



HAL
open science

A note on Apresjan Hierarchies and Linkage-Based Hierarchical Clustering

Patrice Bertrand, Jean Diatta

► **To cite this version:**

Patrice Bertrand, Jean Diatta. A note on Apresjan Hierarchies and Linkage-Based Hierarchical Clustering. 2025. <hal-05285900>

HAL Id: hal-05285900

<https://hal.science/hal-05285900v1>

Preprint submitted on 14 Oct 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

A note on Apresjan Hierarchies and Linkage-Based Hierarchical Clustering

Patrice Bertrand¹ and Jean Diatta²

¹Ceremade, Université Paris-Dauphine-PSL, Paris, France

²LIM-EA2525, Université de La Réunion, Sainte Clotilde, France

Abstract

The model of an Apresjan cluster is a formal account of what the intuitive notion of cluster amounts to. The set of all Apresjan clusters forms a hierarchy, called the Apresjan hierarchy. In this note, we identify a general class of linkage functions for which the Hierarchical Agglomerative Clustering (HAC) algorithm generates hierarchies containing the Apresjan hierarchy. We deduce that, if the HAC algorithm is applied with a standard linkage function not based on centroids, then every generated hierarchy contains the Apresjan hierarchy. We also present preliminary results related to the HAC algorithm and arbitrary linkage functions, e.g. the characterization of all subsets belonging to all hierarchies generated by the HAC algorithm when this algorithm is applied to a given dataset and used with an arbitrarily chosen linkage function.

Keywords— Apresjan hierarchy, Hierarchical Agglomerative Clustering, Linkage function.

1 Introduction

Given a dissimilarity d defined on a dataset S , an *Apresjan cluster of d* is a subset of S , in which the degree of dissimilarity of any two elements is less than the degree of dissimilarity of each of these two elements with every external element (cf. e.g. [3]). This definition guarantees both homogeneity and isolation of Apresjan clusters, and as it reflects the intuition behind the cluster concept, several authors pointed out that Apresjan clusters are natural clusters (e.g. [4, 6, 11]).

Apresjan clusters were defined (often independently) and studied by a number of authors, initially by Parker-Rhodes and Needham [20] who called them K -clumps, then by Apresjan [3], Jardine [17] and Bandelt and Dress [5], as strong-clusters, Bock [9] and Gordon [16], as K -groups, Benzecri [6, 7], Bryant and Berry [11] and Devauchelle *et al.* [12], as Apresjan clusters, Balcan *et al.* [4] as the strict separation property, and by Ackerman *et al.* [1] as a nice clustering for the partitioning problem. An attractive property of Apresjan clusters is that the set of all Apresjan clusters of d forms a hierarchy called the *Apresjan hierarchy of d* (see, e.g. [3, 7, 11]). Moreover, it has been proved that the Apresjan hierarchy is contained in the hierarchies produced by the well-known Hierarchical Agglomerative Clustering (HAC) algorithm when it is used with the single-linkage, the complete-linkage, and the average-linkage (see [6, 11]).

Recall that the popular HAC method consists of a repeated, step-wise and bottom-up cluster merging which is achieved on the basis of the values taken by a linkage function D formerly chosen by the user. Apart from the singletons, which are considered to be at level 0, the level of the cluster C formed by merging clusters A and B , is $D(A, B)$. For most datasets, applying the HAC method with different linkage functions produces different hierarchies. This constitutes the user's dilemma which has been known for a long time,

see e.g. [14, 21, 13]. Furthermore, apart from the single linkage, repeated runs of the HAC algorithm with identical initialization conditions can produce different hierarchies if arbitrary choices have to be made to select the pair of clusters to be aggregated in case of a tie. Together with the user’s dilemma, this raises the question of the relevance of the formed clusters, in particular when arbitrary aggregation choices have been made. Then, in this note, as the Apresjan criterion guarantees both homogeneity and isolation of Apresjan clusters, we aim to determine a general framework that enables to identify those standard and non-standard dissimilarity-based linkage functions with which each hierarchy produced by the HAC algorithm, contains all the Apresjan clusters of the underlying dissimilarity function. In addition, we will demonstrate some preliminary properties defined in a general statement that contributes to increasing knowledge about the HAC algorithm and linkage functions. We provide a four-fold contribution which can be summarized as follows:

- (1) A necessary and sufficient condition for an arbitrary nonempty subset of the dataset S to be formed as a cluster of each hierarchy generated by the HAC method;
- (2) The definition of an Apresjan cluster of an arbitrary linkage function, say D , and the fact that the set of Apresjan clusters of D , is a hierarchy which is included in every hierarchy generated by the HAC algorithm when the linkage function D is used.
- (3) A general and sufficient condition to ensure the inclusion of the Apresjan hierarchy in each possibly pruned (i.e. by removing any cluster contained in another one of equal level) hierarchy obtained by applying the HAC algorithm to the underlying dissimilarity;
- (4) The list of standard linkage functions for which each (possibly pruned) hierarchy obtained by the HAC method, contains the Apresjan hierarchy.

The rest of the paper is organized as follows. Section 2 presents a summary of the HAC algorithm, and the formal setting of Apresjan clusters. In Section 3, using a general setting, we prove preliminary properties that concern both the HAC algorithm and arbitrary linkage functions. In particular, this section provides a necessary and sufficient condition for a nonempty subset of S to be formed as a cluster by the HAC method. In Section 4, we propose a general sufficient condition so that all Apresjan clusters of a dissimilarity are not only formed by the HAC method, but also never pruned by removing clusters formed at the same level as their supercluster. Section 5 aims to determine those standard dissimilarity-based linkage functions for which the HAC method always produces hierarchies that contain the Apresjan hierarchy of the underlying dissimilarity. Moreover, this section presents counter-examples for standard dissimilarity-based linkage functions for which a produced hierarchy may fail to contain the Apresjan hierarchy of the underlying dissimilarity.

2 Background

In this section, we review the HAC method and set the definition of an Apresjan cluster of a dissimilarity. In what follows, S will denote the ground set assumed to be nonempty and finite, and endowed with a (proper) dissimilarity d . Recall that a (proper) dissimilarity d on S is a map $d : S \times S \rightarrow \mathbb{R}^+$ satisfying

$$d(x, y) = d(y, x) > d(x, x) = 0,$$

for all distinct elements x, y in S . We first recall the usual terminology of the HAC algorithm. At the initialization step, the set of clusters is such that every datapoint is, in its own, a cluster. Then, the algorithm successively merges two clusters that minimize a given linkage function, say D , until all the datapoints are merged into a single cluster. A linkage function D on S is a symmetric function from $2^S \times 2^S$ into \mathbb{R}^+ , whose value $D(A, B) = D(B, A)$ measures a dissimilarity degree between every two disjoint subsets A and B . In the following, we extend the definition of a linkage D to the diagonal of $2^S \times 2^S$ and set $D(\{x\}, \{x\}) = 0$ for all $x \in S$. In this text, we focus on linkage functions D , implicitly or explicitly, defined from the dissimilarity d , with in addition the following constraint that for all $x, y \in S$:

$$D(\{x\}, \{y\}) = d(x, y).$$

We will then say that D is a *dissimilarity-based linkage function*, more precisely a *d-based linkage function*. The HAC algorithm starts at step $t = 0$ with the collection \mathcal{C}_0 of all the singletons of S . At each step $t \geq 1$, let \mathcal{C}_{t-1} be the set of all maximal (by inclusion) clusters among all clusters created during the first $t - 1$ steps. The algorithm merges two clusters, say A and B , that realize the minimum of D among the pairs of clusters in \mathcal{C}_{t-1} . Therefore:

$$\mathcal{C}_t = \mathcal{C}_{t-1} \cup \{A \cup B\} \setminus \{A, B\}.$$

One can notice that \mathcal{C}_t is a partition of S at each step $t \geq 0$. The algorithm stops when \mathcal{C}_t is reduced to $\{S\}$, or, equivalently, after $n - 1$ steps, where n denotes the size of S . The collection $\mathcal{C} = \bigcup_{t=0}^{n-1} \mathcal{C}_t$ of clusters generated by the HAC algorithm is a *hierarchy on S* , i.e. a collection of subsets of S , containing S and its singletons, and such that any two members are either disjoint or nested. Notice that the hierarchy \mathcal{C} depends not only on the linkage function D , but also on the pairs of clusters merged at all steps where several choices of merging are possible. Finally, the hierarchy \mathcal{C} is endowed with the *index map* $f_D : \mathcal{C} \rightarrow \mathbb{R}^+$ defined by:

$$f_D(\{x\}) = 0, \text{ for all } x \in S \text{ and } f_D(A \cup B) = D(A, B),$$

when $A \cup B$ is the cluster formed by merging clusters A and B . It is said that the pair (\mathcal{C}, f_D) is an *indexed hierarchy* or that *the hierarchy \mathcal{C} is indexed by f_D* if:

$$\text{for all } A, B \in \mathcal{C}: A \not\subseteq B \Rightarrow f_D(A) < f_D(B). \quad (1)$$

Recall that a pair of clusters (A, B) is called an *inversion* (also referred to as a *reversal* (e.g. [18]) if:

$$A \not\subseteq B \text{ and } f_D(A) > f_D(B).$$

It may be noticed that there is no inversion if for all $A, B \in \mathcal{C}$, $f_D(A) \leq f_D(B)$ whenever $A \not\subseteq B$. Moreover, at the end of the process of cluster merging, the obtained collection \mathcal{C} is *pruned* by removing from \mathcal{C} all clusters A for which a cluster B exists in \mathcal{C} such that

$$A \not\subseteq B \text{ and } f_D(A) = f_D(B). \quad (2)$$

After the *pruning step*, and in case of no inversion, the pair (\mathcal{C}, f_D) is clearly an indexed hierarchy in the sense of (1). An indexed hierarchy (\mathcal{C}, f_D) is graphically represented by means of a *dendrogram*: see Figure 2 for an example of two dendrograms. In view of our objective, we now introduce a little more terminology:

- A *D-linkage hierarchy* will denote any binary hierarchy generated by applying the HAC algorithm to S endowed with the dissimilarity d , and by using the linkage function D without any pruning. More generally, a *HAC-linkage hierarchy* will denote any hierarchy which coincides with a *D-linkage hierarchy* for some linkage function D .
- A *D-linkage cluster* will denote any member of a *D-linkage hierarchy*.

