A Novel Approach to the Problem of Non-uniqueness of the Solution in Hierarchical Clustering

Isabella Cattinelli, Giorgio Valentini, Eraldo Paulesu, and N. Alberto Borghese, *Member, IEEE*

***Abstract*—The existence of multiple solutions in clustering, and in hierarchical clustering in particular, is often ignored in practical applications. However, this is a non-trivial problem, as different data orderings can result in different interpretations of the same data. The method presented here offers a solution to this issue. It is based on the definition of an equivalence relation over dendrograms that allows developing all and only the significantly different dendrograms for the same dataset, thus reducing to polynomial the computational complexity from the exponential one obtained when all possible dendrograms are considered. Experimental results in the neuroimaging and bioinformatics domains show the effectiveness of the proposed method.**

***Index Terms*—Hierarchical clustering, dendrogram equivalence relation, bioinformatics, neuroimaging**

I. Introduction

DISCOVERING similarities in the real world is a fundamental task for both humans and machines, as it allows, for instance, reasoning by categories [1]. This task can be carried out by *clustering*, that groups elements into subsets, called clusters, according to some homogeneity measure, so that objects inside a cluster are more similar among them, and more dissimilar to objects belonging to other clusters [2]–[4].

Although other clustering algorithm families have been proposed (e.g. spectral [5]), two main families are usually identified: hierarchical and partitional clustering. In hierarchical clustering (HC), a progressive partitioning of the data elements is achieved by iterative operations (either merging or splitting) on the data set aimed at grouping pairs of elements that are closest according to a given similarity measure. In partitional clustering, instead, a set of prototypes is positioned and moved inside the data space to obtain the best representation of the input data according to a specified cost function.

The algorithms of both families suffer from several issues, most notably the optimal value of the cost function is rarely reached. In partitional clustering, local minima cannot be easily escaped. Mechanisms proposed in the literature include convex relaxations [6], the costly stochastic optimization [7], and careful initialization of cluster prototypes [8], [9], but they do not provide a general solution to this problem. Similarly, in (agglomerative) HC, although each processing step can be locally optimal since the pair of elements to be merged is chosen so as to minimize a dissimilarity function, the global optimality of the clustering solution cannot be guaranteed [10, p. 330]. Although statistical methods can be employed for both obtaining and validating clusters from data whose distribution is known, or can be reasonably assumed, in the most general case (i.e. no assumptions can be safely made) there is still lack of clear theoretical foundations for clustering [11].

A reasonable requirement for clustering is that the returned solution is unique. However, in HC multiple solutions can be returned for the same dataset, depending on the input data order [12], [13]. This problem *"certainly is not widely known"* [14] and it is usually disregarded. Thus, actual conclusions drawn from clustering may be only the result of a particular presentation order of the input data. The large data sets avalable today lead to a possible explosion of the number of clustering solutions, as the authors themselves experienced when working on real-world datasets (like the BioGRID one [15], and the neuroimaging activation data [20]; see Results section). This provides a well-grounded practical motivation for the present research.

Few attempts have been made to solve this problem. In [14] it is suggested to run the clustering process on different permutations of the input data and to choose the solution that minimizes the defined cost function. However, it cannot be guaranteed that a different data permutation would not produce an even better solution. On the other side, an exhaustive generation and exploration of all alternative solutions associated with a dataset is computationally infeasible. A solution based on simultaneously merging all those clusters sharing the same minimal distance into one "supercluster" has been advanced [16]. However, this choice may not produce the same clusters that would be obtained merging two clusters at a time.

We propose here a novel approach that addresses directly this problem. It is based on generating only the subset of *significantly different* solutions, thus keeping the computational load relatively low but, at the same time, ensuring that no interesting solution is missed. The algorithm is based on an equivalence definition of HC solutions associated with a dataset. The method has been extensively applied to the analysis of both bioinformatics and functional neuroimaging data achieving a dramatic reduction in the number of solutions generated (Sect. IV), demonstrating the relevance and practical utility of the approach.

I. Cattinelli is with the Department of Computer Science, Università degli Studi di Milano, Milan, 20135, Italy, and with the Department of Psychology, Università degli Studi di Milano-Bicocca, 20126 Italy.

G. Valentini and N. A. Borghese are with the Department of Computer Science, Università degli Studi di Milano, Milan, 20135, Italy.

E. Paulesu is with the Department of Psychology, Università degli Studi di Milano-Bicocca, 20126, Italy.

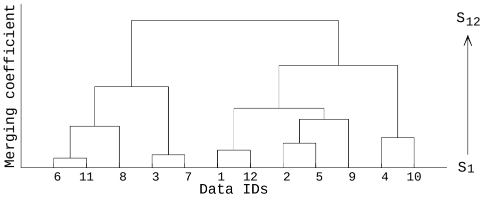


Fig. 1. At start, each input element (IDs on the x-axis) is assigned to a singleton cluster (partition S_1). At each step, the two closest clusters are merged, decreasing the number of clusters by one. At the last step, only one large cluster is obtained (partition S_{12}). The sequence of merging steps is represented in a tree structure, called a dendrogram. The height of the horizontal segments representing merging steps is the dissimilarity value of the two clusters being merged. The dendrogram is then cut at the desired level to get the final clustering solution.

