Anthropometry: An R Package for Analysis of Anthropometric Data

Guillermo Vinué
Department of Statistics and O.R., University of Valencia, Valencia, Spain.

Abstract

The development of powerful new 3D scanning techniques has enabled the generation of large up-to-date anthropometric databases which provide highly valued data to improve the ergonomic design of products adapted to the user population. As a consequence, Ergonomics and Anthropometry are two increasingly quantitative fields, so advanced statistical methodologies and modern software tools are required to get the maximum benefit from anthropometric data.

This paper presents a new R package, called Anthropometry, which is available on the Comprehensive R Archive Network. It brings together some statistical methodologies concerning clustering, statistical shape analysis, statistical archetypal analysis and the statistical concept of data depth, which have been especially developed to deal with anthropometric data. They are proposed with the aim of providing effective solutions to some common anthropometric problems, such as clothing design or workstation design (focusing on the particular case of aircraft cockpits). The utility of the package is shown by analyzing the anthropometric data obtained from a survey of the Spanish female population performed in 2006 and from the 1967 United States Air Force survey.

This manuscript is contained in Anthropometry as a vignette.

Keywords: R, anthropometric data, clustering, statistical shape analysis, archetypal analysis, data depth.

1. Introduction

Ergonomics is the science that investigates the interactions between human beings and the elements of a system. The application of ergonomic knowledge in multiple areas such as clothing and footwear design or both working and household environments is required to achieve the best possible match between the product and its users. To that end, it is fundamental to know the anthropometric dimensions of the target population. Anthropometry refers to the study of the measurements and dimensions of the human body and is considered a very important branch of Ergonomics because of its significant influence on the ergonomic design of products (Pheasant 2003).

A major issue when developing new patterns and products that fit the target population well is the lack of up-to-date anthropometric data. Improvements in health care, nutrition and living conditions as well as the transition to a sedentary life style have changed the body dimensions of people over recent decades. Anthropometric databases must therefore be updated regularly. Traditionally, human physical characteristics and measurements have been manually taken
using rudimentary methods like calipers, rulers or measuring tapes (Simmons and Itook 2003; Lu and Wang 2008; Shu, Wuhrer, and Xi 2011). These procedures are simple (user-friendly), non-invasive and not particularly expensive. However, obtaining a statistically useful sample of thousands of people by hand is time-consuming and error-prone: the set of measurements obtained, and therefore the shape information, is usually imprecise and inaccurate.

In recent years, the development of new three-dimensional (3D) body scanner measurement systems has represented a huge step forward in the way anthropometric data are collected and updated. This technology provides highly detailed, accurate and reproducible anthropometric data from which 3D shape images of the people being measured can be obtained (Itook and Hwang 2001; Lerch, MacGillivray, and Domina 2007; Wang, Wu, Lin, Yang, and Lu 2007; D’Apuzzo 2009). The great potential of 3D body scanning techniques constitutes a true breakthrough in realistically characterizing people and has made it possible to conduct new large-scale anthropometric surveys in different countries (for instance, in the USA, the UK, France, Germany and Australia). Within this context, the Spanish Ministry of Health sponsored a 3D anthropometric study of the Spanish female population in 2006 (Alemany, González, Nácher, Soriano, Arnaiz, and Heras 2010). A sample of 10,415 Spanish females from 12 to 70 years old, randomly selected from the official Postcode Address File, was measured. Associated software provided by the scanner manufacturers made a triangulation based on the 3D spatial location of a large number of points on the body surface. A 3D binary image of the trunk of each woman (white pixel if it belongs to the body, otherwise black) is produced from the collection of points located on the surface of each woman scanned, as explained in Ibáñez, Simó, Domingo, Durá, Ayala, Alemany, Vinué, and Solves (2012a). The two main goals of this study, which was conducted by the Biomechanics Institute of Valencia, were as follows: firstly, to characterize the morphology of females in Spain in order to develop a standard sizing system for the garment industry and, secondly, to encourage an image of healthy beauty in society by means of mannequins that are representative of the population. In order to tackle both these objectives, Statistics plays an essential role.

In every methodological and practical anthropometric problem, body size variability within the user population is characterized by means of a limited number of anthropometric cases. This is what is called a user-centered design process. An anthropometric case represents the set of body measurements the product evaluator plans to accommodate in design (HFES 300 Committee 2004). A case may be a particular human being or a combination of measurements. Depending on the features and needs of the product being designed, three types of cases can be distinguished: central, boundary and distributed. If the product being designed is a one-size product (one-size to accommodate people within a predetermined portion of the population), as may be the case in working environment design, the cases are selected on an accommodation boundary. However, if we focus on a multiple-size product (n sizes to fit n groups of people within a predetermined portion of the population), clothing design being the most apparent example, central cases are selected. Regarding distributed cases, they are spread throughout the distribution of body dimensions. Central and boundary cases can be considered special types of distributed cases, but distributed cases might not include them. Distributed cases represent an alternative when it is necessary to have a greater number of cases that covers the entire distribution. In such a situation, a small number of central (or boundary) cases would not be sufficient, since they are only spread toward the middle (or edges) of the distribution. The statistical methodologies that we have developed seek to define central and boundary cases to tackle the clothing sizing system design problem and the workplace design problem
Clothing sizing systems divide a population into homogeneous subgroups based on some key anthropometric dimensions (size groups), in such a way that all individuals in a size group can wear the same garment (Ashdown 2007; Chung, Lin, and Wang 2007). An efficient and optimal sizing system must accommodate as large a percentage of the population as possible, in as few sizes as possible, that best describes the shape variability of the population. In addition, the garment fit for accommodated individuals must be as good as possible. Each clothing size is defined from a person who is near the center for the dimensions considered in the analysis. This central individual, which is considered as the size representative (the size prototype), becomes the basic pattern from which the clothing line in the same size is designed. Once a particular garment has been designed, fashion designers and clothing manufacturers hire fit models to test and assess the size specifications of their clothing before the production phase. Fit models have the appropriate body dimensions selected by each company to define the proportional relationships needed to achieve the fit the company has determined (Ashdown 2005; Workman and Lentz 2000; Workman 1991). Fit models are usually people with central measurements in each body dimension. The definition of an efficient sizing system depends to a large extent on the accuracy and representativeness of the fit models.

Clustering is the statistical tool that classifies a set of individuals into groups (clusters), in such a way that subjects in the same cluster are more similar (in some way) to each other than to those in other clusters (Kaufman, L. and Rousseeuw, P. J. 1990). In addition, clusters are represented by means of a representative central observation. Therefore, clustering comes up naturally as a useful statistical approach to try to define an efficient sizing system and to elicit prototypes and fit models. Specifically, five of the methodologies that we have developed are based on different clustering methods. Four of them are aimed at segmenting the population into optimal size groups and obtaining size prototypes. The first one, hereafter referred to as trimowa, has been published in Ibáñez, Vinué, Alemany, Simó, Epifanio, Domingo, and Ayala (2012b). It is based on using a special distance function that mathematically captures the idea of garment fit. The second and third ones (called CCbiclustAnthropo and TDDclust) belong to a technical report (Vinué and Ibáñez 2014), which can be accessed on the author’s website, http://www.uv.es/vivigui/docs/biclustDepth. The CCbiclustAnthropo methodology adapts a particular clustering algorithm mostly used for the analysis of gene expression data to the field of Anthropometry. TDDclust uses the statistical concept of data depth (Liu, Parelius, and Singh 1999) to group observations according to the most central (deep) one in each cluster. As mentioned, traditional sizing systems are based on using a suitable set of key body dimensions, so clustering must be carried out in the Euclidean space. In the three previous procedures, we have always worked in this way. Instead, in the fourth and last one, hereinafter called as kmeansProcrustes, a clustering procedure is developed for grouping women according to their 3D body shape, represented by a configuration matrix of anatomical markers (landmarks). To that end, the statistical shape analysis (Dryden and Mardia 1998) will be fundamental. This approach has been published in Vinué, Simó, and Alemany (2014b). Lastly, the fifth clustering proposal is presented with the goal of identifying accurate fit models and is again used in the Euclidean space. It is based on another clustering method originally developed for biological data analysis. This method, called hipamAnthropom, has been published in Vinué, León, Alemany, and Ayala (2014a). Well-defined fit models and prototypes can be used to develop representative and precise mannequins of the population.

A sizing system is intended only to cover what is known as the “standard” population, leaving...
out the individuals who might be considered outliers with respect to a set of measurements. In this case, outliers are called disaccommodated individuals. Clothing industries usually design garments for the standard sizes in order to optimize market share. The four aforementioned methods concerned with apparel sizing system design (trimowa, CCbiclustAnthropo, TDDclust and kmeansProcrustes) take into account this fact. In addition, because hipamAnthropom is based on hierarchical features, it is capable of discovering and returning true outliers.

Unlike clothing design, where representative cases correspond to central individuals, in designing a one-size product, such as working environments or the passenger compartment of any vehicle, including aircraft cockpits, the most common approach is to search for boundary cases. In these situations, the variability of human shape is described by extreme individuals, which are those that have the smallest or largest values (or extreme combinations) in the dimensions considered in the study. These design problems fall into a more general category: the accommodation problem. The supposition is that the accommodation of boundaries will facilitate the accommodation of interior points (with less-extreme dimensions) (Bertilsson, Högborg, and Hanson 2012; Parkinson, Reed, Kokkolaras, and Papalambros 2006; HFES 300 Committee 2004). For instance, a garage entrance must be designed for a maximum case, while for reaching things such as a brake pedal, the individual minimum must be obtained. In order to tackle the accommodation problem, two methodological contributions based on statistical archetypal analysis are put forward. An archetype in Statistics is an extreme observation that is obtained as a convex combination of other subjects of the sample (Cutler and Breiman 1994). The first of these methodologies has been published in Epifanio, Vinué, and Alemany (2013), and the second has been published in Vinué, Epifanio, and Alemany (2015), which presents the new concept of archetypoids.