The next notation is related to the notion of multiset. It will be useful for defining some linkage functions.

Notation 2.1. Let $X = \{x_1, x_2, \dots, x_n\}$ be a (finite) multiset of real numbers, i.e. a collection of real numbers where repeated elements are allowed. For example, $\{1, 1, 1, 3, 4, 4\}$ is a multiset. The notations $\min X$, $\text{med } X$ and $\max X$ will denote, respectively, the minimum, the median and the maximum of X . If $A, B \subseteq S$ and d is a dissimilarity on S , then the multiset $\{d(a, b) \mid a \in A, b \in B\}$ will be denoted as $d(A, B)$. Furthermore, given $a \in S$ and $B \subseteq S$, $d(a, B) = d(B, a)$ will stand for $d(\{a\}, B) = d(B, \{a\})$.

Example 2.2. Consider the dissimilarity d_1 defined on the 8-element set $S_1 = abcdefgh$, represented in Figure 1, where the label of the edge between two pairs of subsets, say X and Y , denotes the dissimilarity

degree between any element of X and any element of Y . For instance, for $X = \{a, b\}$ and $Y = \{g, h\}$, then $d_1(a, g) = d_1(b, g) = d_1(a, h) = d_1(b, h) = 5$.

Now, define the linkage function D_1 by

$$D_1(X, Y) = \begin{cases} \max d_1(X, Y) & \text{if } \lfloor \max d_1(X, Y) \rfloor \text{ is odd,} \\ \max d_1(X, Y) + 1 & \text{else,} \end{cases}$$

where $\lfloor x \rfloor$ denotes the largest integer less than or equal to x . Only two D_1 -linkage hierarchies can be obtained when the HAC algorithm is run with the linkage function D_1 . Both of these D_1 -linkage hierarchies are represented in Figure 2.

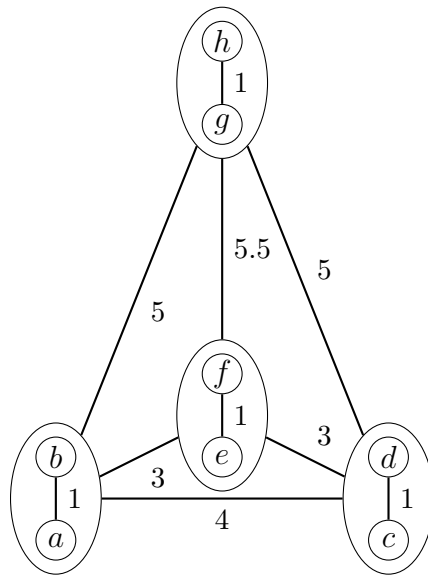


Figure 1: Dissimilarity function d_1

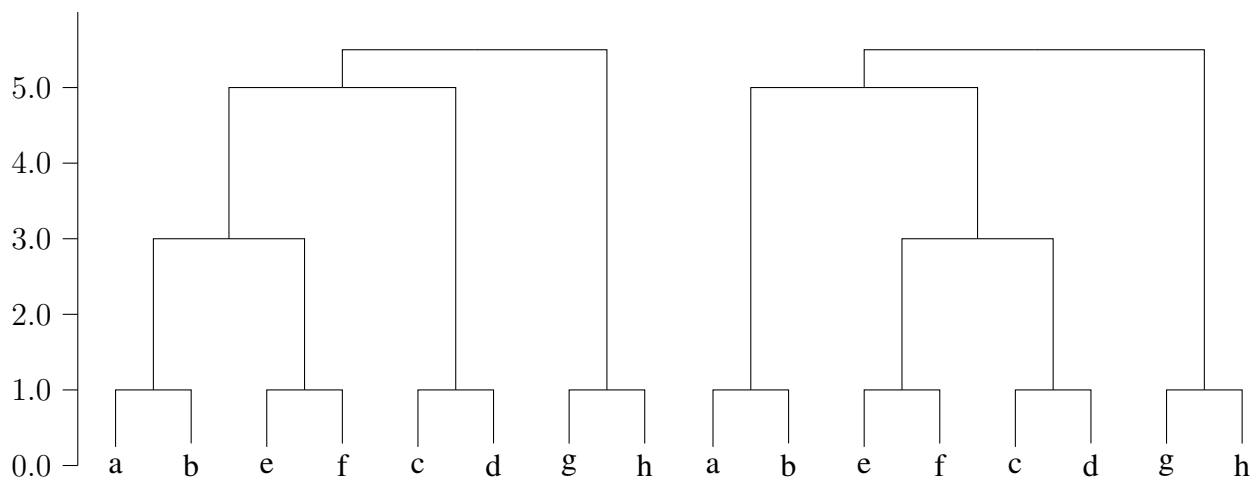


Figure 2: Dendrograms of all the (indexed) D_1 -linkage hierarchies.

Let us now take a brief look at the notion of Apresjan cluster. A nonempty subset C of S is an *Apresjan cluster* of a dissimilarity d or an *d -Apresjan cluster* if the mutual dissimilarity degree between two arbitrary

elements of C is less than the mutual dissimilarity degree between either of these two elements and any element outside C [3]. In other words, if

$$\text{for all } x, y \in C: d(x, y) < \min_{z \notin C} (\min\{d(x, z), d(y, z)\}). \quad (3)$$

Equivalently, C is a d -Apresjan cluster if

$$\text{for all } (x, y, z) \in C \times C \times (S \setminus C): d(x, y) < d(x, z). \quad (4)$$

Either of the equivalent characterizations (3) and (4) accounts for both the homogeneity and the isolation of an Apresjan cluster insofar as it shows that each element of the cluster has the following property: its highest degree of mutual dissimilarity with elements inside the cluster is lower than its lowest degree of mutual dissimilarity with elements outside. The set of Apresjan clusters of a dissimilarity d forms a hierarchy hereafter called the *Apresjan hierarchy* of d (cf. e.g. [3, 7]).

Example 2.3. Consider the dissimilarity d_1 given in Example 2.2. Figure 3 represents the Apresjan hierarchy of d_1 . It may be observed that the Apresjan hierarchy of d_1 is contained in each of the two D_1 -linkage hierarchies displayed in Figure 2. More precisely, here, the Apresjan hierarchy of d_1 coincides with the intersection of these two D_1 -linkage hierarchies.

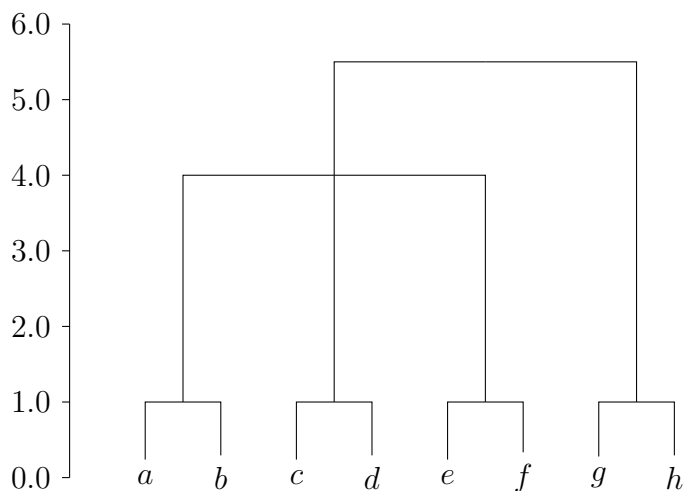


Figure 3: The Apresjan hierarchy of d_1

The definition of Apresjan clusters makes them interesting in the sense that they are homogeneous and well isolated. However, several authors (cf. [6, 11]) note that this definition is very restrictive to the point of being generally satisfied only by very small sized clusters (size around 2), apart from the trivial clusters that are singletons and the ground set S itself. This often results in Apresjan hierarchies with few nontrivial clusters, as noticed in [6, 11]. Nevertheless, it has been proved that if d is ultrametric then the Apresjan hierarchy of d is equal to the single-linkage hierarchy built from d [8]. Thus, by the argument of continuity the closer the dissimilarity d is to an ultrametric (even if this is true only for a subset of the dataset, whose size is significant), the larger the size of nontrivial Apresjan clusters (cf. [8]). As an example, let us consider the dataset of Example 2.2, for which the percentage of ultrametric triples of d_1 is approximately 57% (more precisely, 32 out of the 56 possible triples in $S = abcdefgh$, are ultrametric): this percentage is unusually high and then consistent with the fact that $abcdef$, which is of large size relatively to the size of S , is an Apresjan cluster of d_1 . Consequently, this puts in perspective the assessment of the scarcity of the nontrivial Apresjan clusters. Combined with the quality of its clusters, these characteristics motivate interest in identifying richer cluster systems that contain the Apresjan hierarchy of a given dissimilarity. There are proofs that the Apresjan hierarchy is contained in the single-linkage hierarchy [6, 11], in the complete-linkage hierarchy [6] and in the average-linkage hierarchy [11]. In this note, we introduce a unifying framework for identifying those standard or non standard dissimilarity-based linkage functions for which the HAC algorithm generates hierarchies containing the Apresjan hierarchy of the underlying dissimilarity.