II. Hierarchical Clustering

Let us assume that $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_j}, \ldots, \mathbf{x_N}\}$, with $\mathbf{x_j} = (x_j^1, x_j^2, \ldots, x_j^D)$ is a dataset of N elements that belong to the multi-dimensional space $\mathbb{R}^D$. The $\mathbf{x_j}$ can be seen as the set of features of a given pattern, or, equivalently, as the position of a point inside $\mathbb{R}^D$.

Suppose also that a dissimilarity function, $d : \mathbf{X} \times \mathbf{X} \rightarrow \mathbb{R}$, is defined over this space such that for every $\mathbf{x}, \mathbf{y} \in \mathbf{X}$:

$$d(\mathbf{x}, \mathbf{y}) \geq 0, \text{ and } d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x}). \tag{1}$$

If reflexivity ($d(\mathbf{x}, \mathbf{y}) = 0$ iff $\mathbf{x} = \mathbf{y}$) and triangle inequality ($d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})\ \forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{X}$) also hold, then d is a metric. $d(.)$ defines the degree of similarity between pairs of input elements, and different functions, with different properties, have been proposed to implement it (for a review, see [2], [10]). For instance, l_1 metric (city-block distance) limits the impact of outliers, while l_∞ metric attributes to outliers a very high weight.

Our goal is to partition $\mathbf{X}$ into M sets, called clusters, $S = \{C_1, C_2, \ldots, C_k, \ldots, C_M\}$, such that each of them is composed of elements that are closer to each other and farther from elements in any other cluster according to the chosen dissimilarity measure.

Agglomerative HC partitions the data as follows. At start, each element is assigned to a different cluster (partition S_1). At each step, two clusters are merged, and a new data partition is generated. The procedure is repeated iteratively until a partition containing a single cluster is obtained (S_N). The result is a hierarchy of nested clustering solutions (i.e., partitions of the data), $T = \{S_1, S_2, \ldots, S_N\}$, where S_m is the clustering solution obtained after m steps and it is constituted of $N - m + 1$ clusters. The hierarchy of partitions can be represented in a tree-like structure, called dendrogram (Fig. 1). The final clusters are obtained by cutting the dendrogram at the proper level according to the given criterion, like, for instance, the number of desired clusters or the average intra-cluster variance. The cut can be performed climbing up the dendrogram, starting from the leaves, and stopping just before the figure of merit exceeds the defined threshold. Different HC algorithms have been proposed, each employing a different dissimilarity measure between clusters; among these, single linkage, complete linkage, (weighted) group average linkage, centroid linkage, and Ward's method are the most popular ones [2]. For instance, single linkage defines the dissimilarity between two clusters C_i and C_j as the minimum dissimilarity between pairs of elements $\mathbf{x} \in C_i, \mathbf{y} \in C_j$; that is, $D(C_i, C_j) = \min_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} d(\mathbf{x}, \mathbf{y})$. Notice that we have two notions of dissimilarity measure: one defined over single input elements ($d(.)$), and one defined over clusters ($D(.)$). The values of D are stored in a matrix H, called dissimilarity matrix; at each step, the pair of clusters with the minimum dissimilarity value is merged into a new cluster, and the dissimilarity value between this new cluster and any other existent cluster is computed. The dissimilarity value for the merged clusters is referred to as *merging coefficient* for that time step. The update of H can be conveniently carried out by employing the Lance-Williams formula [17]:

$$\begin{aligned} D(C_k, \{C_i, C_j\}) &= \alpha_i D(C_k, C_i) + \alpha_j D(C_k, C_j) + \\ &+ \beta D(C_i, C_j) + \gamma \left| D(C_k, C_i) - D(C_k, C_j) \right| \end{aligned} \tag{2}$$

where C_i and C_j are the two clusters joined to form the new cluster, and C_k is any other cluster ($k \neq i, j$). Different values of $\alpha_i, \alpha_j, \beta$, and γ are associated with different HC methods. For instance, with $\alpha_i = \alpha_j = 1/2$, $\gamma = -1/2$, and $\beta = 0$, single linkage HC is obtained. In this paper, we will focus on Ward's dissimilarity measure [18] but we demonstrate in the Supplemental Material that it can be applied to other dissimilarity measures. The effect of using Ward's method, along with the use of the squared Euclidean distance as a measure of dissimilarity between elements, is to obtain compact (i.e., having low within-cluster variance), spherical clusters, which is especially desirable when clustering cerebral activation peaks (see Section IV). In Ward's method, the dissimilarity between two clusters is defined as the increase in the total error sum of squares due to the merging of those two clusters. Thus, at each step, the measure being minimized is

$$\Delta ESS_{i,j} = ESS_{i,j} - ESS_i - ESS_j \tag{3}$$

with $ESS_k = \sum_{\mathbf{x} \in C_k} (\mathbf{x} - \boldsymbol{\mu}_k)^2$ where $\boldsymbol{\mu}_k$ is the centroid of cluster C_k. Thus, in Ward's method the dissimilarity between two clusters is a measure of their (collective) variance. As a result, each solution S_m in the final hierarchy is an approximation to the m-partition of the input data having minimum total intra-cluster variance, ESS:

$$ESS = \sum_{k=1}^{|C|} ESS_k \tag{4}$$

with $|C|$ the number of clusters. To implement Ward's method through the Lance-Williams formula, the following coefficients are used:

$$\begin{aligned} &\alpha_i = (n_k + n_i)/(n_k + n_i + n_j) \\ &\alpha_j = (n_k + n_j)/(n_k + n_i + n_j) \\ &\beta = -n_k/(n_k + n_i + n_j) \\ &\gamma = 0 \end{aligned} \tag{5}$$

where n_x is the cardinality of cluster C_x. It can be proved that, if $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$, then the above formula yields $D(C_i, C_j) = 2\Delta ESS_{i,j}$.

