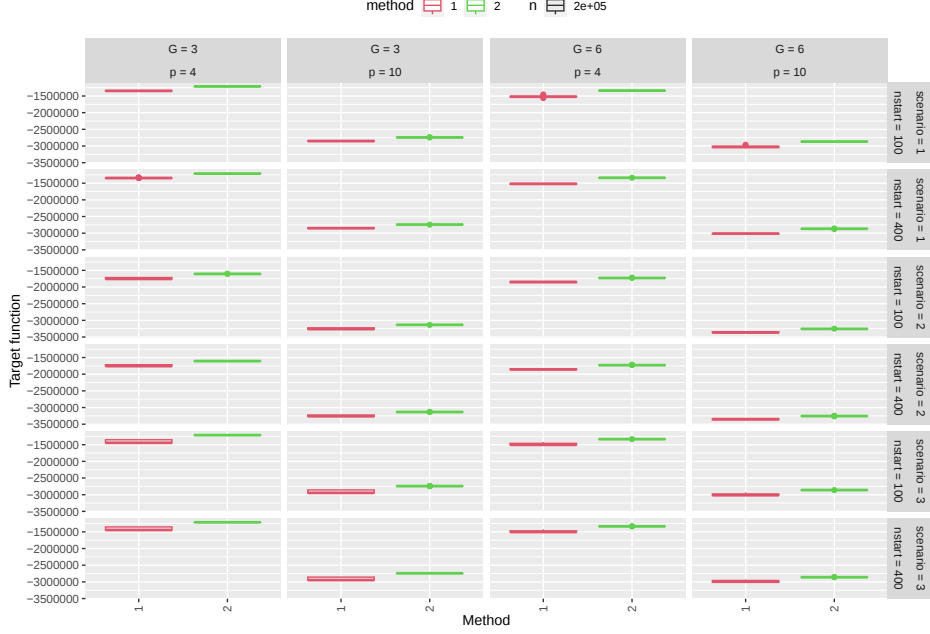


Figure A.10: Target function values when combining subsampling and ensemble initialization with $n = 200000$.

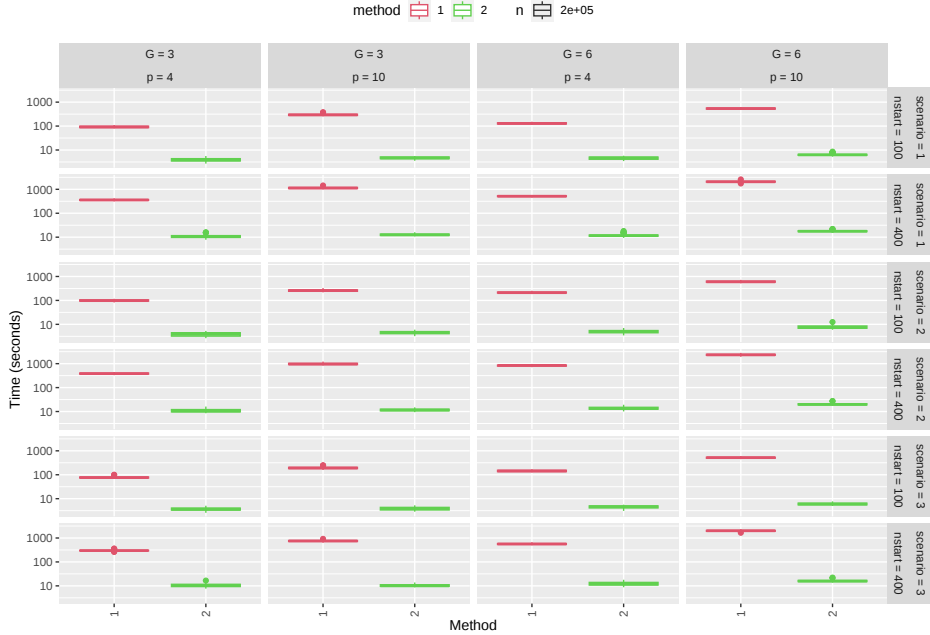


Figure A.11: Computing times (in logarithmic scale) when combining subsampling and ensemble initialization with $n = 200000$.