3 Preliminary results

This section provides preliminary properties that concern, in a general setting, the HAC algorithm and both standard and arbitrary linkage functions. We first prove a necessary and sufficient condition for a nonempty subset to belong to a given D -linkage hierarchy, together with a necessary and sufficient condition for a nonempty subset to belong to each D -linkage hierarchy. Within this aim, we introduce the so-called closed sets for a given D -linkage hierarchy and D -min-closed sets, the latter turn out to coincide with the subsets belonging to each D -linkage hierarchy, as proved by Theorem 3.4.

Definition 3.1 (Closed subset for a D -linkage hierarchy). A nonempty subset C of S is said to be *closed for a D -linkage hierarchy \mathcal{C}* if, we have:

$$X \not\subseteq C \implies Y \not\subseteq C,$$

for all clusters X and Y merged at any step $t \geq 0$ of the HAC algorithm that generates \mathcal{C} .

Definition 3.2 (D -min-closed subset). A nonempty subset C of S is said to be *D -min-closed* if for all D -linkage hierarchy \mathcal{C} and for an arbitrary step $t \geq 0$ of the HAC algorithm that generates \mathcal{C} ,

$$X \not\subseteq C \implies Y \not\subseteq C,$$

for all $(X, Y) \in \arg \min_{A, B \in \mathcal{C}_t} D(A, B)$.

Clearly, a subset C is D -min-closed if and only if it is closed for all D -linkage hierarchies.

Lemma 3.3. Let \mathcal{C} be a D -linkage hierarchy. If C is closed for \mathcal{C} , $A \in \mathcal{C}$ and $A \not\subseteq C$, then there exists a sequence of members of \mathcal{C} , say A_1, \dots, A_j , such that

- (α) $A_1 = A$ and $A_j = C$.
- (β) For all $i \in [1, j - 1]$, A_i is merged with $(A_{i+1} \setminus A_i) \in \mathcal{C}$.

Proof. Let $A_1 = A$. Observe that $A \in \mathcal{C}$ is such that $A \not\subseteq C \subseteq S$, so that there exists a step t_1 at which A_1 is merged with a member of \mathcal{C}_{t_1} , say B_1 . As C is closed for \mathcal{C} , $B_1 \not\subseteq C$. Let $A_2 = A_1 \cup B_1$. If $A_2 = C$, then the sequence (A_1, A_2) fulfills conditions (α) and (β). Otherwise, as $A_2 \not\subseteq C$, there exists a step t_2 at which A_2 is merged with a member of \mathcal{C}_{t_2} , say B_2 . In the same way, we define recursively A_{i+1} and B_{i+1} from $A_i \not\subseteq C$ and $B_i \not\subseteq C$ that are two merged to form A_{i+1} , whenever $A_i \cup B_i \not\subseteq C$ for all $i \geq 1$. Now, the sequence $(A_i)_{i \geq 1}$ being strictly increasing and bounded by the finite set C according to set inclusion order, there must exist $j \geq 1$ such that $A_j = A_{j-1} \cup B_{j-1} = C$. Therefore, there exists in all cases a sequence $(A_i)_{1 \leq i \leq j}$ of members of \mathcal{C} , that verifies (α) and (β). \square

Theorem 3.4. Let D be a linkage function, C a nonempty subset of S and \mathcal{C} a D -linkage hierarchy.

- (i) $C \in \mathcal{C}$ if and only if C is closed for \mathcal{C} .
- (ii) C belongs to each D -linkage hierarchy if and only if C is D -min-closed.

Proof. (i) Assume that $C \in \mathcal{C}$. If $|C| = 1$ or $C = S$, then C is clearly closed for \mathcal{C} . Suppose now that $1 < |C| < |S|$. Let $t \geq 0$ be an arbitrary step of the HAC algorithm with linkage function D , when it generates the hierarchy \mathcal{C} . Moreover, denote as (X, Y) the pair which is chosen to be merged at step t . Suppose that $X \not\subseteq C$. Assume that $Y \not\subseteq C$ does not hold. As $X \not\subseteq C$, we have that C intersects $X \cup Y$. Now $C, X \cup Y \in \mathcal{C}$, so that $C \not\subseteq (X \cup Y)$, for otherwise $(X \cup Y) \subseteq C$ would imply $Y \not\subseteq C$. Consequently, $X \not\subseteq C \not\subseteq (X \cup Y)$ which shows that C intersects Y . It results that $C \subseteq Y$, since $Y \not\subseteq C$ does not hold and two elements of \mathcal{C} are

either disjoint or nested. But this implies that $X \not\subseteq C \subseteq Y$, which contradicts $X, Y \in \mathcal{C}_t$ since the elements of \mathcal{C}_t . We conclude that $Y \not\subseteq C$ must hold true. Thus $X \not\subseteq C$ implies that $Y \not\subseteq C$. Hence C is closed for \mathcal{C} .

Assume now that C is closed for \mathcal{C} . If $|C| = 1$ (resp. $C = S$), then C is formed at the first (resp. last step). Suppose that $1 < |C| < |S|$. Let $x \in C$ and $A = \{x\}$. Then $A \in \mathcal{C}_0 \subseteq \mathcal{C}$ and $A \not\subseteq C$. As C is closed for \mathcal{C} , it results that $C \in \mathcal{C}$ by Lemma 3.3.

(ii) Consider the property (\mathcal{P}) asserting that C belongs to each D -linkage hierarchy. By (i), this amounts to assert that C is closed for any D -linkage hierarchy. In other words, (\mathcal{P}) asserts that at each step t of any run of HAC algorithm that generates some D -linkage hierarchy, if X and Y are merged, then $X \not\subseteq C$ implies $Y \not\subseteq C$. Finally, this is equivalent to assert that at each step t of the HAC algorithm that generates some D -linkage hierarchy \mathcal{C} , if $(X, Y) \in \arg \min_{A, B \in \mathcal{C}_t} D(A, B)$ then $X \not\subseteq C$ implies $Y \not\subseteq C$. In other words, (\mathcal{P}) is equivalent to assert that C is D -min closed. \square

Example 3.5. Consider the linkage function D_1 defined in Example 2.2. It is easy to check that $A_1 = abcdef$ is D_1 -min-closed. In contrast, $abef$ and $cdef$ are not D_1 -min-closed. Indeed, at the step where the four subsets ab, cd, ef and gh are maximal, ef realizes the minimum of D_1 with both of ab and cd , so that neither of $abef$ and $cdef$ is D_1 -min-closed, since $ab \not\subseteq cdef$ and $cd \not\subseteq abef$.

It results from Theorem 3.4-(ii) that the collection of all D -min-closed subsets of S forms a hierarchy on S . Indeed, this collection coincides with the intersection of all D -hierarchies, which is therefore a hierarchy.

Let us now consider the index f_D defined on a D -linkage hierarchy \mathcal{C} , by $f_D(\{x\}) = D(\{x\}, \{x\}) = 0$ and $f_D(A \cup B) = D(A, B)$ if $A \cup B$ is a D -linkage cluster formed by merging A and B . Note that it may exist an inversion, *i.e.*, two D -linkage clusters A and B may exist in \mathcal{C} such that $A \not\subseteq B$ and $f_D(A) > f_D(B)$.

Bruynooghe has introduced the *reducibility property* in order to significantly reduce the complexity of the HAC algorithm, and has pointed out that this property ensures the absence of inversion in every D -linkage hierarchy [10]. A linkage function D is said to satisfy the reducibility property if for all nonempty pairwise disjoint subsets X, Y, Z of S , we have:

$$D(X, Y) \leq \min\{D(X, Z), D(Y, Z)\} \Rightarrow \min\{D(X, Z), D(Y, Z)\} \leq D(X \cup Y, Z).$$

Hereafter, we provide a proof of the absence of inversion when the property of reducibility is verified.