As far as we know, there is currently no reference in the literature related on Anthropometry or Ergonomics that provides the programming of the proposed algorithms. In addition, to the best of our knowledge, with the exception of modern human modelling tools like “Jack” and “Ramsis”, which are two of the most widely used tools by a broad range of industries (Blanchonette 2010), there are no other general software applications or statistical packages available on the Internet to tackle the definition of an efficient sizing system or the accommodation problem. Within this context, this paper introduces a new R package (R Development Core Team 2015) called Anthropometry, which brings together the algorithms associated with all the above-mentioned methodologies. All of them were applied to the anthropometric study of the Spanish female population and to the 1967 United States Air Force (USAF) survey. Anthropometry includes several data files related to both anthropometric databases. All the statistical methodologies, anthropometric databases and this R package were announced in the author’s PhD thesis (Vinué 2014), which is freely available in a Spanish institutional open archive. The latest version of Anthropometry is always available from the Comprehensive R Archive Network at http://cran.r-project.org/package=Anthropometry. The package version 1.6 (or greater) is needed to reproduce the examples of this manuscript.

The outline of the paper is as follows: Section 2 describes all the data files included in Anthropometry. Section 3 is intended to guide users in their choice of the different methods presented. Section 4 gives a brief explanation of each statistical technique developed. In Section 5 some examples of their application are shown, pointing out at the same time the consequences of choosing different argument values. Section 6 provides a discussion about the practical usefulness of the methods. Finally, concluding remarks are given in Section 7. One appendix describes the algorithms listings related to each methodology.
2. Data

2.1. Spanish anthropometric survey

The Spanish National Institute of Consumer Affairs (INC according to its Spanish acronym), under the Spanish Ministry of Health and Consumer Affairs, commissioned a 3D anthropometric study of the Spanish female population in 2006, after signing a commitment with the top Spanish companies in the apparel industry. The Spanish National Research Council (CSIC in Spanish) planned and developed the design of the experiment and received advice on Anthropometry from the Complutense University of Madrid. The study itself was conducted by the Biomechanics Institute of Valencia (Alemany et al. 2010). The target sample was made up of 10,415 women grouped into 10 age groups ranging from 12 to 70 years, randomly chosen from the official Postcode Address File.

As an illustrative example of the full Spanish survey, Anthropometry contains a database called sampleSpanishSurvey, made up of a sample of 600 Spanish women and their measurements for five anthropometric variables: bust, chest, waist and hip circumferences and neck to ground length. These variables are chosen for three main reasons: they are recommended by experts, they are commonly used in the literature and they appear in the European standard on sizing systems. Size designation of clothes. Part 2: Primary and secondary dimensions (European Committee for Standardization 2002).

As mentioned above, the women’s shape is represented by a set of landmarks, specifically 66 points. A data file called landmarksSampleSpaSurv contains the configuration matrix of landmarks for each of the 600 women. As also noted above, a 3D binary image of each woman’s trunk is available. Hence, the dissimilarity between trunk forms can be computed and a distance matrix between women can be built. The distance matrix used in Vinué et al. (2015) is included in Anthropometry and is called descrDissTrunks.

2.2. USAF survey

This database contains the information provided by the 1967 United States Air Force (USAF) survey. It can be downloaded from http://www.dtic.mil/dtic/. This survey was conducted in 1967 by the anthropology branch of the Aerospace Medical Research Laboratory (Ohio). A sample of 2420 subjects of the Air Force personnel, between 21 and 50 years of age, were measured at 17 Air Force bases across the United States of America. A total of 202 variables were collected. The dataset associated with the USAF survey is available on USAFSurvey. In the methodologies related to archetypal analysis, six anthropometric variables from the total of 202 will be selected. They are the same as those selected in Zehner, Meindl, and Hudson (1993) and are called cockpit dimensions because they are critical in order for designing an aircraft cockpit.

2.3. Geometric figures

Two geometric figures, a cube and a parallelepiped, made up of 8 and 34 landmarks, are available in the package as cube8landm, cube34landm, parallelep8landm and parallelep34landm, respectively.
3. Anthropometric problems and their algorithmic solutions

In the Anthropometry R package five clustering methods are available (trimowa, CCbiclus\text{t}Anthropo, TDDclus, hipamAnthropom and kmeansProcrustes), each offering a different theoretical foundation and practical benefits. In addition, the archetypoid algorithm is included. The purpose of this Section is to provide users with insights that can enable them to make a suitable selection of the proposed methods. Clustering methodologies have been developed to obtain central cases. On the other hand, methods based on archetype and archetypoid analysis aim to identify boundary cases. Figure 1 shows a decision tree which indicates when each approach is best suited to obtain representative central or boundary cases.

![Decision tree](Figure 1: Decision tree for case selection methods.)

Regarding clustering methods, the main difference between them is their practical objective. This is the first key to finding out which method is right for the user. If the goal of the practitioner is to obtain representative fit models for apparel sizing, the hipamAnthropom algorithm must be used. Otherwise, if the goal is to create clothing size groups and size prototypes, the other four methods are suitable. If the user wanted to design lower body garments, CCbiclus\text{t}Anthropo should be chosen, while for designing upper body garments, trimowa, TDDclus and kmeansProcrustes are suitable. Choosing one of the latter three methods depends on the kind of data being collected. If the database contains a set of 3D landmarks representing the shape of women, the kmeansProcrustes method must be applied. On the other hand, trimowa and TDDclus can be used when the data are 1D body measurements. For illustrative purposes, Figure 2 shows a decision tree that helps the user to decide which clustering approach is best suited.
As a conclusion to this discussion, an illustrative comparison of the outcomes of using trimowa and TDDclust on a random sample subset is given below. We restrict our attention to these two methods because both of them have the same intention. Table 1 shows, in blue and with a frame box, the upper prototypes obtained with TDDclust and with trimowa, respectively (the R code used to obtain these results is available in http://www.uv.es/vivigui/softw/sect3.R). In this case, two of the three prototypes match. However, it is worth pointing out that in another case it is possible that none of them would match. This is because of the different statistical foundation of each approach. At this point, it would be recommendable to use the trimowa methodology because it has been developed further than TDDclust, returns outcomes with a significantly lower computational time, regardless of the sample size, and is endorsed by a scientific publication.

<table>
<thead>
<tr>
<th>Label</th>
<th>women</th>
<th>neck to ground</th>
<th>waist</th>
<th>bust</th>
</tr>
</thead>
<tbody>
<tr>
<td>92</td>
<td></td>
<td>134.3</td>
<td>71.1</td>
<td>82.7</td>
</tr>
<tr>
<td>480</td>
<td></td>
<td>133.1</td>
<td>96.8</td>
<td>106.5</td>
</tr>
<tr>
<td>340</td>
<td></td>
<td>136.3</td>
<td>85.9</td>
<td>95.9</td>
</tr>
<tr>
<td>377</td>
<td></td>
<td>136.1</td>
<td>87.6</td>
<td>97.9</td>
</tr>
</tbody>
</table>

Table 1: Upper size prototypes obtained by TDDclust (in blue) and by trimowa (frame box).
4. Statistical methodologies

4.1. Anthropometric dimensions-based clustering and shape analysis

For practical guidance, Algorithm 1 explains the workflow for clustering-based approaches from an Anthropometry point of view, followed by the description of individual algorithms. See Appendix A for details about the algorithm listings.

1. The data matrix is segmented using a primary control dimension (bust or waist).

   **Note 1:** The segmentation is done according to the classes suggested in the European standard on sizing systems. Size designation of clothes. Part 3: Measurements and intervals (European Committee for Standardization 2005). This standard is drawn up by the European Union and is a set of guidelines for the textile industry to promote the implementation of a clothing sizing system, that is adapted to users.

2. A further clustering segmentation is carried out using other secondary control anthropometric variables.

   if bust was selected in 1. then
   use one of the following methodologies:
   - trimowa (see Algorithm 3).
   - TDDclust (see Algorithm 5).
   - hipamAnthropom (see Algorithm 6).
   - kmeansProcrustes (see Algorithm 9, 10 and 11).
   else
   if waist was selected in 1. then
   use CCbiclustAnthropo (see Algorithm 4).
   end if
   end if

   **Note 2:** In this way, the first segmentation provides a first easy input to choose the size, while the resulting clusters (subgroups) for each bust (or waist) and other anthropometric measurements optimize the sizing. From the point of view of clothing design, by using a more appropriate statistical strategy, such as clustering, homogeneous subgroups are generated taking into account the anthropometric variability of the secondary dimensions that have a significant influence on garment fit.

   **Algorithm 1:** Workflow for clustering-based approaches.

The **trimowa methodology**

The aim of a sizing system is to divide a varied population into groups using certain key body dimensions (Ashdown 2007; Chung et al. 2007). Three types of approaches can be distinguished for creating a sizing system: traditional step-wise sizing, multivariate methods and optimization methods. Traditional methods are not useful because they use bivariate distributions to define a sizing chart and do not consider the variability of other relevant anthropometric dimensions. Recently, more sophisticated statistical methods have been developed, especially using principal component analysis (PCA) and clustering (Gupta and Gangadhar 2004; Hsu 2009b; Luximon, Zhang, Luximon, and Xiao 2012; Hsu 2009a; Chung et al. 2007; Zheng, Yu, and Fan 2007; Bagherzadeh, Latifi, and Faramarzi 2010). Peter Tryfos was the first to suggest an optimization method (Tryfos 1986). He developed an integer programming procedure to maximize garment sales. Later, McCulloch et al. (McCulloch, Paal, and Ashdown 1998) modified Tryfos’ approach by focusing the problem on maximizing
the quality of fit instead of on the sales. The sizes were determined by means of a nonlinear optimization problem. The objective function measured the misfit between a person and the prototype, using a particular dissimilarity measure and removing from the data set a prefixed proportion of the sample.

The first clustering methodology proposed, called trimowa, is closed to the one developed in McCulloch et al. (1998), in terms of maximizing the quality of fit by using a dissimilarity measure to compare individuals and prototypes and by leaving out some individuals of the data set. However, there are two main differences. First, when searching for k prototypes, a more statistical approach is assumed. To be specific, a trimmed version of the partitioning around medoids (PAM or k-medoids) clustering algorithm is used. The trimming procedure allows us to remove outlier observations (García-Escudero, Gordaliza, Matrán, and Mayo-Iscar 2008; García-Escudero, Gordaliza, and Matrán 2003).

