Journal of Classification (2022) 39:553–577
https://doi.org/10.1007/s00357-022-09419-7

Hierarchical Means Clustering

Maurizio Vichi[1] · **Carlo Cavicchia**[2] · **Patrick J. F. Groenen**[2]

Accepted: 15 August 2022 / Published online: 23 September 2022

Abstract
In the cluster analysis literature, there are several partitioning (non-hierarchical) methods for clustering multivariate objects based on model estimation. Distinct to these methods is the use of a system of n nested statistical models and the optimization of a loss function to best-fit a clustering model to observed data. Many hierarchical clustering methods are not model-based where hierarchy is obtained using a divisive or agglomerative greedy procedure. This paper aims to fill this gap by proposing a novel hierarchical cluster analysis methodology called Hierarchical Means Clustering. HMC produces a set of nested partitions with a centroid-based model estimated via least-squares by minimizing the total within-cluster deviance of the n partitions in the hierarchy. Hierarchical Means Clustering produces a hierarchy formed by n-1 nested partitions from 2 to n clusters with minimal total cluster deviance. Six real data examples are featured, and key links to k-means, Ward's method, Bisecting k-means and model-based hierarchical agglomerative clustering methods are discussed.

Keywords Clustering · Hierarchy · k-means · Hierarchical clustering

This paper is dedicated to the memory of Allan Gordon, a pioneer and a great scientist in the field of Clustering and Classification. He was President of International Federation of Classification Societies and I consider him in this field my great Master. Allan passed away in October 2021 and will be sorely missed and fondly remembered. Maurizio Vichi.

✉ Carlo Cavicchia
cavicchia@ese.eur.nl

Maurizio Vichi
maurizio.vichi@uniroma1.it

Patrick J. F. Groenen
groenen@ese.eur.nl

[1] Department of Statistical Sciences, University of Rome La Sapienza, Rome, Italy

[2] Econometric Institute, Erasmus University Rotterdam, Rotterdam, Netherlands

1 Introduction

Hierarchical clustering methods are famed for yielding a *hierarchy* of partitioned objects (Hartigan, 1975; Gordon, 1999; Müllner, 2011). They start from dissimilarity data between pairs of n objects and produce a nested set of $n-1$ partitions. Most commonly used hierarchical clustering methods are *agglomerative* where pairs of objects or clusters are merged into larger ones until a single cluster emerges. In contrast, *divisive* clustering methods split the whole set of objects or its successively divided clusters until all clusters are singletons.

These agglomerative and divisive clustering methods feature well-known weaknesses. First, they are recursive, greedy algorithms that can never undo prior steps. This implies that once clusters are merged or split, they are recursively detected and may not prove optimal at any successive or final step in the algorithm, producing a worse overall hierarchical solution. Second, they are not switching algorithms (Hartigan, 1975; Gordon, 1999), thus ruling out any improvement of homogeneity criterion — something possible for iterative relocation partitioning methods (Gordon, 1999). Third, they are usually heuristic and lack either a statistical model for the dissimilarity data or an objective function that can be optimized to find the best hierarchy. Fourth, they typically use dissimilarity data requiring massive memory — $O(n^2)$ — and computational complexity — $O(n^3)$, even if the latter can be reduced by a heap to $O(n^2 \log n)$ as reported in Cormen et al. (1990). Fifth, they can suffer from outliers that mask clusters and from noise objects that chain clusters according to the distance formula chosen for merging clusters. Sixth, the interpretation of all $n-1$ partitions in the hierarchy might be difficult, especially when n is large. Gordon has observed that ultra-large sets of hierarchical partitions and clusters tend to escape comprehensive treatment by investigators and can blur interpretation.

One solution for this last weakness is the so-called parsimonious tree that limits the hierarchy to a reduced number of internal nodes K where $K < n$. Parsimonious trees thus view clusters appearing in excess of K as very close to each other, appearing almost indistinguishable in the dendrogram. Several algorithms have been proposed for seeking parsimonious trees directly from (dis)similarity data (Hartigan, 1967; Sriram, 1990).

However, as far as we are aware, model-based hierarchical algorithms are not available for parsimonious trees treating rectangular multivariate data. We thus propose a new hierarchical cluster analysis methodology named *Hierarchical Means Clustering* (HMC) that starts with the rectangular data matrix (objects by variables) or the squared Euclidean distance matrix. HMC indeed differs from the Direct Optimization Methods (Gordon, 1999) that transform the dissimilarity matrix into the best ultrametric distance matrix using least-squares estimation. HMC is founded on the system of n centroid-based statistical models which is defined by n equations representing a set of nested partitions. The simultaneous least-squares estimation of these models defines a quadratic loss function totaling the within-cluster deviance of $n-1$ partitions that is minimized while requiring clusters of partitions to be either disjoint or nested. While HMC identifies the complete hierarchy's $n-1$ internal nodes, it also facilitates the construction of a parsimonious tree.

HMC uses an efficient block-coordinate descent algorithm (Zangwill, 1969) that minimizes the total deviance of the $n-1$ nested partitions by alternating three steps: (a) assigning each of the n objects to the nearest cluster in the current partition of K clusters to next compute the related centroids, (b) amalgamating the K clusters with minimum within-cluster deviance until the whole set of objects is obtained, and (c) splitting the cluster having the largest within-cluster deviance from $K+1$ to n. In sum, concentrate the dense; disperse the scattered. This methodology can be repeated for different values of K from 1 to n in

order to find the optimal solution. Our suggestion is to restrict the search from 1 to m, with $m \ll n$, and consider a parsimonious tree with m or fewer nodes. Some information is lost in this process, but the main heterogeneity of the data is represented more clearly with a tree of m nodes. This views clusters from m to n as neither truly isolated from each other nor relevant to retain.

It is important to underscore that HMC gives a solution to several weaknesses in hierarchical clustering. It optimizes an objective, least-squares loss function of the system of n nested statistical models. HMC allows objects to switch from one cluster to another using the iterative relocation of step (a) described earlier. It treats object by variable matrices or the squared Euclidean distances matrix according to need. HMC yields, if requested, a parsimonious tree that averts intensive computation.

Launching the hierarchy from a pre-specified K cluster partition to then work upward and downward is one useful new addition to the cluster analysis toolbox. This means starting not necessarily from the entire set of objects or from n singleton clusters (as classical divisive or agglomerative methods do) but also allowing for many further options. For example, a researcher may begin with a partition: obtained by another clustering application, derived from theory, or provided by a categorical variable belonging to the dataset. Construction of an entire dendrogram around this partition may subsequently assess and rank the hierarchical relations of the starting partition. Also, instead of a single final hierarchy obtained by classical hierarchical clustering methods, HMC can generate a series of hierarchies to compare enlisting different serial values of K. Finally, the launch from a K-cluster partition allows identifying the most relevant hierarchical relations around a given K-cluster partition from $1 \leq K \leq m$ with 1 and m closely bracketing K.

HMC uniquely relates with *k-means* (MacQueen, 1967), *Ward's agglomerative method* Ward (1963) and the *Bisecting k-means* Gordon (1981); Steinbach et al. (2000). In particular, when the iterative relocation of objects from one cluster to another proves trivial where step (a) is moot in cases $K = n$ or $K = 1$, HMC nicely reduces to Ward's method or Bisecting k-means, respectively. HMC further reduces to k-means when a partition in k clusters is required, while the merging in step (b) and splitting in step (c) may be skipped. HMC may thus be seen as a derivative of Ward's method and Bisecting k-means invoking special cases of $K = n$ and $K = 1$ where the solution of HMC can never degrade beyond these.

Furthermore, HMC has several connections to the class of model-based hierarchical agglomerative clustering methods proposed by Fraley (1998). Assuming a Gaussian mixture model for the data, these methods are based on a dissimilarity measure corresponding to the decrease in the likelihood function when merging two clusters (Banfield and Raftery, 1993). By imposing four different covariance structures at each hierarchical stage, several criteria can be minimized (see Fraley, 1998, Table 1). Here, two spherical and two non-spherical models are considered by Fraley. In general, HMC is related to the spherical family of these models since minimizing the within-cluster deviance (i.e., k-means criterion) equates to maximizing the log-likelihood in one of these hierarchical Gaussian mixture models (Celeux and Govaert, 1995) — the latter again reducing to the Ward's method and thus is directly connected to HMC. Still, such hierarchical clustering mixture models, like classical approaches, are sequential and cannot optimize an objective function for the complete hierarchy.

The HMC algorithm is "NP-hard" since it includes the solution of partitioning problems known to be NP-hard (Křivánek and Morávek, 1986). Hence, the global optimum solution is not guaranteed in a single run. To increase the probability of finding a global optimum, it is wise to run the algorithm starting from different values of K, generally in a restricted interval with a multi-start procedure. HMC may appear computationally complex, but each

run of the HMC algorithm quickly converges to a local minimum in a few steps. Even with a multi-start procedure, the algorithm results fast, especially when the divisive step (c) is not needed and a parsimonious tree is requested.

This paper is organized as follows. Section 2 specifies notation and basic clustering ideas. Section 3 presents the HMC model, while Section 4 is devoted to its least-squares estimation and the corresponding algorithms. Section 5 features six real data examples where HMC is numerically illustrated and compared with Ward's method, Bisecting k-means, and three model-based hierarchical agglomerative clustering methods. In Section 5.2, two examples using HMC appear: an eased interpretation of a data matrix through a parsimonious tree, and a build-out of an entire dendrogram around a given partition to rank its hierarchical relations. The paper concludes with Section 6.