Proposition 3.6. *Let D be a linkage function. If D satisfies the reducibility property, then each D -linkage hierarchy is free from inversions.*

Proof. Let D be a linkage function that satisfies the reducibility property, and \mathcal{C} a D -linkage hierarchy. Let $X, Y \in \mathcal{C}$ such that $X \not\subseteq Y$. Our aim is to prove that $f_D(X) \leq f_D(Y)$.

If $|X| = 1$, then $f_D(X) = 0 \leq f_D(Y)$. Assume now that $|X| > 1$. Then $X \in \mathcal{C}$ is generated by merging two disjoint clusters, say A and $X \setminus A$. By theorem 3.4, Y is closed for \mathcal{C} since $Y \in \mathcal{C}$. Moreover, $A \not\subseteq X \not\subseteq Y$. Therefore, by Lemma 3.3, there exists a sequence of elements of \mathcal{C} , say A_1, A_2, \dots, A_k , such that $A_1 = A, A_k = Y$ and A_i is merged with $A'_i = A_{i+1} \setminus A_i$ for all $i \in \{1, \dots, k-1\}$. Given $2 \leq i \leq k$, consider $A_i = A_{i-1} \cup A'_{i-1}$. Since D satisfies the reducibility property, the following holds for all subset Z disjoint from A_i : $D(A_{i-1}, A'_{i-1}) \leq \min\{D(A_{i-1}, Z), D(A'_{i-1}, Z)\} \leq D(A_{i-1} \cup A'_{i-1}, Z)$. Applying these inequalities with $Z = A'_i$ which is disjoint from A_i , we deduce that for all $i \in [2, k]$:

$$f_D(A_i) = f_D(A_{i-1} \cup A'_{i-1}) = D(A_{i-1}, A'_{i-1}) \leq D(A_{i-1} \cup A'_{i-1}, A'_i) = D(A_i, A'_i) = f_D(A_{i+1})$$

Therefore $f_D(X) = f_D(A_2) \leq \dots \leq f_D(A_k) = f_D(Y)$. \square

Table 1 below recalls the definitions of most standard dissimilarity-based linkage functions, where A and B stand for two disjoint subsets of S . In Table 1 below, it can be noticed that the WPGMC linkage is sometimes called also median linkage [15, 19], whereas in this text, we define the median linkage of two clusters A and B as the mediane of the set $d(A, B)$, where d denotes an arbitrary dissimilarity on S .

Linkage function D	$D(A, B)$
Single linkage	$\min d(A, B)$
Complete linkage	$\max d(A, B)$
Median linkage	$\text{med } d(A, B)$
Average linkage (UPGMA)	$\frac{1}{ A B } \sum_{a \in A} \sum_{b \in B} d(a, b)$
Weighted average linkage (WPGMA)	$[D(A_1, B) + D(A_2, B)]/2$, where $A = A_1 \cup A_2$ is formed by merging A_1 and A_2
Hausdorff linkage	$\max \left\{ \max_{a \in A} \min d(a, B), \max_{b \in B} \min d(b, A) \right\}$
Minimax linkage [2]	$\min_{x \in A \cup B} \max d(x, A \cup B)$
Ward linkage	$D_{\text{Ward}}(A, B) = \frac{w_A w_B}{w_A + w_B} d^2(c_A, c_B)$, where w_X denotes the weight of X and c_X the centroid of X
Centroid linkage (UPGMC)	$D_{\text{CeL}}(A, B) = d(c_A, c_B)$
Weighted centroid linkage (WPGMC)	$D_{\text{WC}}(A_1 \cup A_2, B) =$ $\frac{1}{2} D_{\text{WC}}(A_1, B) + \frac{1}{2} D_{\text{WC}}(A_2, B) - \frac{1}{4} D_{\text{WC}}(A_1, A_2)$

Table 1: d -based standard linkage functions, where d denotes a dissimilarity on S .

In next Corollary 3.8, we show that standard linkage functions that will be considered in Theorem 5.3 satisfy the reducibility property.

Proposition 3.7. *Let D be one of the following linkage functions: Single linkage, Complete linkage, Median linkage, Average linkage, Weighted Average linkage, Hausdorff linkage and Minimax linkage. If X, Y, Z are arbitrary nonempty subsets of S that are pairwise disjoint, then $D(X \cup Y, Z) \geq \min\{D(X, Z), D(Y, Z)\}$.*

Proof. Let d be a dissimilarity on S and D a d -based linkage function among those listed in the statement of the proposition. We consider arbitrary nonempty subsets $X, Y, Z \subseteq S$ of S that are pairwise disjoint. We aim to prove that $D(X \cup Y, Z) \geq \min\{D(X, Z), D(Y, Z)\}$.

Single linkage. $D(X \cup Y, Z) = \min d(X \cup Y, Z) = \min\{D(X, Z), D(Y, Z)\}$.

Complete linkage. $D(X \cup Y, Z) = \max d(X \cup Y, Z) = \max\{D(X, Z), D(Y, Z)\}$.

Median linkage. The linkage function D is defined by $D(X \cup Y, Z) = \text{med } d(X \cup Y, Z)$. Thus

$$\begin{aligned} D(X \cup Y, Z) &\geq \min\{\text{med } d(X, Z), \text{med } d(Y, Z)\} \\ &= \min\{D(X, Z), D(Y, Z)\}. \end{aligned}$$

Average linkage. From the definition of the average linkage, we deduce:

$$\begin{aligned} D(X \cup Y, Z) &= \frac{1}{(|X| + |Y|)|Z|} (|X||Z| D(X, Z) + |Y||Z| D(Y, Z)) \\ &\geq \frac{|X||Z| + |Y||Z|}{(|X| + |Y|)|Z|} \min \{D(X, Z), D(Y, Z)\} = \min \{D(X, Z), D(Y, Z)\} \end{aligned}$$

Weighted average linkage. $D(X \cup Y, Z) = \frac{1}{2}[D(X, Z) + D(Y, Z)] \geq \min \{D(X, Z), D(Y, Z)\}$.

Hausdorff linkage. Let D be the Hausdorff link. We have:

$$\begin{aligned} D(X \cup Y, Z) &= \max \left[\max_{u \in X \cup Y} \min d(u, Z), \max_{z \in Z} \min d(z, X \cup Y) \right], \\ D(X, Z) &= \max \left[\max_{x \in X} \min d(x, Z), \max_{z \in Z} \min d(z, X) \right], \\ D(Y, Z) &= \max \left[\max_{y \in Y} \min d(y, Z), \max_{z \in Z} \min d(z, Y) \right]. \end{aligned}$$

On the one hand,

$$\max \left\{ \max_{x \in X} \min d(x, Z), \max_{y \in Y} \min d(y, Z) \right\} \leq \max_{u \in X \cup Y} \min d(u, Z) \leq D(X \cup Y, Z),$$

and, on the other hand,

$$\max \left\{ \max_{z \in Z} \min d(z, X), \max_{z \in Z} \min d(z, Y) \right\} = \max_{z \in Z} \min d(z, X \cup Y) \leq D(X \cup Y, Z).$$

It results that $\min \{D(X, Z), D(Y, Z)\} \leq D(X \cup Y, Z)$.

Minimax linkage. In this case, $D(X \cup Y, Z) = \min_{u \in X \cup Y \cup Z} \max d(u, X \cup Y \cup Z)$. Furthermore,

$$D(X, Z) = \min_{u \in X \cup Z} \max d(u, X \cup Z) \leq \min_{u \in X \cup Y \cup Z} \max d(u, X \cup Y \cup Z).$$

Similarly, we have $D(Y, Z) \leq \min_{u \in Y \cup Z} \max d(u, X \cup Y \cup Z)$, which implies:

$$\begin{aligned} \min \{D(X, Z), D(Y, Z)\} &\leq \min \left\{ \min_{u \in X \cup Z} \max d(u, X \cup Y \cup Z), \min_{u \in Y \cup Z} \max d(u, X \cup Y \cup Z) \right\} \\ &= \min_{u \in X \cup Y \cup Z} \max d(u, X \cup Y \cup Z) = D(X \cup Y, Z). \end{aligned}$$

□

As a direct consequence of Proposition 3.7, we deduce the Corollary 3.8. The standard linkage functions mentioned in Corollary 3.8, will be considered in Theorem 5.3.

Corollary 3.8. *The following linkage functions satisfy the reducibility property: Single linkage, Complete linkage, Median linkage, Average linkage, Weighted Average linkage, Hausdorff linkage and Minimax linkage.*

4 A sufficient condition for the Apresjan hierarchy to be included in each hierarchy generated by the HAC method

In this section, given a dissimilarity d on S , we determine a sufficient general condition for a d -based linkage function D to ensure that each D -linkage hierarchy contains the Apresjan hierarchy of d .

We now investigate the properties of clusters that are created by using common dissimilarity-based linkage functions, w.r.t. Apresjan clusters.

Remark 4.1. As a consequence of Theorem 3.4, we deduce that if D is a d -based linkage, then the Apresjan hierarchy of d is contained in each D -hierarchy if and only if each d -Apresjan cluster is D -min-closed.