Second, the dissimilarity measure defined in McCulloch et al. (1998) is modified using an OWA (ordered weighted average) operator to consider the user morphology. This approach was published in Ibáñez et al. (2012b) and it is implemented in the trimowa function. Next, the mathematical details behind this procedure are briefly explained. A detailed exposition is given in Ibáñez et al. (2012b); Vinué (2014). The dissimilarity measure is defined as follows. Let \( x = (x_1, \ldots, x_p) \) be an individual of the user population represented by a feature vector of size \( p \) of his/her body measurements. In the same way, let \( y = (y_1, \ldots, y_p) \) be \( p \) measurements of the prototype of a particular size. Then, \( d(x, y) \) measures the misfit between a particular individual and the prototype. In other words, \( d(x, y) \) indicates how far a garment made for prototype \( y \) would be from the measurements for a given person \( x \). In McCulloch et al. (1998) the dissimilarity measure in each measurement has the following expression:

\[
d_i(x_i, y_i) = \begin{cases} a_i^b (\ln(y_i) - b_i^b\ln(x_i)) & \text{if } \ln(x_i) < \ln(y_i) - b_i^b \\ 0 & \text{if } \ln(y_i) - b_i^b \leq \ln(x_i) \leq \ln(y_i) + b_i^b \\ a_i^b (\ln(x_i) - b_i^b\ln(y_i)) & \text{if } \ln(x_i) > \ln(y_i) + b_i^b \\ \end{cases} 
\]

where \( a_i^b, b_i^b, a_i^h \) and \( b_i^h \) are constants for each dimension and have the following meaning: \( b_i \) corresponds to the range in which there is a perfect fit; \( a_i \) indicates the rate at which fit deteriorates outside this range, i.e., it reflects the misfit rate. In McCulloch et al. (1998) the global dissimilarity is merely defined as a sum of squared discrepancies over each of the \( p \) body measurements considered:

\[
d(x, y) = \sum_{i=1}^{p} (d_i(x_i, y_i))^2
\]

Because the different dissimilarities \( d_i(x_i, y_i)'s \) are being aggregated (summed), an OWA operator can be used. Let \( d_1, \ldots, d_p \) the values to be aggregated. An OWA operator of dimension \( p \) is a mapping \( f : \mathbb{R}^p \to \mathbb{R} \) where \( f(d_1, \ldots, d_p) = w_1b_1 + \ldots + w_pb_p \), being \( b_j \) the \( j \)th largest element in the collection \( d_1, \ldots, d_p \) (i.e., these values are ordered in decreasing order) and \( W = (w_1, \ldots, w_p) \) an associated weighting vector such that \( w_i \in [0, 1], 1 \leq i \leq p \) and \( \sum_{j=1}^{p} w_j = 1 \). Because the OWA operators are bounded between the max and min operators, a measure called orness was defined in Yager (1988) to classify the OWA operators.
between those two. The orness quantity adjusts the importance to be attached to the values \(d_1, \ldots, d_p\), depending on their ranks:

\[
\text{orness}(W) = \frac{1}{p-1} \sum_{i=1}^{p} (p-i)w_i. \tag{3}
\]

On consequence, the dissimilarity used in \textit{trimowa} and also in \textit{hipamAnthropom} is defined as follows:

\[
d(x,y) = \sum_{i=1}^{p} w_i(d_i(x_i, y_i))^2 \tag{4}
\]

In short, the dissimilarity presented in Equation 4 is defined as a sum of squared discrepancies over each of the \(p\) body measurements considered, adjusting the importance of each one of them by assigning to each one of them a particular OWA weight. The set of weights \(W = (w_1, \ldots, w_p)\) is based on using a mixture of the binomial \(Bi(p-1, 1.5 - 2 \cdot \text{orness})\) and the discrete uniform probability distributions. Specifically, each weight is calculated as:

\[
w_i = \lambda \cdot \pi_i + (1-\lambda) \cdot \frac{1}{p}, \]

where \(\pi_i\) is the binomial probability for each \(i = 0, \ldots, p-1\). The algorithm associated with the \textit{trimowa} methodology is summarized in Algorithm 3 (the number of clusters is labeled \(k\) as in the \(k\)-medoids algorithm).

Our approach allows us to obtain more realistic prototypes (medoids) because they correspond to real women from the database and the selection of individual discommodities. In addition, the use of OWA operators has resulted in a more realistic dissimilarity measure between individuals and prototypes. We learned from this situation that there is an ongoing search for advanced statistical approaches that can deliver practical solutions to the definition of central people and optimal size groups. Consequently, we have come across two different statistical strategies in the literature and have aimed to discuss their potential usefulness in the definition of an efficient clothing sizing system. These approaches are based on biclustering and data depth and will be summarized below.

**The \textit{CCbiclustAnthropo} methodology**

Given a data set with a number of rows and columns, conventional clustering can be applied to either the rows or the columns of the data matrix, separately. In a traditional row cluster, each row is defined using all the columns of the data matrix. Something similar would occur with a column cluster. Biclustering is a novel clustering approach that consists of simultaneously partitioning the set of rows and the set of columns into subsets. With biclustering, each row in a bicluster is defined using only a subset of columns and vice versa. Therefore, clustering provides a global model but biclustering defines a local one (Madeira and Oliveira 2004). This interesting property made us think that biclustering could perhaps be useful for obtaining efficient size groups, since they would only be defined for the most relevant anthropometric dimensions that describe a body in the detail necessary to design a well-fitting garment.

Recently, a large number of biclustering methods have been developed. Some of them are implemented in different sources, including R. Currently, the most complete R package for biclustering is \textbf{biclust} (Kaiser and Leisch 2008; Kaiser, Santamaria, Khamiakova, Sill, Theron, Quintales, and Leisch 2013). The usefulness of the approaches included in \textbf{biclust} for dealing with anthropometric data was investigated in Vinué (2012). Among the conclusions reached,
the most important was concerned with the possibility of considering the Cheng & Church biclustering algorithm (Cheng and Church 2000) (referred to below as CC) as a potential statistical approach to be used for defining size groups. Specifically, in Vinué (2012) an algorithm to find size groups (biclusters) and disaccommodated women with CC was set out. This methodology is called CCbiclustAnthropo and it is implemented in the CCbiclustAnthropo function. Next, the mathematical details behind the CCbiclustAnthropo procedure are briefly described. First of all, the CC algorithm must be introduced (see Cheng and Church 2000; Vinué 2014; Kaiser and Leisch 2008, for more details). The CC algorithm searches for biclusters with constant values (in rows or in columns). To that end, it defines the following score:

$$H(I, J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2,$$

where $a_{iJ}$ is the mean of the $i$th row of the bicluster, $a_{Ij}$ is the mean of the $j$th column of the bicluster and $a_{IJ}$ is its overall mean. Then, a subgroup is called a bicluster if the score is below a value $\delta \geq 0$ and above an $\alpha$-fraction of the score of the whole data ($\alpha > 1$).

The CC algorithm implemented in the biclust function of the biclust package requires three arguments. Firstly, the maximum number of biclusters to be found. We propose that this number should be fixed for each waist size according to the number of women it contains: For less than 150, fix 2 biclusters; between 151-300, 3; between 351-450, 4; greater than 415, 5. Secondly, the $\alpha$ value. Its default value (1.5) is maintained. Finally, the $\delta$ value. Because CC is nonexhaustive, i.e., it might not group every woman into a bicluster, the value of $\delta$ can be iteratively adapted to the number of disaccommodated women we want to discard in each size. The proportion of the trimmed sample is prefixed to 0.01 per size. In this way, a number of women between 0 and the previous fixed proportion will not be assigned to any group. The algorithm associated with the CCbiclustAnthropo methodology is summarized in Algorithm 4.

Designing lower body garments depends not only on the waist circumference (the principal dimension in this case), but also on other secondary control dimensions (for upper body garments only the bust circumference is usually needed). Biclustering produces subgroups of objects that are similar in one subgroup of variables and different in the remaining variables. Therefore, it seems more interesting to use a biclustering algorithm with a set of lower body dimensions. For that purpose, all the body variables related to the lower body in the Spanish anthropometric survey were chosen (there were 36). An efficient partition into different biclusters was obtained with promising results. All individuals in the same bicluster can wear a garment designed for the specific body dimensions (waist and other variables) which were the most relevant for defining the group. Each group is represented by the median woman.

The main interest of this approach was descriptive and exploratory and the important point to note here is that CCbiclustAnthropo cannot be used with sampleSpanishSurvey, since this data file does not contain variables related to the lower body in addition to waist and hip. However, this function is included in the package in the hope that it could be helpful or useful for other researchers. All theoretical and practical details are given in Vinué and Ibáñez (2014), Vinué (2014) and Vinué (2012).
The **TDDclust** methodology

The statistical concept of data depth is another general framework for descriptive and inferential analysis of numerical data in a certain number of dimensions. In essence, the notion of data depth is a generalization of standard univariate rank methods in higher dimensions. A depth function measures the degree of centrality of a point regarding a probability distribution or a data set. The highest depth values correspond to central points and the lowest depth values correspond to tail points (Liu et al. 1999; Zuo and Serfling 2000). Therefore, the depth paradigm is another very interesting strategy for identifying central prototypes.

The development of clustering and classification methods using data depth measures has received increasing attention in recent years (Dutta and Ghosh 2012; Lange, Mosler, and Mozharovskyi 2012; López and Romo 2010; Ding, Dang, Peng, and Wilkins 2007). The most relevant contribution to this field has been made by Rebecka Jörnsten in Jörnsten (2004) (see Jörnsten, Vardi, and Zhang 2002; Pan, Jörnsten, and Hart 2004, for more details). She introduced two clustering and classification methods (**DDclust** and **DDclass**, respectively) based on $L_1$ data depth (see Vardi and Zhang 2000, for more details). The **DDclust** method is proposed to solve the problem of minimizing the sum of $L_1$-distances from the observations to the nearest cluster representatives. In clustering terms, the $L_1$ data depth is the amount of probability mass needed at a point $z$ to make $z$ the multivariate $L_1$-median (a robust representative) of the data cluster.