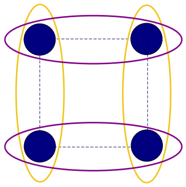


Fig. 2. Four data points lie at the corners of a square: pairs on each side have the same (Euclidean) distance, which leads to four minimal-dissimilarity (MD) pairs. If we run a HC algorithm on this dataset and cut the resulting dendrogram to get a 2-clusters solution, two different solutions are obtained (shown in yellow and purple, respectively), according to which pair of points is selected first.

A. The non-uniqueness of the solution in HC

HC can return different solutions depending on the order in which the input data are presented (see Fig. 2). This is due to the presence of *ties* in the dissimilarity matrix at a given step, that is the minimum dissimilarity value, v, is shared by more than one cluster pair.

Definition II.1. *Let $v = \min_{C_i,C_j} D(C_i, C_j)$, where C_i and C_j are clusters available at the current processing step, t. We call* ***minimal-dissimilarity pair*** *(MD pair) each pair of clusters $p = (C_i, C_j)$ such that $p \in P_t = \arg\min_{C_i,C_j} D(C_i, C_j)$; that is, $D(p) = v$ for each minimal distance pair p.*

At each step t (i.e., every time the dissimilarity matrix is updated), we might have more than one MD pair; that is, $|P_t| > 1$. The order in which the input data points are presented to the algorithm determines the order whereby cluster pairs are found inside matrix H; current algorithms just select the first MD pair encountered when browsing H. Therefore, a different permutation of the input data points can lead to the selection of a different MD pair, and this, in turn, can produce a different dendrogram. It also turns out that different dendrograms and associated solutions can be associated with different interpretations for the same set of data: therefore, the existence of ties in the dissimilarity matrix can potentially lead to unstable, and unreliable, conclusions about the structure underlying a dataset. Ties can occur quite frequently, especially at the initial merging stages, when dealing with discrete data (although they cannot be completely ruled out even for real-valued applications). Current hierarchical clustering algorithms, lacking any control on the existence of multiple solutions, choose arbitrarily a feasible dendrogram, according to one of the possible permutations of the input data. This may lead to different clustering solutions and to possible misinterpretations of clustering results.

III. Algorithm Description

The solution proposed here is based on identifying what we have called the *significantly different* alternative dendrograms, that result from the selection of different MD pairs. This requires making a first distinction between *critical* and *non-critical* MD pairs; whereas differences in clustering introduced by non-critical pairs can safely be disregarded, critical pairs require more attention. Secondly, to reduce complexity, a further distinction on critical pairs is introduced, aimed at identifying equivalence classes within them: in this way, only one representative per class can be fully developed.

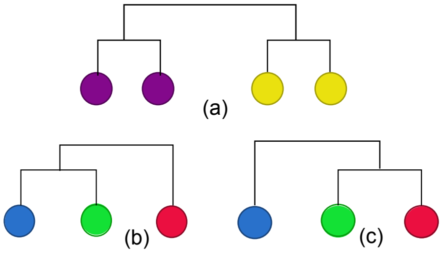


Fig. 3. The purple and the yellow pairs in panel (a) are non-critical pairs, that is MD pairs that have no cluster in common: whichever pair of points is merged first, the final dendrogram is the same. On the other hand, the three elements in panels (b) and (c) produce two critical pairs (blue-green and green-red) from which two different dendrograms are obtained, depending on which pair is selected first.

The notion of *non-critical* pairs follows from the observation that, in some cases, the choice between different MD pairs, although leading to different merging sequences, do not result in different dendrograms (see Fig. 3a).

Definition III.1. *A MD pair $p = (C_i, C_j)$ is a* ***non-critical pair*** *if $\forall p' = (C_{i'}, C_{j'})$, $p' \neq p$ being a MD pair, $i \neq \{i', j'\}$ and $j \neq \{i', j'\}$ hold.*

Non-critical pairs are therefore those pairs that do not share any element with other MD pairs. The choice of merging one non-critical pair in place of another does not affect the shape of the resulting dendrogram because these choices are not mutually exclusive: the choice of a non-critical pair leaves other non-critical pairs available for subsequent merging.