2 Notation and Background of Cluster Analysis Notions

For the readers' convenience, an overview of the notation used in this paper is listed here.

n, k, p	number of objects; number of clusters of the partition of objects; number of variables.
$O = \{o_1, \dots, o_n\}$	set of n objects to be classified or simply its sets of indices (labels) $1, 2, \dots, n$.
C^k	partition $C^k = \{C_1^k, \dots, C_k^k\}$ of O, where C_h^k is the h-th cluster of the partition C^k of objects of O in k clusters.
n_h^k	size of cluster C_h^k.
$\mathbf{1}_p, \mathbf{I}_p$	the p-dimensional unitary vector of which all elements equal to 1, and the $(p \times p)$ identity matrix, respectively.
$\mathbf{X} = [x_{ij}]$	$(n \times p)$ data matrix, where x_{ij} is the value observed on the i-th object for the j-th variable.
$\mathbf{D} = [d_{il}]$	$(n \times n)$ dissimilarity matrix, where d_{il} is a dissimilarity between objects (o_i, o_l) $i, l = 1, \dots, n$.
$\mathbf{D}^B = [d_{lh}^B]$	$(k \times k)$ between-cluster isolation matrix, where $d_{lh}^B \geq 0$ denotes a measure of isolation between clusters C_l^k and C_h^k $l, h = 1, \dots, k$ of the partition C^k; hence $d_{hh}^B = 0$ for all h. Given a partition in clusters, a measure of isolation is a measure of distance between clusters. The latter is generally evaluated: (i) as a function of the distances between all pairs of objects, one belonging to a cluster and the other belonging to the other cluster; or, (ii) as the distance between the centroids of the two clusters. In this paper we use (i) with the square euclidean distance, so as to define the between-cluster deviance.
$\mathbf{D}^W = [d_{lh}^W]$	$(k \times k)$ within-cluster heterogeneity (diagonal) matrix, where $d_{lh}^W = 0$ for all $l \neq h$, and $d_{hh}^W \geq 0$ is a measure of heterogeneity of cluster C_h^k, $h = 1, \dots, k$. Given a partition in clusters, a measure of heterogeneity is a measure of distance between objects within the clusters. The latter is generally evaluated: (i) as a function of the distances between all pairs of objects, all belonging to the same cluster; or, (ii) as the distance between objects and the centroids of a cluster. In this paper we use (i) with the square euclidean distance, so as to define the within-cluster deviance.

$\mathbf{U}_k = [u_{ih}^k,\ i = 1, \ldots, n,\ h = 1, \ldots, k]$	$(n \times k)$ membership matrix, where $u_{ih}^k = 1$ if the i-th object o_i belongs to the h-th cluster C_h^k, $u_{ih}^k = 0$ otherwise, therefore the column $\mathbf{u}_h^k$ $(h = 1, \ldots, k)$ of $\mathbf{U}_k$ represents the h-th cluster of the partition C^k of objects in k clusters, and the matrix $\mathbf{U}_k$ specifies the partition C^k. Matrix $\mathbf{U}_k$ is binary and row-stochastic and thus has only one nonzero element per row.
$\mathbf{U}_k'\mathbf{U}_k = \text{diag}(n_{1,k}, \ldots, n_{k,k})$	$(k \times k)$ diagonal matrix whose diagonal elements are the sizes of the k clusters.
$\mathbf{U}_k^+$	is the Moore-Penrose inverse of the matrix $\mathbf{U}_k$, it is equal to $(\mathbf{U}_k'\mathbf{U}_k)^{-1}\mathbf{U}_k$.
$\mathbf{M}_k = [m_{hj}^k,\ h = 1, \ldots, k,\ j = 1, \ldots, p]$	$(k \times p)$ prototype/centroid matrix associated to a partition C^k and a membership matrix $\mathbf{U}_k$. $\mathbf{M}_k$ represents both the matrix whose rows are the general points in $\mathbb{R}^p$ that serve as prototypes for the clusters, and the special situation where these centers are the centroids for the clusters, i.e., the average of all cluster members.
$\mathbf{G} = [g_{il}]$	$(n \times n)$ ultrametric matrix, where g_{il} is the ultrametric distance between objects.
$\mathbf{E}_k = [e_{ij}^k,\ i = 1, \ldots, n,\ j = 1, \ldots, p]$	$(n \times p)$ matrix of error terms.
$\|\mathbf{X}\|^2$	is the sum of squares of $\mathbf{X}$, it is equal to $\text{tr}(\mathbf{X}'\mathbf{X}) = \sum_{i=1}^n \sum_{j=1}^p x_{ij}^2$.

2.1 The Centroid-Based Model of *k*-Means

The k-means model casts each observation as equaling one of the k centroids — one centroid of each cluster perturbed by error in measuring the features. In matrix terms, this model is

$$\mathbf{X} = \mathbf{U}_k\mathbf{M}_k + \mathbf{E}_k, \tag{1}$$

where $\mathbf{U}_k$, $\mathbf{M}_k$, $\mathbf{E}_k$ are matrices of membership, prototype and error, respectively. Here, the usual least-squares estimation partitions the p-dimensional space into a Voronoi tessellation where centroid-based Model (1) leads to the k-means optimization problem formalized as:

$$\|\mathbf{X} - \mathbf{U}_k\mathbf{M}_k\|^2 = \text{tr}[(\mathbf{X} - \mathbf{U}_k\mathbf{M}_k)'(\mathbf{X} - \mathbf{U}_k\mathbf{M}_k)] \rightarrow \min_{\mathbf{U}_k, \mathbf{M}_k}$$

subject to

$$u_{ih}^k \in \{0, 1\},\ \forall i, h$$
$$\mathbf{U}_k\mathbf{1}_k = \mathbf{1}_n.$$

The solution of the minimization problem w.r.t. $\mathbf{M}_k$ is $\mathbf{M}_k = \mathbf{U}_k^+\mathbf{X}$, where the rows of the solution $\mathbf{M}_k$ are the cluster centroids, i.e., the averages of the data points in the clusters.

2.2 The Parsimonious Tree: a *k*-Cluster Hierarchical Partition

Since the notion of the parsimonious tree is not well-known, a brief overview appears here. The hierarchical classifications produced by clustering algorithms usually comprise partitions into k clusters for all values of k between 1 and n depicted by *dendrograms* featuring $n - 1$

internal nodes. Several authors have noted that the complete sets of partitions and clusters go untreated by investigators, even hindering interpretation (Gordon, 1999). One approach for resolving this difficulty has involved the construction of *parsimonious trees* that contain a limited number of internal nodes. Some information is lost here, but the main features of the data are represented more clearly (Gordon, 1999).

A parsimonious tree (Vichi, 2008) induces a *hierarchical partition* $HP(k)$ that partitions data into k clusters and their hierarchy of "nested partitions" in $k-1, k-2, \ldots, 1$ clusters formed when passing from k to $k-1$ by these same clusters, unless one is the amalgamation of two. By considering two well-known properties characterizing the clusters of a partition, namely isolation and heterogeneity as described by Cormack (1971), $HP(k) = \{C^k = \{C_1^k, \ldots, C_k^k\}, C_{k+1}^k, \ldots, C_{2k-1}^k\}$ is formed by $2k-1$ clusters, where the first k, $C_1^k, \ldots, C_k^k$ represent a partition C^k of O with heterogeneity d_{hh}^W $(h = 1, \ldots, k)$, and the remaining $C_{k+1}^k, \ldots, C_{2k-1}^k$ are obtained by $k-1$ pairwise possible amalgamations of subsets of C^k with isolation between clusters d_{lh}^B $(l, h = 1, \ldots, k;\ h \neq l)$, such that

$$\max(d_{hh}^W : k = 1, \ldots, k) \leq \min(d_{lh}^B : l, h = 1, \ldots, k;\ h \neq l). \tag{2}$$

Here, the largest heterogeneity of a cluster of C^k never exceeds the smallest isolation between two clusters of C^k. This property is known to define a *well-structured partition* C^k Rubin (1967). Thus, for each pair $C_h^k,\ C_m^k \in C^k \rightarrow (C_m^k \cap C_h^k) \in (C_m^k, C_h^k, \emptyset)$, that is, for each pair of clusters belonging to a hierarchical partition, either they are nested, or they are disjoint. The hierarchical partition can also be expressed in terms of three matrices $\mathbf{U}_k$, $\mathbf{D}^W$ and $\mathbf{D}^B$ Vichi (2008),

$$\mathbf{G} = \mathbf{U}_k \mathbf{D}^B \mathbf{U}_k' + \mathbf{U}_k \mathbf{D}^W \mathbf{U}_k' - \text{diag}(\mathbf{U}_k \mathbf{D}^W \mathbf{U}_k'), \tag{3}$$

where matrix $\mathbf{D}^B$ of order k is an ultrametric matrix, its triplets ensuring ultrametric inequality. It follows that $\mathbf{D}^B$ has at most $k-1$ different off-diagonal values. Matrix $\mathbf{G}$ is a $(2k-1)$-ultrametric matrix, a square dissimilarity matrix of order n with off-diagonal elements that can assume one of at most $(2k-1)$ different values $0 < d_{kk}^W \leq d_{lh}^B$ $(l, h = 1, \ldots k;\ h \neq l)$, and whose triplets also satisfy the ultrametric inequality. $\mathbf{G} = [g_{il}], g_{ii} = 0,\ g_{il} \geq 0,\ g_{il} = g_{li}, g_{il} \leq \max(g_{iv}, g_{lv})\ \forall(i, l, v)$; furthermore $g_{il} \in \{0, d_{hh}^W, d_{lh}^B\}$, with $0 < d_{hh}^W \leq d_{lh}^B\ \forall(l, h : h \neq l)$.

Any $(2K-1)$-ultrametric matrix can be represented by a dendrogram not exceeding $2K-1$ levels (heights of the tree), concisely named $(2K-1)$-dendrogram. The following walk-through of the parsimonious tree representation using the following synthetic dataset considers the partition $C^3 = \{C_1^3 = \{1, 2, 4, 5, 6\},\ C_2^3 = \{7, 9\},\ C_3^3 = \{3, 8, 10\}\}$ of 10 objects in 3 clusters with associated matrices $\mathbf{D}^B$, $\mathbf{D}^W$, and $\mathbf{U}_3$, that is,

$$\mathbf{D}^B = \begin{bmatrix} 0 & 4 & 5 \\ & 0 & 5 \\ & & 0 \end{bmatrix},\ \mathbf{D}^W = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{bmatrix},\ \mathbf{U}_3' = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Here, equation (2) is satisfied, and C^3 is thus a partition in 3 well-structured clusters. Indeed, $4 = \min\{d_{lh}^B : h, l = 1, \ldots, 3\} > \max\{d_{hh}^W : h = 1, \ldots, 3\} = 3$. Furthermore, it can be verified where matrix $\mathbf{D}^B$ is ultrametric that matrix

$$\mathbf{G} = \begin{bmatrix} 0 & 1 & 5 & 1 & 1 & 1 & 4 & 5 & 4 & 5 \\ 1 & 0 & 5 & 1 & 1 & 1 & 4 & 5 & 4 & 5 \\ 5 & 5 & 0 & 5 & 5 & 5 & 5 & 2 & 5 & 2 \\ 1 & 1 & 5 & 0 & 1 & 1 & 4 & 5 & 4 & 5 \\ 1 & 1 & 5 & 1 & 0 & 1 & 4 & 5 & 4 & 5 \\ 1 & 1 & 5 & 1 & 1 & 0 & 4 & 5 & 4 & 5 \\ 4 & 4 & 5 & 4 & 4 & 4 & 0 & 5 & 3 & 5 \\ 5 & 5 & 2 & 5 & 5 & 5 & 5 & 0 & 5 & 2 \\ 4 & 4 & 2 & 5 & 5 & 5 & 5 & 0 & 5 & 2 \\ 5 & 5 & 2 & 5 & 5 & 5 & 5 & 2 & 5 & 0 \end{bmatrix}$$

is also ultrametric and has associated the 5-dendrogram with 5 different nodes in Fig. 1.