By extending characterization (4) to any linkage function D , we obtain the so-called D -Apresjan subsets that are defined similarly to their analogues for dissimilarities.

Definition 4.2 (D -Apresjan subset). Let D be a linkage function on S . A nonempty subset A of S will be said to be D -Apresjan if the following two conditions are verified:

1. for all $X \not\subseteq A$, all $Y \subseteq A \setminus X$ and all $Z \subseteq S \setminus A$, we have $D(X, Y) < D(X, Z)$,
2. for all $x \in A$ and all $Y \subseteq S \setminus A$, we have $D(\{x\}, \{x\}) < D(\{x\}, Y)$.

Proposition 4.3. Let D be a linkage function based on a proper dissimilarity on S . The set of D -Apresjan clusters is a hierarchy on S .

Proof. Clearly, S and the singletons of S are D -Apresjan. Assume that subsets A and B are D -Apresjan and intersect properly. Thus, each of the sets $A \setminus B$, $A \cap B$ and $B \setminus A$ is nonempty. Since A is D -Apresjan, $D(A \setminus B, A \cap B) < D(A \cap B, B \setminus A)$. Since B is D -Apresjan, we have $D(A \cap B, B \setminus A) < D(A \cap B, A \setminus B)$, which is contradictory. Therefore, there is no D -Apresjan cluster that properly intersects another one. Thus, the set of D -Apresjan clusters is a hierarchy on S . \square

In what follows, the set of D -Apresjan clusters will then be called the D -Apresjan hierarchy or the Apresjan hierarchy of D . Beyond the relation mentioned in Remark 4.4, between D -Apresjan subsets and Apresjan clusters of the dissimilarity underlying a linkage function D , we show in Proposition 4.7 that every D -Apresjan subset is D -min-closed.

Remark 4.4. If D is a d -based linkage function, then every D -Apresjan subset of S is an Apresjan cluster of d . The converse is not true, as shown by Counter-example 4.5 below. In other words, the Apresjan hierarchy of D is included in the Apresjan hierarchy of d , and this inclusion is strict for some particular datasets.

Counter-example 4.5. Consider the dissimilarity d_1 and the d_1 -based linkage function D_1 both defined in Example 2.2. Subset $A_1 = abcdef$ is clearly an Apresjan cluster of d_1 . However, A_1 is not D_1 -Apresjan since, for instance, $D_1(ab, gh) = D_1(ab, cd) = 5$.

As observed earlier, at any step t of the HAC algorithm that generates \mathcal{C} , the collection \mathcal{C}_t partitions the ground set S . For any nonempty subset $A \subseteq S$ of S , let us denote $\mathcal{T}(A, \mathcal{C}_t) = \{X \in \mathcal{C}_t : X \cap A \neq \emptyset\}$, i.e. the collection of D -linkage clusters that are maximal at step t and intersect A^\dagger . The following lemma is instrumental for the proof of Proposition 4.7.

Lemma 4.6. Let D be a linkage function on S and A a nonempty subset of S . If A is D -Apresjan, then at each step t of the HAC algorithm, the collection $\mathcal{T}(A, \mathcal{C}_t)$ either partitions A or is reduced to a single member containing A .

Proof. Let A be a D -Apresjan subset of S , and denote as $P_A(t)$ the property: “ $\mathcal{T}(A, \mathcal{C}_t)$ either partitions A or is reduced to a single element containing A .” Our aim is to prove that $P_A(t)$ is true for all steps t of the HAC algorithm.

- If $A = \{a\}$, then $\mathcal{T}(A, \mathcal{C}_t) = \{X_a\}$ where X_a is the unique element of \mathcal{C}_t containing a : such an element X_a exists since \mathcal{C}_t is a partition of S for all steps t of the HAC algorithm. If $A = S$, then $\mathcal{T}(A, \mathcal{C}_t) = \mathcal{C}_t$ which partitions $S = A$ for all steps t .

- Assume now that $1 < |A| < |S|$. We proceed by induction on the step t of the HAC-algorithm, with $0 \leq t \leq n-1$. At step $t = 0$, $\mathcal{C}_0 = \{\{x\} : x \in S\}$ so that $\mathcal{T}(A, \mathcal{C}_0) = \{\{x\} : x \in A\}$ partitions A . Let us suppose

[†]Considering the hypergraph $H = (S, \mathcal{C}_t)$ of vertex set S and hyperedge set \mathcal{C}_t , it may noticed that the set $\mathcal{T}(A, \mathcal{C}_t)$ is the maximum collection $\mathcal{E} \subseteq \mathcal{C}_t$ of hyperedges of H such that A is a transversal of the partial hypergraph (S, \mathcal{E}) .

that $P_A(t)$ holds up to some step t . If the collection $\mathcal{T}(A, \mathcal{C}_t)$ is reduced to a single member X containing A , then for all $t' \geq t$, $\mathcal{T}(A, \mathcal{C}_{t'})$ is clearly reduced to a single member Y such that $A \subseteq X \subseteq Y \in \mathcal{C}_{t'}$, so that $P_A(t')$ holds. If the collection $\mathcal{T}(A, \mathcal{C}_t)$ is not reduced to a single member containing A , let t_1 denote the first step for which there exists $X_1 \in \mathcal{T}(A, \mathcal{C}_{t'})$, which is merged with a member Y_1 of $\mathcal{C}_{t'}$ for all $t \leq t' \leq t_1$. One may observe that $\mathcal{T}(A, \mathcal{C}_{t'}) = \mathcal{T}(A, \mathcal{C}_t)$ for all $t \leq t' < t_1$. Moreover, $\mathcal{T}(A, \mathcal{C}_t)$ being a partition of A , which is not reduced to $\{A\}$, there exists $Z \in \mathcal{T}(A, \mathcal{C}_t) \setminus \{X_1\}$. Let us assume that Y_1 does not intersect A , i.e. $Y_1 \subseteq S \setminus A$. Then, $D(X_1, Z) < D(X_1, Y_1)$ since A is D -Apresjan, which contradicts the fact that X_1 and Y_1 are merged at step t_1 . Therefore, Y_1 intersects A , and consequently $Y_1 \in \mathcal{T}(A, \mathcal{C}_t)$ which is a partition of A , distinct from $\{A\}$. Thus we have $Y_1 \not\subseteq A$ and $X_1 \cup Y_1 \in \mathcal{C}_{t_1}$. This implies that $\mathcal{T}(A, \mathcal{C}_{t_1}) = \mathcal{T}(A, \mathcal{C}_t) \setminus \{X_1, Y_1\} \cup \{X_1 \cup Y_1\}$ is a partition of A , eventually reduced to $\{A\}$. Therefore $P_A(t_1)$ is true, with $t_1 > t$. By the argument of induction, we conclude that $P_A(t)$ is true for all step t of the HAC algorithm. \square

Proposition 4.7 and Counter-example 4.8 show that the condition defining of a D -Apresjan subset is more restrictive than the one defining a D -min-closed subset.

Proposition 4.7. *For all linkage functions D on S , every D -Apresjan subset of S is D -min-closed.*

Proof. Let A be a D -Apresjan subset of S . It is clear that A is D -min-closed if $|A| = 1$ or $A = S$. Assume now that $1 < |A| < |S|$. First, observe that there exists a pair $\{X_0, Y_0\} \subseteq \mathcal{C}_t$ that realizes the minimum of D at step t of the HAC algorithm, with $X_0 \not\subseteq A$. Therefore, $X_0 \in \mathcal{T}(A, \mathcal{C}_t)$ which is therefore not reduced to a single member containing A . Since A is D -Apresjan, by Lemma 4.6, $\mathcal{T}(A, \mathcal{C}_t)$ partitions A . Moreover, again because A is D -Apresjan, we have, $D(X_0, Y) < D(X_0, Z)$ for all $Y \in \mathcal{T}(A, \mathcal{C}_t) \setminus \{X_0\}$ and all $Z \in \mathcal{C}_t \setminus \mathcal{T}(A, \mathcal{C}_t)$. It follows that Y_0 belongs necessarily to $\mathcal{T}(A, \mathcal{C}_t)$. Therefore, Y_0 is strictly included in A , showing that A is D -min-closed. \square

Counter-example 4.8. *Consider the linkage function D_1 defined in Example 2.2. As respectively observed in Example 3.5 and Counter-example 4.5, the subset $A_1 = abcdef$ is D_1 -min-closed but not D_1 -Apresjan. This shows that the converse of Proposition 4.7 is not true.*

According to Theorem 3.4 and Proposition 4.7, we have the following corollary.

Corollary 4.9. *For all linkage functions D on S , every D -Apresjan subset of S belongs to each D -linkage hierarchy. In other words, the Apresjan hierarchy of D is contained into each D -linkage hierarchy.*

Remark that next Proposition 4.10 implies that if the linkage function D satisfies the reducibility property, then no D -Apresjan subset should be removed by pruning any D -linkage hierarchy.

