CONTAMINATION-SOURCE BASED K-SAMPLE CLUSTERING

Xavier Milhaud, Denys Pommeret, Yahia Salhi, Pierre Vandekerkhove

CONTAMINATION-SOURCE BASED K-SAMPLE CLUSTERING

Xavier Milhaud,[(1)] Denys Pommeret,[(1,2)] Yahia Salhi[(2)] and Pierre Vandekerkhove[(3)]

[(1)]Aix-Marseille University, Campus de Luminy, 13288 Marseille cedex 9, France

[(2)]University Claude Bernard Lyon 1, UCBL, ISFA LSAF EA2429, F-69007, Lyon, France

[(3)]University Gustave Eiffel, LAMA (UMR 8050), 77420 Champs-sur-Marne, France

June 14, 2023

Abstract

We investigate in this work the K-sample clustering of populations issued from contamination phenomenon. A contamination model is a two-component mixture model in which one component is known (standard behaviour) when the second one, modelling a departure from the standard behaviour, is unknown. When K populations from such a model are observed we propose a semiparametric clustering methodology to detect, for coordinated diagnosis and/or best practices sharing purpose, which populations are impacted by the same type of contamination. We prove the consistency of our approach under the existence of true clusters and show the performances of our methodology through an extensive Monte Carlo study. We finally apply our methodology, implemented in the `admix`[1] R package, to a European countries COVID-19 excess of mortality dataset for which we aim to cluster countries similarly impacted by the pandemic over classes of age.

AMS 2000 subject classifications. Primary 62G05, 62G20; secondary 62E10.
Keywords. Admixture; Clustering; Contamination; Hypothesis Testing; Semiparametric Mixture.

[1]See `https://CRAN.R-project.org/package=admix` for more information about the package on CRAN.

1 Introduction

Let consider the two-component mixture model with cumulative distribution function

$$L(x) = (1-p)G(x) + pF(x), \tag{1}$$

for all $x \in \mathbb{R}$, where G is a known cumulative distribution function and the unknown parameters are the mixture proportion $p \in]0,1[$ along with the cumulative distribution function of the unknown component F. This model, usually called contamination or admixture model, has been largely studied in the last decades and is related to various applications, see Shen et al. (2018) for a good survey about this topic. This model is of particular interest when considering generic situations distorted by an unexpected event, such as: i) the mortality excess due to the COVID-19 crisis, see Milhaud et al. (2023); ii) the presence of diseased tissues in microarray analysis, see McLachlan et al. (2006), iii) variables observation, such as metallicity and radial velocity of stars, in the background of the Milky Way, see Walker et al. (2009); iv) trees diameters modelling in presence of extra varieties, see Podlaski and Roesch (2014). In this paper, the data of interest is made of $K \geq 2$ independent samples $X^{(i)} = (X_1^{(i)}, \ldots, X_{n_i}^{(i)})$, for $i = 1, \ldots, K$, which are assumed to be internally independent and identically distributed with respective cumulative distribution function

$$L_i(x) = (1-p_i)G_i(x) + p_i F_i(x), \tag{2}$$

for all $x \in \mathbb{R}$, where the p_i's and the F_i's are respectively the unknown mixture proportions and the cumulative distribution function of the unknown component associated to the ith sample. In practice, the G_i's are associated to a well known population, as for instance in the real-life mortality excess due to COVID-19 application of Section 7.1, the historical national mortality profile for a given country, when the unknown F_i's are associated to a new subpopulation, which can be the specific mortality profile associated to the pandemic. Generally speaking, the F_i's represent a raising phenomena not modelled yet which makes this model of particular interest for generic crisis or population transformation modelling.

The aim of this paper is to provide a clustering methodology to detect subgroups among the K existing samples having possibly similar unknown, sometimes called nodular, components for obvious coordinated diagnosis and/or best practices sharing interest, based on the type of contamination impacting each population. To answer this original problem we will adopt a testing approach in the sense that samples will be collected in the same group if the equality of their unknown component according to an ad. hoc. test and a level $0 < \alpha < 1$ (to be setup) cannot be rejected. More formally, we suppose the existence of N true clusters denoted by $\mathcal{G}_s$, $1 \leq s \leq N \leq K$ and defined by

$$\mathcal{G}_s = \left\{ i_{s,1} \leq j \leq K : \ F_j = F_{i_{s,1}} \right\}, \tag{3}$$

where we denote by $F_{i_{s,1}}$ the first representative of group $\mathcal{G}_s$, when running increasingly through the set of indices $\{1, \ldots, K\}$, in the family of nodular components $\mathcal{N} = \{F_i, \ i = 1, \ldots, K\}$. For convenience we will also denote by n_s the cardinal of $\mathcal{G}_s$ giving the opportunity to number the elements of $\mathcal{G}_s$ as follows $\mathcal{G}_s = \{i_{s,1}, i_{s,2}, \ldots, i_{s,n_s}\}$. Obviously we have the following partition

$$\{1, \ldots, K\} = \cup_{s=1}^{N} \mathcal{G}_s, \quad \mathcal{G}_s \cap \mathcal{G}_{s'} = \emptyset, \qquad (1 \leq s \neq s' \leq N), \tag{4}$$

with the group separation assumption given below.

(GS) There exists real-sets $A_{s,s'} \subseteq \mathbb{R}$ with $\mu(A_{s,s'}) \neq 0$ such that for all $x \in A_{s,s'}$:

$$F_{i_{s,1}}(x) \neq F_{i_{s',1}}(x), \qquad (1 \leq s \neq s' \leq N),$$

where μ denotes a reference measure on the support of the F_i's (e.g. Lebesgue measure over $\mathbb{R}$, counting measure over $\mathbb{N}$). Given the above framework our clustering strategy will consist in identifying recursively over $s \in \{1, \ldots, N\}$ the first representative $F_{s,1}$ of group $\mathcal{G}_s$ along with the whole group itself. In this work, similarly to Patra and Sen (2016) or Milhaud et al. (2023), we will consider situations where the G_i's and F_i's distributions are: either i) absolutely continuous with respect to the Lebesgue measure, supported over $\mathbb{R}$, $\mathbb{R}^+$ or intervals of $\mathbb{R}$; or ii) finite discrete or $\mathbb{N}$-discrete distributions such as Poisson or Binomial. All our results will be still valid in such setups provided that the G_i's are all distinct. If certain pairs (G_i, G_j), $1 \leq i \neq j \leq K$, are possibly equal a distinct procedure will then be implemented, see for details Appendix D of the Supplement in . Given the above model, we will have to answer first the basic statistical problem

$$H_0 : \ F_1 = \cdots = F_k \qquad \text{against} \qquad H_1 : \ F_i \neq F_j \text{ for some } 1 \leq i \neq j \leq k \ , \tag{5}$$

without assigning any specific parametric family to the F_i's. Our clustering methodology will be grounded on the above k-sample testing problem with the k value possibly evolving from 1 to K along an algorithm scheme. When $k = 2$ the above problem has been addressed in Milhaud et al. (2022) under restrictive shape constraints such as the zero-symmetry of the F_i's. More recently the two-sample testing problem has been revisited by Milhaud et al. (2023), who propose the so-called IBM (Inversion-Best Matching) testing approach requiring very relaxed identifiability and regularity conditions making, as a consequence, this methodology much more suitable for real-life applications. Our contribution is twofold: i) on the one hand we aim to generalize the work of Milhaud et al. (2023) to the k-sample case, when k is greater than 2; ii) on the other hand our objective is to derive a handy clustering algorithm grounded on the previous k-sample testing procedure, as described in (3)–(4). For that purpose we develop a data-driven methodology, inspired from Schwarz (1978) or Kallenberg and Ledwina (1995), allowing to select the most different populations pairs among all the possible pairs. More precisely we introduce the following set of pair indices: $\mathcal{S}(k) = \{(i,j) \in \mathbb{N}^2; 1 \leq i < j \leq k\}$. Clearly $\mathcal{S}(k)$ contains $d(k) = k(k-1)/2$ elements that can be lexicographically ordered as follows: we denote $(i,j) < (i',j')$ if $i < i'$, or if $i = i'$ and $j < j'$, and we denote by $r_k[(i,j)]$ the associated rank of (i,j) in the set $S(k)$. This ordering will be used to sum the test statistics over all the pairs of populations, and can be considered as the natural ordering over the elements of an upper triangle $k \times k$ matrix. For instance we have across the first row $r_k[(1,2)] = 1$, $r_k[(1,3)] = 2$, and so on, when across the second row we have $r_k[(2,3)] = k$, $r_k[(2,4)] = k+1$, and so on. For $(i,j) \in \mathcal{S}(k)$ we denote by $T_{i,j}$ the two-sample statistic used in Milhaud et al. (2023) to compare populations i and j, for $1 \leq i \neq j \leq k$. For simplicity matters, we drop the dependence on n since the statistic $T_{i,j}$ stands for T_n defined in the paragraph following expression (12) in Milhaud et al. (2023).

We can then build-up a sequence of statistics by slicing the set of index as follows: for slices s numbered from 1 to $k-1$, we define couples of index delimiters $(b_s^-, b_s^+) = \big(1 + (s-1)k - \frac{s(s-1)}{2}), sk -$

$\frac{s(s+1)}{2}\big)$ with $b_{s+1}^- = b_s^+ + 1$. This enables to define the sequence of embedded statistics U_r, the n-dependence dropped again for simplicity matters, as follows

$$\begin{aligned}
\text{slice } 1: \quad U_r &= \sum_{i=1}^{r} T_{1,1+i}, \qquad (b_1^- = 1 \le r \le k-1 = b_1^+),\\
\text{slice } 2: \quad U_r &= U_{k-1} + \sum_{i=1}^{r-(k-1)} T_{2,2+i}, \qquad (b_2^- = k \le r \le 2k-3 = b_2^+),\\
&\vdots \\
\text{slice } s: \quad U_r &= U_{b_{s-1}^+} + \sum_{i=1}^{r-(b_s^- -1)} T_{s,s+i}, \qquad (b_s^- \le r \le b_s^+),\\
&\vdots \\
\text{slice } k-1: \quad U_r &= U_{b_{k-2}^+} + T_{k-1,k}, \qquad (r = d(k)).
\end{aligned} \tag{6}$$

By construction U_1 compares the first two populations $(1,2)$, U_2 compares simultaneously the first two pairs of populations $(1,2)$ and $(1,3)$, and more generally U_r with r in slice of index s compares simultaneously the populations from 1 to $s-1$ with populations of upper ranks pairwizely through $U_{b_{s-1}^+}$ and population s with upper ranks lying in $\{s+1,\ldots,s+r-(b_s^- -1)\}$ through $\sum_{i=1}^{r-(b_s^- -1)} T_{s,s+i}$. Clearly, since the test statistics $T_{i,j}$ are positive, each statistic U_r is a sum of such r positive quantities and we have with probability 1 that $U_1 \le \cdots \le U_{d(k)}$. We then propose a penalized rule inspired from Schwarz (1978) criteria to select the most sensitive rank r given by $S(n)$ in expressions (8) or (9) of Section 3.1 . Under the null, we prove that the asymptotic limit distribution of our procedure coincides with the one obtained in the two sample case given by the less penalized statistic $T_{1,2}$. It is also shown that our test statistic goes to infinity with n under the alternative. Our procedure is then adapted to construct a data-driven clustering algorithm able to classify the populations with equal unknown components. In order to pre-select a natural cluster to be tested by the k-sample test, we investigate the "closest" populations based on their pairwise associated (distance-based) statistics. We propose in addition a self-tuning method for the penalization term involved in our k-sample test statistic that yields to an automated and easy to implement clustering procedure. The only required parameter is the asymptotic test level used to accept or not a cluster. This method is illustrated through an extensive Monte Carlo experiment including very diverse situations and applied to a real life dataset dealing with the post COVID-19 mortality rates across a panel of 29 European countries.