This can be also seen by analyzing H. Let us suppose that H contains n_p entries that have the same MD value, v, and that these entries are distributed such that for each row and column at most one entry is equal to v (it can be shown that this is another way to state Def. III.1). Whenever one MD pair, say (C_i, C_j), is merged, dissimilarity values for clusters C_i and C_j are discarded, which corresponds to deleting the i-th row and the jth column[1]. None of the other MD pairs would be touched by such operation. Therefore, at the subsequent clustering step, one of the remaining MD pairs, with $D(.) = v$, would be selected for merging, and so on, until all the non-critical pairs with $D(.) = v$ have been merged. Since all these pairs have merging coefficient equal to v, the same dendrogram is obtained regardless of the specific merging sequence. In other words, the choice among non-critical pairs cannot open new scenarios where a new MD pair appears, which would make the order whereby non-critical pairs are selected relevant. This is guaranteed by the following fact:

Theorem III.2. *Let v be the minimum value in the dissimilarity matrix and therefore the merging coefficient in the current clustering step; let C_i and C_j be the clusters being merged. In a HC algorithm employing Ward's method, each new dissimilarity value v' for the newly created cluster $\{C_i, C_j\}$ is such that $v' \geq v$. If (C_i, C_j) is a non-critical pair, then $v' > v$.*

[1] We implicitly assume here that H is stored as a triangular matrix.

Proof: According to the Lance-Williams updating equation for Ward's method (see (2) and (5)), for a generic cluster C_k $(k \neq i, j)$ the dissimilarity value v' of C_k from the new cluster $\{C_i, C_j\}$ is computed as

$$v' = \frac{1}{n_i + n_j + n_k}\left(z(n_k + n_i) + w(n_k + n_j) - v(n_k)\right) = \\ = \frac{1}{n_i + n_j + n_k}\left(n_k(z + w - v) + n_i z + n_j w\right)$$

where $z = D(C_k, C_i)$, $w = D(C_k, C_j)$. Since v is the minimum value in the dissimilarity matrix, and (C_i, C_j) is a non-critical pair, $z > v$ and $w > v$ hold; that is, (C_k, C_i) and (C_k, C_j) cannot be MD pairs; otherwise, (C_i, C_j) would not be non-critical by definition. Therefore we can write $z = v + \varepsilon$, $w = v + \eta$ ($\epsilon > 0$, $\eta > 0$), and

$$v' = \frac{1}{n_i + n_j + n_k}\left(n_k(v + \varepsilon + \eta) + n_i(v + \varepsilon) + n_j(v + \eta)\right) = \\ = \frac{(n_i + n_j + n_k)v}{n_i + n_j + n_k} + \frac{n_k\varepsilon + n_k\eta + n_i\varepsilon + n_j\eta}{n_i + n_j + n_k}$$

from which $v' > v$ follows[2]. ■

At each merging step all the MD pairs are identified and distinguished into non-critical and critical ones. As the merging sequence of non-critical pairs is not relevant, they are simply merged in a random order, producing a single dendrogram. On the other hand, a separate dendrogram can be developed for each alternative choice of a critical pair. In this way, the number of dendrograms that must be generated is reduced with respect to an exhaustive exploration of the space of all alternative dendrograms; however, in many practical situations, such reduction is not large enough (cf. Sect. IV). For this reason, we introduce an equivalence relation on the dendrograms (Figs. 4a–b) so that the dendrogram space can be shrunk and only one representative for equivalence class can be fully developed. We explicitly remark that equivalent dendrograms are not *identical* dendrograms.

Definition III.3. *Let $p = (C_i, C_j)$ and $p' = (C_j, C_k)$ be two critical pairs for the current step, and C, C′ the clusters resulting from their choice. We say that p and p′ are* ***equivalent pairs*** *if*

$$C_k = \arg\min_{C_x} D(C, C_x) \text{ and } C = \arg\min_{C_x} D(C_k, C_x) \quad (6a)$$

$$C_i = \arg\min_{C_x} D(C', C_x) \text{ and } C' = \arg\min_{C_x} D(C_i, C_x) \quad (6b)$$

$$D(C, C_k) = D(C', C_i). \quad (6c)$$

Theorem III.4. *When Ward's method is used, (6c) directly follows from the definition of critical pair (Def. III.1).*

Proof: Since $p = (C_i, C_j)$ and $p' = (C_j, C_k)$ are critical pairs, then $D(C_i, C_j) = D(C_j, C_k) = v$, where v is the minimum value in the current dissimilarity matrix. Then, by applying the Lance-Williams formula for Ward's method ((2) and (5)), we get:

$$D(\{C_i, C_j\}, C_k) = \\ = \frac{(n_i + n_k)D(C_i, C_k) + (n_j + n_k)v - n_k v}{n_i + n_j + n_k} = \\ = \frac{(n_i + n_k)D(C_i, C_k) + n_j v}{n_i + n_j + n_k}$$

and

$$D(\{C_j, C_k\}, C_i) = \\ = \frac{(n_j + n_i)v + (n_k + n_i)D(C_i, C_k) - n_i v}{n_i + n_j + n_k} = \\ = \frac{(n_i + n_k)D(C_i, C_k) + n_j v}{n_i + n_j + n_k}$$

from which property 6c follows. ■

We can restate Def. III.3 as follows. Considering the three clusters C_i, C_j, and C_k, we can refer to C_k as the excluded element when pair p is chosen, and to C_i as the excluded element when pair p' is selected. Conditions 6a and 6b state that p and p' are equivalent if p **and its excluded element are closer to each other than to any other cluster**, and the same holds for p'. This means that, whichever pair we select for generating a new cluster, the next merging step involving that cluster will group it with its excluded element. That is, although the shapes of the dendrograms corresponding to p and p' temporarily diverge, they do converge to the same clustering solution (Figs. 4a–b); if p and p' are non-equivalent the shape of their corresponding dendrograms cannot be guaranteed to converge (Figs. 4c–d).