There exists a one-to-one correspondence between hierarchical partitions and $(2k-1)$ -ultrametric matrices. Let $\mathcal{C}_H$ be the set of hierarchical partitions of O, and let $\mathcal{D}_{HP}$ be the set of $(2k-1)$-ultrametric matrices $\mathbf{G}$. Then there exists a one-to-one correspondence between $\mathcal{C}_H$ and $\mathcal{D}_{HP}$ Vichi (2008). Each hierarchical partition thus maps to a matrix $\mathbf{G}$ of the form (3) that satisfies inequality (2). The complete dendrogram of the n objects is given by $HP(n)$.

3 The HMC Model

So far, almost all hierarchical clustering methods are based on heuristics. Consequently, the fit to the data of a hierarchy produced by such a heuristic is unknown. In addition, it remains unclear how its quality can be compared to a hierarchy obtained by a different heuristic. To solve these issues, we propose a single loss function in this section and an algorithm that optimizes the loss function.

Given the $(n \times p)$ data matrix $\mathbf{X}$, the $(n \times k)$ membership matrix $\mathbf{U}_k = [u_{ih}^k : i = 1, \ldots, n, h = 1, \ldots, k]$, the $(k \times p)$ prototype matrix $\mathbf{M}_k$, and the residuals matrix $\mathbf{E}_k$, the model corresponding to the *Hierarchical Means Clustering* (HMC) is specified by the following system of equations

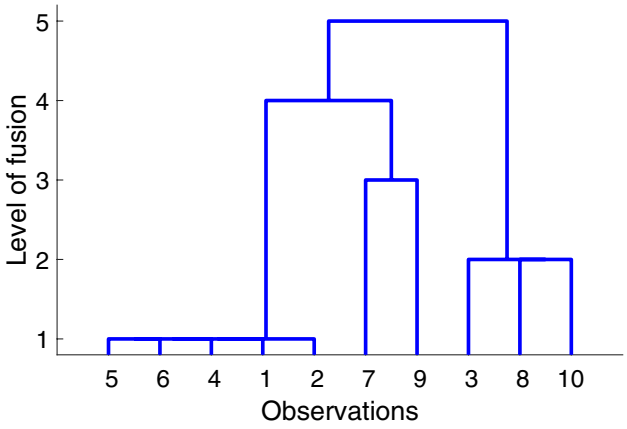


Fig. 1 5-dendrogram corresponding to the ultrametric distance matrix $\mathbf{G}$

$$\left.\begin{array}{l}\left.\begin{array}{rl}\mathbf{X} &= \mathbf{U}_1\mathbf{M}_1 + \mathbf{E}_1\\ \mathbf{X} &= \mathbf{U}_2\mathbf{M}_2 + \mathbf{E}_2\\ &\ldots \quad \ldots \quad \ldots\\ \mathbf{X} &= \mathbf{U}_k\mathbf{M}_k + \mathbf{E}_k\\ &\ldots \quad \ldots \quad \ldots\\ \mathbf{X} &= \mathbf{U}_K\mathbf{M}_K + \mathbf{E}_K\end{array}\right\} \text{Parsimonious tree: Hierarchical partition in K clusters}\\ \quad \ldots \quad \ldots \quad \ldots\\ \mathbf{X} = \mathbf{U}_n\mathbf{M}_n + \mathbf{E}_n\end{array}\right\} \text{Complete tree} \tag{4}$$

subject to the constraints

$$\mathbf{U}_k = [u_{ih}^k \in \{0,1\} : i = 1,\ldots,n, h = 1,\ldots,k] : k = 1,\ldots,n \text{ (binary)} \tag{5}$$

$$\mathbf{U}_k\mathbf{1}_k = \mathbf{1}_n \text{ (row stochastic)} \tag{6}$$

$$\begin{array}{c}\mathbf{U}_{k-1} = [\mathbf{U}_k \setminus \{\mathbf{u}_{k-1}^k, \mathbf{u}_k^k\}, \mathbf{u}_{k-1}^{k-1}] \text{ (nested partitions)}\\ \mathbf{u}_{k-1}^{k-1} = \mathbf{u}_{k-1}^k + \mathbf{u}_k^k.\end{array} \tag{7}$$

The first equation of model (4) does not identify clusters since $k = 1$ specifies a single cluster and $\mathbf{M}_1$ corresponds to the mean vector of the entire dataset. Furthermore, the last equation of model (4) with $k = n$ yields the trivial partition of n objects in n clusters. We include these two cases for sake of simplicity without loss of generality that will be clearer in Section 4.

Matrix $\mathbf{U}_k$, for $k = 3,\ldots,n$, has $k-2$ columns equal to $\mathbf{U}_{k-1}$ — assumed without loss of generality to be the first $k-2$ columns of $\mathbf{U}_{k-1}$. The last column of $\mathbf{U}_{k-1}$, $\mathbf{u}_{k-1}^{k-1}$, equals the sum of the last two columns of $\mathbf{U}_k$, $\mathbf{u}_{k-1}^k + \mathbf{u}_k^k$, for $k = 3,\ldots,n-1$. This assumption is crucial to understand the notation used in equation (7), where the matrix $\mathbf{U}_{k-1}$ is equal to $\mathbf{U}_k$ deprived of the two columns $\mathbf{u}_{k-1}^k$ and $\mathbf{u}_k^k$, and to which their sum ($\mathbf{u}_{k-1}^{k-1}$) is attached.

It can be observed with an agglomerative approach that $\mathbf{U}_{k-1}$ is obtained from $\mathbf{U}_k$ by summing $\mathbf{u}_{k-1}^k + \mathbf{u}_k^k$, which indeed implies the merging of the two corresponding clusters. Vice-versa using a divisive approach, $\mathbf{U}_k$ is obtained from $\mathbf{U}_{k-1}$ by splitting the last column of $\mathbf{U}_{k-1}$, $\mathbf{u}_{k-1}^{k-1}$, into two columns $\mathbf{u}_{k-1}^k$ and $\mathbf{u}_k^k$.

If a parsimonious tree for a hierarchical partition in K clusters is required, model (4) is limited to the first K equations that allow the estimation of the partition in K clusters with the nested partitions in $K-1, K-2,\ldots,1$ clusters. Once all n equations are considered, a complete tree or hierarchical classification is obtained.

The parameters to estimate in model (4) are the n prototype matrices $\mathbf{M}_k$ with $k = 1,\ldots,n$ and the n membership matrices (or corresponding classification variables) $\mathbf{U}_k$ with $k = 1,\ldots,n$ constrained as being binary, row-stochastic, and forming a set of nested partitions through constraints (5)–(7). The least-squares estimation of model (4) can therefore be obtained by minimizing $\sum_{k=1}^n \|\mathbf{E}_k\|^2$ with respect to the binary membership matrices $\mathbf{U}_k$ and the continuous prototype matrices $\mathbf{M}_k$, that is,

$$F(\mathbf{U}_1, \ldots, \mathbf{U}_n, \mathbf{M}_1, \ldots, \mathbf{M}_n) = \sum_{k=1}^{n} \|\mathbf{X} - \mathbf{U}_k\mathbf{M}_k\|^2 = \left\| \begin{array}{c} \mathbf{X} - \mathbf{U}_1\mathbf{M}_1 \\ \vdots \\ \mathbf{X} - \mathbf{U}_n\mathbf{M}_n \end{array} \right\|^2. \tag{8}$$

To simplify from here, we will refer to $F(\mathbf{U}_1, \ldots, \mathbf{U}_n, \mathbf{M}_1, \ldots, \mathbf{M}_n)$ as F. It is important to focus on two special terms in the previous sum (8): for $k = 1$, we have that $\|\mathbf{X} - \mathbf{U}_1\mathbf{M}_1\|^2$ is the deviance of $\mathbf{X}$; for $k = n$, $\|\mathbf{X} - \mathbf{U}_n\mathbf{M}_n\|^2 = \|\mathbf{X} - \mathbf{X}\|^2 = 0$, since $\mathbf{U}_n = \mathbf{I}_n$. If a parsimonious tree is required, the sum in equation (8) is restricted to $k = 1, \ldots, K$ with $K < n$.

The rationale for choosing F as objective function for HMC is summarized as follows.

1. The new performance criterion F allows assessing the fit of the final cluster hierarchy to the given data. Indeed, goodness of fit can be computed.
2. The new methodology using function F finds a strong connection between agglomerative and divisive clustering; extant methods are only agglomerative or divisive. The new methodology is hybrid: part-agglomerative and part-divisive.
3. Function F finds another strong connection between hierarchical (Ward's method and Bisecting k-means) and non-hierarchical clustering (k-means).