The paper is organized as follows: In Section 2 we review recent results about the two-sample case making the paper self-contained. Section 3 is devoted to the penalized testing rule and contains the main results of the paper. In Section 4 we develop a tuning method making our approach data-driven. The clustering algorithm is described in Section 5. Section 6 is devoted to an extensive simulation study covering the empirical level and power behaviour of our k-sample test procedure along with the numerical performances of our test-based clustering method. Section 7 ends the paper with a study dealing with the excess of mortality due to COVID-19 over a panel of European countries during the early times of the pandemic. The proofs of our theorems and proposition are relegated in Appendix.

2 Mathematical background

In this section along with Section 3 we consider k, $2 \leq k \leq K$, samples among the K original samples still denoted for simplicity and without loss of generality $X^{(i)} = (X_1^{(i)}, \ldots, X_{n_i}^{(i)})$, for $i = 1, \ldots, k$. In the spirit of Milhaud et al. (2023), we consider $n = \min_{1 \leq i \leq k} n_i$, and define $\kappa_i \geq 1$ such that $n_i = \kappa_i n$, for all $i = 1, \ldots, k$. For $i \neq j \in \{1, \ldots, k\}$, we denote $\theta_{ij} = (p_i, p_j) \in \Theta_i \times \Theta_j$ the pair of unknown proportions associated to the ith and jth populations, respectively.

(A0) Assume that Θ_i is a $[\delta_1, \delta_2]$-type compact set satisfying $0 < \delta_1 < 1 < \delta_2$, for all $i = 1, \ldots, k$.

Similarly to Milhaud et al. (2023), we notice that the unknown component associated with sample i can be recovered under the correct parameter p_i by using, for all $x \in \mathbb{R}$, the following inversion formula

$$F_i(x, L_i, p_i) = \frac{L_i(x) - (1 - p_i)G_i(x)}{p_i}, \qquad (i = 1, \ldots, k).$$

To compare populations i and j we define the sub-(i, j) testing problem

$$H_0(i, j) : \ F_i = F_j \qquad \text{against} \qquad H_1(i, j) : \ F_i \neq F_j \ ,$$

and consider the following discrepancy measure and its empirical counterpart

$$\begin{aligned} d[i, j](\theta_{ij}) &= \int_{\mathbb{R}} \Big(F_i(x, L_i, p_i) - F_j(x, L_j, p_j) \Big)^2 dH(x) \\ d_n[i, j](\theta_{ij}) &= \int_{\mathbb{R}} \Big(F_i(x, \widehat{L}_i, p_i) - F_j(x, \widehat{L}_j, p_j) \Big)^2 dH(x), \end{aligned}$$

for $\theta_{ij} = (p_i, p_j)$ fixed in $\Theta_{ij} = \Theta_i \times \Theta_j$, where H is a positive measure over $\mathbb{R}$ that allows to weight the square of the difference between F_i and F_j along the real line, and $\widehat{L}_i$ denotes the empirical cdf associated to the sample $X^{(i)}$. In practice we choose for H a uniform distribution when the support of the L_i's is bounded or a probability distribution having a density supported by $\mathbb{R}$ in the unbounded case, see also Appendix F of the Supplement in Milhaud et al. (2023) for further discussion about the choice of H. In the discrete case we simply choose for H the counting measure over the observations support.

We introduce now two assumptions connected to the identifiability and definite positiveness of the d-Hessian matrix. These assumptions are based on a cross-model identifiability condition inspired from the identifiability Theorem 1 in Teicher (1963).

(A1) Under $H_0(i, j)$ ($F_i = F_j = F_{ij}$), there exists at least three points $(x_1[i, j], x_2[i, j], x_3[i, j]) \in \mathbb{R}^3$ such that

$$\det \begin{pmatrix} G_i(x_1[i,j]) & G_j(x_1[i,j]) & F_{ij}(x_1[i,j]) \\ G_i(x_2[i,j]) & G_j(x_2[i,j]) & F_{ij}(x_2[i,j]) \\ G_i(x_3[i,j]) & G_j(x_3[i,j]) & F_{ij}(x_3[i,j]) \end{pmatrix} \neq 0.$$

(A2) Under $H_1(i,j)$ ($F_i \neq F_j$) , there exists at least four points $(x_1[i,j], x_2[i,j], x_3[i,j], x_4[i,j]) \in \mathbb{R}^4$ such that

$$\det \begin{pmatrix} G_i(x_1[i,j]) & G_j(x_1[i,j]) & F_i(x_1[i,j]) & F_j(x_1[i,j]) \\ G_i(x_2[i,j]) & G_j(x_2[i,j]) & F_i(x_2[i,j]) & F_j(x_2[i,j]) \\ G_i(x_3[i,j]) & G_j(x_3[i,j]) & F_i(x_3[i,j]) & F_j(x_3[i,j]) \\ G_i(x_4[i,j]) & G_j(x_4[i,j]) & F_i(x_4[i,j]) & F_j(x_4[i,j]) \end{pmatrix} \neq 0.$$

The above pairwise-model conditions are stated and discussed in Milhaud et al. (2023).

For all $i \neq j \in \{1, \ldots, k\}$ we consider

$$\widehat{\theta}_{ij} = \arg \min_{\theta_{ij} \in \Theta_{ij}} d_n[i,j](\theta_{ij}),$$

which is the estimated pair of parameters (p_i, p_j) that makes the unknown components F_i and F_j look the more similar according to the d discrepancy measure, which is then basically evaluated by

$$d_n[i,j](\widehat{\theta}_{ij}) = \int_{\mathbb{R}} \left(F_i(x, \widehat{L}_i, \widehat{p}_i) - F_j(x, \widehat{L}_j, \widehat{p}_j) \right)^2 dH(x).$$

Remark 1. *As described in Milhaud et al. (2023), under $H_0(i,j)$, $\widehat{\theta}_{ij} \to \theta^*_{ij} = (p^*_i, p^*_j)$ almost surely, with $d(\theta^*_{ij}) = 0$, where p^*_i and p^*_j are respectively the true value of the proportions involved in the $X^{(i)}$ and $X^{(j)}$ models, see expression (2). In contrast under $H_1(i,j)$, $\widehat{\theta}_{ij} \to \theta^c_{ij} = (p^c_i, p^c_j)$ almost surely, a local minima of $\theta \mapsto d(\theta)$ with $d(\theta^c_{ij}) > 0$ and generally $\theta^c_{ij} \neq \theta^*_{ij}$.*

We recall here the main result of Milhaud et al. (Theorem 2, 2023) that we use to construct our k-sample test. For $(i,j) \in \mathcal{S}(k)$ we consider

$$T_{i,j} = n d_n[i,j](\widehat{\theta}_{ij}), \tag{7}$$

the estimator of the n-discrepancy measure between population i and j, where $\widehat{\theta}_{ij} = (\widehat{p}_i, \widehat{p}_j)$.

Lemma 1. *Assume that* **(A1-2)** *hold.*

i) Then under $H_0(i,j)$, the statistic $T_{i,j} = U^0_n(i,j)$ converges in distribution towards $U^0(i,j)$, as $n \to +\infty$, where the limiting random variable $U^0(i,j)$ is fully identified (closed form stochastic integral) and tabulated.

ii) Then under $H_1(i,j)$, the statistic $T_{i,j} = U^1_n(i,j) + V^1_n(i,j)$, where $U^1_n(i,j)$ converges in distribution towards $U^1(i,j)$, as $n \to +\infty$, where the limiting random variable $U^1(i,j)$ is fully identified and tabulated when $V^1_n(i,j) = \lambda[i,j] \times n + o_{a.s.}(n)$ is a drift term, where

$$\lambda[i,j] = \int_{\mathbb{R}} \left(F_i(x, L_i, p^c_i) - F_j(x, L_j, p^c_j) \right)^2 dH(x) > 0.$$

Remark 2. *In order to get our n-asymptotic results, we need to slightly adapt the matrices involved in the identification of the final covariance matrix $\Sigma_W[i,j] = M_{i,j}(\theta_{ij}^c,\cdot)\Sigma_{i,j}M_{i,j}(\theta_{ij}^c,\cdot)^T$, $1 \leq i < j \leq k$, of Theorem 2 in Milhaud et al. (2023). In the k-sample setup involving multiple n_i-sample sizes, we must define*

$$\Sigma_{i,j}(x,y) = \begin{bmatrix} \Sigma_i(x,y) & 0_{3\times 3} \\ 0_{3\times 3} & \Sigma_j(x,y) \end{bmatrix}, \quad \textit{and} \quad M_{i,j}(\theta_{ij}^c,\cdot) = L_{i,j}(\cdot,\theta_{ij}^c)\, J_{i,j}^{-1}(\theta_{ij}^c)\, \mathcal{C}_{i,j},$$

where, since $n^{1/2} = (\kappa_i n)^{1/2}\kappa_i^{-1/2} = n_i^{1/2}\kappa_i^{-1/2}$, $i = 1,\ldots,K$, we can denote $\zeta_i = \kappa_i^{-1/2}$ and get

$$\mathcal{C}_{i,j} = \begin{bmatrix} -\zeta_i & 0 & 0 & 0 & -\zeta_j & 0 \\ 0 & -\zeta_i & 0 & -\zeta_j & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \; J_{i,j}(\theta) = \begin{bmatrix} \ddot{d}[i,j](\theta) & 0_{2\times 2} \\ 0_{2\times 2} & Id_{2\times 2} \end{bmatrix},$$

and

$$L_{i,j}(\cdot,\theta) \;=\; \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{L_i(\cdot)-G_i(\cdot)}{p_i^2} & \frac{L_j(\cdot)-G_j(\cdot)}{p_j^2} & \frac{\zeta_i}{p_i} & -\frac{\zeta_j}{p_j} \end{bmatrix}.$$

This way any sample size departures between samples can be automatically handled.

3 The k-sample test

3.1 Main results

Let us remind that we aim to test condition (5) based on the observation of any k samples, $X^{(i)} = (X_1^{(i)},\ldots,X_{n_i}^{(i)})$, $i = 1,\ldots,k$, picked from the original K-sample.