An extension of **DDclust** is introduced which incorporates a trimmed procedure, aimed at segmenting the data into efficient size groups using central (the deepest) people. This methodology will be referred to below as **TDDclust** and it can be used within **Anthropometry** by using a function with the same name. Next, the mathematical details behind the **TDDclust** procedure are briefly described. A thorough explanation is given in Vinué and Ibáñez (2014); Vinué (2014). First, the $L_1$ multivariate median (from now on, $L_1$-MM) is defined as the solution of the Weiszfeld problem (Vardi and Zhang 2000). Vardi et al. (Vardi and Zhang 2000) proved that the depth function associated with the $L_1$-MM, called $L_1$ data depth, is:

$$D(y) = \begin{cases} 
1 - ||\bar{e}(y)|| & \text{if } y \notin \{x_1, \ldots, x_m\}, \\
1 - \left(||\bar{e}(y)|| - f_k\right) & \text{if } y = x_k.
\end{cases}$$

(5)

where $e_i(y) = (y - x_i)/||y - x_i||$ (unit vector from $y$ to $x_i$) and $\bar{e}(y) = \sum_{x_i \neq y} e_i(y) f_i$ (average of the unit vectors from $y$ to all observations), with $f_i = \eta_i / \sum_{j=1}^k \eta_j$ ($\eta_i$ is a weight for $x_i$) and $||\bar{e}(y)||$ is close to 1 if $y$ is close to the edge of the data, and close to 0 if $y$ is close to the center.

The **DDclust** method is proposed to solve the problem of minimizing the sum of $L_1$-distances from the observations to the nearest cluster representatives. Specifically, **DDclust** iterates between median computations via the modified Weiszfeld algorithm (Weiszfeld and Plastria 2009) and a Nearest-Neighbor allocation scheme with simulation annealing. The clustering criterion function used in **DDclust** is the maximization of:

$$C(I^K_1) = \frac{1}{N} \sum_{k=1}^{K} \sum_{i \in I(k)} (1 - \lambda) s_i l_i + \lambda R e D_i$$

(6)
with respect to a partition \( I^K_1 = \{I(1), \ldots, I(K)\} \). For each point \( i \), \( \text{sil}_i \) is the silhouette width, \( \text{Re}D_i \) is the difference between the within cluster \( L_1 \) data depth and the between cluster \( L_1 \) data depth, and \( \lambda \in [0, 1] \) is a parameter that controls the influence the data depth has over the clustering. Following Zuo (2006), for any \( 0 < \alpha < \alpha^* = \sup_x(D(x)) \leq 1 \), the \( \alpha \)-th trimmed depth region is:

\[
D^\alpha = \{x : D(x) \geq \alpha\}.
\]  

The idea behind \textit{TDDclust} is to define trimmed regions at each step of the iterative algorithm and to apply the \textit{DDclust} algorithm to the remaining set of observations. The algorithm associated with the \textit{TDDclust} methodology is summarized in Algorithm 5.

The \textit{hipamAnthropom} methodology

Representative fit models are important for defining a meaningful sizing system. However, there is no agreement among apparel manufacturers and almost every company employs a different fit model. Companies try to improve the quality of garment fit by scanning their fit models and deriving dress forms from the scans (Ashdown 2007; Song and Ashdown 2010). A fit model’s measurements correspond to the commercial specifications established by each company to achieve the company’s fit (Loker, Ashdown, and Schoenfelder 2005; Workman and Lentz 2000; Workman 1991). Beyond merely wearing the garment for inspection, a fit model provides objective feedback about the fit, movement or comfort of a garment in place of the consumer. The \textit{hipamAnthropom} methodology is proposed in order to provide new insights about this problem. This methodology is available in the \textit{hipamAnthropom} function. It consists of two classification algorithms based on the hierarchical partitioning around medoids (HIPAM) clustering method presented in Wit and McClure (2004), which has been modified to deal with anthropometric data. HIPAM is a divisive hierarchical clustering algorithm using PAM. This procedure was published in Vinué \textit{et al.} (2014a). The outputs of the two algorithms include a set of central representative subjects or medoids taken from the original data set, which constitute our fit models. They can also detect outliers. The first one, called \textit{HIPAM}_MO, is a slightly modification of the HIPAM that uses the dissimilarity defined in Equation 4. \textit{HIPAM}_MO uses the average silhouette width (asw, see Kaufman, L. and Rousseeuw, P. J. (1990)) as a measure of cluster structure and the maximization of the asw as the rule to subdivide each already accepted cluster. The use of asw could be too restrictive. That’s why a second algorithm, \textit{HIPAM}_IMO, is proposed, where the differences regarding the original HIPAM are even deeper. It incorporates a different criterion: the INCA statistic criterion (Irigoien and Arenas 2008; Arenas and Cuadras 2002; Irigoien, Sierra, and Arenas 2012) to decide the number of child clusters and as a stopping rule. In short, INCA is defined as the probability of properly classified individuals and it is estimated with the following expression:

\[
\text{INCA}_k = \frac{1}{k} \sum_{j=1}^{k} \frac{N_j}{n_j}
\]  

where \( N_j \) is the total number of units in a cluster \( C_j \) which are well classified and \( n_j \) is the size of cluster \( C_j \). Next, a briefly exposure about the details behind \textit{HIPAM}_MO and
**Anthropometry: An R Package for Analysis of Anthropometric Data**

**HIPAM** is given. Let’s start with **HIPAM**: The output of a HIPAM algorithm is represented by a classification tree where each node corresponds to a cluster. The end nodes give us the final partition. The highest or top node, **T**, corresponds to the whole database. For a given node **P**, the algorithm must decide if it is convenient to split this (parent) cluster into new (child) clusters, or stop. If |**P**| ≤ 2, then it is an end (or terminal) node. If not, a **PAM** is applied to **P** with **k**1 groups, where **k**1 is chosen by maximizing the asw of the new partition. After a post-processing step, a partition **C** = {**C**1, ..., **C**k} is finally obtained from **P** (k is not necessarily equal to **k**1). Next, the mean silhouette width of **C** (or aswC) is obtained, and then the same steps used to generate **C** are applied to each **C**i to obtain a new partition. If we denote by SSi the asw of the new partition with i = 1, ..., k (if |**C**i| ≤ 2 then SSi = 0), then the Mean Split Silhouette (MSS) is defined as the mean of the SSi’s. If MSS(k) < aswC, then these new k child clusters of the partition **C** are included in the classification tree. Otherwise, **P** is a terminal node. On the other hand, the algorithm **HIPAM** is summarized in Algorithm 6. The main difference between **HIPAM** and **HIPAM** is in the use of the INCA criterion:

1. At each node **P**, if there is k such that INCAk > 0.2, then we select the k prior to the first largest slope decrease.
2. On the other hand, if INCAk < 0.2 for all k, then **P** is a terminal node.

However, this procedure does not apply either to the top node **T**, or to the generation of the new partitions from which the MSS is calculated. In this case, even when all INCAk < 0.2, we fix k = 3 as the number of groups to divide and proceed.

**The kmeansProcrustes methodology**

The clustering methodologies explained so far use a set of control anthropometric variables as the basis for a different type of sizing system in which people are grouped in a size group based on a full range of measurements. Consequently, clustering is done in the Euclidean space. The shape of the women recruited into the Spanish anthropometric survey is represented by a configuration matrix of correspondence points called landmarks. Taking advantage of this fact, we have adapted the k-means clustering algorithm to the field of statistical shape analysis, to define size groups of women according to their body shapes. The representative of each size group is the average woman. This approach was published in Vinué et al. (2014b).

We have adapted both the original Lloyd and Hartigan-Wong (H-W) versions of k-means to the field of shape analysis and we have demonstrated, by means of a simulation study, that the Lloyd version is more efficient for clustering shapes than the H-W version. The function that uses the Lloyd version of k-means adapted to shape analysis (what we called kmeansProcrustes) is LloydShapes. The function that uses the H-W version of k-means adapted to shape analysis is HartiganShapes. A trimmed version of kmeansProcrustes can be also executed with trimmedLloydShapes.

To adapt k-means to the context of shape analysis, we integrated the Procrustes distance and Procrustes mean into it. A glossary of the concepts of shape analysis used is provided below. The following general notation will be used: n refers to the number of objects, h to the number of landmarks and m to the number of dimensions (in our case, m = 3). Then, each object is described by an h × m configuration matrix X containing the m Cartesian coordinates of
its $h$ landmarks. The pre-shape of an object is what is left after allowing for the effects of translation and scale. The shape of an object is what is left after allowing for the effects of translation, scale and rotation. The shape space $\Sigma_h$ (named Kendall shape space) is the set of all possible shapes. The Procrustes distance is the square root of the sum of squared differences between the positions of the landmarks in two optimally (by least-squares) superimposed configurations at centroid size (the centroid size is the most commonly used measure of size for a configuration). The Procrustes mean is the shape that has the least summed squared Procrustes distance to all the configurations of a sample. Algorithms 9, 10 and 11 show the algorithms behind LloydShapes, trimmedLloydShapes and HartiganShapes, respectively.

4.2. Archetypal analysis

Regarding the methodologies using archetypal analysis, for practical guidance, Algorithm 2 explains their corresponding workflow, followed by the description of individual algorithms. See Appendix A for details about the algorithm listings.

1. Depending on the problem, the data may or may not be standardized.
2. An accommodation subsample is selected.
3. A number $k$ of archetypes is obtained (see Algorithm 12).
4. The nearest individuals to the archetypes are computed.
5. A number $k$ of archetypoids is obtained (see Algorithm 13).

Algorithm 2: Workflow for archetypal-based approaches.