3.1 Properties of HMC

In this section, the loss function (8) is rewritten to ease and improve efficiency of computation, expressing it as a weighted sum of rises in within-cluster deviance when merging two clusters from two subsequent partitions. This saves unnecessary operations. We recall that (8) sums the within-cluster deviance of single clusters within each partition of the hierarchy. The total within-cluster deviance of the partition in k clusters can be expressed by

$$\|\mathbf{X} - \mathbf{U}_k\mathbf{M}_k\|^2 = \sum_{h=1}^{k} \sum_{i \in C_h} \|\mathbf{x}_i - \mathbf{m}_h^k\|^2 = \sum_{h=1}^{k} w_h^k, \tag{9}$$

where $\mathbf{m}_h^{k\prime}$ is the h-th row of $\mathbf{M}_k$, i.e., the cluster prototype for C_h^k in $\mathbf{U}_k$, and, w_h^k is the within-cluster deviance of cluster h of the partition in k clusters. The loss (8) can therefore be rewritten as

$$F = \sum_{k=1}^{n} \|\mathbf{X} - \mathbf{U}_k\mathbf{M}_k\|^2 = \sum_{k=1}^{n} \sum_{h=1}^{k} w_h^k. \tag{10}$$

Since the partitions are nested, each partition $\mathbf{U}_k$ for $k = 3, \ldots, n$ has $k - 2$ clusters equal to the previous one in the hierarchy ($\mathbf{U}_{k-1}$). This means we can further simplify the loss (10) by introducing the difference between two subsequent partitions $\mathbf{U}_k$ and $\mathbf{U}_{k-1}$

$$I(\mathbf{U}_k, \mathbf{U}_{k-1}) = \|\mathbf{X} - \mathbf{U}_{k-1}\mathbf{M}_{k-1}\|^2 - \|\mathbf{X} - \mathbf{U}_k\mathbf{M}_k\|^2 = \sum_{h=1}^{k-1} w_h^{k-1} - \sum_{h=1}^{k} w_h^k. \tag{11}$$

$I(\mathbf{U}_k, \mathbf{U}_{k-1})$ measures the rise in the deviance when merging two clusters from the partition in k clusters to obtain the partition in $k - 1$ clusters. This depends only on two clusters: $\mathbf{u}_{k-1}^k$

and $\mathbf{u}_k^k$. To proceed from $\mathbf{U}_k$ to $\mathbf{U}_{k-1}$, two clusters $\mathbf{u}_{k-1}^k$, $\mathbf{u}_k^k$ are merged to form $\mathbf{u}_{k-1}^{k-1}$. This necessarily incurs a gain in the deviance equal to the deviance between the two merged clusters. Indeed, the deviance w_{k-1}^{k-1} of the merged cluster $\mathbf{u}_{k-1}^{k-1}$ equals the sum of the two deviances $w_{k-1}^k + w_k^k$ of the two clusters to be merged $\mathbf{u}_{k-1}^k$, $\mathbf{u}_k^k$, plus their between-deviance. Since the partitions $\mathbf{U}_k$ and $\mathbf{U}_{k-1}$ have $k-2$ clusters in common, say these are the first $k-2$ clusters, then the difference $I(\mathbf{U}_k, \mathbf{U}_{k-1})$ can be simplified into

$$I(\mathbf{U}_k, \mathbf{U}_{k-1}) = I(\mathbf{u}_{k-1}^{k-1}; \mathbf{u}_{k-1}^k, \mathbf{u}_k^k) = w_{k-1}^{k-1} - w_{k-1}^k - w_k^k, \tag{12}$$

which represents the deviance between clusters $\mathbf{u}_{k-1}^k$ and $\mathbf{u}_k^k$.

Proposition 1 *The loss function* (8) *might be written as a weighted sum of* $I(\mathbf{U}_k, \mathbf{U}_{k-1})$ *where weights are larger for the differences between partitions having more clusters* (*i.e.*, *the larger k is*, *the larger the weight*):

$$F = \sum_{k=2}^{n} (k-1)(w_{k-1}^{k-1} - w_{k-1}^k - w_k^k). \tag{13}$$

Proof

$$\begin{aligned}
F &= \sum_{k=1}^{n} \|\mathbf{X} - \mathbf{U}_k \mathbf{M}_k\|^2 \\
&= \|\mathbf{X} - \mathbf{U}_1 \mathbf{M}_1\|^2 + \cdots + \|\mathbf{X} - \mathbf{U}_n \mathbf{M}_n\|^2 \\
&= \|\mathbf{X} - \mathbf{U}_1 \mathbf{M}_1\|^2 - \|\mathbf{X} - \mathbf{U}_2 \mathbf{M}_2\|^2 + 2(\|\mathbf{X} - \mathbf{U}_2 \mathbf{M}_2\|^2 - \|\mathbf{X} - \mathbf{U}_3 \mathbf{M}_3\|^2) + \ldots \\
&\quad + (n-1)(\|\mathbf{X} - \mathbf{U}_{n-1} \mathbf{M}_{n-1}\|^2 - \|\mathbf{X} - \mathbf{U}_n \mathbf{M}_n\|^2) \\
&= \sum_{k=2}^{n} (k-1) I(\mathbf{U}_k, \mathbf{U}_{k-1}) \\
&= \sum_{k=2}^{n} (k-1)(w_{k-1}^{k-1} - w_{k-1}^k - w_k^k).
\end{aligned}$$

To thus minimize (8) subject to (13), it is necessary to minimally increase the difference $I(\mathbf{u}_{k-1}^{k-1}; \mathbf{u}_{k-1}^k, \mathbf{u}_k^k)$ when two clusters of the partition in k clusters are merged to form one of the partition in $k-1$ clusters. Thus, merged clusters must be the two nearest, those with minimum between-deviance. Note that to move from a $k-1$ partition to a k cluster solution by splitting one of the clusters into two, function (9) must be maximized while the partition with the maximal between-deviance yields the new pair having minimum within-deviance. □

3.2 HMC Formulation for Squared Euclidean Distances

So far, HMC has been cast as a model for rectangular data requiring less memory and computational effort versus dissimilarity data modeling. Yet, an alternative formulation of HMC is possible by considering the squared Euclidean distances between $\mathbf{x}_i$ and $\mathbf{x}_j$, that is, $q_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$ $(i, j = 1, \ldots, n)$, instead of the data matrix. The objective function (8) can now be rewritten as

$$
\begin{aligned}
F &= \sum_{k=1}^{n} \|\mathbf{X} - \mathbf{U}_k\mathbf{M}_k\|^2 \\
&= \sum_{k=1}^{n} \mathrm{tr}[(\mathbf{X} - \mathbf{U}_k\mathbf{M}_k)'(\mathbf{X} - \mathbf{U}_k\mathbf{M}_k)] \\
&= \sum_{k=1}^{n} \sum_{h=1}^{k} \frac{1}{2n_h^k} \sum_{i \in C_h^k} \sum_{j \in C_h^k} q_{ij}.
\end{aligned} \tag{14}
$$

Thus, the HMC objective function can also be computed by considering only the matrix of squared Euclidean distances between the rows of $\mathbf{X}$ and the clustering in C^k.

Note that instead of squared Euclidean distances also other dissimilarity measures could be used, much in the same way as for classical hierarchical clustering methods that use dissimilarities. A dissimilarity measure different from the squared Euclidean distance would lead to another overall objective function than F.

4 The HMC Algorithm

We provide two algorithms that minimize (13): the unconstrained version in the next subsection and the constrained one later in Section 4.2.

4.1 Unconstrained HMC

To minimize (13), we propose Algorithm 1 that builds the clustering hierarchy by alternating four phases: (1) partition computation of n objects and K clusters, (2) the related centroids computation, (3) agglomeration clustering steps from $K-1$ to 1, and (4) division clustering steps from $K+1$ to n, or to a limited number of nodes m. The HMC algorithm is destined to find the optimal tree under this local search. When $1 < K < n$, high quality local minima emerge from the alternating agglomerative and divisive clustering steps. For $K = n$, these two steps lead to a fixed path of hierarchical partitioning. The algorithm seeks the optimal solution for $K = 2, \ldots, n$ according to the evaluation of F at Step 5, where the optimal values of F and K, F^{opt} and K^{opt}, respectively, are updated and stored. However, it is possible to set a maximum number of clusters $m < n$ for cases not needing all the partitions through n.

The first step computes membership matrix $\mathbf{U}_K$ by assigning each object to cluster k that minimizes $\|\mathbf{x}_i - \mathbf{m}_k\|^2$ where $\mathbf{x}_i'$ and $\mathbf{m}_k'$ are the i-th and k-th row of $\mathbf{X}$ and $\mathbf{M}_K$, respectively. This step also includes the constraint of non-emptiness for each cluster, thus ensuring that each column of $\mathbf{U}_K$ is not empty. The second step computes the centroid matrix $\mathbf{M}_K$ by the least-squares solution $\mathbf{M}_K = \mathbf{U}_K^+\mathbf{X}$. The third step enacts the agglomerative phase of the algorithm: starting from $\mathbf{U}_k$, the membership matrix $\mathbf{U}_{k-1}$ $(k = K, K-1, \ldots, 2)$ is obtained by merging two columns of $\mathbf{U}_k$ so that $I(\mathbf{U}_k, \mathbf{U}_{k-1})$ is minimal. Matrix $\mathbf{U}_{k-1}$ is thus obtained after finding the cluster pair incurring the minimum value of the increase $I(\mathbf{U}_k, \mathbf{U}_{k-1})$ among the $k(k-1)/2$ amalgamations of two columns in $\mathbf{U}_k$. This step further implies estimation of $\mathbf{M}_k$ and $\mathbf{M}_{k-1}$.

Finally, the fourth step enacts the divisive phase of the algorithm: starting from $\mathbf{U}_K$, the membership matrix $\mathbf{U}_{k+1}, (k = K, \ldots, n-1)$ is obtained by reassigning the objects belonging to one of the k columns of $\mathbf{U}_k$ to a new cluster pair so that this produces

the largest value of $I(\mathbf{U}_{k+1}, \mathbf{U}_k)$. The new matrix $\mathbf{U}_{k+1}$ is formed by $k-1$ columns of matrix $\mathbf{U}_k$ plus two new columns whose sum equals the remaining column of $\mathbf{U}_k$. Matrix $\mathbf{U}_{k+1}$ is the one yielding the split cluster pair with *maximal between-cluster* deviance among k columns of $\mathbf{U}_k$. This cluster split induces the partition in $k+1$ clusters with *minimum within-cluster* deviance. The split of the former cluster into two new ones can be achieved by applying k-means while fixing $k=2$ to only those objects that belong to each of the k clusters.

Algorithm 1: The HMC algorithm.

General initialization.$F^{\text{opt}} = \text{Inf.}$;
for $K = 2, \cdots, n-1$ **do**
  Initialization.Generate a random centroid $(K \times p)$matrix $\mathbf{M}_K$.
  Step 1. *Allocation*. Compute the membership matrix $\mathbf{U}_K$.
  Step 2. *Centroid computation*. Compute the centroid matrix $\mathbf{M}_K$.
  Step 3. *Agglomerative*. Get hierarchy from K to 2 clusters by recursively merging the clusters that give the minimum increase $I(\mathbf{U}_k, \mathbf{U}_{k-1})$.
  Step 4. *Divisive*. Get hierarchy from K to $n-1$ clusters by recursively splitting the clusters that give the maximum value of $I(\mathbf{U}_{k+1}, \mathbf{U}_k)$.
  Step 5. Compute $F^{(K)}$. If $F^{(K)} < F^{\text{opt}}$, then $F^{\text{opt}} = F^{(K)}$, $K^{opt} = \text{K}$ and $\mathbf{U}_k^{\text{opt}} = \mathbf{U}_k$ for $k = 2, \cdots, n-1$.
end

Note that the HMC algorithm does not ensure a halt at a globally optimal solution of the problem as each partitioning problem is known to be NP-hard (Křivánek and Morávek, 1986). To avoid local minima, we advise launching the algorithm from different random partitions into K clusters and retaining the best solution from the random starts. In our experiments, we set the number of random starts equal to 20.