To solve this problem we propose to generalize the two-sample case by considering series of embedded statistics defined by (7), each new of them including a new pair of populations to be compared. To choose automatically the appropriate number of pairs of populations we introduce the following penalization procedure, in the spirit of the Schwarz (1978) criteria procedure. The principle of the penalized rule consist in selecting the rank s for which the penalized statistic U_s is the greatest. We introduce more specifically a sensitive rank defined by

$$S(n) = \min\left\{ \arg\max_{1\leq r\leq d(k)} \left(U_r - r \sum_{(i,j)\in S(k)} \ell_n(i,j)\mathbb{I}_{\{r_k(i,j)=r\}} \right) \right\}, \tag{8}$$

where $\ell_n(i,j)$ is a penalty term, and $\mathbb{I}_{r_k(i,j)=r}$ is 1 if $r_k(i,j) = r$ and 0 otherwise, indicating that we consider only the pair (i,j) associated to the order r. In the sequel we consider a penalty term independent from the population, i.e. $\ell_n(i,j) = \ell_n$ for all $i,j = 1,\ldots,k$. Finally, the data driven selection can simply be rewritten as

$$S(n) = \min\left\{ \arg\max_{1\leq r\leq d(k)} \{U_r - r\ell_n\} \right\}, \tag{9}$$

each statistic U_r being penalized by ℓ_n and by the number r, as a scale factor, of pairs of populations in it, according to the standard parcimony principle introduced by Schwarz (1978). In this sense, the sensitive rank $S(n)$ will select automatically the rank associated to the most significant group of $T_{i,j}$'s statistics incorporated cumulatively in $U_{S(n)}$, see slicing scheme (6). We assume now that

(B) $\ell_n = n^{\varepsilon}$, with $0 < \varepsilon < 1$.

Since under the null each test statistic is a $O_P(1)$ when $\ell_n \to +\infty$ as shown in Lemma 1, it is expected that only the first statistic will be kept. The following result shows that under the null as defined in Problem (5), the penalty effectively allows to select the first element of $\mathcal{S}(k)$ asymptotically.

Theorem 3. *Assume that* **(A1-2)** *and* **(B)** *hold. Under* H_0, $S(n)$ *converges in probability towards 1, as* $n \to +\infty$.

Theorem 4. *Assume that* **(A1-2)** *and* **(B)** *hold. Under* H_0, $U_{S(n)}$ *converges in distribution towards* $U^0(1,2)$ *given in Lemma 1, as* $n \to +\infty$.

Then our data driven test statistic is

$$\widetilde{U}_n = U_{S(n)}. \tag{10}$$

From Theorem 4, the asymptotic distribution of $\widetilde{U}_n$ under H_0 is exactly the null limit distribution studied in the two sample case and given in Lemma 1. We can use a tabulation of the random variable $U^0(1,2)$ which corresponds to a parametrized closed form stochastic integral, see Theorem 2 in Milhaud et al. (2023), that can be easily and consistently sampled, see Section 5 in Milhaud et al. (2023). By considering an empirical sample based $(1-\alpha)$-quantile, denoted $\widehat{q}_{1-\alpha}$, of the stochastic integral we decide to consider the following H_0-rejection rule

$$\widetilde{U}_n \geq \widehat{q}_{1-\alpha} \quad \Rightarrow \quad H_0 \text{ is rejected.} \tag{11}$$

3.2 Alternatives

We consider the following series of alternative hypothesis

$$\begin{array}{rcl} H_1(1) & : & F_1 \neq F_2, \\ H_1(r) & : & F_i = F_j \text{ for } r_k(i,j) < r \quad \text{and} \quad F_i \neq F_j \quad \text{for} \quad r_k(i,j) = r, \end{array}$$

with $1 < r \leq d(k)$. The hypothesis $H_1(r)$ means that the ith and jth populations such that $r_k(i,j) = r$ are the first (in the $S(k)$ ordering sense) with different unknown components.

Theorem 5. *Assume that* **(A1-2)** *and* **(B)** *hold. Under* $H_1(r)$, $S(n)$ *converges in probability towards* r, *as* $n \to +\infty$, *and* $\widetilde{U}_n$ *goes to* $+\infty$ *in probability, that is,* $\mathbb{P}(\widetilde{U}_n < \xi) \to 0$ *for all* $\xi > 0$.

4 Real world and finite samples: test statistic tuning

Experiments show that using (11) with small samples often leads to unsatisfactory results. We thus present here additional tools to improve the quality of our testing procedure in cases where the asymptotic regime is clearly not achieved.

4.1 About the penalty term ℓ_n

Since all our results are asymptotic we can replace Assumption **(B)** by

$$\ell_n(C) = Cn^{\varepsilon}, \quad \text{with } 0 < \varepsilon < 1,$$

where $C > 0$ is any positive constant that will be used as a tuning parameter to adjust the test level (type-I error). The choice of ε is important for small and moderate sample sizes. Indeed a value ε close to 1 will favour a smaller $S(n)$ value and a smaller value of the test statistic, with a lower rejection rate, while a value close to 0 will clearly empower the test. In fact, in the latter case, the divergence of $\widetilde{U}_n$ is less likely to be compensated by the penalty term. The limit case $\varepsilon = 0$ coincides with a constant penalty which is the Akaike procedure, see Akaike (1974). Following Inglot and Ledwina (2006), we propose a rule to select ε based on the data itself. To introduce this rule, consider first the two-sample case. One can write

$$\begin{aligned} F_1(x,\widehat{L}_1,\widehat{p}_1) - F_2(x,\widehat{L}_2,\widehat{p}_2) &= (F_1(x,\widehat{L}_1,\widehat{p}_1) - F_1(x,L_1,p_1^c)) \\ &\quad -(F_2(x,\widehat{L}_2,\widehat{p}_2) - F_2(x,L_2,p_2^c)) \\ &\quad +(F_1(x,L_1,p_1^c) - F_2(x,L_2,p_2^c)) \\ &= A(x) - B(x) + C(x), \end{aligned}$$

where $\theta^c = (p_1^c, p_2^c)$ is the minimizer of the contrast $d(\cdot)$, see expression (10) and (11) in Milhaud et al. (2023), with the property (p_1^c, p_2^c) equal to the true value of the proportion parameters $\theta^* = (p_1^*, p_2^*)$, under H_0 which makes $C(x) = 0$, for all $x \in \mathbb{R}$, under the null. For all $x \in \mathbb{R}$, a straightforward expansion of $A(x)$ is

$$A(x) = \frac{1}{p_1^c}\left(\widehat{L}_1(x) - L_1(x)\right) + \frac{1}{p_1^c\widehat{p}_1}\left(\widehat{p}_1 - p_1^c\right)\left(\widehat{L}_1(x) - G_1(x)\right),$$

where $(p_1^c, \widehat{p}_1) \in [\delta_1, \delta_2]^2$, see Assumption **(A0)** about the parametric space to which the proportion parameters belong, and $\widehat{L}_1$, respectively G_1, are cdfs which difference in modulus is bounded by 1. We then obtain

$$\begin{aligned} \sup_{x\in\mathbb{R}}\left(n^{1/2}|A(x)|\right) &\leq \frac{1}{\delta_1}\sup_{x\in\mathbb{R}}\left(n^{1/2}\left|\widehat{L}_1(x) - L_1(x)\right|\right) + \frac{1}{\delta_1^2}\left|n^{1/2}\left(\widehat{p}_1 - p_1^c\right)\right| \\ &= A_1 + A_2. \end{aligned}$$

By the law of the iterated logarithm for empirical processes, see Shorack and Wellner (1986), we have $A_1 = O_P((\log\log(n))^{1/2})$ and by Theorem 1 of Milhaud et al. (2023), which establishes the central limit theorem of $\widehat{p}_1$ towards p_1^c, we have that $A_2 = o_P((\log\log(n))^{1/2})$. Similarly we obtain $\sup_{x\in\mathbb{R}}(n^{1/2}|B(x)|) = O_P((\log\log(n))^{1/2})$. It follows that under the null we have

$$\begin{aligned} \sup_{x\in\mathbb{R}}\left(n^{1/2}\left|F_1(x,\widehat{L}_1,\widehat{p}_1) - F_2(x,\widehat{L}_2,\widehat{p}_2)\right|\right) &\leq \gamma\sup_{x\in\mathbb{R}}\left(n^{1/2}|A(x)|\right) + \sup_{x\in\mathbb{R}}\left(n^{1/2}|B(x)|\right) \\ &= O_P\left((\log\log(n))^{1/2}\right). \end{aligned}$$

Under $H_1(1)$ there exists at least a real x such that $C(x) \neq 0$. In that case we have for all $\gamma > 0$ and for all positive sequence b_n such that $b_n \to +\infty$

$$\mathbb{P}\left(\sup_{x\in\mathbb{R}}\left(b_n|C(x)|\right) \leq \gamma\right) \to 0.$$

In particular, choosing $b_n = n^{1/2}(\log(n))^{-1}$, it follows that under $H_1(1)$ we have

$$\mathbb{P}\left(\sup_{x\in\mathbb{R}}\left(n^{1/2}\left|F_1(x,\widehat{L}_1,\widehat{p}_1)-F_2(x,\widehat{L}_2,\widehat{p}_2)\right|\right)\leq\gamma\log(n)\right) \quad\to\quad 0,$$

as $n\to+\infty$, while this probability goes to 1 under H_0. To generalize this principle to the k-sample case we can consider

$$S_{i,j}=\sup_{x\in\mathbb{R}}\left(n^{1/2}\left|F_i(x,\widehat{L}_i,\widehat{p}_i)-F_j(x,\widehat{L}_j,\widehat{p}_j)\right|\right). \tag{12}$$

Therefore, the expected conclusion is that small values of $\max_{i,j} S_{i,j}$, overs $(i,j)\in\mathcal{S}(k)$, indicate that the unknown distributions over the considered k-sample is close to the null hypothesis while large values indicate an $H_1(r)$-type alternative. To take into account this information we set

$$I_n(\gamma) \quad=\quad \mathbb{I}\{\max_{(i,j)\in\mathcal{S}(k)} S_{i,j}\leq\gamma\log(n)\}, \tag{13}$$

for some positive constant $\gamma>0$. Under the null, from the above computations we can see that $S_{i,j}=O_P(1)$ for all $(i,j)\in\mathcal{S}(k)$. Under $H_1(r)$ as seen previously $\mathbb{P}\left(S_{i,j}\leq\gamma\log(n)\right)\to 0$ for $r_k(i,j)=r$. We can deduce, as $n\to+\infty$, the convergence in probability

$$\left\{\begin{array}{lll} I_n(\gamma)\to 1 & , & \text{under } H_0,\\ I_n(\gamma)\to 0 & , & \text{under } H_1(r).\end{array}\right. \tag{14}$$

We then define a new penalty term by

$$\ell_n(C,\gamma) \quad=\quad C\left(I_n(\gamma)\,n^{\varepsilon_0}+(1-I_n(\gamma))\,n^{\varepsilon_1}\right), \tag{15}$$

where $\varepsilon_0\approx 1$ and ε_1 is small enough in order to keep acceptable test levels even in the case of a wrong $I_n(\gamma)$ selection, privileging respectively the null or the alternative. The corresponding new selection rule is

$$\widetilde{S}(n) \quad=\quad \min\left\{\arg\max_{1\leq r\leq d(k)}\{U_r-r\ell_n(C,\gamma)\}\right\}. \tag{16}$$

In practice we have obtained very good performances with the following values $\varepsilon_0=0.99$ and $\varepsilon_1=0.75$. At this stage, it now remains to explain how to pick appropriate tuning parameters C and γ. To do this we will use the information given both by (14) and by Theorem 3.