In ergonomic-related problems, where the goal is to create more efficient people-machine interfaces, a small set of extreme cases (boundary cases), called human models, is sought. Designing for extreme individuals is appropriate where some limiting factor can define either a minimum or maximum value which will accommodate the population. The basic principle is that accommodating boundary cases will be sufficient to accommodate the whole population. For too long, the conventional solution for selecting this small group of boundary models was based on the use of percentils. However, percentils are a kind of univariate descriptive statistic, so they are suitable only for univariate accommodation and should not be used in designs that involve two or more dimensions. Furthermore, they are not additive (Zehner et al. 1993; Robinette and McConville 1981; Moroney and Smith 1972). Today, the alternative commonly used for the multivariate accommodation problem is based on PCA (Friess and Bradtmiller 2003; Hudson, Zehner, and Meindl 1998; Robinson, Robinette, and Zehner 1992; Bittner, Glenn, Harris, Iavecchia, and Wherry 1987). However, it is known that the PCA approach presents some drawbacks (Friess 2005). In Epifanio et al. (2013), a different statistical approach for determining multivariate limits was put forward: archetypal analysis (Cutler and Breiman 1994), and its advantages regarding over PCA were demonstrated. The theoretical basis of archetype analysis is as follows. Let $X$ be an $n \times m$ matrix that represents a multivariate dataset with $n$ observations and $m$ variables. The goal of archetype analysis is to find a $k \times m$ matrix $Z$ that characterizes the archetypal patterns in the data, such that data can be represented as mixtures of those archetypes. Specifically, archetype analysis is aimed at obtaining the two $n \times k$ coefficient matrices $\alpha$ and $\beta$ which minimize the residual sum of squares that arises from combining the equation that shows $x_i$ as being approximated by a linear combination of $z_j$’s (archetypes) and the equation that shows $z_j$’s
as linear combinations of the data:

\[ \|x_i - \sum_{j=1}^{k} \alpha_{ij} z_j\|^2 \]

\[ z_j = \sum_{l=1}^{n} \beta_{jl} x_l \]

\[ \Rightarrow RSS = \sum_{i=1}^{n} \|x_i - \sum_{j=1}^{k} \alpha_{ij} z_j\|^2 = \sum_{i=1}^{n} \|x_i - \sum_{j=1}^{k} \alpha_{ij} \sum_{l=1}^{n} \beta_{jl} x_l\|^2, \quad (9) \]

under the constraints

1) \[ \sum_{j=1}^{k} \alpha_{ij} = 1 \] with \( \alpha_{ij} \geq 0 \) and \( i = 1, \ldots, n \) and

2) \[ \sum_{l=1}^{n} \beta_{jl} = 1 \] with \( \beta_{jl} \geq 0 \) and \( j = 1, \ldots, k \).

On the one hand, constraint 1) tells us that the predictors of \( x_i \) are finite mixtures of archetypes, \( \hat{x}_i = \sum_{j=1}^{k} \alpha_{ij} z_j \). Each \( \alpha_{ij} \) is the weight of the archetype \( j \) for the individual \( i \), that is to say, the \( \alpha \) coefficients represent how much each archetype contributes to the approximation of each individual. On the other hand, constraint 2) implies that archetypes \( z_j \) are convex combinations of the data points, \( z_j = \sum_{l=1}^{n} \beta_{jl} x_l \). Algorithm 12 shows an outline of the archetypal algorithm, following Eugster and Leisch (2009).

The function that allows us to reproduce the results discussed in Epifanio et al. (2013) is archetypesBoundary (use set.seed(2010) to obtain the same results).

According to the previous definition, archetypes computed by archetypal analysis are a convex combination of the sampled individuals, but they are not necessarily real observations. The archetypes would correspond to specific individuals when \( z_j \) is an observation of the sample, that is to say, when only one \( \beta_{jl} \) is equal to 1 in constraint 2) for each \( j \). As \( \beta_{jl} \geq 0 \) and the sum of constraint 2) is 1, this implies that \( \beta_{jl} \) should only take on the value 0 or 1. In some problems, it is crucial that the archetypes are real subjects, observations of the sample, and not fictitious. To that end, we have proposed a new archetypal concept: the archetypoid, which corresponds to specific individuals and each observation of the data set can be represented as a mixture of these archetypoids. In the analysis of archetypoids, the original continuous optimization problem therefore becomes:

\[ RSS = \sum_{i=1}^{n} \|x_i - \sum_{j=1}^{k} \alpha_{ij} z_j\|^2 = \sum_{i=1}^{n} \|x_i - \sum_{j=1}^{k} \alpha_{ij} \sum_{l=1}^{n} \beta_{jl} x_l\|^2, \quad (10) \]

under the constraints

1) \[ \sum_{j=1}^{k} \alpha_{ij} = 1 \] with \( \alpha_{ij} \geq 0 \) and \( i = 1, \ldots, n \) and
2) \( \sum_{l=1}^{n} \beta_{jl} = 1 \) with \( \beta_{jl} \in \{0, 1\} \) and \( j = 1, \ldots, k \) i.e. \( \beta_{jl} = 1 \) for one and only one \( l \) and \( \beta_{jl} = 0 \) otherwise.

This new concept archetypoids is introduced in a paper published in Vinué et al. (2015). We have developed an efficient computational algorithm based on PAM to compute archetypoids (called archetypoid algorithm), we have analyzed some of their theoretical properties, we have explained how they can be obtained when only dissimilarities between observations are known (features are unavailable) and we have demonstrated some of their advantages regarding over classical archetypes.

The archetypoid algorithm has two phases: a BUILD phase and a SWAP phase, like PAM. In the BUILD step, an initial set of archetypoids is determined, made up of the nearest individuals to the archetypes computed in the first instance. This set can be defined in three different ways: The first possibility consists in computing the Euclidean distance between the \( k \) archetypes and the individuals and choosing the nearest ones, as mentioned in Epifanio et al. (2013) (set \( \text{cand}_{\alpha} \)). The second choice identifies the individuals with the maximum \( \alpha \) value for each archetype, i.e. the individuals with the largest relative share for the respective archetype (set \( \text{cand}_{\alpha} \), used in Eugster (2012) and Seiler and Wohlrabe (2013)). The third choice identifies the individuals with the maximum \( \beta \) value for each archetype, i.e., the major contributors in the generation of the archetypes (set \( \text{cand}_{\beta} \)). Accordingly, the initial set of archetypoids is \( \text{cand}_{\alpha}, \text{cand}_{\alpha} \) or \( \text{cand}_{\beta} \). The aim of the SWAP phase of the archetypoid algorithm is the same as that of the SWAP phase of PAM, but the objective function is now given by Equation 10 (see Vinué et al. 2015; Vinué 2014, for more details). Algorithm 13 shows an outline of the archetypoid algorithm.

5. Applications

This Section presents a detailed explanation of the numerical and graphical outcome provided by each method by means of several examples. In addition, some relevant comments are given about the consequences of choosing different argument values in each case.

First of all, Anthropometry must be loaded into R:

```r
library("Anthropometry")
```

5.1. Anthropometric dimensions-based clustering and shape analysis

The trimowa methodology

The following code executes the trimowa methodology. A similar code was used to obtain the results described in Ibáñez et al. (2012b). We use `sampleSpanishSurvey` and its five anthropometric variables. The bust circumference is used as the primary control dimension. Twelve bust sizes (from 74 cm to 131 cm) are defined according to the European standard on sizing systems. Size designation of clothes. Part 3: Measurements and intervals (European Committee for Standardization 2005).
Anthropometry: An R Package for Analysis of Anthropometric Data

The aggregation weights of the OWA operator are computed. They are used to calculate the global dissimilarity between the individuals and the prototypes. We give orness a value of 0.7 in order to highlight the largest aggregated values, that is to say, the largest discrepancies between the women’s body measurements and those of the prototype. An orness value close to 1 gives more importance to the worst fit, whilst an orness value close to 0 gives more importance to the best fit (see Vinué (2014, p. 27-31) for details).

The trimmed proportion, alpha, is prefixed to 0.01 per segment (therefore, the accommodation rate in each bust size will be 99%). This selection allows us to accommodate a very large percentage of the population in the sizing system. A larger trimmed proportion would result in a smaller amount of accommodated people. The number of random initializations is 10 (niter), with seven steps per initialization (algSteps). These values are small in the interests of a fast execution. The more random repetitions, the more accurate the prototypes and the more representative of the size group. In Ibáñez et al. (2012b), the number of random initializations was 600.

In addition, a vector of five constants (one per variable) is needed to define the dissimilarity. The numbers collected in the ah argument are related to the particular five variables selected in sampleSpanishSurvey. Different body variables would require different constants (see McCulloch et al. 1998; Vinué 2014, for further details).

To reproduce results, a seed for randomness is fixed. This will also be done with the other methods presented below.

```r
#suppressWarnings(RNGversion("3.5.0"))
#set.seed(2014)
numSizes <- bustSizes$nsizes - 1
res_trimowa <- computSizesTrimowa(dataTrimowa, bust, bustSizes$bustCirc, numSizes, weightsTrimowa, numClust, alpha, niter, algSteps, ah, FALSE)
```
The prototypes are the clustering medoids. The `anthrCases` generic function allows us to obtain the estimated cases by each method.

```
prototypes <- anthrCases(res_trimowa, numSizes)
```

Figure 3 shows the scatter plots of bust circumference against neck to ground with the three prototypes obtained for each bust class without (left) and with (right) the prototypes defined by the European standard. The prototypes color and the plot title must be provided. Unlike the European standard prototypes, which are strictly defined for any database, our prototypes are better adapted to the particular body measurements of the sample of individuals belonging to each size.

```
bustVariable <- "bust"
xlim <- c(72, 132)
color <- c("black", "red", "green", "blue", "cyan", "brown", "gray",
          "deepink3", "orange", "springgreen4", "khaki3", "steelblue1")
variable <- "necktogram"
ylim <- c(116, 156)
title <- "Prototypes \n bust vs neck to ground"
plotPrototypes(dataTrimowa, prototypes, numSizes, bustVariable,
                  variable, color, xlim, ylim, title, FALSE)
plotPrototypes(dataTrimowa, prototypes, numSizes, bustVariable,
                  variable, color, xlim, ylim, title, TRUE)
```

Figure 3: Bust vs. neck to ground, jointly with our medoids (left) and the prototypes defined by the European standard (right).
The **TDDclust** methodology

A basic example of the **TDDclust** methodology is shown here. Computing data depth is very demanding. As an illustration, only 25 individuals are selected. In addition, the neck to ground, waist and bust variables are selected.

```r
dataTDDcl <- sampleSpanishSurvey[1:25, c(2, 3, 5)]
dataTDDcl_aux <- sampleSpanishSurvey[1:25, c(2, 3, 5)]
```

In line with **trimowa**, three size groups are calculated (**numClust**) and a trimmed proportion is fixed to 0.01 (**alpha**). The **lambda** controls the influence the data depth has over the clustering. If **lambda** is 1, the clustering criterion is equivalent to the average silhouette width. On the contrary, if **lambda** is 0, it is given by the average relative data depth. Fixing **lambda** to 0.5 is an intermediate and suggested scenario. A detailed explanation of the consequences of different **lambda** values is given in Jörnsten (2004). Because the depth computation is costly, we only run the algorithm for five iterations (**niter**).