Proposition 4.10. *Let D be a linkage function that satisfies the reducibility property and C be a D -Apresjan subset of S . For each D -linkage cluster C' such that $C \not\subseteq C'$, we have $f_D(C) < f_D(C')$. As a consequence, the Apresjan hierarchy of D is indexed by f_D .*

Proof. Let C be a D -Apresjan subset of S and \mathcal{C} a D -linkage hierarchy. By Corollary 4.9, C belongs to \mathcal{C} . If $C = S$, it obviously satisfies the condition of the proposition. If $C = \{x\}$ for some $x \in S$, then, since C is D -Apresjan, we have, for all D -linkage cluster $Y \subseteq S \setminus \{x\}$ merged with $\{x\}$:

$$f_D(C) = f_D(\{x\}) = D(\{x\}, \{x\}) < D(\{x\}, Y) = f_D(\{x\} \cup Y). \quad (5)$$

Let us consider $C' \in \mathcal{C}$ such that $C = \{x\} \not\subseteq C'$. Notice that $\{x\} \cup Y$ is the smallest cluster of \mathcal{C} that contains $C = \{x\}$. Since C' is a D -linkage cluster that contains C , it results that $\{x\} \cup Y \subseteq C'$. Now, D satisfies the reducibility property, thus $f_D(\{x\} \cup Y) \leq f_D(C')$ by Proposition 3.6. Using the strict inequality (5), it results that $f_D(C) < f_D(C')$.

Assume now $1 < |C| < |S|$. Let X and Y be the two D -linkage clusters merged to form C . Denote as Z the D -linkage cluster which is merged to C . As C is D -Apresjan, and $Z \subseteq S \setminus C$, we have: $D(X, Y) < \min\{D(X, Z), D(Y, Z)\}$, so that, by the reducibility property,

$$f_D(C) = D(X, Y) < \min\{D(X, Z), D(Y, Z)\} \leq D(X \cup Y, Z) = f_D(C \cup Z). \quad (6)$$

Let $C' \in \mathcal{C}$ such that $C \not\subseteq C'$. Since $C \cup Z$ is the smallest cluster of \mathcal{C} containing C , and $C' \in \mathcal{C}$ contains C , it results that $C \cup Z \subseteq C'$. Thus, $f_D(C \cup Z) \leq f_D(C')$, by Proposition 3.6 as D satisfies the reducibility property. Using the strict inequality (6), it results that $f_D(C) < f_D(C')$. \square

Next Theorem 4.11 is a direct consequence of Corollary 4.9 and Proposition 4.10.

Theorem 4.11. *Let D a d -based linkage function. If each Apresjan cluster of d is D -Apresjan, then the Apresjan hierarchy of d is contained in every D -linkage hierarchy. If, in addition, D satisfies the reducibility property, then the Apresjan hierarchy of d is indexed by f_D .*

Remark 4.12. According to Remark 4.4, the first part of Theorem 4.11 is equivalent to assert that if the Apresjan hierarchy of d coincides with the Apresjan hierarchy of D , then the Apresjan hierarchy of d is contained in every D -linkage hierarchy. Note that this condition is sufficient but not necessary: as observed in Example 2.3, the subset $A_1 = abcdef$ is d_1 -Apresjan, but is not D_1 -Apresjan (cf. Counter-example 4.5). However the Apresjan hierarchy of d_1 is included in each D_1 -linkage hierarchy (cf. Example 2.3).

5 Standard linkage functions for which the Apresjan hierarchy is included in each hierarchy generated by the HAC method

In this section, given a dissimilarity d on S , we provide, on the one hand, a list of standard d -based linkage functions whose associated HAC-linkage hierarchies contain the Apresjan hierarchy of d , and, on the other hand, counter-examples that illustrate that for some dissimilarities d and some standard d -based linkage functions D , the obtained D -linkage hierarchies may fail to contain the Apresjan hierarchy of d . The following two lemmas are instrumental for the proof of Theorem 5.3.

Lemma 5.1. *Let \mathcal{P} be a partition of a nonempty finite subset U of \mathbb{R} . Then we have*

$$\min_{V \in \mathcal{P}} \text{med } V \leq \text{med } U \leq \max_{V \in \mathcal{P}} \text{med } V.$$

Proof. Recall that either $\text{med } U \in U$ or $\text{med } U$ is the middle of two consecutive elements of U . Assume that $\text{med } U < \min_{V \in \mathcal{P}} \text{med } V$. Let V be an arbitrary class of \mathcal{P} . Then, there would be more elements of V , greater or equal than $\text{med } V$, than there are elements of V lower or equal than $\text{med } V$. Thus, the elements of U which are greater or equal than $\text{med } U$, outnumber the elements of U lower or equal than $\text{med } U$. This contradicts the definition of $\text{med } U$. Therefore, $\min_{V \in \mathcal{P}} \text{med } V \leq \text{med } U$. The other inequality proves in a similar way. \square

The proof of Lemma 5.2 is straightforward.

Lemma 5.2. *Let d be a dissimilarity on S and $X, Y, Z \subseteq S$ nonempty subsets of S . If $d(u, v) < d(u, w)$ for all $(u, v, w) \in X \times Y \times Z$, then for all $x \in X$, we have:*

$$\min d(x, Y) \leq \frac{1}{|Y|} \sum_{y \in Y} d(x, y) \leq \max d(x, Y) < \min d(x, Z).$$

Theorem 5.3. *Let D one of the following d -based linkage functions: Single linkage, Complete linkage, Median linkage, Average linkage, Weighted Average linkage, Hausdorff linkage and Minimax linkage. Then the Apresjan hierarchy of d is contained in each D -linkage hierarchy, and is indexed by f_D .*

Proof. Let D a d -based linkage function among those listed in the statement of the theorem. According to Theorem 4.11, it is sufficient to show, on the one hand, that each Apresjan cluster of d is D -Apresjan, and, on the other hand, that D satisfies the reducibility property. By Corollary 3.8, D satisfies the reducibility property, so that we are left to prove that each Apresjan cluster of d is D -Apresjan. For this, we consider,

w.l.o.g., an Apresjan cluster of d , say A , such that $1 < |A| < |S|$, nonempty subsets $X, Y, Z \subseteq S$ of S such that $X, Y \not\subseteq A$, $X \cap Y = \emptyset$, $Z \subseteq S \setminus A$, and aim to show the strict inequality $D(X, Y) < D(X, Z)$.

Single linkage. $D(X, Y) = \min d(X, Y)$. The subset A being an Apresjan cluster of d , we deduce from (4) that $d(u, v) < d(u, w)$ for all $(u, v, w) \in X \times Y \times Z$. We then deduce from Lemma 5.2 that

$$D(X, Y) = \min_{x \in X} \min d(x, Y) < \min_{x \in X} \min d(x, Z) = D(X, Z).$$

Complete linkage. $D(X, Y) = \max d(X, Y)$. We similarly deduce from (4) and Lemma 5.2 that

$$D(X, Y) = \max_{x \in X} \max d(x, Y) < \max_{x \in X} \min d(x, Z) \leq \max_{x \in X} \max d(x, Z) = D(X, Z).$$

Median linkage. $D(X, Y) = \text{med } d(X, Y)$. We proceed by induction on $|Y| \geq 1$. Assume first that $|Y| = 1$ and let $Y = \{y\}$. Let $X = \{x_1, \dots, x_n\}$ with $n = |X|$ and $d(x_1, y) \leq \dots \leq d(x_n, y)$. Furthermore, we sort all the elements (u, z) of $X \times Z$ in the following way: $d(u_1, z_1) \leq \dots \leq d(u_m, z_m)$, where $m = |d(X, Z)|$.

Since A is an Apresjan cluster of d , note that for each integer $1 \leq i \leq n$ we have $d(x_i, y) < \min_{z \in Z} d(x_i, z)$. Using the fact that $d(x_i, y) \leq d(x_{i+1}, y)$, it results that for all $j \in \{1, \dots, n\}$,

$$d(x_j, y) < \min_{i=j}^n d(x_i, Z). \quad (7)$$

Assume first that $n = |X|$ is even. Thus $\text{med } d(X, y) = \frac{d(x_k, y) + d(x_{k+1}, y)}{2}$ with $k = n/2$. Taking into account that $d(X, Z)$ is a multiset, we have $|d(X, Z)| = \sum_{z \in Z} |d(X, z)| = 2k|Z| = 2\ell$, with $\ell = k|Z|$. Then

$\text{med } d(X, Z) = \frac{d(u_\ell, z_\ell) + d(u_{\ell+1}, z_{\ell+1})}{2}$. Observe that $|\bigcup_{i \geq k} d(x_i, Z)| = (k+1)|Z| > k|Z| = \ell$, so that

$\min_{i=k}^n d(x_i, Z) \leq d(u_\ell, z_\ell)$. From (7), we deduce that $d(x_k, y) < d(u_\ell, z_\ell)$.