4.2 Data-driven choice for the parameter γ based on (14)

Assume that we want to test the equality of k populations. From (14)–(15), a small value of γ yields a smaller penalty and then a more powerful test. But at the same time we want under the null

$$I_n(\gamma) \quad=\quad 1, \tag{17}$$

which is more likely achieved for large values of γ. Thus to optimize the power of the test we search for the smallest γ which guarantees (17) under H_0. For this we create a dummy H_0-setup as follows: by splitting a population in two we obtain two identical sub-populations. Since such sub-populations are identically distributed the γ associated to their test statistic should satisfy (17). And to optimize the power we choose the smaller γ satisfying this equality. We repeat this procedure b times and we obtain Algorithm 1.

Algorithm 1: Tuning of the parameter γ.

for $i = 1, \cdots, k$ **do**

  Trick: split randomly the ith sample $X^{(i)}$ into two subpopulations, namely $X^{(i,1)}$ and $X^{(i,2)}$, of equal size $n_i/2$. Compute $q_i^* = \sup_{x\in\mathbb{R}} \left((n_i/2)^{1/2} \left| F_{i,1}(x, \widehat{L}_{i,1}, \widehat{p}_{i,1}) - F_{i,2}(x, \widehat{L}_{i,2}, \widehat{p}_{i,2}) \right| \right)$, where the index i, j refers to the subpopulation $X^{(i,j)}$. /* spirit of (12) */

  Repeat b times steps 2 and 3 to get b subpopulations under the null, and b values of q_i^* for each sample $X^{(i)}$. Write $\bar{q}_i^*$ the mean of the q_i^* over the b repetitions.

Now, we have obtained k mean values for $q_i^*,\ i = 1, ..., k$. Since all q_i^* are obtained under the null, based on (13) and (14) we estimate γ: $\widehat{\gamma} = \max_{1\le i\le k}(\bar{q}_i^*/\log(n_i/2))$.

4.3 Data-driven choice for the parameter C based on Theorem 3

While the tuning of γ is based on the property (14), the tuning of the parameter C will use the result given by Theorem 3. From (15), a smaller value of C coincides with a smaller penalty yielding a larger test statistic and finally a larger power. Moreover, from Theorem 3 under the null we would expect

$$\widetilde{S}(n) \quad = \quad 1. \tag{18}$$

We then exploit this property, choosing the larger C such that (18) is satisfied. In this way we can split a population into k' sub-populations, creating an artificial null hypothesis for which we modify C to get (18). The simplest choice of k' is $k' = 3$, which gives $d(k') = 3$ and seems to tune correctly the test procedure described in Algorithm 2.

Algorithm 2: Tuning of the parameter C.

for $i = 1, \cdots, k$ **do**

  Trick : split randomly the ith sample $X^{(i)}$ into $k^{'}$ subpopulations, of equal size $n_i/k^{'}$. We obtain k new k'-sample problems under the null.

  for $j = 1, ..., d(k')$ **do**

    Compute U_j^i, where i refers to the ith population and j plays the role of r in (6).

  Choose C^i such that $U_1^i - C^i(n_i/k')^{\varepsilon_0} > U_j^i - jC^i(n_i/k')^{\varepsilon_0}$, for $j = 1, \ldots, d(k')$.

  /* Choose C_i such that $S(n) = 1$ in every case */

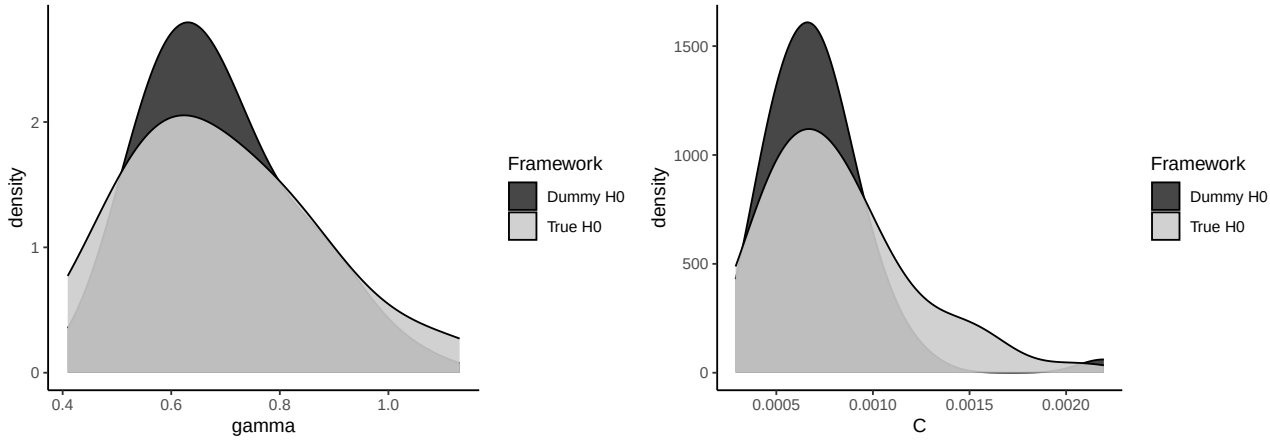
  Equivalently, we have $C^i = \max_j \left(\dfrac{U_j^i - U_1^i}{(j-1)(n_i/k')^{\varepsilon_0}} \right)$, for $j = 1, \ldots, d(k')$.

Finally, choose $\widehat{C} = \min_i C^i$.

In a nutschell, we first tune γ and C, which allows to deduce ℓ_n in (15). Hence, we get the order $S(n)$ through Equation (9). Finally, the test statistic given by Equation (10) is used in the test procedure (11).

Remark 6. *The trick used here, consisting in splitting one given sample into several (at least two) sub-samples, leads to a dummy H_0-framework. However, this framework is clearly different from the real-life*

situation where F_i would be equal to F_j considering two different samples. In particular, the known component G_i of the ith contamination model should be different from G_j in full generality. Instead, the trick causes $G_i = G_j$, in addition to the fact that the observations originally come from the same sample. This makes the estimation process and thus the testing procedure slightly different, see Appendix D of the Supplement in Milhaud et al. (2023). It is therefore important to check whether this artificial procedure does not strongly affect the choice of parameters γ and C, as compared to the parameters that would be selected by the tuning process in a real-life situation under the null. In this spirit, we repeat 100 times the following simulation scheme under H_0: (i) simulate 4 samples (populations) following contamination/admixture models, (ii) use Algorithms 1 and 2 to get the distributions of γ and C under the dummy H_0 setting, (iii) still consider Algorithms 1 and 2 in a simplified version (delete step 2 and consider the samples themselves in the process, since we are under the null) to get the distributions of γ and C (without the trick). Finally, compare the obtained distributions. Keeping in mind that this was tested in many other frameworks, Figure 1 shows that our tuning process embedding the trick remains consistent. Indeed, despite that the distributions of the parameters γ and C are slightly different, they look similar with the same mode.



(a) Tuned γ distribution.

(b) Tuned C distribution.

Figure 1: Distributions of selected tuning parameters obtained from Algorithms 1 and 2 over 100 repetitions, under the dummy H_0 framework and the true one.

5 Clustering strategy

In the sequel we propose to adapt the previous test procedure to obtain a data-driven method to cluster K populations into N subgroups characterized by a common unknown nodular component. The novelty here lies in the fact that we will be able to cluster unlabeled behaviours presenting similar distributions, contrary to classical existing clustering strategies that are based on directly/fully observed phenomenons. Moreover it is worth to notice that the number N of clusters is not assumed at the beginning of the procedure but is automatically deduced at the end of a run.

Assume that we observe K independent samples $X^{(i)} = (X_1^{(i)}, \dots, X_{n_1}^{(i)})$, $i = 1, \dots, K$, made separately of independent and identically distributed observations.

To build the first cluster we consider the two closest populations in terms of the statistics $T_{i,j}$, $i \neq j \in \{1, \cdots, K\}$. Two populations are thus proposed to be merged to create the first group $\mathcal{G}_1$. We test their equality according to the testing procedure (11) to confirm the construction of such a cluster. We continue to add populations to the group until the test rejects equality. Once this first cluster $\mathcal{G}_1$ is fully identified: close the cluster, remove clustered samples from the initial collection of samples, and create a new cluster $\mathcal{G}_2$. Then look for still unclustered neighboors from the last studied sample that led to reject H_0. Again we select the biggest collection of samples, among the remaining pool, that is tested to share a common unknown component with the latter. This creates our second cluster. One can iterate this several times until every sample is associated with a cluster. Algorithm 3 describes our so-called KCMC (K-sample Contamination Model Clustering) algorithmic clustering strategy, with $S = \{1, \dots, K\}$ the set of population indices, c the cluster id and S_c the members of cluster c. The procedure stops when all populations are merged or when all populations have been considered by the algorithm. Note that since the first two populations selected are the closest, if the test rejects their equality then the algorithm stops and returns as many clusters as populations. The procedure is straightforward since the parameters γ and C are data-driven (up to the prior choice of b and k', see Section 4.2 and Algorithms 1 and 2). Furthermore, we deduce from Theorem 5 the following property.

Proposition 1. *With a probability that tends to 1 as $n \to +\infty$, the number N^* of groups detected by Algorithm 3 satisfies $1 \leq N \leq N^* \leq K$, where N denotes the true unknown number of groups.*

From Proposition 1 we know that the number of clusters obtained from the KCMC algorithm is potentially greater than N. Moreover, from Theorem 5, asymptotically all distributions of each clusters

Algorithm 3: K-sample Contamination Model Clustering (KCMC).

Initialization: create the first cluster to be filled, *i.e.* $c = 1$. By convention, $S_0 = \emptyset$.
Select $(x, y) = \text{argmin}\{nd_n[i, j](\widehat{\theta}_{ij}); i \neq j \in S \setminus \bigcup_{m=1}^{c} S_{m-1}\}$.
Test H_0 between x and y (two-sample test). `/* using (11) */`
. **if** H_0 *is not rejected* **then**
    $S_1 = \{x, y\}$ `/* fill in the first cluster */`
else
    $S_1 = \{x\}$, $S_{c+1} = \{y\}$ and then $c = c + 1$ `/* close, open new one */`
while $S \setminus \bigcup_{m=1}^{c} S_m \neq \emptyset$ **do**
    `/* seek unclustered neighboors, select the closest one */`
    Select $u = \text{argmin}_j\{nd_n[i, j](\widehat{\theta}_{ij}); i \in S_c, j \in S \setminus \bigcup_{m=1}^{c} S_m\}$
    Test H_0 the simultaneous equality of all the F_j, $j \in S_c$:
    if H_0 *not rejected* **then**
        put $S_c = S_c \bigcup \{u\}$
    else
        $S_{c+1} = \{u\}$ and $c = c + 1$

are equal. Thus the only possible error is that a real group has been splitted into several other groups, which can happen because we have an asymptotic test level $\alpha = 5\%$, see testing rule (11). This parameter clearly reflects the threshold for accepting the creation of a group. One way to check the stability of the clusters is to change this threshold, for example by decreasing α to see if the groups merge then. We illustrate this point in our real world application, see Section 7. The tuning strategy of Sections 4.1 and 4.3 allows us to achieve very good performances across our simulation study. In particular the detected number N^* of clusters often does not exceed the actual number N.