The HMC algorithm features the following properties.

Property 1 *The HMC reduces to k-means when only Steps* 1 *and* 2 *are performed excluding Steps* 3 *and* 4.

Proof If Steps 3 and 4 are excluded, Algorithm 1 finds the partition in K clusters by alternating Steps 1 and 2 (the two steps of k-means) absent any agglomeration or division phases. The final solution is then a k-cluster partition identical to that obtained by k-means. □

Property 2 *The HMC algorithm for $K = n$ reduces to the well-known agglomerative hierarchical cluster analysis method of Ward* (1963). *Here, no division occurs in the algorithm, only a full agglomerative phase starting from n clusters to settle at one cluster. Each time, the two merged clusters minimally increase $I(\mathbf{U}_k, \mathbf{U}_{k-1})$. The minimum is found among the possible agglomerations of $k(k-1)/2$ yielding the least gain in the between-cluster deviance.*

Proof The initial cluster distance in Ward's is defined to be the squared Euclidean distances among objects q_{ij}, $(i, j = 1, \ldots, n)$ Murtagh and Legendre (2014). Distance q_{ij} is the within-cluster deviance where clusters C_{iK} and C_{jK} containing n_i and n_j objects are combined. If the two merged clusters $(C_{iK} \cup C_{jK})$ join some other cluster C_{lK} having n_l objects, then the rise in the deviance $q_{(ij)l}$ is given by the recursive formula of Lance and Williams (1967):

$$q_{(ij)l} = \frac{n_i + n_l}{n_i + n_j + n_l} q_{ik} + \frac{n_j + n_l}{n_i + n_j + n_l} q_{jk} - \frac{n_l}{n_i + n_j + n_l} q_{ij}.$$

In the agglomerative part of the HMC starting from $\mathbf{U}_n = \mathbf{I}_n$, the membership matrix $\mathbf{U}_{n-1}$ is obtained by merging two columns of $\mathbf{U}_n$ via addition so that the minimum increase of $I(\mathbf{U}_n, \mathbf{U}_{n-1})$ is achieved. The matrix $\mathbf{U}_{n-1}$ is obtained by searching among the $n(n-1)/2$ amalgamations of two columns in $\mathbf{U}_n$ for the one producing the minimum of $I(\mathbf{U}_n, \mathbf{U}_{n-1})$, and thus the minimum of F. Since the within-cluster deviance of the merged clusters is q_{ij}, then the minimum of F is achieved by considering the minimum in $\mathbf{Q} = [q_{ij}]$ outside the main diagonal. The algorithm continues compounding clusters with minimum within-cluster deviance until all clusters are amalgamated. □

Property 3 *The HMC algorithm for $K = 1$ reduces to the Bisecting K-means result. As such, the agglomerative part of the algorithm is disabled while each step sequentially applies the divisive part of the HMC featuring the k-means algorithm path to the cluster pair that most decreases function F.*

Proof Bisecting k-means starts with all data as a single cluster, then splitting it in two using k-means with $k = 2$. Next, the cluster that leads to the maximum decrease of within-cluster deviance, once split, is detected and divided in two clusters using k-means where $k = 2$. This operation is repeated until n singletons are found. The HMC algorithm for $K = 1$ thus echoes Bisecting K-means since the cluster yielding the maximum decrease of within-cluster deviance is the one that most decreases function F or, alternatively, the one that produces the largest value of $I(\mathbf{U}_{k+1}, \mathbf{U}_k)$. □

It is interesting to examine the computational complexity of HMC reported in Table 1. The run time of the iterative relocation Steps 1 and 2 of the Algorithm 1 — the HMC algorithm — is $O(nKpt)$, similar to Lloyd's algorithm (Lloyd, 1982) where t is the number of iterations needed for convergence. Generally, t is assumed fixed at 10, and the time complexity is thus linear in n. For agglomerative Step 3 of the HMC algorithm, the total computational complexity requires $\sum_{k=2}^{K-1} k(k-1)/2 = O(K^3)$ agglomerations and loss function evaluations for each K. Yet, complexity can be reduced to $O(K^2 \log K)$ using a heap. For divisive Step 4 of the HMC algorithm, the total complexity requires $n - K$ splits, each defined by K-means with $K = 2$ clusters, making the total complexity for the $n - K$ splits $O(n^2)$. However, this complexity can be avoided using a parsimonious tree, or curtailed after a given K, say equal to m, when clusters in the hierarchy begin looking smeared and indistinct in the dendrogram, especially for large n.

Table 1 The computational complexity for each step in the HMC algorithm for a specific K

Step	Complexity
1-2	$O(n)$
2	$O(K^2 \log K)$
3	$O(n^2)$
Total	$O(n^2)$

4.2 A Constrained Algorithm for HMC

In general, there is no reason why the optimal least-squares estimation of the model in (4) should include the k-means solution. However, since k-means is one of the most used clustering algorithms, the idea to link k-means to the hierarchical classification is attractive. Therefore, we define a new constrained algorithm for HMC (Algorithm 2). If we replace the first two steps of Algorithm 1 with the obtained k-means solution (i.e., the partition of n objects into K clusters obtained by k-means at convergence with the corresponding centroids in $\mathbf{M}_K$), we actually force the solution of HMC to include the best k-means solution for each given K. In Algorithm 2, Step 1 is performed as k-means++ Arthur and Vassilvitskii (2007). Step 2 is equivalent to Step 3 of Algorithm 2 and invokes Ward's starting from the solution for K clusters obtained in the first step — dubbed as Partial Ward's method. Finally, Step 3 corresponds to Step 4 of Algorithm 2, also known as Bisecting k-means.

Notably, this constrained algorithm also allows applying HMC to reasonably large datasets, since k-means is known to be a fast algorithm and avoids computing the entire hierarchy at every iteration. Algorithm 2 thus demands less computation than Algorithm 1 since the latter builds a hierarchy (Steps 3 and 4) for *each new allocation* of the observations in K clusters (Steps 1 and 2), while Algorithm 2 builds *only one hierarchy* upon the K-means solution in K clusters. Moreover, if only the agglomerative part of HMC needs to be applied to evaluate the most relevant nested relations between clusters, it is possible to omit the divisive step. Since K is generally not so large, the parsimonious tree proves computationally swift.

Algorithm 2: A constrained algorithm for HMC.

Initialization.$F^{\text{opt}} = \text{Inf.}$;
for $K = 2, \cdots, n-1$ **do**
  Step 1. $K-$*means step*.Compute the membership matrix $\mathbf{U}_K$ and the centroid matrix $\mathbf{M}_K$.
  Step 2. *Agglomerative*.Get hierarchy from K to 2 clusters by partial Ward$'$s method.
  Step 3. *Divisive*.Get hierarchy from K to $n-1$ clusters by Bisecting $K-$means.
  Step 4. Compute $F^{(K)}$. If $F^{(K)} < F^{\text{opt}}$, then $F^{\text{opt}} = F^{(K)}$, $K^{opt} = \text{K}$ and $\mathbf{U}_k^{\text{opt}} = \mathbf{U}_k$ for $k = 2, \cdots, n-1$.
end

Example 1 An example may help grasp Algorithm (2). For a small dataset formed by the 7 objects and 2 variables (Gordon, 1999), the membership matrix represents the best K-means partition for $K = 4$, that is, clusters $C_1^4 = \{1\}$, $C_2^4 = \{2\}$, $C_3^4 = \{3, 6, 7\}$, $C_4^4 = \{4, 5\}$ and the corresponding centroids:

$$\mathbf{X} = \begin{bmatrix} 9 & 33 \\ 18 & 7 \\ 24 & 23 \\ 25 & 40 \\ 32 & 47 \\ 34 & 30 \\ 40 & 16 \end{bmatrix}, \mathbf{U}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \mathbf{M}_4 = \begin{bmatrix} 9.00 & 33.00 \\ 18.00 & 7.00 \\ 32.67 & 23.00 \\ 28.50 & 43.50 \end{bmatrix}.$$

The algorithm next computes the rise in deviance (i.e., sum of squares) when two clusters are amalgamated using function (11)

$$\begin{array}{c} \\ C_2^4 \\ C_3^4 \\ C_4^4 \end{array} \begin{array}{c} \begin{array}{ccc} C_1^4 & C_2^4 & C_3^4 \end{array} \\ \begin{bmatrix} 757.0 & 990.2 & 654.0 \\ & 706.7 & 1923.3 \\ & & 1050.3 \end{bmatrix} \end{array}.$$

This is the dissimilarity matrix defined in part III of Table 4.4 (page 93) of Gordon (1999). In fact, the k-means partition in 4 clusters is exactly the partition in 4 clusters obtained by Ward's method. Thus, the amalgamation sequence corresponds to that obtained by Ward's. The trees obtained by HMC are displayed in Fig. 2.

5 Real Data Examples

To see how HMC performs, we enlist six oft-used empirical datasets known to be composed by K^* clusters. We compare the complete HMC (for K from 2 to 30) to Ward's method, Bisecting k-means, and three model-based hierarchical agglomerative clustering methods, that is, 1) Model VII, where $\boldsymbol{\Sigma}_k$ is diagonal but may vary among clusters (i.e., $\boldsymbol{\Sigma}_k = \lambda_k \mathbf{I}_p$), 2) Model EEE, where $\boldsymbol{\Sigma}_k$ is fixed across all clusters without structural constraints (i.e., $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$), and 3) Model VVV, where $\boldsymbol{\Sigma}_k$ is estimated per cluster separately. Model EII, where covariance matrix $\boldsymbol{\Sigma}_k$ for segment k is constrained as diagonal and fixed across all clusters k (i.e., $\boldsymbol{\Sigma}_k = \lambda \mathbf{I}_p$), is excluded as fully reducing to Ward's Fraley (1998).