Similarly, $|\bigcup_{i=k+1}^n d(x_i, Z)| = k|Z| = \ell$, so that $\min_{i=k+1}^n d(x_i, Z) \leq d(u_{\ell+1}, z_{\ell+1})$. From (7), we deduce that $d(x_{k+1}, y) < d(u_{\ell+1}, z_{\ell+1})$. Therefore:

$$\text{med } d(X, y) = \frac{d(x_k, y) + d(x_{k+1}, y)}{2} < \frac{d(u_\ell, z_\ell) + d(u_{\ell+1}, z_{\ell+1})}{2} = \text{med } d(X, Z).$$

Assume that $n = |X|$ is odd. Thus $|d(X, y)|$ is odd and $\text{med } d(X, y) = d(x_{k+1}, y)$ with $k = (n-1)/2$. Furthermore, $|d(X, Z)| = \sum_{z \in Z} |d(X, z)| = (2k+1)|Z|$ which is even if $|Z|$ is even, and odd otherwise.

If $|Z|$ is even, then $\text{med } d(X, Z) = \frac{d(u_\ell, z_\ell) + d(u_{\ell+1}, z_{\ell+1})}{2}$ with $\ell = \frac{|d(X, Z)|}{2} = k|Z| + \frac{|Z|}{2}$.

Since $n = 2k+1$, we have $|\bigcup_{i \geq k+1} d(x_i, Z)| = (k+1)|Z| > k|Z| + \frac{|Z|}{2} = \ell$, so that $\min_{i=k+1}^n d(x_i, Z) \leq d(u_\ell, z_\ell)$.

From (7), we deduce that $d(x_{k+1}, y) < d(u_\ell, z_\ell)$. Moreover, $d(u_\ell, z_\ell) \leq d(u_{\ell+1}, z_{\ell+1})$. Therefore, we deduce $\text{med } d(X, y) = d(x_{k+1}, y) < [d(u_\ell, z_\ell) + d(u_{\ell+1}, z_{\ell+1})]/2 = \text{med } d(X, Z)$.

If $|Z|$ is odd, then $\text{med } d(X, Z) = d(u_{\ell+1}, z_{\ell+1})$ with $\ell = \frac{|d(X, Z)| - 1}{2} = k|Z| + \frac{|Z| - 1}{2}$.

Observe again that $|\bigcup_{i \geq k+1} d(x_i, Z)| = (k+1)|Z|$. Now $(k+1)|Z| \geq \ell + 1 = k|Z| + \frac{|Z| + 1}{2}$, for $|Z| \geq 1$.

Thus $\min_{i=k+1}^n d(x_i, Z) \leq d(u_{\ell+1}, z_{\ell+1})$. By (7), $\text{med } d(X, y) = d(x_{k+1}, y) < d(u_{\ell+1}, z_{\ell+1}) = \text{med } d(X, Z)$.

We conclude that, in all cases, $D(X, y) = \text{med } d(X, y) < \text{med } d(X, Z) = D(X, Z)$.

Assume now $|Y| > 1$. We assume that $D(X, A) = \text{med } d(X, A) < \text{med } d(X, Z) = D(X, Z)$ for all $A \not\subseteq Y$, and aim to prove that $D(X, Y) < D(X, Z)$. By applying Lemma 5.1 and the induction hypothesis, it results

that, for all $y \in Y$, we have:

$$D(X, Y) = \text{med } d(X, Y) \leq \max \{ \text{med } d(X, Y \setminus \{y\}), \text{med } d(X, \{y\}) \} < \text{med } d(X, Z) = D(X, Z).$$

Average linkage. $D(X, Y) = \frac{1}{|X||Y|} \sum_{x \in X} \sum_{y \in Y} d(x, y)$. Let x_0 be an arbitrary element of X . Then $d(x_0, y) < d(x_0, z)$ for all $(y, z) \in Y \times Z$. Therefore

$$\sum_{y \in Y} |Z| d(x_0, y) = \sum_{y \in Y} \sum_{z \in Z} d(x_0, y) < \sum_{z \in Z} \sum_{y \in Y} d(x_0, z) = \sum_{z \in Z} |Y| d(x_0, z),$$

which implies $\frac{1}{|Y|} \sum_{y \in Y} d(x_0, y) < \frac{1}{|Z|} \sum_{z \in Z} d(x_0, z)$. Finally,

$$D(X, Y) = \frac{1}{|X||Y|} \sum_{x \in X} \sum_{y \in Y} d(x, y) < \frac{1}{|X||Z|} \sum_{x \in X} \sum_{z \in Z} d(x, z) = D(X, Z).$$

Weighted average linkage. $D(X, Y) = \frac{1}{2} [D(X_1, Y) + D(X_2, Y)]$ where $X = X_1 \cup X_2$ was formed by merging X_1 and X_2 . The proof is by induction on iteration number $t = 0, 1, \dots, n-1$ of the HAC algorithm. If $t = 0$, there is little to prove, since, in this case, X, Y, Z are singletons. Assume then that $t \geq 1$. By the induction hypothesis, we are left to prove $D(X, Y) < D(X, Z)$ only if (exactly) one of the clusters X, Y or Z is formed at iteration t .

Assume first that at iteration t , $X = X_1 \cup X_2$ is formed by merging X_1 and X_2 . By the induction hypothesis,

$$D(X, Y) = \frac{1}{2} [D(X_1, Y) + D(X_2, Y)] < \frac{1}{2} [D(X_1, Z) + D(X_2, Z)] = D(X, Z).$$

Assume now that Y is formed by merging Y_1 and Y_2 at iteration t . Then, by induction,

$$D(X, Y) = \frac{1}{2} [D(X, Y_1) + D(X, Y_2)] < \frac{1}{2} [D(X, Z) + D(X, Z)] = D(X, Z).$$

Assume finally that Z_1 and Z_2 are the clusters merged to form Z at iteration t . By the induction hypothesis, $D(X, Y) < D(X, Z_1)$ and $D(X, Y) < D(X, Z_2)$, so that

$$D(X, Y) < \frac{1}{2} [D(X, Z_1) + D(X, Z_2)] = D(X, Z).$$

Therefore in either case, $D(X, Y) < D(X, Z)$, as required.

Hausdorff linkage. $D(X, Y) = \max [\max_{x \in X} \min d(x, Y), \max_{y \in Y} \min d(y, X)]$. By Lemma 5.2, we have, on the one hand, $\min d(x, Y) < \min d(x, Z)$ for all $x \in X$, so that

$$\max_{x \in X} \min d(x, Y) < \max_{x \in X} \min d(x, Z). \quad (8)$$

On the other hand, assume that $\max_{y \in Y} \min d(y, X) = d(x_m, y_m)$ for some $(x_m, y_m) \in X \times Y$. Then

$$\max_{y \in Y} \min d(y, X) = d(x_m, y_m) < \min d(x_m, Z) \leq \max_{x \in X} \min d(x, Z). \quad (9)$$

Therefore, combining (8) and (9), we have:

$$\begin{aligned} D(X, Y) &= \max [\max_{x \in X} \min d(x, Y), \max_{y \in Y} \min d(y, X)] < \max_{x \in X} \min d(x, Z) \\ &\leq \max [\max_{x \in X} \min d(x, Z), \max_{z \in Z} \min d(z, X)] = D(X, Z), \end{aligned}$$

as required.

Minimax linkage. $D(X, Y) = \min_{u \in X \cup Y} \max d(u, X \cup Y)$. If $u \in X$, then as $d(u, x) < d(u, z)$ for all $(x, z) \in X \times Z$, we have $\max d(u, X \cup Z) = \max d(u, Z) = d(u, z_u)$ for some $z_u \in Z$.

If $u \in Z$, then $\max d(u, X \cup Z) \geq \max d(u, X) = d(u, x_u)$ for some $x_u \in X$.

Thus, for all $u \in X \cup Z$, we have $\max d(u, X \cup Z) = d(\tilde{x}_u, \tilde{z}_u)$ for some $\tilde{x}_u \in X, \tilde{z}_u \in Z$. It follows that

$$D(X, Z) = \min_{u \in X \cup Z} \max d(u, X \cup Z) = d(\tilde{x}, \tilde{z}),$$

for some $\tilde{x} \in X$ and $\tilde{z} \in Z$. Therefore,

$$D(X, Y) \leq \min_{x \in X} \max d(x, X \cup Y) \leq \max d(\tilde{x}, X \cup Y) < d(\tilde{x}, \tilde{z}) = D(X, Z).$$

□

Remark 5.4. Let d be a dissimilarity on S and D one of the following d -based linkage functions: Ward linkage, Centroid linkage and Weighted-centroid linkage. Then, a D -linkage hierarchy may fail to contain the Apresjan hierarchy of d , as shown by counter-examples below.

Counter-example 5.5 (A Ward-linkage hierarchy may not contain the Apresjan hierarchy). The Ward linkage based on the Euclidean distance d is defined by

$$D_{\text{Ward}}(A, B) = \frac{w_A w_B}{w_A + w_B} d^2(c_A, c_B),$$

where w_X denotes the weight of X (e.g. the number of objects in X if each object has a unit weight) and c_X the centroid of X . The following example of dataset shows that a D_{Ward} -linkage hierarchy may fail to contain the Apresjan hierarchy of d . Indeed, let $S = X \cup Y \cup Z$ where X, Y, Z are the one-dimensional sets defined by $X = \{4\}$, $Y = \{0.1, 0.2\}$ and $Z = \{8\}$. Let d be the Euclidean distance on S . It can be easily observed that $C = X \cup Y$ is an Apresjan cluster of d since,

$$\max_{x, y \in C} d(x, y) = 3.9 < \min_{x \in C, z \notin C} d(x, z) = 4.$$

However, $C = X \cup Y$ cannot belong to a D_{Ward} -linkage hierarchy since, for the unit weight,

$$D_{\text{Ward}}(X, Y) \approx 9.8 > D_{\text{Ward}}(X, Z) = 8,$$

which shows that X and Y cannot be merged before X and Z .