6 Simulation study

All our numerical experiments were performed thanks to the `admix` R package developed and implemented for estimation, test and clustering of populations coming from admixture models. To begin with, we test the influence of the number of populations under consideration, to see whether this affects the quality of our k-sample testing procedure. For this purpose, we let k vary from 2 to 10. The populations are drawn from different distributions supported by various types of real-sets. We provide here the results for distributions supported over $\mathbb{R}$ (Gaussian mixtures), but simulations were extended to other supports such as $\mathbb{N}$ (Poisson mixtures) or $\mathbb{R}^+$ (Gamma mixtures) with very similar conclusions. The proportions of the unknown components are fixed all along the simulation scheme for easier comparisons. To evaluate the empirical level (and power) of the k-sample test, we use a Monte-Carlo approach where each of the B experiments is performed in the same way. We also make the sample size vary to illustrate the asymptotic properties of our results. Unless otherwise stated, all our simulations were performed with fixed values $\varepsilon_0 = 0.99$ and $\varepsilon_1 = 0.75$ in (15) and (16) (meaning that we use the tuning process described in Sections 4.1 and 4.3). As expected, the tuning process reveals to be decisive to improve the power of the test, but has no real influence under the null. This is in line with common sense, since tuning parameters γ and C are estimated under the null. Once the quality of the k-sample test will be validated, we will derive extra simulations to assess the performance of our clustering algorithm itself.

6.1 Empirical level of our k-sample testing procedure

We draw k populations from two-component Gaussian mixtures, where the k simulated known components are distributed according different Gaussian distributions. On the contrary, those k populations share the same unknown component distribution (namely a standard normal distribution). For each simulation being part of the Monte-Carlo procedure, we implement the following steps: (i) generate the k populations, each one following an admixture model; (ii) perform the k-sample test; (iii) retrieve which penalty rule (similarly which ε, either ε_0 or ϵ_1) and which rank $\widetilde{S}(n)$ have been chosen, as well as the p-value of the test. We repeat this simulation scheme $B = 100$ times in order to estimate the empirical level of our test procedure (11). Table 1 reports the parameters involved in each simulated population for three different sample sizes (about 400, 1000 and 3000 observations), as well as the results related to the main indicators showing how efficient our procedure is. More comprehensively, Table 1 stores in its last four columns how often the right penalty rule (15) has been chosen (in percent), the 90%-percentile of the distribution of the selected order $\widetilde{S}(n)$ (16), the mean of the 100 p-values obtained when testing, and finally the empirical level of the test.

Table 1: k-sample test (Gaussian mixtures). Reported $\widetilde{S}(n)$ corresponds to the 90%-percentile of the distribution of $\widetilde{S}(n)$ over the 100 experiments, and p-val is the average of obtained p-values

	Samples													
i	1	2	3	4	5	6	7	8	9	10				
p_i	0.3	0.8	0.6	0.4	0.9	0.2	0.4	0.15	0.7	0.5	Pen.			Emp.
G_i	$\mathcal{N}(2,0.7)$	$\mathcal{N}(4,1.1)$	$\mathcal{N}(3,0.8)$	$\mathcal{N}(-1,0.3)$	$\mathcal{N}(-3,0.2)$	$\mathcal{N}(-5,0.4)$	$\mathcal{N}(3.5,0.1)$	$\mathcal{N}(-4,0.7)$	$\mathcal{N}(-2.5,1)$	$\mathcal{N}(1.5,0.3)$	rule	$\widetilde{S}(n)$	p-	level
n_i	347	449	308	382	426	372	440	447	474	424	(%)		value	(10^{-2})
	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}				
k=2	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$									100	1	0.53	5
k=4	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$							98	1	0.74	3
k=6	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$					96	1	0.76	4
k=8	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$			92	1	0.83	6
k=10	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	95	1	0.9	5
n_i	1011	1027	1077	1019	903	942	971	1065	1071	1068				
	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}				
k=2	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$									100	1	0.4	7
k=4	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$							100	1	0.77	2
k=6	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$					100	1	0.8	4
k=8	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$			87	1	0.8	8
k=10	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	86	1	0.83	6
n_i	3187	2847	3189	3175	3042	2989	3184	2868	2998	3193				
	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}				
k=2	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$									100	1	0.48	6
k=4	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$							100	1	0.71	3
k=6	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$					98	1	0.78	4
k=8	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$			93	1	0.81	6
k=10	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	94	1	0.92	3

Given that the selected sample-based quantile considered in (11) was fixed as the 95%-percentile of the tabulated distribution, it is expected that the empirical level of the test (last column) stays close to 5%. Looking at the results, our test procedure looks to be globally efficient. Most of time, the right penalty rule and the right testing rank have been selected. Indeed, in more than 90 of the 100 experiments, the selected order $\widetilde{S}(n)$ has the correct value (equal to 1, since we are under H_0). Moreover, the number k of populations involved in the k-sample test does not seem to impact our testing procedure. Even when some populations have overlapping components, the quality of the test remains satisfactory. Also, the same simulations were performed without using the tuning process, with almost no impact on test levels. In that latter case, we have set $\varepsilon = 0.87$ for the penalty given by Assumption **(B)** in (9), as this value lies exactly in the middle of $[\varepsilon_1, \varepsilon_0] = [0.75, 0.99]$. Setting ε to way lower values led to seriously deteriorate the test levels in finite samples applications, which validates the need to keep a strong penalty under the null. Of course, this global picture may change depending on the chosen parameters to conduct the simulation study. For instance, much higher variances for the mixture components would clearly affect our results.

Remark 7. *We observed higher empirical levels when the population to test is strongly under-represented. If the product $n_i p_i$ is low, say around 30, the estimation of the mixture weight p_i deteriorates. This spreads out to the computation of supremum in (12), which mechanically increases and leads to the wrong choice in the penalization rule, i.e. taking ε_1 instead of ε_0.*

6.2 Empirical power

Now, we aim to study the power of our testing strategy, that is to say how our k-sample test performs in detecting that (at least) two of the k populations have different unknown component distributions. For ease of comparisons, we keep the same known component distributions and unknown component proportions as previously. The different parameters involved in Gaussian mixtures are stored in Table 2,

Table 2: k-sample test under the alternative H_1, with emphasis on different settings when $K = 10$. Interpretation of the last four columns is identical to Table 1 (*n.a.* stands for *not applicable*)

	Samples													
i	1	2	3	4	5	6	7	8	9	10				
p_i	0.3	0.8	0.6	0.4	0.9	0.2	0.4	0.15	0.7	0.5	Pen.			Emp. power
G_i	$\mathcal{N}(2, 0.7)$	$\mathcal{N}(4, 1.1)$	$\mathcal{N}(3, 0.8)$	$\mathcal{N}(-1, 0.3)$	$\mathcal{N}(-3, 0.2)$	$\mathcal{N}(-5, 0.4)$	$\mathcal{N}(3.5, 0.1)$	$\mathcal{N}(-4, 0.7)$	$\mathcal{N}(-2.5, 1)$	$\mathcal{N}(1.5, 0.3)$	rule	$\widetilde{S}(n)$	p-	Tune/NoTune
n_i	347	449	308	382	426	372	440	447	474	424	(%)		value	(10^{-2})
	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}				
k=2	$\mathcal{N}(0, 1)$	$\mathcal{N}(0.3, 1)$									*n.a.*	1	0.24	*n.a.* / 36
k=4	$\mathcal{N}(0, 1)$	$\mathcal{N}(1, 1)$	$\mathcal{N}(0.3, 1)$	$\mathcal{N}(0, 1)$							15	6	0.46	**20** / 11
k=7	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(1, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0.5, 1)$	$\mathcal{N}(0, 1)$				30	21	0.7	**26** / 3
k=10	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0.5, 0.5)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	30	1	0.8	**7** / 1
k=10	$\mathcal{N}(0, 1)$	$\mathcal{N}(-0.5, 1)$	$\mathcal{N}(0.3, 0.5)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0.7, 0.7)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(-0.2, 1)$	25	1	0.72	**9** / 1
k=10	$\mathcal{N}(0, 1)$	$\mathcal{N}(-0.5, 1)$	$\mathcal{N}(0.3, 0.5)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(1, 1)$	$\mathcal{N}(0.7, 0.7)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(-0.4, 1)$	$\mathcal{N}(0.9, 3)$	$\mathcal{N}(-0.2, 1)$	70	45	0.26	**63** / 2
n_i	1011	1027	1077	1019	903	942	971	1065	1071	1068				
	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}				
k=2	$\mathcal{N}(0, 1)$	$\mathcal{N}(0.3, 1)$									*n.a.*	1	0.09	*n.a.* / 74
k=4	$\mathcal{N}(0, 1)$	$\mathcal{N}(1, 1)$	$\mathcal{N}(0.3, 1)$	$\mathcal{N}(0, 1)$							41	6	0.22	**50** / 18
k=7	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(1, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0.5, 1)$	$\mathcal{N}(0, 1)$				95	21	0.04	**95** / 1
k=10	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0.5, 0.5)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	95	45	0.23	**72** / 1
k=10	$\mathcal{N}(0, 1)$	$\mathcal{N}(-0.5, 1)$	$\mathcal{N}(0.3, 0.5)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0.7, 0.7)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(-0.2, 1)$	85	45	0.13	**84** / 2
k=10	$\mathcal{N}(0, 1)$	$\mathcal{N}(-0.5, 1)$	$\mathcal{N}(0.3, 0.5)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(1, 1)$	$\mathcal{N}(0.7, 0.7)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(-0.4, 1)$	$\mathcal{N}(0.9, 3)$	$\mathcal{N}(-0.2, 1)$	100	45	0	**99** / 1
n_i	3187	2847	3189	3175	3042	2989	3184	2868	2998	3193				
	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}				
k=2	$\mathcal{N}(0, 1)$	$\mathcal{N}(0.3, 1)$									*n.a.*	1	0.009	*n.a.* / 96
k=4	$\mathcal{N}(0, 1)$	$\mathcal{N}(1, 1)$	$\mathcal{N}(0.3, 1)$	$\mathcal{N}(0, 1)$							99	6	0.001	**99** / 12
k=7	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(1, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0.5, 1)$	$\mathcal{N}(0, 1)$				100	21	0	**98** / 1
k=10	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0.5, 0.5)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	100	45	0	**100** / 3
k=10	$\mathcal{N}(0, 1)$	$\mathcal{N}(-0.5, 1)$	$\mathcal{N}(0.3, 0.5)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0.7, 0.7)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(-0.2, 1)$	100	45	0	**99** / 1
k=10	$\mathcal{N}(0, 1)$	$\mathcal{N}(-0.5, 1)$	$\mathcal{N}(0.3, 0.5)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(1, 1)$	$\mathcal{N}(0.7, 0.7)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(-0.4, 1)$	$\mathcal{N}(0.9, 3)$	$\mathcal{N}(-0.2, 1)$	100	45	0	**99** / 2

showing that some considered frameworks corresponds to critical situations where mixture components can be highly overlapping, see for instance when $k = 2$. As already mentioned, the tuning process is essential here to correctly detect the alternative. Indeed, the penalty term should not compensate the explosion of the test statistic, which means that taking $\varepsilon = 0.75$ (i.e. $\varepsilon = \varepsilon_1$) instead of $\varepsilon = 0.87$ leads to very distinct results. Let us focus here on the case $k = 10$, and emphasize the different possibilities depending on the number of different unknown component distributions involved across those k populations.