The variables of all datasets are standardized to z-score. Performance for the six methods is evaluated both by detecting the proposed partition in K^* clusters and in terms of overall hierarchical clustering solution. To rank the partitions obtained by the six methods for K^*, the Adjusted Rand Index (ARI, Hubert and Arabie, 1985) is computed between each and the one proposed in the dataset. Note that ARI is the corrected-for-chance version of the Rand index Rand (1971) having zero as the expected value where the identified partition is random, and capped by 1 for perfect agreement with the given partitions. To assess the similarity in hierarchies obtained by HMC, Ward's method, Bisecting k-means and three model-based hierarchical agglomerative clustering methods, we computed the cophenetic correlation (r_{coph}) between the trees proposed by Sokal and Rohlf (1962) and the graphical representations proposed in the dentexend package Galili (2015) for R R Core Team (2021).

In detail, we consider the Wine dataset — available in the UCI Machine Learning data repository and as part of the gclus package Hurley (2004) for R — that results from a chemical analysis of Italian wines grown in Piedmont. 13 attribute measurements for 178 observations represent distinct constituents found in 3 types of wine (Barolo, Grignolino and Barbera). The attributes are as follows: Alcohol, Malic acid, Ash, Alkalinity of ash, Magnesium, Total phenols, Flavonoids, Non-Flavonoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, and Proline. Second is the Archaeology dataset for the chemical composition of glass belonging to the Roman era found in Norway (Baxter, 1994; Everitt et al., 2011). Although comprising 19 observations of 8 variables *not* composed by distinct groups, Archeology is still used to illustrate clustering techniques, and Baxter (1994) found 2 main clusters. The third dataset is the Coffee data Der heutige stand der kaffeechemie (Streuli 1973) featuring 43 samples divided in 2 varieties, namely Arabica and Robusta. Twelve chemical constituents of coffee are the features measured: Water, pH value, Fat, Chlorogenic acid, Bean weight, Free acid, Caffeine, Neochlorogenic acid, Extract yield, Mineral content, Trigonelline, and Isochlorogenic acid. The

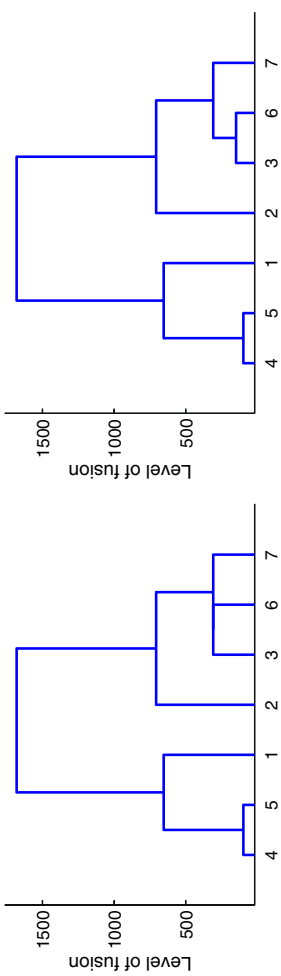


Fig. 2 Trees obtained by HMC of the dataset given in Gordon (1999) for $K = 4$. It can be observed that the parsimonious dendrogram (a), starting from $K = 4$, corresponds to the one given by the Ward's method

fourth dataset is the Ruspini dataset Ruspini (1970), a data frame with 75 observations on 2 variables specifying coordinates of the points divided in 4 clusters. Fifth is the Thyroid dataset Coomans et al. (1983) — available as part of the R package mclust (Scrucca et al., 2016) — sampling 215 observations divided in 3 clusters on 5 variables from laboratory tests diagnosing thyroid gland patients. Finally, we consider the Seeds data containing 210 observations of 7 measured geometrical properties of kernels belonging to 3 distinct varieties of wheat. This dataset is commonly available in the UCI Machine Learning data repository.

HMC solves a hierarchical cluster problem by optimizing an objective function and is thus expected to outrun both Ward's method and Bisecting k-means, overcoming a major limitation in hierarchical methods for clustering analysis. Indeed, the loss function F for HMC always outperforms Ward's method, Bisecting k-means, EEE, VII and VVV. Results listed in Tables 2 and 3 confirm HMC as always outperforming all rivals, both in detecting the "true" partition of the data, and as the overall hierarchical clustering solution (i.e., by evaluation of the objective function F). The latter result confirms Ward's method and the three aggregative model-based hierarchical models as sub-optimal due to their misclassifications in the lowest part of the hierarchy that notably spoil the final hierarchy. Results further show that a fully divisive approach, such as Bisecting k-means, is sub-optimal in finding the overall hierarchical clustering solution. In the next section, our treatment of the Wine dataset is presented as a walk-through. The graphical representations for the results for the other datasets are provided in the Electronic Supplementary Material.

5.1 Wine Dataset

To exemplify the results in Tables 2 and 3, our treatment of Wine is detailed next. The best HMC solution in terms of the objective function is found for $K = 17$ scoring 46678.5, a value smaller than those computed for the solution of Ward's method ($F = 46843.3$), Bisecting k-means ($F = 48272.7$), EEE ($F = 85911.97$), VII ($F = 66195.97$) and VVV ($F = 85149.12$). This means that, apart from Ward's method, EEE, VII and VVV, the objective function for the entire hierarchy secured by HMC appears untainted by misclassifications in its sublevels — the same deduction holding for the sub-optimal solution found by Bisecting k-means. Outperformance by HMC stands tall in detecting a partition of three clusters. Indeed,

Table 2 Summary of the applications' results for HMC

Dataset	n	K^*	K	ARI	F
Wine	178	3	17	.87	46678.5
Archaeology	19	2	5	1.00	548.6
Coffee	43	2	8	1.00	3947.4
Ruspini	75	4	30	1.00	337.4
Thyroid	215	3	25	.62	7858.3
Seeds	210	3	12	.82	10956.7

K^* represents the number of clusters provided by the dataset and the number of clusters for which the Adjusted Rand Index (ARI) is evaluated, while K is the number of clusters found under the best solution for HMC. Column F lists the value of the objective function (13) computed for each method

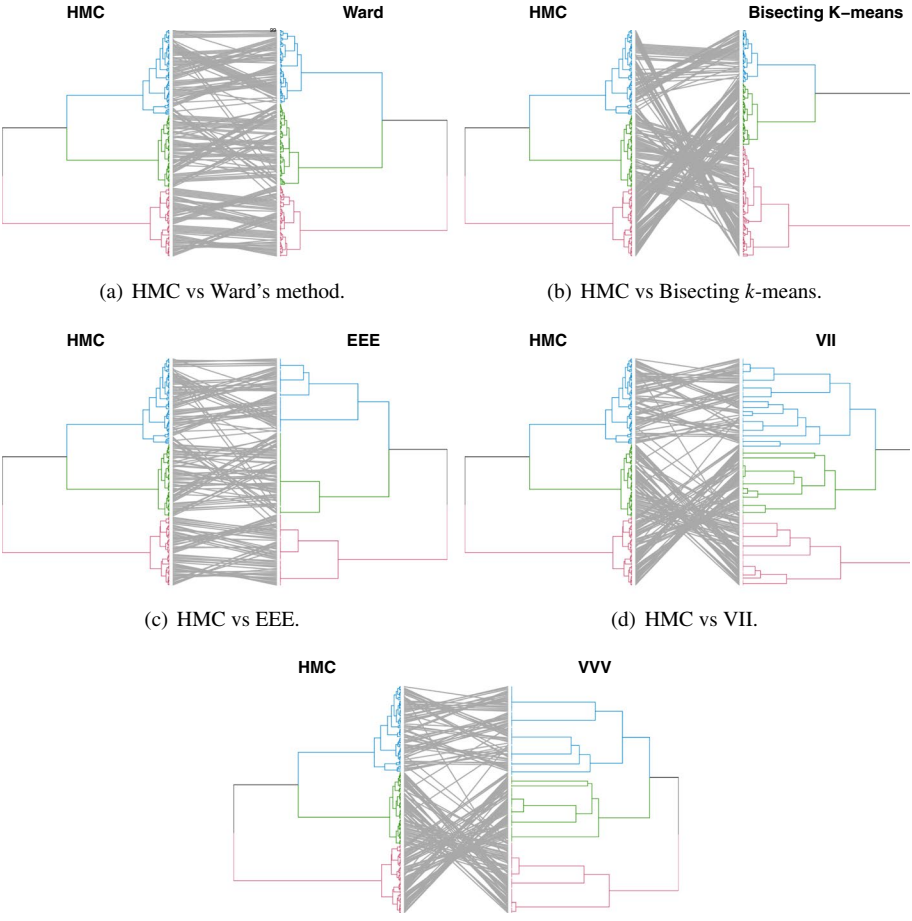
Table 3 Comparison of the HMC clustering results for the applications with the corresponding results for Ward's method, Bisecting k-means, EEE, VII and VVV

	HMC	Ward's	Bisecting k-means	EEE	VII	VVV
Dataset	ARI					
Wine	.87	.79	.59	.79	.85	.85
Archaeology	1.00	.79	1.00	.79	.79	.79
Coffee	1.00	1.00	1.00	1.00	1.00	1.00
Ruspini	1.00	1.00	1.00	1.00	1.00	1.00
Thyroid	.62	.59	.60	.59	.47	.47
Seeds	.82	.80	.74	.80	.61	.61
	F					
Wine	46678.5	46843.3	48272.7	85912.0	66196.0	85149.1
Archaeology	548.6	550.8	551.6	931.3	720.9	757.9
Coffee	3947.4	4015.4	4021.5	5020.9	5010.8	5900.4
Ruspini	337.4	337.5	338.0	365.1	515.1	472.8
Thyroid	7858.3	7890.6	7921.1	12276.6	12213.0	12418.0
Seeds	10956.7	10985.4	11299.2	28075.7	14518.3	22727.2
	r_{coph}					
Wine	1.00	.92	.58	.89	.65	.65
Archaeology	1.00	.79	.99	.79	.75	.79
Coffee	1.00	.97	.99	.78	.92	.90
Ruspini	1.00	1.00	.99	.95	.74	.99
Thyroid	1.00	.96	.97	.89	.53	.54
Seeds	1.00	.85	.86	.79	.74	.70

The Adjusted Rand Index (ARI) is evaluated for number of clusters provided by the dataset (Table 2). F represents the value of the objective function (13) computed for each method. The cophenetic correlation (r_{coph}) with the tree obtained by HMC is reported for each model

ARI between the partition of HMC(17) with 3 clusters and the partition proposed in the Wine dataset is $\text{ARI}(\mathbf{U}_{\text{HMC}}, \mathbf{U}_{\text{Wine}}) = 0.87$. Remaining indices compute as: $\text{ARI}(\mathbf{U}_{\text{Ward's}}, \mathbf{U}_{\text{Wine}}) = 0.79$, $\text{ARI}(\mathbf{U}_{\text{bKm}}, \mathbf{U}_{\text{Wine}}) = 0.59$, $\text{ARI}(\mathbf{U}_{\text{EEE}}, \mathbf{U}_{\text{Wine}}) = 0.79$, $\text{ARI}(\mathbf{U}_{\text{VII}}, \mathbf{U}_{\text{Wine}}) = 0.85$ and $\text{ARI}(\mathbf{U}_{\text{VVV}}, \mathbf{U}_{\text{Wine}}) = 0.85$. These results evidently show that HMC outperforms rival methods in better detecting the partitioning in three clusters for the Wine data set.