The other arguments are given by default. A different value for **Th** may result in the optimum clustering not being found (see Jörnsten 2004, page 75). A different simulated annealing parameter (**T0** and **simAnn**) may change the clustering results obtained.

```r
numClust <- 3 ; alpha <- 0.01 ; lambda <- 0.5 ; niter <- 5
Th <- 0 ; T0 <- 0 ; simAnn <- 0.9
#suppressWarnings(RNGversion("3.5.0"))
#set.seed(2014)
res_TDDcl <- TDDclust(dataTDDcl, numClust, lambda, Th, niter, T0, simAnn, alpha, dataTDDcl_aux, verbose = FALSE)
```

The following code statements allow us to analyze the clustering results, the final value of the optimal partition and the iteration in which the optimal partition was found, respectively.

```r
table(res_TDDcl$NN[1,])
#1 2 3
#5 10 9
res_TDDcl$Cost
#[1] 0.3717631
res_TDDcl$klBest
#[1] 3
```

The prototypes are obtained with **anthrCases**. In addition, the **trimmOutl** generic function allows us to obtain the trimmed or outlier observations discarded by each method.

```r
prototypes <- anthrCases(res_TDDcl)
trimmed <- trimmOutl(res_TDDcl)
```

The **hipamAnthropom** methodology
The following code statements illustrate how to use the `hipamAnthropom` methodology. The same twelve bust segments as in `trimowa` are used.

```r
dataHipam <- sampleSpanishSurvey
bust <- dataHipam$bust
bustSizes <- bustSizesStandard(seq(74, 102, 4), seq(107, 131, 6))
```

In this situation, where `hipamAnthropom` is applied to a sequence of body sizes, this algorithm is used inside the helper function `computSizesHipamAnthropom`. The `HIPAM_{IMO}` algorithm is used. It was verified in Vinué et al. (2014a) that `HIPAM_{IMO}` showed better performance for finding representative prototypes. The maximum number of clusters that any cluster can be divided into is fixed to five (`maxsplit`). In the HIPAM algorithm the number of sub-clusters that any cluster is potentially divided into is between 2 and `maxsplit`. A larger `maxsplit` than five could result in too many clusters, which is not interesting from the point of view of the strategy used by companies to design sizes.

The same orness and vector of constants as in `trimowa` are used.

```r
type <- "IMO"
maxsplit <- 5 ; orness <- 0.7
ah <- c(23, 28, 20, 25, 25)
#suppressWarnings(RNGversion("3.5.0"))
#set.seed(2013)
numSizes <- bustSizes$nsizes - 1
res_hipam <- computSizesHipamAnthropom(dataHipam, bust, bustSizes$bustCirc, 
                                        numSizes, maxsplit, orness, type, 
                                        ah, FALSE)
```

The fit models are the clustering medoids and the outliers are the discarded observations.

```r
fitmodels <- anthrCases(res_hipam, numSizes)
outliers <- trimmOutl(res_hipam, numSizes)
```

Figure 4 displays the fit models (left) and the outliers (right) corresponding to each bust size. The fit models color and the plot title must be provided. The important point to note here is the fact that each bust segment has a small sample size. This might explain the fact that this algorithm (and also `HIPAM_{MO}`) does not find large homogeneous clusters and therefore identifies a lot of women as outliers in each class for this database. One of the features of the HIPAM algorithm is that it is a very sensitive algorithm for identifying outliers. A broad discussion, analysis and thoughts on the anthropometric meaning of these outliers is given in Vinué et al. (2014a) (including the supplementary material).

```r
bustVariable <- "bust"
xlim <- c(72, 132)
color <- c("black", "red", "green", "blue", "cyan", "brown", "gray",
         "deeppink3", "orange", "springgreen4", "khaki3", "steelblue1")
```
variable <- "hip"
ylim <- c(83, 153)
title <- "Fit models HIPAM.IMO \n bust vs hip"
title_outl <- "Outlier women HIPAM.IMO \n bust vs hip"
plotPrototypes(dataHipam, fitmodels, numSizes, bustVariable, variable, color, xlim, ylim, title, FALSE)
plotTrimmOutl(dataHipam, outliers, numSizes, bustVariable, variable, color, xlim, ylim, title_outl)

Figure 4: Bust vs. hip with the medoids (left) and with the outliers (right) obtained using HIPAM.IMO.

The \textit{kmeansProcrustes} methodology

To conclude this Section, the use of the \textit{kmeansProcrustes} methodology is illustrated. For the sake of simplicity of the computation involved only a small sample (the first 50 individuals) is selected. When there are missing values (NA’s, not available numbers), they are removed.

landmarksNoNa <- na.exclude(landmarksSampleSpaSurv)
numLandmarks <- (dim(landmarksNoNa)[2]) / 3
landmarksNoNa_First50 <- landmarksNoNa[1 : 50, ]
numIndiv <- dim(landmarksNoNa_First50)[1]

We have to define an array with the 3D landmarks of the sample objects.

array3D <- array3Dlandm(numLandmarks, numIndiv, landmarksNoNa_First50)

Again, three size groups are calculated (numClust) and a trimmed proportion is fixed to 0.01 (alpha). The \textit{trimmedLloydShapes} algorithm is used with only five iterations and five
steps per initialization in the interests of a fast execution. A larger number of repetitions is suggested to obtain more optimal results. The default relative stopping criteria is 0.0001. Using this small value ensures that the algorithm stops when the decrease in the objective function is hardly visible. A larger stopping value could prematurely stop the algorithm (but the decrease in the objective function should have been taken into account).

```r
numClust <- 3 ; alpha <- 0.01 ; algSteps <- 5
niter <- 5 ; stopCr <- 0.0001

#suppressWarnings(RNGversion("3.5.0"))
#set.seed(2013)
res_kmProc <- trimmedLloydShapes(array3D, numIndiv, alpha, numClust,
                                 algSteps, niter, stopCr,
                                 verbose = FALSE)
```

The clustering results are obtained in the following way:

```r
clust_kmProc <- res_kmProc$asig
table(clust_kmProc)
#1 2 3
#19 18 12
```

The optimal prototypes and the trimmed individuals of the optimal iteration can be also identified:

```r
prototypes <- anthrCases(res_kmProc)
trimmed <- trimmOutl(res_kmProc)
```

In order to examine the differences between clusters for some key anthropometric dimensions, their boxplots can be represented. To do this, we need to identify the first 50 individuals in `sampleSpanishSurvey` and to remove the trimmed ones. Figure 5 (left) displays the boxplots for neck to ground measurement for the three clusters calculated.

```r
data_First50 <- sampleSpanishSurvey[1 : 50, ]
data_First50_notrimm <- data_First50[-trimmed, ]
boxplot(data_First50_notrimm$necktoground ~ as.factor(clust_kmProc),
        main = "Neck to ground")
```

In addition, Figure 5 (right) displays the projection on the $xy$-plane of the recorded points and mean shape for cluster 1.

```r
projShapes(1, array3D, clust_kmProc, prototypes)
legend("topleft", c("Registered data", "Mean shape"),
       pch = 1, col = 1:2, text.col = 1:2)
title("Procrustes registered data for cluster 1 \nwith its mean shape superimposed", sub = "Plane xy")
```
5.2. Archetypal analysis

We focus on the cockpit design problem. The accommodation of boundaries (our archetypes) ensures the accommodation of interior points in the cockpit. We use the USAFSurvey database. Again, as an illustrative example only the first 50 individuals are chosen. From the total variables, the six so-called cockpit dimensions are selected. They are thumb tip reach, buttock-knee length, popliteal height sitting, sitting height, eye height sitting and shoulder height sitting. We convert the variables from mm into inches in order to compare our results with those discussed in Zehner et al. (1993) (see Epifanio et al. 2013, for more details). The procedure followed for each problem is as follows:

Before computing archetypes and archetypoids, the data must be preprocessed. Firstly, depending on the problem, the user must decide whether or not to standardize the data. Secondly, it must be decided whether to use the Mahalanobis distance or a depth procedure to establish the percentage of the population to accommodate (see Epifanio et al. 2013, Section 2.2.2 for more details). Both actions are done with the following preprocessing function. In this case, the variables are standardized, as they measure different dimensions. This is indicated with the first TRUE argument in preprocessing. On the other hand, when designing a workspace, it has typically been a requirement that 95 percent of the relevant population are accommodated (value 0.95 in the third argument). Finally, the second TRUE (and fourth and final parameter) indicates that the Mahalanobis distance is used to remove the most extreme 5% data.

```r
USAFSurvey_First50 <- USAFSurvey[1 : 50, ]
variabl_sel <- c(48, 40, 39, 33, 34, 36)
USAFSurvey_First50_inch <- USAFSurvey_First50[,variabl_sel] / (10 * 2.54)
```
USAFSurvey_preproc <- preprocessing(data = USAFSurvey_First50_inch, 
  stand = TRUE, percAccomm = 0.95, 
mahal= TRUE)

Next, archetypes must be calculated. In the archetypes R package (Eugster, Leisch, and Seth 2014; Eugster and Leisch 2009) this is done with the stepArchetypes function, which executes the archetype algorithm repeatedly. However, this function standardizes the data by default and this option is not always desired. To overcome this, a new R function called stepArchetypesRawData has been written, which results from modifying and adapting the original stepArchetypes to apply the archetype algorithm to raw data. In this way, the archetype algorithm is run repeatedly from 1 to numArch archetypes. The user can decide how many archetypes are to be considered. We chose numArch equal to 10 because a larger number of boundary cases may overwhelm the designer and therefore be counterproductive.

The argument numRep specifies the number of repetitions of the algorithm. Choosing twenty repetitions ensures that the best possible archetypes are obtained.