Table 2 leads to several interesting conclusions. First, the power of the test is much more sample size sensitive than its level. This is not surprising: detecting departures from the null hypothesis requires strong evidence that two unknown component distributions are different, which is far from being obvious when considering mixtures with overlapping components and moderate sample sizes. However, as soon as the product $n_i p_i$ becomes large enough, say around 1000, the power of the test gets close to 1.

With small sample sizes, it looks tricky to select the right penalty rule in most of cases (except when $k = 2$). In practice, one tends to select ε_0 instead of ε_1. Indeed, dividing the population at the beginning of Algorithm 1 leads to lower the original sample size, which is likely to create higher variability for $S_{i,j}$. Mechanically, the quantity $\max(S_{i,j})/\log(n)$ increases, and thus also γ does. Finally, one gives more importance to the sample size in (15), which globally increases the penalty term in (16) and is more likely to compensate the increase of the contrast that normally indicates the departure from the null hypothesis. The testing procedure thus tends to believe that we are under the null, as can be seen looking at the 90%-percentile of $\widetilde{S}(n)$ (which sometimes equals 1 with $k = 10$). Stopping at order $\widetilde{S}(n) = 1$ prevents from detecting the alternative, given that there exists at least two populations with same unknown component in our k-sample settings and that the implemented algorithm starts by testing the two *closest* populations in the $nd(\cdot)$ discrepancy sense. The consideration of moderate sample sizes clearly helps to improve the quality of the test. Indeed, both the penalty rule and the right order tend to be correctly selected.

As expected, the higher the number of different unknown components among the k populations is, the more powerful the test is. Notice that when $k = 2$, Algorithms 1 and 2 are useless since they only affect the penalty rule that helps to detect the right number of summands in the test statistics. Indeed, in this case, there is only one summand by construction. Finally, Table 2 also reports in its last column the improvement, in terms of test power, obtained using the tuning process, which validates that the power of the k-sample test strongly depends on its use. Of course, these results may differ with different simulation parameters but we tried to consider a large class of simulation setups in order to challenge the robustness of our procedure.

6.3 Clustering

Hereafter, we are willing to cluster the unknown components F_i's over K populations under study, having only observed the admixture L_i of the known G_i's with the unknown F_i's. We begin with the description of our clustering frameworks, before discussing our results.

Clustering schemes description. We dedicated the previous section to the study of the performance of the k-sample testing procedure ($2 \leq k \leq K$), since the quality and the robustness of our clustering algorithm strongly relies on it. Now, we would like to recover simulated clusters over $K = 10$ populations. Various frameworks are investigated, from the extreme cases of one single cluster up to ten clusters. In-between, we also study situations where we have both, sizewize speaking, unbalanced and balanced clusters. Figure 2 illustrates the four considered settings. In the first case (one single cluster) the common unknown component is distributed according to $F_i \sim \mathcal{N}(7, 0.5)$, $i = 1, \ldots, K$, whereas two clusters appears for the second case. The densities of populations underlying these two clusters are depicted through different line types (plain and dotted). In the third case, the densities associated to the three balanced clusters are displayed with different line types and widths. Clustering the unknown components of these populations having only the knowledge of the known components is not straightforward. Indeed, there are overlapping components among the populations, see for instance the 3rd and 4th populations in the 1-cluster example. Moreover, some of the clusters can be close from one to another, see for instance the third case where two of the three clusters are not well separated because of close means and higher variances. All the parameters involved in those simulations are stored in Table 3.

Table 3: Parameters for clustering. The two clusters are composed of populations (1,2,5,6,8,9,10) and (3,4,7), and the three clusters embed populations (1,3,4,7), (2,6,9) and (5,8,10)

Populations i	1	2	3	4	5	6	7	8	9	10
Weight p_i	0.3	0.8	0.6	0.4	0.9	0.2	0.4	0.15	0.7	0.5
Sample size n_i	312	271	293	322	289	282	279	280	294	324
1 cluster G_i	$\mathcal{N}(16, 0.7)$	$\mathcal{N}(22, 1)$	$\mathcal{N}(6, 2)$	$\mathcal{N}(8, 1.2)$	$\mathcal{N}(2, 0.2)$	$\mathcal{N}(3, 0.3)$	$\mathcal{N}(-3, 0.4)$	$\mathcal{N}(-5, 0.5)$	$\mathcal{N}(-1, 0.1)$	$\mathcal{N}(11, 0.7)$
G_i (2&3 clusters)	$\mathcal{N}(16, 0.7)$	$\mathcal{N}(22, 1)$	$\mathcal{N}(6, 2)$	$\mathcal{N}(8, 1.2)$	$\mathcal{N}(2, 0.2)$	$\mathcal{N}(3, 0.3)$	$\mathcal{N}(4, 0.4)$	$\mathcal{N}(5, 0.5)$	$\mathcal{N}(6, 0.6)$	$\mathcal{N}(7, 0.7)$
F_i (2 clusters)	$\mathcal{N}(7, 0.5)$	$\mathcal{N}(7, 0.5)$	$\mathcal{N}(15, 1.1)$	$\mathcal{N}(15, 1.1)$	$\mathcal{N}(7, 0.5)$	$\mathcal{N}(7, 0.5)$	$\mathcal{N}(15, 1.1)$	$\mathcal{N}(7, 0.5)$	$\mathcal{N}(7, 0.5)$	$\mathcal{N}(7, 0.5)$
F_i (3 clusters)	$\mathcal{N}(15, 1.1)$	$\mathcal{N}(7, 0.5)$	$\mathcal{N}(15, 1.1)$	$\mathcal{N}(15, 1.1)$	$\mathcal{N}(17, 0.7)$	$\mathcal{N}(7, 0.5)$	$\mathcal{N}(15, 1.1)$	$\mathcal{N}(17, 0.7)$	$\mathcal{N}(7, 0.5)$	$\mathcal{N}(17, 0.7)$
G_i (10 clusters)	$\mathcal{N}(16, 0.7)$	$\mathcal{N}(22, 1)$	$\mathcal{N}(6, 2)$	$\mathcal{N}(8, 1.2)$	$\mathcal{N}(2, 0.2)$	$\mathcal{N}(3, 0.3)$	$\mathcal{N}(-3, 0.4)$	$\mathcal{N}(5, 0.5)$	$\mathcal{N}(-1, 0.1)$	$\mathcal{N}(7, 0.7)$
F_i (10 clusters)	$\mathcal{N}(7, 0.5)$	$\mathcal{N}(6, 0.6)$	$\mathcal{N}(15, 1.1)$	$\mathcal{N}(12, 0.05)$	$\mathcal{N}(3, 2)$	$\mathcal{N}(-4, 0.9)$	$\mathcal{N}(-8, 1.1)$	$\mathcal{N}(0, 0.5)$	$\mathcal{N}(17, 0.4)$	$\mathcal{N}(-5, 0.2)$

Performance of the clustering. We still use a Monte Carlo approach here, meaning that we perform the clustering task B times for each of the four cases aforementioned. As the clustering process is

computationally intensive (it requires to perform many k-sample tests), we set $B = 20$. However, we also considered $B = 50$ for some of our examples, which led to very minor modifications of the results without changing the global picture. Given the parameters of the simulations, see Table 3, in the four studied frameworks further denoted (a) to (d), we expect our procedure to find respectively the following clusters: (1,2,3,4,5,6,7,8,9,10); (1,2,5,6,8,9,10) and (3,4,7); (1,3,4,7), (2,6,9) and (5,8,10); and finally (1),(2),(3),(4),(5),(6),(7),(8),(9),(10).

In practice, there exists many ways for the clustering algorithm to reach wrong conclusions. Either it detects the right number of clusters but does not affect the right populations to the right clusters, which should not happen asymptotically, or it selects straight out a wrong number of clusters. In the latter case the algorithm tends to overestimate the correct number of clusters, leading to clusters with wrong sizes and isolated populations.

Basically, it is difficult to summarize all possible encountered wrong answers through one single indicator. In our case, we have chosen to measure the performance of the clustering algorithm through classification matrices (also called heatmaps). Indeed, it seems to us that it is an efficient and yet simple indicator. Figures 3 and 4 display examples of such matrices, in our simulation setups. The reading of heatmaps is easy. First, they are symmetric, with errors stored in the off-diagonal terms (of course a given population is always clustered with itself). For these non-diagonal terms, one counts how many times (among B experiments) the clustering algorithm clustered each population with the other ones.

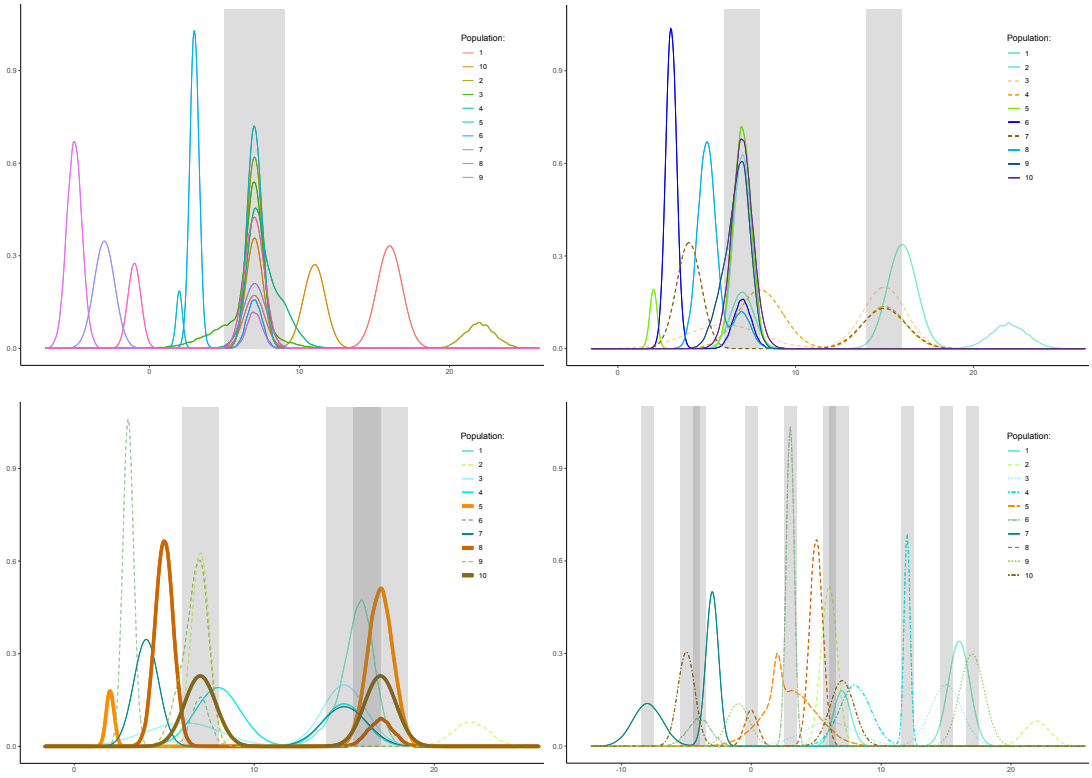
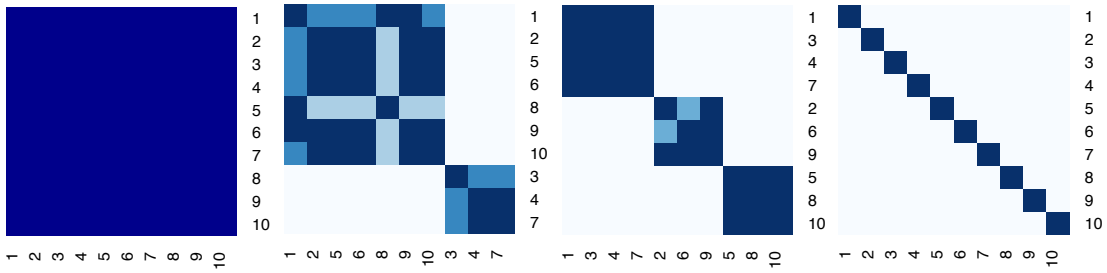