Figure 3 graphically compares the resulting dendrograms. The dendrograms obtained by HMC and Ward's (Fig. 3(a)) are far more similar — cophenetic correlation between trees $r_{\text{coph}} = 0.916$ — than those for HMC and Bisecting k-means (Fig. 3(b), $r_{\text{coph}} = 0.576$). Among the model-based hierarchical agglomerative clustering methods, the one producing spherical clusters (i.e., EEE) is the most similar to HMC ($r_{\text{coph}} = 0.89$). These results clearly signal that observations are classified by HMC distinctly from the other five methods when looking at partitions assigned in 3 clusters (i.e., in three different colors, respectively). The crossed gray lines connecting identical objects in each compared pair of dendrograms unveil a different ordering and classification that allows HMC to improve the fit. In addition, HMC shows very distinct, small linkages corresponding to within-clusters objects versus large linkages between clusters. Such contrast is less evident, especially for VII and VVV.



(a) HMC vs Ward's method.

(b) HMC vs Bisecting k-means.

(c) HMC vs EEE.

(d) HMC vs VII.

(e) HMC vs VVV.

Fig. 3 Wine data: comparison between dendrograms

5.2 Codon Application: Parsimonious Tree

HMC is applied to the genetic dataset available on the UCI Machine Learning Repository at archive.ics.uci.edu/ml/datasets/Codon+usage and recently used by Khomtchouk (2020). This HMC application shows the potential of parsimonious dendrograms tackling reasonably large datasets, particularly when the complete sets of partitions and clusters go unconsidered by users, even leading to a poor interpretation of the main clusters. We applied the prepossessing steps from Khomtchouk (2020) toward a dataset of 12964 observations (i.e., organisms) across 64 variables (i.e., codons) with variables standardized to z-score and truncated to reside within the interval $[-2, 2]$.

For both observations (rows) and columns (variables), separate solutions were computed for the complete HMC with K from 2 to 30. The best K in terms of objective function was determined equal to 28 for the rows and 9 for the columns. We visually compare our parsimonious solutions (i.e., starting from 28–9 clusters) with dendrograms from the

complete linkage clustering as proposed by Khomtchouk (2020). Here, Figs. 4 and 5 each display two dendrograms, one above for the variables (columns) and one leftward for the observations (rows), as well as the heatmap of the data matrix. Dendrograms in Fig. 4 are parsimonious, thereby improving the visual representation and interpretation of results. In that figure, we can easily distinguish 6 homogeneous clusters each of observations and variables in color. We used the same coloring for the complete linkage dendrograms (albeit

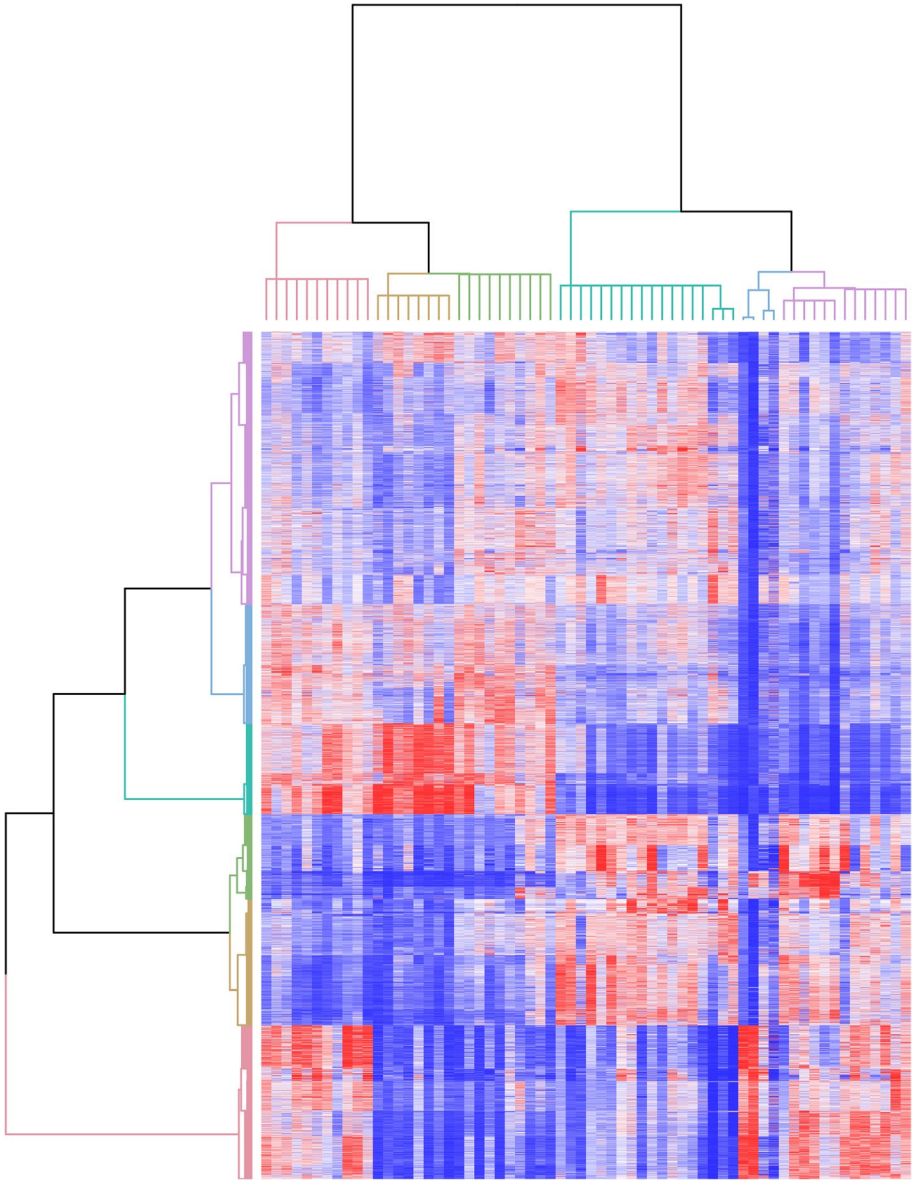


Fig. 4 Heatmap and dendrograms of HMC on rows and columns of the codon data. Values are displayed as colors ranging from blue (−2) to red (2). Coloring is used in the dendrograms for the first 6 clusters

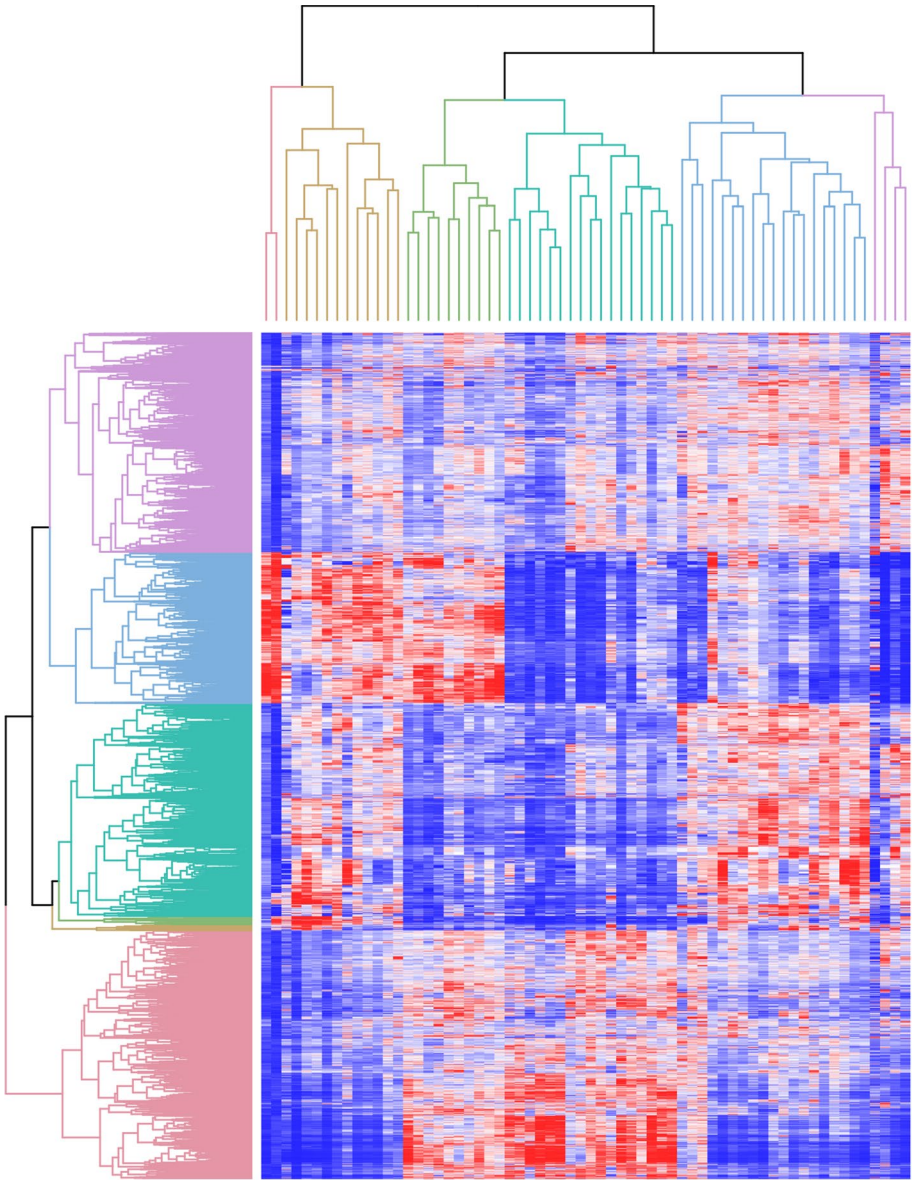


Fig. 5 Heatmap and dendrograms of complete linkage method on rows and columns of the codon data. Values are displayed as colors ranging from blue (−2) to red (2). The partitioning of the first six clusters is colored

with some uncertainty). Stubby linkages represent *within-cluster* distances that signal homogeneity. These are clearly distinguishable from the long, spacious linkages representing high distances (isolation) between clusters.