Def. III.3 establishes an equivalence relation over dendrograms. In particular, (6c) guarantees that equivalent dendrograms – those associated with equivalent pairs – have the same sequence of merging coefficients. This allows us to fully develop only one representative dendrogram from each equivalence class. This drastically reduces the number of dendrograms to be fully built, making the problem computationally affordable.

Once all *non-equivalent* dendrograms (that is, the significantly different ones) have been generated, the corresponding solutions can be obtained cutting each dendrogram according to the user-designated strategy. Among these solutions, the best one according to the defined quality criterion is identified. In the applications presented here, the between-cluster error sum of squares has been adopted. The maximization of this measure favors a better separation among clusters:

$$bESS = \sum_{k=1}^{|C|} n_k(\boldsymbol{\mu}_k - \boldsymbol{\mu}_X)^2 \quad (7)$$

where $|C|$ is the number of clusters in the solution, n_k and $\boldsymbol{\mu}_k$ are the number of elements and the mean of cluster C_k, and $\boldsymbol{\mu}_X$ is the grand mean of the dataset X[3]. Let us remark that

[2] We notice that the case $v' = v$ can only occur when both $D(C_k, C_i)$ and $D(C_k, C_j)$ are equal to $v = D(C_i, C_j)$, that is when the three clusters are equidistant from each other (with dissimilarity v), but in such case they would not qualify as non-critical pairs.

[3] Notice that $bESS = ESS_{dataset} - ESS$, where ESS is the total within-cluster error-sum-of-squares introduced in (4), and with $ESS_{dataset}$ we refer to the error-sum-of-squares over the whole dataset, considered as a unique cluster.

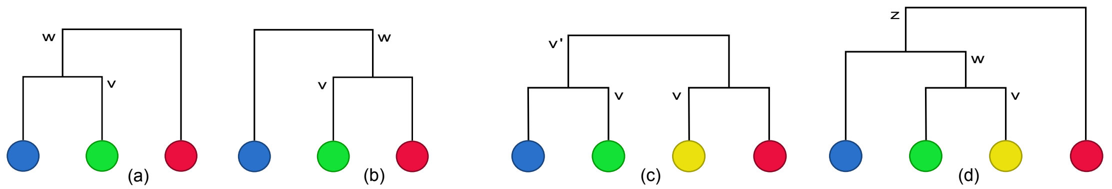


Fig. 4. Panels (a) and (b) show two equivalent pairs: blue-green and green-red. The element closest to the first pair is the red one, and the one closest to the second pair is the blue one. This guarantees that these three elements will be grouped in the same cluster: although the two dendrograms are different, they are equivalent. Notice that in intermediate steps the clusters obtained are different, but the sequence of merging coefficients (v, w) is the same independently of which pair is first merged. The blue-green and the green-yellow pairs shown in panels (c) and (d) are non-equivalent pairs. If the blue-green pair is merged first, the yellow element will then be merged with the red one as this is closer to it than the newly created cluster. The 2-clusters solution $\{\text{blue} \cup \text{green}\}$ and $\{\text{yellow} \cup \text{red}\}$ is obtained. If we first merge the green-yellow pair, instead, we would obtain a different 2-clusters solution: $\{\text{blue} \cup \text{green} \cup \text{yellow}\}$ and $\{\text{red}\}$. Notice that the merging coefficients are different in the two cases: v, v, v' and v w, z respectively.

other user-defined measures could be employed to evaluate the different clustering solutions.

The end result of our method is a clustering solution that is unique, up to equivalences. It is also optimal, with respect to the desired measure of quality, among the alternative solutions that the HC algorithm would return with different orderings of the input data. The operation flow of the proposed algorithm is summarized in Fig. 5. The key element is the state of the clustering process, that is saved each time a new dendrogram has to be developed; specifically, the state contains the current step t, the dissimilarity matrix, the non-equivalent pairs still to be examined, the parent dendrogram from which a new one will be developed, the current merging coefficient, and additional information about current clusters (their number, cardinalities, and indexes).

IV. Results

The algorithm presented here has been extensively applied in two different domains in which ties often occur: analysis of neuroimaging data and of protein-protein interactions[4].

HC has been recently introduced in the field of functional neuroimaging as a tool for a meta-analysis of large sets of brain activation sites that are reported in a broad collection of studies investigating different aspects of a specific cognitive function [19]. In this context, the result of HC is used to identify groups of anatomically close activation peaks that may represent functionally meaningful brain regions inside specific networks of cortical and subcortical areas involved in the cognitive function of interest. In particular, we have investigated the possible cortical network involved in single word reading through a meta-analysis with HC of a set of 1176 activation peaks collected over 35 different studies. 128 *significantly different* dendrograms were found. These were cut at the level where the average standard deviation over the clusters in any of the three directions raised above $\sigma = 7.5$; this value was set in agreement with [19], to comply to the standard resolution of functional images, of about 15 mm. Cutting produced four different solutions, the optimal one being composed of 57 clusters. The statistical analysis of the solution allowed us to identify the putative functional role of each cluster, and thus of its corresponding brain area (see [20] for details). Here we want to remark that the optimal clustering found by our algorithm allows distinguishing between a more lateral region (in the Angular gyrus) showing a preference for word stimuli and an occipital one that is less sensitive to lexicality, whereas in other (non-optimal) solutions this distinction would be lost (Fig. 6a–b).