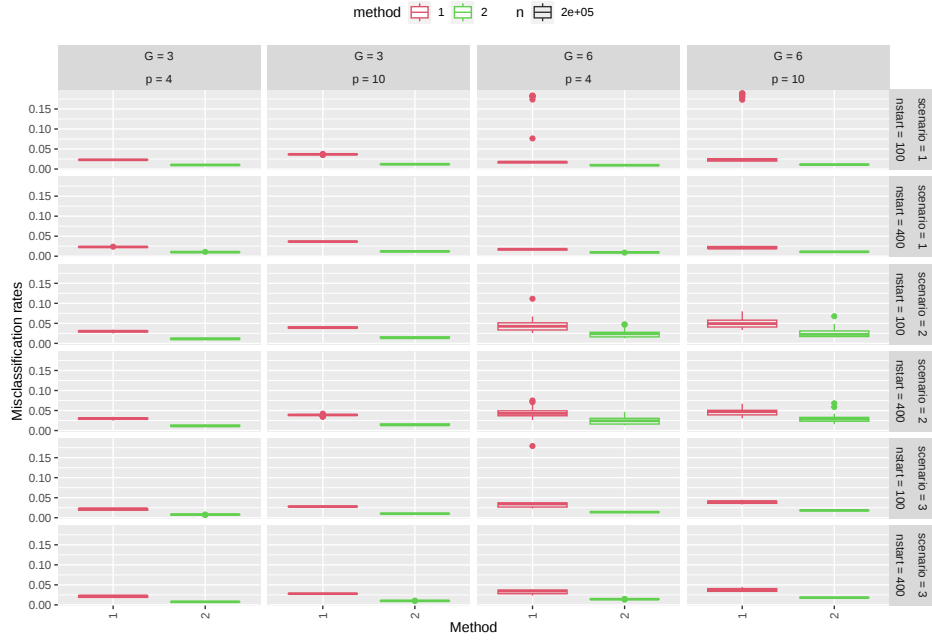


Figure A.12: Misclassification rates when combining subsampling and ensemble initialization with $n = 200000$.

4 Additional results for the “olive oil” dataset

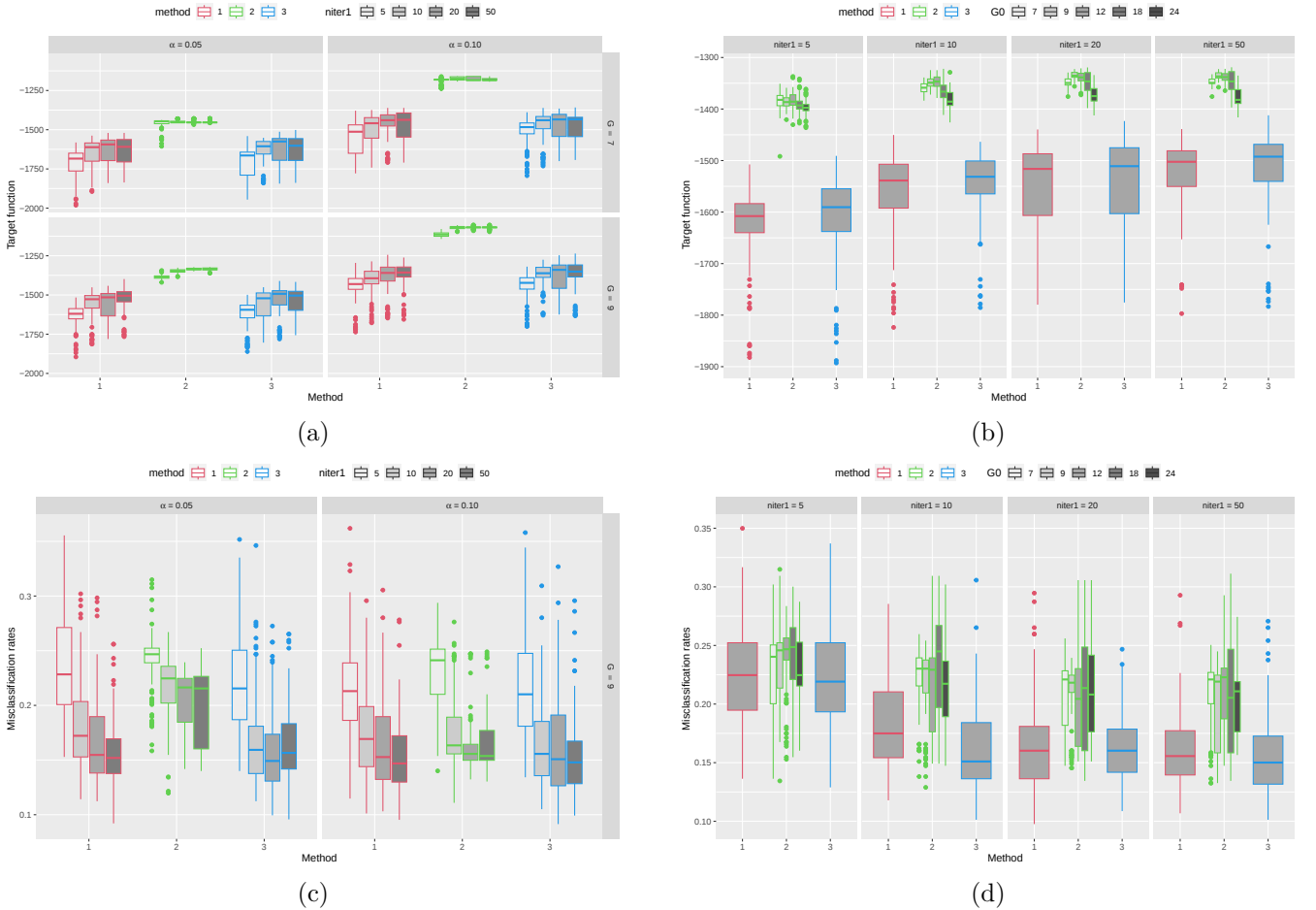
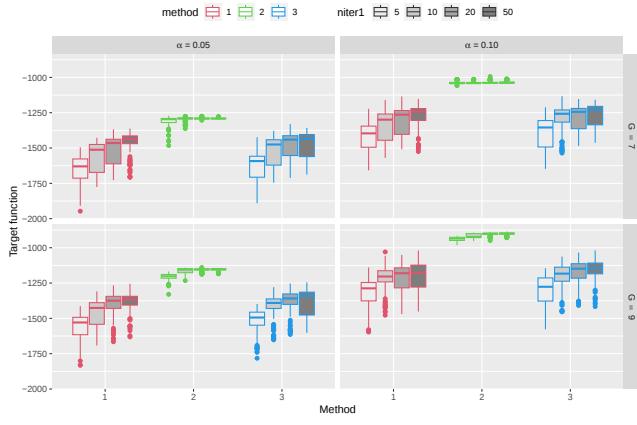
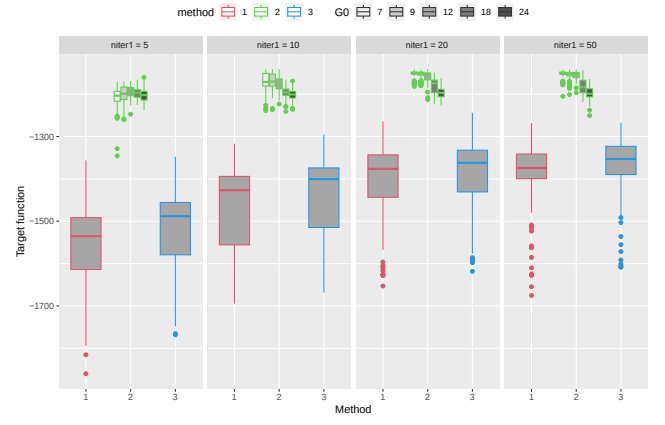