These parsimonious dendrograms obtained by HMC allow enhanced visibility of main clusters tied by clear, large inter-cluster linkages and woven by small intra-cluster linkages.

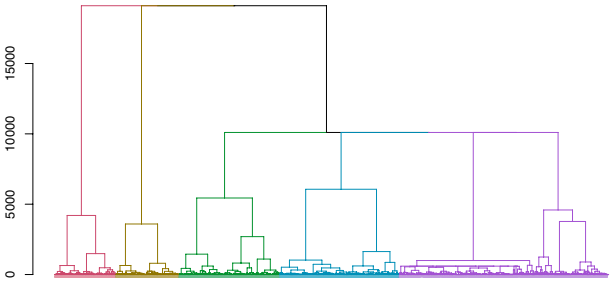


Fig. 6 HMC dendrogram of constrained HMC on the rows of the codon data using the natural partition given by the variable kingdom as the start. The coloring is based on the five categories of kingdom

The complete linkage method, however, obscures the difference between inter-cluster and intra-cluster linkages in its unclear solution. In other words, the dendrograms obtained by the complete linkage method are more difficult to cut in order to find the desired partition since this clutters the detection of cluster homogeneity and isolation. The complete linkage dendrogram in Fig. 5 highlights the much more confused situation versus the HMC dendrogram in Fig. 4. Similar conclusions may apply for the variables where the situation is however slightly better.

The dataset used in Khomtchouk (2020) also includes the five-category domain class (Archea, Bacteria, Eukaryota, Bacteriophage and Virus) that biologically corresponds to the highest taxonomic rank per Luketa (2012), but with Prionobiota replacing Bacteriophage. This category variable is called *kingdom* and is here used to natural partition around which to construct the optimal hierarchy. This represents a quite common situation in numerous fields (e.g., genetics) where a natural partition exists, and the researcher wishes to explore the hierarchical relations around it.

Thus, we apply HMC, constrained to start from the natural partition given by the variable *kingdom*, and we visually trace the entire dendrogram in Fig. 6. By cutting the dendrogram near the 10, 000 level, a crisp partition into 6 clusters emerges where the domain Eukaryota splits in two clusters: Diphoda and Amorphea. Cutting the dendrogram at the 15, 000 mark, the three-domain classification traces the partition formed by Archea, Bacteriophage, and a larger cluster that includes Eukaryota, Bacteria and Virus. An earlier three-domain classification of the living world was proposed by Woese et al. (1990). HMC thus shows remarkable flexibility to include one theoretical partition — here, a 5 class kingdom of life and its observed dendrogram — as well as derived information, such as the partitions for 3 and 6 clusters compatible with the original taxonomy in 5 classes.

6 Conclusion

In this paper, we proposed a new method for hierarchical cluster analysis modeling that optimizes a least-square loss function tracking the within-cluster deviance of nested partitions. The methodology named Hierarchical Means Clustering (HMC) overcomes notable weaknesses in popular agglomerative or divisive clustering algorithms. Agglomerative methods are greedy heuristic algorithms that form hierarchies by recursively merging the two nearest clusters per a prefixed isolation (dissimilarity) measure between clusters. Divisive treatments recursively split the most heterogeneous cluster according to some criterion of heterogeneity. The typology of these algorithms forbids the undoing of prior amalgamations or divisions, and classification errors made in the recursive (frequently first) steps thus remain uncorrectable. Furthermore, classical approaches do not enlist an explicit loss

function-based clustering method for the dissimilarity data and do not directly optimize an objective function for detecting the best hierarchy. Finally, researchers have frequently observed that the complete dendrogram, with all n nested partitions, is rarely used. Yet, a parsimonious dendrogram with a reduced set of internal nodes is preferable since it includes a limited number of nested partitions easier to interpret.

The new method HMC proposed here aims to address the limitations of these classical hierarchical algorithms. HMC operates directly on the (objects by variables) data matrix, thus averting the use, computation and storage of a dissimilarity matrix. Yet, it remains possible to implement HMC on the matrix of squared Euclidean distances that pursues a dissimilarity-based approach. HMC is founded on the system of n centroid-based statistical models which is defined by n equations representing a set of nested partitions. Its least-squares estimation yields a loss function that tracks the total within-cluster deviance — a measure that can be minimized to best identify the hierarchy formed by nested clusters with lowest total deviance. This loss function-based clustering method ensures "goodness of fit" assessment of the hierarchical model to the observed data.

An aptly parsimonious version of HMC can be obtained that favors the interpretation of a reduced number of partitions under far less computational complexity. HMC minimizes the total within-cluster deviance of $n-1$ nested partitions from 1 to n clusters. Here, the objective function can be seen as the sum of the minimal incremental deviances in the $n-1$ partitions. An efficient, three-stage algorithm has been proposed featuring an iterative relocation, plus an agglomerative and a divisive phase. The computation complexity of HMC includes $O(n)$ for the relative relocation part, $O(K^2 \log K)$ for the agglomerative step, and $O(n^2)$ for the divisive stage. HMC is certainly an evolution of Ward's method and Bisecting k-means since these two techniques are special cases of HMC for $K = n$ or 1, respectively. Therefore, the solution of HMC can never fare worse than under Ward's method or Bisecting k-means.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00357-022-09419-7.

Data Availability The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of Interest The authors declare no competing interests.

References

Arthur, D., & Vassilvitskii, S. (2007). K-Means++: The Advantages of Careful Seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. SODA '07. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, pp. 1027-1035. ISBN: 9780898716245

Banfield, J. D., & Raftery, A. E. (1993). Model-based gaussian and non-gaussian clustering. *Biometrics, 49*, 803–821.

Baxter, M. J. (1994). *Exploratory multivariate analysis in archaeology*. Edinburgh: Edinburgh University Press. . ISBN: 0748604235

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

Coomans, D. et al. (1983). Comparison of multivariate discrimination techniques for clinical data-application to the thyroid functional state. In: Methods of information in medicine 22.2, pp. 93-101.

Cormack, R. M. (1971). A Review of Classification. In: Journal of the Royal Statistical Society. Series A (General) 134.3, pp. 321-367.

Cormen, T. H., Leiserson, C. E., & Rivest R. L. (1990). Introduction to Algorithms. 1st. Cambridge, MA: The MIT Press.

Everitt, B. et al. (2011). Cluster analysis. 5th. Wiley. ISBN: 978-0-470-74991-3.

Fraley, C. (1998). Algorithms for model-based Gaussian hierarchical clustering. *SIAM Journal on Scientific Computing, 20*, 270–281.

Galili, T. (2015). dendextend: an R package for visualizing, adjusting, and comparing trees of hierarchical clustering. In: Bioinformatics. https://doi.org/10.1093/bioinformatics/btv428

Gordon, A. D. (1981). Classification. 1st ed. Chapman and Hall/CRC.

Gordon, A. D. (1999). Classification. 2nd ed. Chapman and Hall/CRC.

Hartigan, J. A. (1967). Representation of similarity matrices by trees. In: Journal of the American Statistical Association 62.320, pp. 1140-1158.

Hartigan, J. A. (1975). *Clustering algorithms*. New York: Wiley.

Hubert, L., Arabie, P. (1985). Comparing partitions. In: Journal of Classification 2.1, pp. 193-218.

Hurley, C. B. (2004). Clustering visualizations of multidimensional data. In: Journal of Computational and Graphical Statistics 13.4, pp. 788-806.

Khomtchouk, B. B. (2020). Codon usage bias levels predict taxonomic identity and genetic composition. In: bioRxiv. https://doi.org/10.1101/2020.10.26.356295.

Křivánek, M. & Morávek, J. (1986). NP-hard problems in hierarchical-tree clustering. In: Acta Informatica 23.3, pp. 311-323.

Lance, G. N., & Williams, W. T. (1967). A general theory of classificatory sorting strategies: 1. *Hierarchical Systems. In: The Computer Journal, 9*(4), 373–380.

Lloyd, S. (1982). Least squares quantization in PCM. In: IEEE Transactions on Information Theory 28.2, pp. 129-137.

Luketa, S. (2012). New views on the megaclassification of life. In: Protistology 7.2, pp. 218-237.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics. Berkeley, Calif.: University of California Press, pp. 281-297.

Müllner, D. (2011). Modern hierarchical, agglomerative clustering algorithms. arXiv. https://doi.org/10.48550/ARXIV.1109.2378

Murtagh, F., & Legendre, P. (2014). Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion? *Journal of Classification, 31*, 274–295.

R Core Team (2021). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. https://www.R-project.org/

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. In: Journal of the American Statistical Association 66.336, pp. 846-850.

Rubin, J. (1967). Optimal classification into groups: An approach for solving the taxonomy problem. In: Journal of Theoretical Biology 15.1, pp. 103-144.

Ruspini, E. H. (1970). Numerical methods for fuzzy clustering. In: Inf. Sci. 2.3, pp. 319-350. ISSN: 0020-0255.

Scrucca, L. et al. (2016). mclust 5: clustering, classification and density estimation using gaussian finite mixture models. In: The R Journal 8.1, pp. 289-317.

Sokal, R. R., & Rohlf, F. J. (1962). The comparison of dendrograms by objective methods. *Taxon, 11*, 33–40.

Sriram, N. (1990). Clique optimization: A method to construct parsimonious ultrametric trees from similarity data. In: Journal of Classification 7.1, pp. 33-52.

Steinbach, M., Karypis, G., & Kumar V. (2000). A comparison of document clustering techniques. In: In KDD Workshop on Text Mining.

Streuli, H. (1973). Der heutige Stand der Kaffee-Chemie. In *Association Scientifique International du Cafe, 6th International Colloquium on Coffee Chemistry, Bogata, Colombia* (pp. 61–72).
Vichi, M. (2008). Fitting semiparametric clustering models to dissimilarity data. In: Advances in Data Analysis and Classification 2.2, pp. 121-161.
Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. In: Journal of the American Statistical Association 58.301, pp. 236-244.
Woese, C. R., Kandler, O., & Wheelis, M. L. (1990). Towards a natural system of organisms: proposal for the domains Archaea, Bacteria, and Eucarya. In: Proceedings of the National Academy of Sciences 87.12, pp. 4576-4579.
Zangwill, W. I. (1969). *Nonlinear programming: a unified approach*. Englewood Cliffs: Prentice-Hall.