Counter-example 5.6 (A centroid-linkage hierarchy may not contain the Apresjan hierarchy). The centroid linkage based on (Euclidean distance) d is defined by

$$D_{\text{CeL}}(A, B) = d(c_A, c_B),$$

where c_X denotes the centroid of X . The following example of dataset shows that a D_{CeL} -linkage hierarchy may fail to contain the Apresjan hierarchy of d . Indeed, let $S = X \cup Y \cup Z$ where X, Y, Z are the two-

dimensional sets defined by $X = \left\{ \begin{pmatrix} 2 \\ 1 \end{pmatrix} \right\}$, $Y = \left\{ \begin{pmatrix} 1.15 \\ 1 \end{pmatrix} \right\}$ and $Z = \left\{ z_1 = \begin{pmatrix} 2.8 \\ 0.65 \end{pmatrix}, z_2 = \begin{pmatrix} 2.8 \\ 1.35 \end{pmatrix} \right\}$. Let d be

the Euclidean distance on S . It can be easily observed that $C = X \cup Y$ is an Apresjan cluster of d since,

$$\max_{x, y \in C} d(x, y) = 0.85 < \min_{x \in C, z \notin C} d(x, z) \approx 0.87.$$

However, $C = X \cup Y$ cannot belong to a D_{CeL} -linkage hierarchy since, on the one hand,

$$D_{\text{CeL}}(z_1, z_2) = 0.7 < D_{\text{CeL}}(X, Y) = 0.85$$

and, on the other hand, $D_{\text{CeL}}(X, Y) = 0.85 > D_{\text{CeL}}(X, Z) = 0.8$, which shows that, when using linkage function D_{CeL} , X and Y cannot be merged before X and Z .

Counter-example 5.7 (A weighted-centroid-linkage hierarchy may not contain the Apresjan hierarchy). The *weighted-centroid linkage*, also known as median method (in the manner of Apollonius’s theorem) or weighted pair group method using centroids (WPGMC), based on (Euclidean distance) d is defined by

$$D_{\text{WC}}(A_1 \cup A_2, B) = \frac{1}{2} D_{\text{WC}}(A_1, B) + \frac{1}{2} D_{\text{WC}}(A_2, B) - \frac{1}{4} D_{\text{WC}}(A_1, A_2),$$

with, in addition, $D_{\text{WC}}(X, Y) = D_{\text{WC}}(Y, X)$ and $D_{\text{WC}}(\{a\}, \{b\}) = d(a, b)$. The following example shows that a D_{WC} -linkage hierarchy may fail to contain the Apresjan hierarchy of d .

As was defined in the previous example given for the centroid linkage, consider again the set $S = X \cup Y \cup Z$ endowed with the Euclidean distance d on S . It was observed that subset $C = X \cup Y$ is an Apresjan cluster of d . However, $C = X \cup Y$ cannot belong to a D_{WC} -linkage hierarchy since, on the one hand,

$$\arg \min_{(a,b) \in S^2, a \neq b} d(a, b) = \{(z_1, z_2)\}$$

and, on the other hand, $D_{\text{WC}}(X, Y) = 0.85 > D_{\text{WC}}(X, Z) \approx 0.7$, which shows that X and Y cannot be merged before X and Z .

6 Final remarks

We have investigated both arbitrary dissimilarity-based and standard linkage functions D in view of identifying those whose D -linkage hierarchies contain the Apresjan hierarchy. First, given an arbitrary linkage function D , we have provided a necessary and sufficient condition for a given nonempty subset of the ground set S to be formed as a cluster of each D -linkage hierarchy. Then, for an arbitrary dissimilarity-based linkage function D , we proposed a sufficient condition for each (possibly pruned) D -linkage hierarchy to contain the Apresjan hierarchy of the underlying dissimilarity. Finally, we have determined the list of standard dissimilarity-based linkage functions D , for which each (possibly pruned) D -linkage hierarchy contains the Apresjan hierarchy of the underlying dissimilarity.

We have observed that a Ward-linkage hierarchy, a centroid-linkage hierarchy and a weighted-centroid-linkage hierarchy may not contain the Apresjan hierarchy for some datasets. Furthermore, notice that Ackerman *et al.* [1] have also pointed out that the Ward-linkage has a specific behavior: more precisely, among the most standard linkage, Ward-linkage is the only one for which the HAC algorithm is weight sensitive for all data sets being processed. In order to provide an insight on user guidelines, an issue could be to determine the dataset characteristics which ensure the inclusion of the Apresjan hierarchy in every HAC-linkage hierarchy produced with a linkage function based on centroids.

References

1. M. Ackerman, S. Ben-David, S. Brânzei, and D. Loker. Weighted clustering: Towards solving the user’s dilemma. *Pattern Recognition*, 120:108152, 2021. [1](#), [16](#)
2. S. I. Ao, K. Yip, M. Ng, D. Cheung, P.-Y. Fong, I. Melhado, and P. C. Sham. Clustag: hierarchical clustering and graph methods for selecting tag snps. *Bioinformatics*, vol. 21(8):1735–1736, 2005. [8](#)
3. J. D. Apresjan. An algorithm for constructing clusters from a distance matrix. *Mashinnyi perevod: prikladnaja lingvistika*, 9:3–18, 1966. [1](#), [5](#)
4. M.-F. Balcan, A. Blum, and S. Vempala. A discriminative framework for clustering via similarity functions. In *STOC '08: Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 671 – 680, 2008. [1](#)

5. H.-J. Bandelt and A.W.M. Dress. Weak hierarchies associated with similarity measures: an additive clustering technique. *Bull. Math. Biology*, 51:133–166, 1989. [1](#)
6. J.-P. Benzécri. *L'Analyse des Données, I la taxinomie*. Dunod, Paris, 1973. [1](#), [5](#)
7. J.P. Benzécri. Description mathématique des classifications. *Revue de Statistique Appliquée*, 20(3):23—56, 1972. [1](#), [5](#)
8. P. Bertrand and J. Diatta. An interval convexity-based framework for multilevel clustering with applications to single-linkage clustering. *Discrete Applied Mathematics*, 342:38–63, 2024. [5](#)
9. H. H. Bock. *Automatische Klassifikation*. Vandenhoeck & Ruprecht, GÄüttingen, 1974. [1](#)
10. M. Bruynooghe. Classification ascendante hiérarchique des grands ensembles de données : un algorithme rapide fondé sur la construction des voisinages réductibles. *Les cahiers de l'Analyse des Données*, 3:7–33, 1978. [7](#)
11. D. Bryant and V. Berry. A structured family of clustering and tree construction methods. *Advances in Applied Mathematics*, 27(4):705–732, 2001. [1](#), [5](#)
12. C. Devauchelle, A. W. M. Dress, A. Grossmann, S. GrÄijnewald, and A. Henaut. Constructing hierarchical set systems. *Annals of Combinatorics*, 8(4):441–456, 2004. [1](#)
13. R. Dubes and A. K. Jain. Clustering techniques: The user’s dilemma. *Pattern Recognition*, 8(4):247–260, 1976. [2](#)
14. L. Fisher and Ness J.V. Admissible clustering procedures. *Biometrika*, 58(1):91–104, 1971. [2](#)
15. M. Gagolewski, A. Cena, S. James, and G. Beliakov. Hierarchical clustering with owa-based linkages, the lance-williams formula, and dendrogram inversions. *Fuzzy Sets and Systems*, vol. 473, 2023. [7](#)
16. A. D. Gordon. Hierarchical classification. In Hubert L. J. Arabie P. and G. DeSoete, editors, *Clustering and Classification*, pages 65–122. World Scientific, London, 1996. [1](#)
17. N. Jardine. Towards a general theory of clustering. *Biometrics*, 25:609–610, 1969. [1](#)
18. P. Legendre and L.F.J. Legendre. Cluster analysis. In *Numerical Ecology. Developments in Environmental Modelling*, volume 24, pages Ebook 289–367. Elsevier, 2012. [3](#)
19. F. Murtagh and P. Contreras. Algorithms for hierarchical clustering: an overview, ii. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(6):e1219, 2017. [7](#)
20. A. F. Parker-Rhodes and R. M. Needham. A reduction method for non-arithmetic data, and its application to thesauric translation. In *Information Processing, Proceedings of the International Conference on Information Processing, Paris, 1960*, pages 321–325. UNESCO, 1960. [1](#)
21. W. E. Wright. A formalization of cluster analysis. *Pattern Recognition*, 5(3):273–282, 1973. [2](#)