#suppressWarnings(RNGversion("3.5.0"))
#set.seed(2010)
numArch <- 10 ; numRep <- 20
oldw <- getOption("warn")
options(warn = -1)
lass <- stepArchetypesRawData(data = USAFSurvey_preproc$data, 
  numArch=1:numArch, numRep = numRep, 
  verbose = FALSE)
options(warn = oldw)
screeplot(lass)

Once the archetypes are obtained, archetypoids are calculated either with the archetypoids function or with the stepArchetypoids function, which is a function based on stepArchetypes to execute the archetypoid algorithm repeatedly. According to the screeplot and following the elbow criterion, we compute three archetypoids (beginning from cand_ns, cand_α and cand_β sets of the nearest observations to the archetypes).

numArchoid <- 3
res_archoids_ns <- archetypoids(numArchoid, USAFSurvey_preproc$data, 
  huge = 200, step = FALSE, ArchObj = lass, 
  nearest = "cand_ns", sequ = TRUE)
res_archoids_alpha <- archetypoids(numArchoid, USAFSurvey_preproc$data, 
  huge = 200, step = FALSE, ArchObj = lass, 
  nearest = "cand_alpha", sequ = TRUE)
res_archoids_beta <- archetypoids(numArchoid, USAFSurvey_preproc$data, 
  huge = 200, step = FALSE, ArchObj = lass, 
  nearest = "cand_beta", sequ = TRUE)

boundaries_ns <- anthrCases(res_archoids_ns)
boundaries_alpha <- anthrCases(res_archoids_alpha)
boundaries_beta <- anthrCases(res_archoids_beta)
In this case, the \textit{cand}_ns, \textit{cand}_\alpha and \textit{cand}_\beta archetypoids match (although the \textit{cand}_ns, \textit{cand}_\alpha and \textit{cand}_\beta archetypes do not), so it is enough to represent a single percentile plot. To that end, the \texttt{percentilsArchetypoid} computes the percentils of the archetypoids for every column of the data frame.

\begin{verbatim}
df <- USAFSurvey_preproc$data
matPer <- t(sapply(1:dim(df)[2], percentilsArchetypoid, boundaries_ns, df, 0))
\end{verbatim}

Figure 6 shows the percentils of three archetypoids.

\begin{verbatim}
barplot(matPer, beside = TRUE, main = paste(numArchoid, " archetypoids", sep = ","),
       ylim = c(0, 100), ylab = "Percentile",
xlab = "Each bar is related to each anthropometric variable selected")
\end{verbatim}

Figure 6: Percentils of three archetypoids, beginning from the \textit{cand}_ns, \textit{cand}_\alpha and \textit{cand}_\beta sets for USAFSurvey. In this case, the \textit{cand}_ns, \textit{cand}_\alpha and \textit{cand}_\beta archetypoids coincide. The anthropometric variables selected are thumb tip reach, buttock-knee length, popliteal height sitting, sitting height, eye height sitting and shoulder height sitting.

6. Discussion

Procedures related to clustering, data depth and shape analysis (\texttt{trimowa}, \texttt{CCbiclustAnthropo}, \texttt{hipamAnthropom}, \texttt{TDDclust} and \texttt{kmeansProcrustes}) are aimed at defining optimal clothing size groups and both 1D and 3D central cases, which are actually representative statistical models or prototypes (and fit models in the case of \texttt{hipamAnthropom}) of the human body of the target population. The five aforementioned methodologies followed the same scheme. Firstly, the selected data matrix was segmented using a primary control dimension (bust or waist) and then a further segmentation was carried out using other secondary control
Figure 7: Practical implementation of the methodologies presented. These are three T-shirts designed from the prototypes obtained. The three T-shirts have the same bust size (primary dimension), but different measurements for other secondary dimensions. This method of designing and labelling may speed up the purchasing process, making it more satisfactory.

anthropometric variables. The number of size groups generally obtained with these methods was three, because this number is quite well aligned to clothing industry practice for the mass production of clothing, where the objective is to optimize sizes by addressing only the most profitable. This procedure can be translated into practice as shown in Figure 7.

For a given bust size, for example, 86-90 cm, the three T-shirts in Figure 7 were designed from the three prototypes obtained by any of the aforementioned methodologies. It can be seen that all three have the same bust size (primary dimension), but different measurements for other secondary dimensions (in this example, waist and neck-to-hip are selected for illustrative purposes). In a commercial situation, a woman in a store would directly select the T-shirts with her bust size and, out of all of them, she would finally choose the one with her same measurements for the other secondary variables. As a result, the customer would have quickly and easily found a T-shirt that fits perfectly. It is believed that the statistical methodologies presented here can speed up the purchasing process, making it more satisfactory. Figure 7 also shows a proposal for garment labelling. Clothing fit depends a lot on better garment labelling. Apparel companies should offer consumers truthful information that is not confusing on the garment sizes that they wish to offer for sale, so that people can easily recognise their size. In addition, the prototypes and fit models obtained can also be used to make more realistic store mannequins, thus helping to offer an image of healthy beauty in society, which is another very useful and practical application.

On the other hand, the two approaches based on archetypal and archetypoid analysis make it possible to identify boundary cases, that is to say, the individuals who present extreme body measurements. The basic idea is that accommodating boundary cases will accommodate the people who fall within the boundaries (less extreme population). This strategy is valuable in all human-computer interaction problems, for example, the design and packaging of plane cockpits or truck cabins. When designing workstations or evaluating manual work, it is common to use only a few human figure models (extreme cases, which would be our archetypoids) as virtual test individuals. These models are capable of representing people with a wide range of body sizes and shapes. Archetypal and archetypoid analysis can be very useful in improving industry practice when using human model tools to design products and work environments.
7. Conclusions

New three-dimensional whole-body scanners have drastically reduced the cost and duration of the measurement process. These types of systems, in which the human body is digitally scanned and the resulting data converted into exact measurements, make it possible to obtain accurate, reproducible and up-to-date anthropometric data. These databases constitute very valuable information to effectively design better-fitting clothing and workstations, to understand the body shape of the population and to reduce the design process cycle. Therefore, rigorous statistical methodologies and software applications must be developed to make the most of them.

This paper introduces a new R package called Anthropometry that brings together different statistical methodologies concerning clustering, the statistical concept of data depth, statistical shape analysis and archetypal analysis, which have been especially developed to deal with anthropometric data. The data used have been obtained from a 3D anthropometric survey of the Spanish female population and from the USAF survey. Procedures related to clustering, data depth and shape analysis are aimed at defining optimal clothing size groups and both central prototypes and fit models. The two approaches based on archetypal analysis are useful for determining boundary human models which could be useful for improving industry practice in workspace design.

The Anthropometry R package is a positive contribution to help tackle some statistical problems related to Ergonomics and Anthropometry. It provides a useful software tool for engineers and researchers in these fields so that they can analyze their anthropometric data in a comprehensive way.

Acknowledgments

The author gratefully acknowledges the many helpful suggestions of I. Epifanio and G. Ayala. The author would also like to thank the Biomechanics Institute of Valencia for providing us with the Spanish anthropometric data set and the Spanish Ministry of Health and Consumer Affairs for having commissioned and coordinated the “Anthropometric Study of the Female Population in Spain”. This paper has been partially supported by the following grants: TIN2009-14392-C02-01, TIN2009-14392-C02-02. The author would also like to thank the editors and referee for their very constructive suggestions, which led to a great improvement of both this paper and the Anthropometry package.

References


A. Algorithm listings

1. Set $k$, number of groups; $algSteps$, number of repetitions to find optimal medoids; and $niter$, number of repetitions of the whole algorithm.
2. Select $k$ starting points that will serve as seed medoids (e.g., draw at random $k$ subjects from the whole data set).
   
   for $r = 1 \rightarrow niter$
   
   for $s = 1 \rightarrow algSteps$
   
   Assume that $x_{i_1}, \ldots, x_{i_k}$ are the $k$ medoids obtained in the previous iteration.
   
   Assign each observation to its nearest medoid:
   
   $$d_i = \min_{j=1, \ldots, k} d(x_i, x_{i_j}), \quad i = 1, \ldots, n,$$
   
   and keep the set $H$ having the $\lceil n(1 - \alpha) \rceil$ observations with lowest $d_i$'s.
   
   Split $H$ into $H = \{H_1, \ldots, H_k\}$ where the points in $H_j$ are those closer to $x_{i_j}$ than to any of the other medoids.
   
   The medoid $x_{i_j}$ for the next iteration will be the medoid of observations belonging to group $H_j$.
   
   Compute
   
   $$F_0 = \frac{1}{\lceil n(1 - \alpha) \rceil} \sum_{j=1}^{k} \sum_{x_i \in H_j} d(x_i, x_{i_j}). \quad (11)$$
   
   if $s == 1$
   
   $F_1 = F_0$.
   
   Set $M$ the set of medoids associated to $F_0$.
   
   else
   
   if $F_1 > F_0$
   
   $F_1 = F_0$.
   
   Set $M$ the set of medoids associated to $F_0$.
   
   end if
   
   end if
   
   end for
   
   if $r == 1$
   
   $F_2 = F_1$.
   
   Set $M$ the set of medoids associated to $F_1$.
   
   else
   
   if $F_2 > F_1$
   
   $F_2 = F_1$.
   
   Set $M$ the set of medoids associated to $F_1$.
   
   end if
   
   end if
   
   end for
   
   return $M$ and $F_2$.

Algorithm 3: trimowa algorithm.