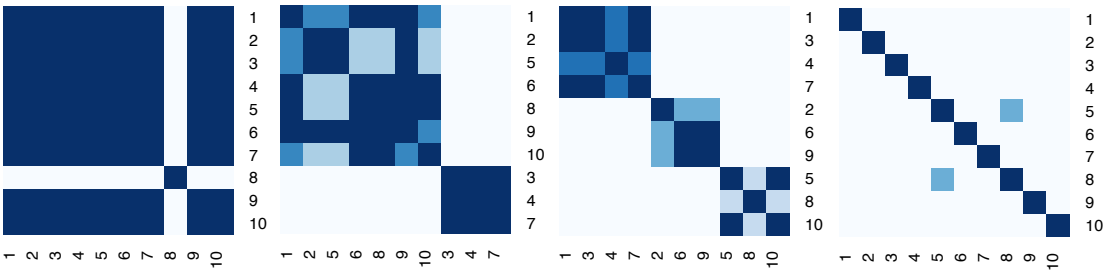
Figure 2: Simulated densities of the 10 populations. From top left to bottom right: 1 cluster, 2 clusters (pop.(1,2,5,6,8,9,10) and (3,4,7)), 3 clusters ((1,3,4,7), (2,6,9) and (5,8,10)), 10 clusters.



(a) 1 cluster. (b) 2 clusters. (c) 3 clusters. (d) 10 clusters.

Figure 3: Heatmap (or classification matrix) describing the efficiency of our clustering algorithm with n around 300, see Table 3. Values for the percentage of right predictions are given from very light to dark blue, which corresponds to no error (0%), $[80\%, 90\%[$, $[90\%, 100\%[$, 100%

Then, comparing this to the expected clusters, it is straightforward to deduce the percentage of correct classifications.



(a) 1 cluster. (b) 2 clusters. (c) 3 clusters. (d) 10 clusters.

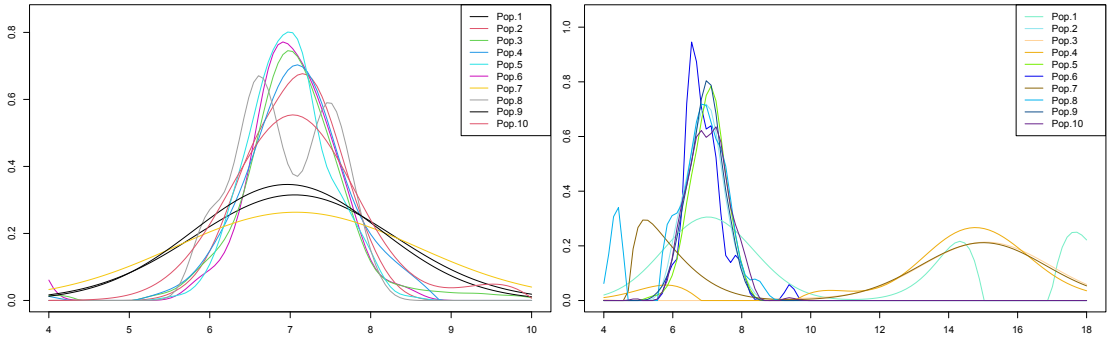
Figure 4: Heatmap with $n = 200$. Percentage of classifications are given from very light to dark blue, which corresponds to no error (0%), $[80\%, 90\%[$, $[90\%, 100\%[$, 100%. For cases (a) and (d), the interpretation differs: populations 5 and 8 were put in the same cluster in 10% of simulations

To simplify, our heatmaps are organized by blocks, each block corresponding to an expected cluster. This means that a perfect clustering has 100% of right classifications for every blocks. This is the case for instance in Fig. 3 concerning the first and fourth settings (cases (a) and (d)). In the two other frameworks, the KCMC algorithm is sometimes mistaking, but results show that these errors remain reasonable. Indeed, the percentage of right classifications does not fall below 80%. Focusing now on the case of two unbalanced clusters (case (b)), it is obvious that certain populations are always found to belong to the same cluster (populations 2, 5 and 6; or populations 4 and 7), whereas others can be detected to be outside the actual cluster (e.g. population 3 which is not clustered everytime with populations 4 and 7). The worst case here lies in the cluster containing population 8. Figure 4 illustrates the phenomenon that was already observed when studying the performance of the k-sample test. Decreasing the sample size has a strong influence on the clustering efficiency. Here, the clusters could be recovered thanks to the

fact that we use B simulations, but the reader has to keep in mind that using this clustering algorithm can reveal tricky in real-life applications containing a low number of observations. Finally, once the clusters are recovered, useful information can be deduced. For instance, knowing that the unknown weights are consistently estimated inside each cluster, it is possible to retrieve the estimated proportions of the unknown perturbation impacting the original population. Table 4 provides such results based on our simulation parameters (not applicable for the case of 10 clusters since weights are not consistenly estimated when there are no equal unknown components). Moreover, the corresponding decontaminated densities can also be nicely illustrated, see Fig. 5.

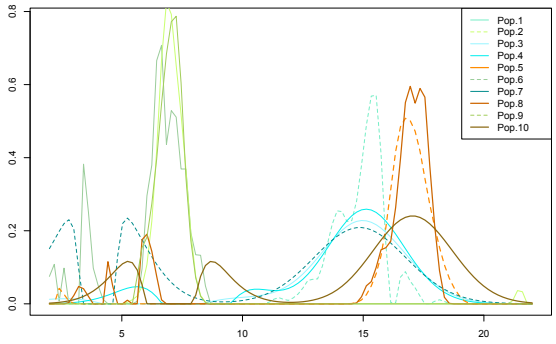
Table 4: Mean of estimated unknown weights of the ten populations under study ($n = 300$), obtained from pairwise IBM testing over each cluster.

Real weight p_i	0.3	0.8	0.6	0.4	0.9	0.2	0.4	0.15	0.7	0.5
Estimated weight $\widehat{p}_i$	$\widehat{p}_1$	$\widehat{p}_2$	$\widehat{p}_3$	$\widehat{p}_4$	$\widehat{p}_5$	$\widehat{p}_6$	$\widehat{p}_7$	$\widehat{p}_8$	$\widehat{p}_9$	$\widehat{p}_{10}$
Case of 1 cluster	0.271	0.798	0.669	0.424	0.894	0.159	0.378	0.174	0.676	0.390
Case of 2 clusters	0.291	0.829	0.603	0.417	0.884	0.253	0.395	0.147	0.701	0.752
Case of 3 clusters	0.367	0.821	0.581	0.441	0.903	0.216	0.448	0.151	0.740	0.494



(a) 1 cluster.

(b) 2 clusters.



(c) 3 clusters.

Figure 5: Decontaminated densities (recovered unknown components) once the clusters identified.

7 Real world Application

7.1 Excess of mortality during pandemics

In Milhaud et al. (2023), a pairwise comparison of COVID-19 excess of mortality has been investigated. This is aimed at identifying countries experiencing similar impact of the COVID-19 on the mortality. More precisely, the age distribution of death is considered and is supposed to exhibit a common component over periods at the country level. This would correspond to the known component in model (1). The COVID-19 component is assumed to be unknown. For instance, the mortality change between 2019 and 2020 is considered as a baseline and serves to assess the excess of the mortality due to COVID-19. The current paper follows the same line as Milhaud et al. (2023) and proposes to cluster countries given the inherent impact of the COVID-19. The datasets of interest came from the Short-Term Mortality Fluctuations (STMF) data series compiled by the Human Mortality Database (HMD). It contains death records aggregated over age groups: 0-14, 15-64, 65-74, 75-85 and 85+. Here, we restrain our study to the four last age classes (given that experts agree to consider that the first one 0-14 was clearly not affected by the pandemic). First, we consider a clustering procedure over the same countries considered in Milhaud et al. (2023) for the first wave. Formally, we study the similarities in terms of the changes for France, Belgium, Germany, Italy, Netherlands and Spain. The known distributions are multinomial ones with four categories here and we compare the unknown multinomial distributions caused by the COVID-19. In Table 5, we report the resulted clusters:

Table 5: Clustering of excess of mortality profile over 2020.

	France	Italy	Netherlands	Belgium	Germany	Spain
Cluster (id)	3	2	2	1	1	1

Two clusters are the same as those already identified by the authors. Namely, France shows a proper COVID-19 impact on its mortality whereas Italy and the Netherlands share the same profile. However, our clustering methodology identifies a third cluster consisting of Germany, Spain and Belgium. In Milhaud et al. (2023), it is shown that Germany and Belgium, on one hand, and Belgium and Spain on the other have similar impacts but the test rejects the null hypothesis for Germany and Spain. This lack of transitivity of the pairwise testing procedure was already discussed in Milhaud et al. (2023) and the K-sample procedure offers an interesting yet robust generalization of the latter.

7.2 Europe-wide clustering of COVID-19 excess of mortality

In the following, we explore a larger clustering scheme and consider 29 European countries, i.e. Austria, Belgium, Bulgaria, Switzerland, Czech Republic, Germany, Denmark, Spain, Estonia, England and Wales, Finland, France, Greece, Croatia, Hungary, Iceland, Ireland, Italy, Lithuania, Luxembourg, Latvia, Netherlands, Norway, Poland, Portugal, Scotland, Slovakia, Slovenia and Sweden. We aim at exploring the impact of the pandemic over these countries in 2020 and identify the clusters. We adopt the same assumptions as described above and proceed to clustering the countries with regard to their shared COVID-19 excess of mortality effect. The known and unknown distributions are multinomials with four categories (the four age classes). The sample sizes are given in the first row of Table 7 and range from

1141 for Iceland to 462577 for Germany. We should recall that this comparison focus on the distribution of the nodular effect of the COVID-19 rather than its dynamics. As soon as the countries under study suffered from the impact of the outbreak of the pandemic at different periods during the year of 2020, we will be more concerned with the impact on the population rather than its magnitude. Hence, the following comparison should be root on the socio-demographic disparities that may exist among the populations as well as the healthcare capacities, public health measures and many other factors. The discussion of the implication of such an impact is, however, beyond the scope of this paper.