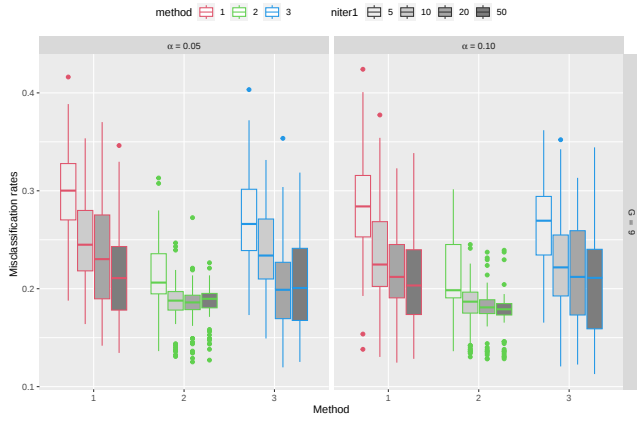
Figure A.13: Results of the experiments as in Section 6 of the manuscript when the restriction factor is set to $c = 128$.



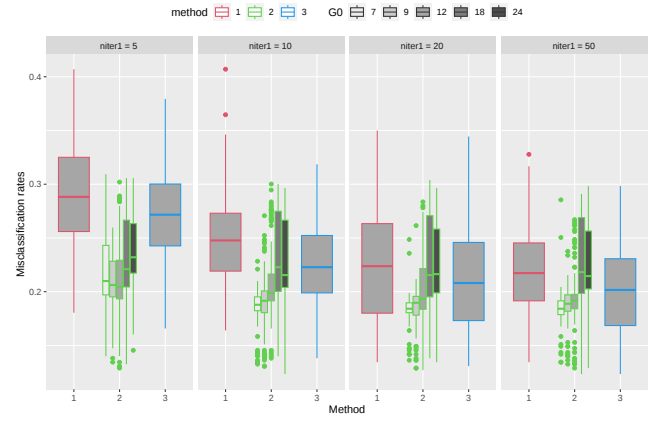
(a)



(b)



(c)



(d)

Figure A.14: Results of the experiments as in Section 6 of the manuscript when the restriction factor is set to $c = 1000$.