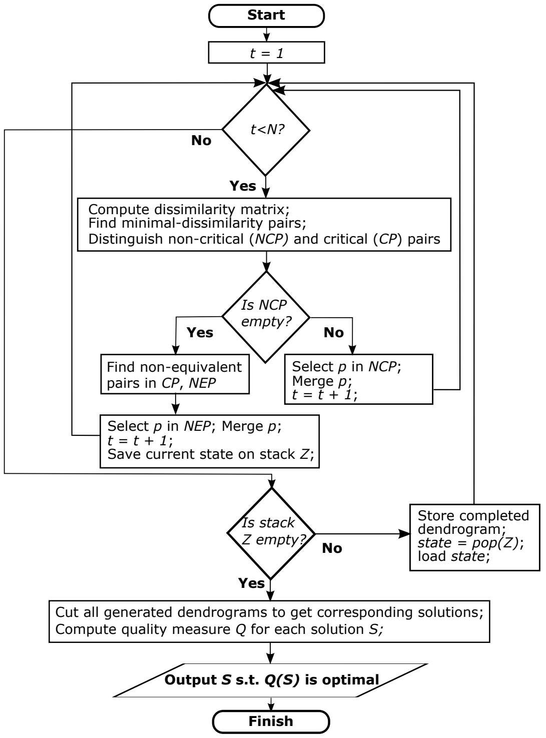
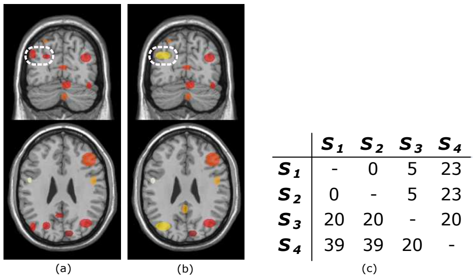

Start
t = 1
t<N?
No
Yes
Compute dissimilarity matrix;
Find minimal-dissimilarity pairs;
Distinguish non-critical (NCP) and critical (CP) pairs
Is NCP empty?
Yes
No
Find non-equivalent pairs in CP, NEP
Select p in NCP;
Merge p;
t = t + 1;
Select p in NEP; Merge p;
t = t + 1;
Save current state on stack Z;
Is stack Z empty?
No
Store completed dendrogram;
state = pop(Z);
load state;
Yes
Cut all generated dendrograms to get corresponding solutions;
Compute quality measure Q for each solution S;
Output S s.t. Q(S) is optimal
Finish

Fig. 5. Flow chart of the proposed algorithm. We employ a stack structure in which the current state of the process is saved when a new dendrogram is generated. When a dendrogram is completed, the state on top of the stack is loaded, and from this the next non-equivalent pair identified at step t is extracted. From this pair a new dendrogram is developed. Notice that NEP contains one representative pair for each equivalence class, identified for a given element shared by the critical pairs at the current step. Pairs that are equivalent to those stored in NEP are just discarded.

The potential risk of misinterpreting data, when multiple clustering solutions are present, can be further demonstrated with respect to bioinformatics data. HC is one of the most used techniques for their analysis, with applications ranging from biomolecular evolution to multiple sequence alignment, functional genomics and DNA microarray data analysis [21]. In several bioinformatics clustering problems data are two-valued (e.g. they represent whether a given property is present

[4] See Supplemental Material for more details on these experiments.



	S_1	S_2	S_3	S_4
S_1	-	0	5	23
S_2	0	-	5	23
S_3	20	20	-	20
S_4	39	39	20	-

Fig. 6. On the left, two alternative clustering solutions for our neuroimaging dataset are shown: (a) the optimal solution ($bESS = 2.4023 \times 10^6$); (b) one of the alternative solutions ($bESS = 2.3977 \times 10^6$). Each cluster is represented by a blob centered on its mean coordinate with semiaxes equal to the cluster standard deviation. The color of a blob codes for its cardinality. Only one section of the cerebral volume is shown. The white box highlights the difference in the two clustering solutions. On the right the comparison of the number of GO BP terms differentially overrepresented at 0.001 significance level between the different clustering solutions S_i.

or not for a given gene or protein) and are characterized by high dimensionality and sparsity. It is therefore likely to obtain dissimilarity matrices with ties, but the consequent problem of non-uniqueness of the solution is largely neglected.