First, we set up the level α to 1% and explore the clusters that formed on the basis of the H_0-rejection rule. At this level, we are left with 11 clusters. In Figure 6, we report these as well as the estimated unknown cumulative distribution functions, averaged over each cluster. First, we can observe that some countries are single isolated clusters. This is the case for Spain, Island, Switzerland, Netherlands and Portugal. On the other hand, we have two large clusters that represent most countries from center and eastern European countries: Lithuania, Latvia, Poland, Hungry, Bulgaria, Slovakia and Estonia. This block is isolated from the geographically adjacent cluster constituted by the Czech Republic and Croatia. Some of the Northern European countries are gathered on two clusters. The largest is constituted of Finland, Austria, Germany, Northern Ireland, Scotland, Sweden and England & Wales. Surprisingly, a common factor, among other things, is the Protestant inheritance. Numerous studies, e.g. Kaklauskas et al. (2022) among others, validated the similarities between the English-speaking and the Protestant European clusters due to their closely related common histories, cultural interactions, similar development levels, and religions. Finally, in order to understand more closely the clusters we refer the reader to the plethora of studies that investigate the factors that influence mortality levels from COVID-19 such as well-functioning healthcare system, prevention measures (e.g. social distancing), and population age structure, among others.

A way to check the stability of the clusters is to change the threshold of the test acceptance, for example by increasing α to see if the groups merge then. In Figure 7 we reported the clusters for different levels, respectively, 1%, 5% and 15%. As noted earlier, this parameter clearly reflects the threshold for accepting the composition of a group. Indeed, we see that these three levels lead respectively, to 11, 14 and 15 clusters. From Proposition 1 the number of groups determined by our algorithm is asymptotically greater than the true (unknown) number of groups. Since the sample sizes are large here we can conclude that 11 groups is a reasonable choice. If we want to obtain greater detail, a larger value for α will enable a more refined clustering, but may look artificial if too many groups are suddenly created.

Acknowledgement

This work was conducted within the Research Chair DIALog under the aegis of the Risk Foundation, an initiative by CNP Assurances.

Figure 6: Clustering of the excess mortality profile due to COVID-19 during the year 2020 over 29 countries (top) and the corresponding unknown cdf (bottom right).

	Population size	10042	307605	2161	26928	39334	462577	7628	28146	41928	261626	1145	294612	62614	73827	32433
Level	15%	SVN	ITA	LUX	FIN	AUT	GER	NIR	SCO	SWE	ENW	ISL	FRA	GRC	NLD	CHE
	5%	SVN	ITA	LUX	FIN	AUT	GER	NIR	SCO	SWE	ENW	ISL	FRA	GRC	NLD	CHE
	1%	SVN	ITA	LUX	FIN	AUT	GER	NIR	SCO	SWE	ENW	ISL	FRA	GRC	NLD	CHE

	Population size	18383	12878	209032	62346	55686	27103	7169	56903	25157	209431	55458	52715	27224	19712
Level	15%	LTU	LVA	POL	HUN	BGR	SVK	EST	CZE	HRV	ESP	PRT	BEL	DNK	NOR
	5%	LTU	LVA	POL	HUN	BGR	SVK	EST	CZE	HRV	ESP	PRT	BEL	DNK	NOR
	1%	LTU	LVA	POL	HUN	BGR	SVK	EST	CZE	HRV	ESP	PRT	BEL	DNK	NOR

Figure 7: Clustering of the excess mortality profile (for the 29 countries) due to COVID-19 for different levels of α: 15% (top), 5% (middle) and 1% (bottom).

8 Appendix

8.1 Proof of Theorem 3

Let us prove that $\mathbb{P}(S(n) \geq 2)$ vanishes as $n \to +\infty$. By definition of $S(n)$ we have

$$\begin{aligned}
\mathbb{P}\left(S(n) \geq 2\right) &= \mathbb{P}\left(\text{it exists } 2 \leq r \leq d(k) :\ U_r - r\ell_n \geq U_1 - \ell_n\right) \\
&\leq \mathbb{P}\left(\text{it exists } 2 \leq r \leq d(k) :\ U_r - U_1 \geq (r-1)\ell_n\right) \\
&= \mathbb{P}\left(\text{it exists } 2 \leq r \leq d(k) : \sum_{(i,j)\in S(k):\ 2\leq r_k(i,j)\leq r} T_{i,j} \geq (r-1)\ell_n\right) \\
&\leq \mathbb{P}\left(\text{it exists } (i,j) \text{ with } 2 \leq r_k(i,j) \leq r \leq d(k) :\ T_{i,j} \geq \ell_n\right) \\
&\leq \sum_{2\leq r_k(i,j)\leq d(k)} \mathbb{P}\left(T_{i,j} \geq \ell_n\right).
\end{aligned}$$

From Lemma 1 we know that under H_0, for all $\varepsilon > 0$, $n^{-\epsilon}T_{i,j} = n^{-\varepsilon} n d_n[i,j](\widehat{\theta}_n(i,j))$ that goes to 0 in probability, as $n \to +\infty$. Since $d(k) = k(k-1)/2$ is fixed we then obtain $\mathbb{P}(S(n) \geq 2) \to 0$ as $n \to +\infty$, which proves the wanted result.

8.2 Proof of Theorem 4

From Theorem 3 we have $\mathbb{P}(S(n) = 1) \to 1$ as $n \to +\infty$, from which we can deduce that for all $\xi > 0$

$$\begin{aligned}
\mathbb{P}(|U_{S(n)} - U_1| \geq \xi) &= \mathbb{P}(|U_{S(n)} - U_1| \geq \xi \cap \{S(n) = 1\}) + \mathbb{P}(|U_{S(n)} - U_1| \geq \xi \cap \{S(n) > 1\}) \\
&= \mathbb{P}(|U_{S(n)} - U_1| \geq \xi \cap \{S(n) > 1\}) \\
&\leq \mathbb{P}(S(n) > 1) \to 0,
\end{aligned}$$

which implies that $U_{S(n)}$ has the same limiting distribution as $U_1 = T_{1,2}$, see Lemma 1.

8.3 Proof of Theorem 5

Consider the general case $H_1(r)$ with $r > 1$, the particular case $H_1(1)$ being similar. We first show that $\mathbb{P}(S(n) \geq r)$ tends to 1 as $n \to +\infty$. Under $H_1(r)$, we have for all $r' < r$

$$\begin{aligned}
\mathbb{P}\left(U_r - r\ell_n \geq U_{r'} - r'\ell_n\right) &= \mathbb{P}\left((U_r - U_{r'}) \geq (r-r')\ell_n\right) \\
&= \mathbb{P}\left(\sum_{r'<r_k(i,j)\leq r} T_{i,j} \geq (r-r')\ell_n\right) \\
&\geq \mathbb{P}\left(T_{i,j}\mathbb{I}_{\{r_k(i,j)=r\}} \geq (r-r')\ell_n\right).
\end{aligned}$$

When $r_k(i,j) = r$, under $H_1(r)$ we have from Lemma 1 $T_{i,j} = U_n^1(i,j) + V_n^1(i,j)$ where $V_n^1(i,j) = \lambda[i,j] \times n + o_{a.s.}(n)$. From **(B)** we know that $l_n = n^{\varepsilon}$ with $\varepsilon < 1$, and we deduce that $\mathbb{P}\Big(T_{i,j}\mathbb{I}_{\{r_k(i,j)=r\}} \geq$

$(r-r')\ell_n\Big) \to 1$, as n tends to infinity, which proves that $\mathbb{P}(S(n) \geq r)$ tends to 1. Under $H_1(r)$, if $r_k(i,j) = r$ we have from Lemma 1 that $T_{i,j} \to U_n^1 + V_n^1$, where $V_n^1 = O(n)$. Since $U_r \geq T_{i,j}$ we obtain $\mathbb{P}(U_r \to +\infty) = 1$. From **(B)** we also have $\mathbb{P}(U_r - r\ell_n \to +\infty) = 1$. It implies that $\mathbb{P}(r \in \arg\max_{1\leq s\leq d(k)}\{U_s - s\ell_n\}) \to 1$ which implies that $\mathbb{P}(S(n) > r)$ tends to 0.

8.4 Proof of Proposition 1

Consider a group with at least two elements, $\mathcal{G}_s$, obtained from Algorithm 3 and assume that there exists $i_{s,j}$ such that $F_{i_{s,j}} \neq F_{i_{s,1}}$. We are then under an alternative of the form $H_1(r)$ which is asymptotically detected from Theorem 5, that is: $\mathbb{P}\left(i_{s,j} \in \mathcal{G}_s\right) \to 0$, as $n \to +\infty$.

References

AKAIKE, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control* **19**, 716–723.

INGLOT, T. & LEDWINA, T. (2006). Towards data driven selection of a penalty function for data driven Neyman tests. *Linear Algebra and its Applications* **417**, 124–133.

KAKLAUSKAS, A, MILEVICIUS, V. AND KAKLAUSKIENE, L. (2022). Effects of country success on COVID-19 cumulative cases and excess deaths in 169 countries. *Ecological indicators* **137**, 108703.

KALLENBERG, W. C. & LEDWINA, T. (1995). Consistency and Monte-Carlo simulation of a data driven version of smooth goodness-of-fit tests. *The Annals of Statistics* **23**, 1594–1608.

MCLACHLAN, G. J., BEAN, R. W. & BEN-TOVIM JONES, L. (2006). A simple implementation of a normal mixture approach to differential gene expression in multiclass microarrays. *Bioinformatics* **22**, 1608–1615.

MILHAUD, X., POMMERET, D., SALHI, Y. & VANDEKERKHOVE, P. (2022). Semiparametric two-sample admixture components comparison test: The symmetric case. *Journal of Statistical Planning and Inference* **216**, 135–150.

MILHAUD, X., POMMERET, D., SALHI, Y. & VANDEKERKHOVE, P. (2023). Two-sample contamination model test. *To appear in Bernoulli*.

PATRA, R. K. AND SEN, B. (2016). Estimation of a Two-component Mixture Model with Applications to Multiple Testing. *J. R. Statist. Soc.* B **78**, 869–893.

PODLASKI, R. & ROESCH, F. A. (2014). Modelling diameter distributions of two-cohort forest stands with various proportions of dominant species: A two-component mixture model approach. *Math. Biosci.* **249**, 60–74.

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

SHEN, Z., LEVINE, M. & SHANG, 7. (2018). An MM algorithm for estimation of a two component semiparametric density mixture with a known component. *Electronic Journal of Statistics* **12**, 1181–1209.

SHORACK, G. R. & WELLNER, J. A. (1986). *Empirical Processes with Applications to Statistics*. Wiley, New York.

TEICHER, H. (1963). Identifiability of finite mixtures. *Ann. Math. Stat.* **34**, 1265–1269.

WALKER, M., MATEO, M., OLSZEWSKI, E., SEN, B. & WOODROOFE M. (2009). Clean kinematic samples in dwarf spheroidals: An algorithm for evaluating membership and estimating distribution parameters when contamination is present. *Astron. J.* **137**, 3109–3138.