1. Set $k$, number of biclusters; $delta$ (initial default value 1); and $disac$, number of women who will not form part of any group (at the beginning, it is equally to the number of women belonging to each size).
2. The proportion of disaccommodated sample is prefixed to 1% per segment.
   
   while $disac > \text{ceiling}(0.01 \times \text{number of women belonging to the size})$
   
   biclust(data, method = BCCC(), delta = delta, alpha = 1.5, number = k)
   
   disac = number of women not grouped.
   
   delta = delta + 1
   
   end while

Algorithm 4: CCbiclustAnthropo algorithm.
1. Start with an initial partition $I^K_1 = \{I(1), \ldots, I(K)\}$ obtained with PAM. Set $\beta = \beta_{\text{init}}$.
2. Compute:
   - The $L_1$-MM of the $K$ clusters, $y_0(1), \ldots, y_0(K)$.
   - The silhouette widths, $sil_i$ for $i = 1, \ldots, n$.
   - The within cluster $L_1$ data depth of $x_i : i \in I(k)$,
     $D^w_i = D(x_i | k)$.
   - The between cluster $L_1$ data depth of $x_i$, $D^b_i = D(x_i | l)$ for $I(l)$ the nearest cluster of $x_i : i \in I(k)$.
   - The relative data depths, $ReD_i = D^w_i - D^b_i$.
   - The total value of the partition, $C(I^K_{\beta})$.
3. Compute $c_i = (1 - \lambda) sil_i + \lambda ReD_i$ for $i = 1, \ldots, n$. Remove $R = \{i : c_i \leq \alpha\}$. Let $R$ be the set of $n(1 - \alpha)$ non-trimmed points.
4. Identify a set of observations $S = \{i \in R : c_i \leq T\}$, where $T$ is a prefixed threshold.
5. For a random subset $E \subset S$, identify the nearest competing clusters. Define the partition with $E$ relocated as $I^K_{\beta}$.
6. Compute the value of the new partition $C(I^K_{\beta})$.
   - If $C(I^K_{\beta}) > C(I^K_{\beta_{\text{init}}})$ then set $I^K_{\beta} = I^K_{\beta_{\text{init}}}$.
   - Else if $C(I^K_{\beta}) \leq C(I^K_{\beta_{\text{init}}})$ then set $I^K_{\beta} = I^K_{\beta_{\text{init}}}$ with probability $Pr(\beta, \Delta(C))$, being $b$ a tuning parameter, and $\Delta(C) = C(I^K_{\beta}) - C(I^K_{\beta_{\text{init}}})$.
   - End if
7. Set $S = S \setminus E$ removing the subset $E$ form $S$.
8. Iterate 5-6 until set $S$ is empty.
9. For $j \in \{1, \ldots, n : x_j \in R\}$ compute $k_j = \arg\max\{c_k\}$ the value of $c_j$ as in Equation 6, assuming that the $j$-th point belongs to cluster $k$. Assign $x_j$ to the $k_j$-th cluster.
10. If no moves were accepted for the last $M$ iterations and $\beta < \infty$, set $\beta = \infty$ and iterate 2-8. If no moves were accepted for the last $M$ iterations and $\beta = \infty$. Otherwise, set $\beta = 2\beta$ and iterate 2-8.

Algorithm 5: TDDclust algorithm.

1. Initialization of the tree:
   Let the top cluster with all the elements be $T$.
   1.1. Initial clustering: Apply a PAM to $T$ with the number of clusters, $k_1$, provided by the INCA statistic with the following rule:
   - If $INCA_{k_1} < 0.2 \forall k_1$ then $k_1 = 3$
   - Else $k_1$ as the value preceding the first biggest slope decrease.
   1.2. Post-processing: Apply several partitioning or collapsing procedures to the $k_1$ clusters to try to improve the asw.
   2. Local HIPAM:
   while there are active clusters do
     Generation of the candidate clustering partition: PHASE I FOR HIPAMI
     Evaluation of the candidate clustering partition: PHASE II FOR HIPAMI
   end while

Algorithm 6: The HIPAMI method of the hipamAnthropom algorithm.
For each cluster, $P$, of a partition:

1. if $|P| \leq 2$ then
   STOP (P is a terminal node).
else
   if $\text{INCA}_{k_1} < 0.2 \forall k_1$ then
      STOP (P is a terminal node).
else
   2. Initial clustering: Apply a PAM to $P$ with the number of clusters, $k_1$, provided by the INCA statistic as the value preceding the first biggest slope decrease. An initial partition with $k_1$ clusters is obtained.
   3. Post-processing: Apply several partitioning or collapsing procedures to the $k_1$ clusters to try to improve the asw. The candidate partition, $C = \{C_1, \ldots, C_k\}$, from $P$ is obtained.

end if
end if

Algorithm 7: PHASE I FOR $HIPAM_{IMO}$.

Let the candidate clustering partition be $C = \{C_1, \ldots, C_k\}$ obtained from $P$.
1. Calculate the asw of $C$, $\text{asw}_C$.
2. For each $C_i$, generate a new partition using the steps 1.1. and 1.2. of the initialization of the tree and calculate its $SS_i$.
3. if $\text{MSS}(k) = \frac{1}{k} \sum_{i=1}^{k} SS_i < \text{asw}_C$ then
   C is accepted.
else
   C is rejected. STOP (P is a terminal node).
end if

Algorithm 8: PHASE II FOR $HIPAM_{IMO}$.

1. Given a vector of shapes $Z = ([Z_1], \ldots, [Z_k])$ with $[Z_i] \in \Sigma_m$, $i = 1, \ldots, k$, we minimize with respect to a $k$-partition $C = (C_1, \ldots, C_k)$, assigning each shape $([X_1], \ldots, [X_n])$ to the class whose centroid has the Procrustes minimum distance to it.
2. Given $C$, we minimize with respect to $Z$, taking $Z = ([\hat{\mu}_1], \ldots, [\hat{\mu}_k])$, and $[\hat{\mu}_i] i = 1, \ldots, k$, the Procrustes mean of shapes in cluster $C_i$.
3. Steps 1. and 2. are repeated until convergence of the algorithm.

Algorithm 9: $LloydShapes$ algorithm.

1. Given a centroid vector $Z = ([Z_1], \ldots, [Z_k])$ with $[Z_i] \in \Sigma_m$, $i = 1, \ldots, k$, we calculate the Procrustes distances of each shape $([X_1], \ldots, [X_n])$ to its closest centroid. The $n \alpha$ shapes with largest distances are removed, the $n(1 - \alpha)$ left are assigned to the class whose centroid has the minimum full Procrustes distance to it.
2. Given $C$, we minimize with respect to $Z$, taking $Z = ([\hat{\mu}_1], \ldots, [\hat{\mu}_k])$, and $[\hat{\mu}_i] i = 1, \ldots, k$, the Procrustes mean of shapes in cluster $C_i$.
3. Steps 1. and 2. are repeated until convergence of the algorithm.

Algorithm 10: $trimmedLloydShapes$ algorithm.
1. Given a centroid vector $Z = ([Z_1], \ldots, [Z_k])$ with $[Z_i] \in \Sigma_n$, for each shape $[X_j]$ ($j = 1, 2, \ldots, n$), find its closest and second closest cluster centroids, and denote these clusters by $C1(j)$ and $C2(j)$, respectively. Assign shape $[X_j]$ to cluster $C1(j)$.

2. Update the cluster centroids to be the Procrustes mean of the shapes contained within them.

3. Initially, all clusters belong to the live set.

4. This stage is called the optimal-transfer stage: Consider each shape $[X_j]$ ($j = 1, 2, \ldots, n$) in turn. If cluster $l$ ($l = 1, 2, \ldots, k$) is updated in the last quick-transfer stage, then it belongs to the live set throughout that stage. Otherwise, at each step, it is not in the live set if it has not been updated in the last $n$ optimal-transfer steps. Let shape $[X_j]$ be in cluster $l_1$. If $l_1$ is in the live set, do Step 4.a. Otherwise, do Step 4.b.

4.a. Compute the minimum of the quantity $R2 = \frac{n_l \|x_j - z_l\|^2}{n_l + 1}$, over all clusters $l$ ($l \neq l_1$, $l = 1, 2, \ldots, k$).

Let $l_2$ be the cluster with the smallest $R2$. If this value is greater than or equal to $\frac{n_l \|x_j - z_l\|^2}{n_l + 1}$, no reallocation is necessary and $C1(j)$ is the new $C2(j)$. Otherwise, shape $[X_j]$ is allocated to cluster $l_2$ and $C1(j)$ is the new $C1(j)$. Cluster centroids are updated to be the Procrustes means of shapes assigned to them if reallocation has taken place. The two clusters that are involved in the transfer of shape $[X_j]$ at this particular step are now in the live set.

4.b. This step is the same as Step (iv-a), except that the minimum $R2$ is only computed over clusters in the live set.

5. Stop if the live set is empty. Otherwise, go to Step 6. after one pass through the data set.

6. This is the quick-transfer stage: Consider each shape $[X_j]$ ($j = 1, 2, \ldots, n$) in turn. Let $l_1 = C1(j)$ and $l_2 = C2(j)$. It is not necessary to check shape $[X_j]$ if both clusters $l_1$ and $l_2$ have not changed in the last $n$ steps. Compute the values $R1 = \frac{n_{l_1} \|x_j - z_{l_1}\|^2}{n_{l_1} + 1}$ and $R2 = \frac{n_{l_2} \|x_j - z_{l_2}\|^2}{n_{l_2} + 1}$. If $R1$ is less than $R2$, shape $[X_j]$ remains in cluster $l_1$. Otherwise, switch $C1(j)$ and $C2(j)$ and update the mean shapes of clusters $l_1$ and $l_2$. The two clusters are also noteworthy for their involvement in a transfer at this step.

7. If no transfer took place in the last $n$ steps, go to Step 4. Otherwise, go to Step 6.

**Algorithm 11: HartiganShapes algorithm.**

Given the number of archetypes $k$:

1. Data preparation and initialization: scale data, add a dummy row and initialize $\beta$ in such a way that the constraints are fulfilled to calculate the starting archetypes $Z$.

2. Loop until RSS reduction is enough small or the number of iterations is reached (see Equation 9):

2.1. Find best $\alpha$ for the given set of archetypes $Z$.

2.2. Recompute archetypes $\tilde{Z}$.

2.3. Find best $\beta$ for the given set of archetypes $\tilde{Z}$.

2.4. Recalculate archetypes $Z$.

2.5. Compute residual sum of squares RSS.


**Algorithm 12: Archetypal algorithm.**
1. BUILD phase: look for a good initial set of \( k \) archetypoids from the \( n \) data points.
2. SWAP phase: for each archetypoid \( a \)
   (a) For each non-archetypoid data point \( o \).
      i. Swap \( a \) and \( o \) and compute the RSS of the configuration (see Equation 10, \( \alpha \) coefficients must be calculated).
3. Select the configuration with the lowest RSS.
4. Repeat steps 2 to 4 until there is no change in the archetypoids.

**Algorithm 13:** Archetypoid algorithm.

**Affiliation:**
Guillermo Vinué  
Department of Statistics and Operations Research  
Faculty of Mathematics  
University of Valencia  
46100 Burjassot, Spain  
E-mail: Guillermo.Vinue@uv.es  
URL: http://www.uv.es/vivigui