As an example, we report here the clusters of functionally related proteins obtained analyzing protein-protein interaction (PPI) data of a random subset (of size 500) of the 5367 proteins downloaded from the *BioGRID* database [15]. 96 significantly different dendrograms were identified and cut with a threshold $\sigma = 5$ on the norm of the vector representing the average cluster standard deviation over each dimension. This left us with 4 unique solutions: S_1 and S_2, including 9 clusters; S_3 and S_4, of 10 clusters. To understand whether the different solutions lead to different biological conclusions w.r.t. protein-protein interactions, we performed a functional enrichment analysis of the different clusterings [22] to assess whether known functional categories are significantly overrepresented in the discovered clusters. We have chosen the Gene Ontology (GO) terms of the Biological Processes (BP) ontology [23] as functional categories: each GO term represents a class of gene/proteins with common functional characteristics (e.g. *catabolic process* or *regulation of translation*). For each clustering solution S_i we merged the GO terms that we found significantly overrepresented in each cluster. Lastly we compared the set of GO terms that biologically characterize each S_i. We found that 233, 233, 248 and 249 GO terms were significantly overrepresented in the unique solutions. These turn out to be quite similar, but with some relevant differences. Although no difference exists between S_1 and S_2, we did find significant differences between all the other solutions: for instance, S_1 and S_4 differ for 23 terms overrepresented in S_1 but not in S_4 and 39 GO BP terms in the opposite direction (Fig. 6c) – that is, about 1/4 of the GO terms identified are different between S_1 and S_4 (the "optimal" solution). In particular, analyzing the GO terms overrepresented in S_4 but not in S_1, we observe that the additional functional classes present in S_4 are characterized by biological processes involved in the structural organization of cellular components and by its related anabolic/catabolic processes (see Supplemental Material, Table 1). This makes the two solutions very different also from the semantic point of view as they lead to different biological characterization of clustering results.

V. Discussion

As shown in Section IV, the non-uniqueness of the solution is a critical problem, since it can make results inconsistent, leading to different interpretations of the same data depending on the order in which the data are presented. To avoid this, *all* the possible dendrograms that result from different MD pairs could be considered, but this is not a feasible approach. In fact, in the worst case, we obtain $p = N/2$ non-critical pairs at the first clustering step, from which $(N/2)!$ dendrograms are generated, leading to a complexity of $O(N!)$.

Our method allows to greatly reduce the number of generated dendrograms, without sacrificing completeness. This is achieved by a careful analysis of the ties that arise in the clustering process. More precisely, we showed that it is possible to identify the equivalence classes over dendrograms, according to Def. III.3, and to generate a single dendrogram per class. One can envision the strategy described here as a shrinkage from a combinatorial space consisting of all possible dendrograms that can stem from ties, to a reduced space where only the most salient dendrograms (those that we called significantly different) are retained.

The reduction in the number of dendrograms is relevant: only 128 dendrograms were generated for the neuroimaging data of Section IV (96 for the PPI data). On the contrary, when considering all MD pairs, or even the critical pairs only (equivalent and non-equivalent ones), the clustering procedure had to be stopped when 100,000 dendrograms were generated, because of memory saturation, confirming the combinatorial explosion due to ties.

This reduction has been obtained by limiting the number of dendrograms that must be fully developed, although for each new dendrogram all the data that identify the clustering state have to be saved. The dominant cost is represented by the dissimilarity matrix, which is $O(N^2)$, at least at the first clustering steps. Overall, the algorithm has therefore a complexity of $O(qN^2)$, where q is the number of non-equivalent pairs encountered along the clustering process (and of the generated dendrograms); we explicitly remark that usually $q << p$. This figure is much smaller than $O(N!)$, obtained when developing all the dendrograms stemming from MD pairs. After all clustering solutions have been generated, an additional step is required to identify the *unique* solutions: in fact, some solutions, although deriving from different dendrograms, may be constituted of the same clusters.

We remark here that equivalent dendrograms are not *identical* dendrograms: by choosing one representative for each equivalence class, we do compress information. Let us consider, for instance, Figs. 4a–b, that show two equivalent dendrograms. If a 2-cluster solution is required, different solutions would be obtained from those dendrograms. Although in both cases the two clusters will be merged into the same cluster in the subsequent clustering steps, these two equivalent clusterings do exist in the intermediate steps. Notice that this is true even for identical dendrograms: the dendrogram shown in

Fig. 3a could have been obtained by either merging the yellow elements first, or the purple ones. According to which pair was selected first a different 3-cluster solution is obtained. In these cases, equivalent pairs should be tracked at each clustering step. More precisely, by introducing an additional "backtracking" step that traces back pairs of equivalent dendrograms, we could explore all the equivalent (but not identical) clusterings associated with a given dendrogram cut (see Supplemental Materials for more details).

Also notice that, at each step, we identify all the MD pairs, but only one pair of clusters is merged. A speed-up could be attained if we merged non-critical pairs in one step. This can be seen as collapsing multiple clustering steps into one. This, in fact, is akin to the strategy taken by [16], where all MD pairs for one level, both critical and non-critical ones, are simultaneously merged in "superclusters", and the result is depicted in one multidendrogram. By doing so, however, some solutions are arbitrarily discarded. Let us consider, for instance, Fig. 4c and 4d, and assume that, whereas $\{\text{blue}, \text{green}\}$ and $\{\text{green}, \text{yellow}\}$ are MD pairs with $d(.) = v$, the pair $\{\text{yellow}, \text{red}\}$ has $d(.) = v+\varepsilon$, with $\varepsilon << v$. In [16], the 2-cluster solution depicted in Fig. 4c, which might turn out the "optimal" one, would never be obtained, as the $\{\text{blue}, \text{green}, \text{yellow}\}$ supercluster would be forced.

The algorithm has been described here with Ward's dissimilarity measure but it can be applied to other measures as well (see Supplemental Material), as long as they are not prone to inversion [13]. This occurs when the sequence of merging coefficients is non-monotonic and, in this case, the fact that Eqs. 6a and 6b hold at the current step does not guarantee equivalence, as a subsequent merging operation could produce a cluster C_z that is closer to $\{C_i, C_j\}$ than C_k. The monotonicity requirement rules out centroid and median linkage clustering, whereas simple, complete, group and weighted group average linkages can be successfully employed with the presented method.

The proposed approach could also be extended to the case of real-valued data sets. Although in this case it is unlikely that exact ties occur, it is possible that the data are affected by noise; more elaboration on this is given in the Supplemental Material.

In conclusion, we notice that although the final solution returned by our method may be called an optimized solution since it is the best one according to the criterion set, among those that have been generated, it cannot be assumed optimal in a global sense. This directly follows from the greedy nature of the HC agglomerative process and represents a distinct well-known problem, which is out of the scope of the present work.

VI. Conclusion

We have discussed how ties in the data can cause HC to yield very different solutions for different permutations of the same input data. We have shown that by defining an adequate equivalence relation over the dendrograms stemming from the data, all the significantly different clusterings can be generated with polynomial complexity. This allows obtaining a unique solution independently of the data presentation order, which guarantees a unique interpretation of the data. The identification of the final unique solution is driven here by the maximum of Eq. 7, but it could also be made by a domain expert based on his experience. As illustrated by the experimental results, this approach could be a valuable choice for several neuroimaging and bioinformatics problems, but it could be suitable also to other application domains in which discrete data values are present that may easily lead to ties in the data.

References

[1] R. L. Goldstone, "The role of similarity in categorization: providing a groundwork," *Cognition*, vol. 52, no. 2, pp. 125–57, 1994.

[2] R. Xu and D. Wunsch, "Survey of clustering algorithms." *IEEE transactions on neural networks*, vol. 16, no. 3, pp. 645–78, 2005.

[3] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. Elsevier Academic Press, 4th Edition, 2009.

[4] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[5] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[6] Y. Guo and D. Schuurmans, "Convex relaxations of latent variable training," in *Advances in Neural Information Processing Systems*, 2008, pp. 601–608.

[7] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by Simulated Annealing." *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[8] T. Su and J. G. Dy, "In search of deterministic methods for initializing K-means and Gaussian mixture clustering," *Intelligent Data Analysis*, vol. 11, no. 4, pp. 319–338, 2007.

[9] S. Ferrari, G. Ferrigno, V. Piuri, and N. A. Borghese, "Reducing and Filtering Point Clouds with Enhanced Vector Quantization," *IEEE Transactions on Neural Networks*, vol. 18, pp. 161–177, 2007.

[10] R. M. Cormack, "A review of classification," *Journal of the Royal Statistical Society*, vol. 134, no. 3, pp. 321–367, 1971.

[11] U. von Luxburg and S. Ben-David, "Towards a statistical theory of clustering," in *PASCAL workshop on Statistics and Optimization of Clustering, London*, 2005.

[12] R. Sibson, "Order invariant methods for data analysis (with discussion)," *J. Royal Statistical Society B*, vol. 34, pp. 311–349, 1972.

[13] B. J. T. Morgan and A. P. G. Ray, "Non-uniqueness and inversions in cluster analysis," *Applied Stat.*, vol. 44, no. 1, pp. 117–134, 1995.

[14] W. A. Van der Kloot, A. M. J. Spaans, and W. J. Heiser, "Instability of Hierarchical Cluster Analysis Due to Input Order of the Data: The PermuCLUSTER Solution," *Psychological Methods*, vol. 10, no. 4, pp. 468–476, 2005.

[15] C. Stark, B. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers, "BioGRID: a general repository for interaction datasets," *Nucleic Acids Res.*, vol. 34, pp. D535–D539, 2006.

[16] A. Fernández and S. Gómez, "Solving non-uniqueness in agglomerative hierarchical clustering using multidendrograms," *Journal of Classification*, vol. 25, pp. 43–65, 2008.

[17] G. N. Lance and W. T. Williams, "A generalized sorting strategy for computer classifications," *Nature*, vol. 212, pp. 218–218, 1966.

[18] J. H. J. Ward, "Hierarchical grouping to optimize an objective function," *Journal American Statistical Association*, vol. 58, pp. 236–244, 1963.

[19] G. Jobard, F. Crivello, and N. Tzourio-Mazoyer, "Evaluation of the dual route theory of reading: a metanalysis of 35 neuroimaging studies," *NeuroImage*, vol. 20, no. 2, pp. 693–712, 2003.

[20] I. Cattinelli, N. A. Borghese, M. Gallucci, and E. Paulesu, "Reading the reading brain: A new meta-analysis of functional imaging data on reading," *Journal of Neurolinguistics*, vol. 26, no. 1, pp. 214–238, 2013.

[21] P. D'haeseleer, "How does gene expression clustering work?" *Nature Biotechnology*, vol. 23, pp. 1499–1501, 2005.

[22] P. Khatri and S. Draghici, "Ontological analysis of gene expression data: current tools, limitations, and open problems," *Bioinformatics*, vol. 21, no. 18, pp. 3587–3595, 2005.

[23] The Gene Ontology Consortium, "Gene ontology: tool for the unification of biology," *Nature Genetics*, vol. 25, pp. 25–29, 2000.