Progress in Artificial Intelligence (2023) 12:33–44
https://doi.org/10.1007/s13748-022-00294-2

REGULAR PAPER

Clustering: an R library to facilitate the analysis and comparison of cluster algorithms

Luis Alfonso Pérez Martos[1] · Ángel Miguel García-Vico[2] · Pedro González[1] · Cristóbal J. Carmona[1]

Received: 21 April 2022 / Accepted: 29 October 2022 / Published online: 17 December 2022

Abstract
Clustering is an unsupervised learning method that divides data into groups of similar features. Researchers use this technique to categorise and automatically classify unlabelled data to reveal data concentrations. Although there are other implementations of clustering algorithms in R, this paper introduces the Clustering library for R, aimed at facilitating the analysis and comparison between clustering algorithms. Specifically, the library uses relevant clustering algorithms from the literature with two objectives: firstly to group data homogeneously by establishing differences between clusters and secondly to generate a ranking between the algorithms and the attributes of a data set to obtain the optimal number of clusters. Finally, it is crucial to highlight the added value that the library provides through its interactive graphical user interface, where experiments can be easily configured and executed without requiring expert knowledge of the parameters of each algorithm.

Keywords Unsupervised learning · Cluster analysis · Clustering algorithms · Cluster quality

1 Introduction

Data clustering [1,2] is one of the main tasks within data mining. Its main aim is to explore the properties of data to generate groups of objects with similar features, where all the attributes are treated in the same way. This allows to extract relevant knowledge about the behaviour of the data. Some common application examples are market segmentation [3], social network analysis [4], or anomaly detection [5], among others. The result of the application of clustering is a concise data model where data are partitioned into groups, called clusters. The clusters must meet two conditions: they must be as different as possible, and the elements they contain must be as similar as possible. These conditions are satisfied by maximising, or minimising, some quality measures related to the clusters data distribution. In the literature, several measures to validate the quality of clusters can be found [6]. The first kind of measure is based on external metrics, which involves evaluating the results of a base algorithm in a pre-specified structure. This is imposed on a data set and reflects our intuition about the structure of the clustering of the data set. The second kind of measure is based on internal metrics, where the results of a clustering algorithm are evaluated in terms of the characteristics of the instances that belong to each cluster, e.g. the proximity matrix.

In the specialised literature, there are many proposals on clustering algorithms. Therefore, a review was carried out of the algorithms available in the R libraries. In fact, the *Clustering* library for R incorporates the most relevant algorithms from the Hierarchical and Partitioning sections of the Clustering Task View[1] based on the number of citations and downloads of the different algorithms. The libraries that implement these algorithms and that are most cited in the Partitional Clustering section are apcluster [8] and cluster [9], while in Hierarchical Clustering the most cited library are: cluster [9], ClusterR [19], and pvclust [10]. In addition,

✉ Luis Alfonso Pérez Martos
lapm0001@gmail.com

Ángel Miguel García-Vico
agvico@decsai.ugr.es

Pedro González
pglez@ujaen.es

Cristóbal J. Carmona
ccarmona@ujaen.es

[1] Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Jaen, Jaén, Spain

[2] Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Granada, Spain

[1] https://cran.r-project.org/web/views/Cluster.html.

Table 1 Comparison of the functionality provided by the packages

Library	Algorithms used	External and internal measures	Validate external measure using the attributes data set	Comparison by algorithm	Comparison by metrics	GUI
pvclust	Pvclust and Pvpick	N/A	N/A	N/A	N/A	N/A
cluster	Agnes, Clara, Diana, Fanny, Mona, Pam	Silhouette	N/A	N/A	N/A	N/A
apcluster	Aggexcluster, Apclusterk, Apclusterl	N/A	N/A	N/A	N/A	N/A
ClusterR	Ap_affinity_propagation, Gmm, Kmeans_arma, Kmeans_rcpp, Mininbatchkmeans	Rand_index, adjusted_rand_index, fowlkes_Mallows_index, mirkin_metric, purity, entropy	N/A	N/A	N/A	N/A
amap	hcluster, Kmeans	N/A	N/A	N/A	N/A	N/A
Clustering	Hcluster, Aggexcluster, Apclusterk, Agnes, Clara, Daisy, Diana, Fanny, Mona, Pam, Gmm, Kmeans_arma, Kmeans_rcpp, Minibatchkmeans, Pvclust, Pvpick	Entropy, Variation Information, Precision, Recall, F-measure, Fowlkes mallows, Silhouette, Dunn, and Connectivity	Yes	Algorithm, Cluster, distance measure, external, and internal measure	Yes	Yes

library amap [7] is also included in our Clustering library because it includes the widely used Kmeans algorithm.

Table 1 shows a comparison of the features offered by the libraries included in the package and our Clustering package. The description of each column of Table 1 is detailed as follows:

- *External and Internal Measures* This column indicates whether the package implements any novel or traditional quality measures. If so, the name of the measures implemented is reflected.
- *Validate external measure using the attributes data set* Traditionally, external validation methods make use of external information to assess quality. To validate external information, it uses the values of the attributes of the data set.
- *Comparison by algorithm* A useful functionality is to be able to compare sets of algorithms by quality measures, and the number of clusters to evaluate the results. In this column, we indicate the fields by which it is possible to compare the results.
- *Comparison by metrics* This column indicates whether comparison by quality measures is possible.

Unfortunately, not much software implements quality criteria in clustering to measure and analyse the quality of different algorithms. In particular, there are several problems associated with current libraries:

- It is not possible to work with different input formats.
- The algorithm mainly focuses on the distribution of the data in the clusters, but they do not show their quality.
- It is not possible to work with a set of data sets, so it is not easy to compare different algorithms.
- Few libraries include a graphical user interface (GUI).

To address these problems, this paper presents the *Clustering* library for R. It is a library that allows for the comparison of multiple clustering algorithms simultaneously while assessing the quality of the clusters extracted. The purpose of this library is to evaluate a set of data sets to determine which attributes are the most suitable for obtaining clusters of interest. Therefore, assessment of the clusters created, how they have been distributed, whether the distributions are uniform, and how they have been categorised from the data can be performed. In addition, the library offers the added value of an easy-to-use and highly helpful GUI, which allows experiments to be quickly set up and run with no need for the user to know the parameters of each algorithm, facilitating the analysis and comparison of the results provided by different algorithms.

The advantages provided by the Clustering library compared to other packages are:

- This library can work with a data set and with a directory of several data sets.
- Putting all this together, the main advantage and novelty appears: users can run an experimental study with

multiple algorithms, measures, number of clusters, and similarity measures where the comparison between the algorithms is based on internal and external measures. In addition, the measurement of the external quality measures in order to determine the optimal number of clusters is performed automatically for each of the attributes in the data set as a parameter.
- Finally, another strong point is the GUI facilitates the use of the library by the user. Nowadays, there are only two libraries that implement GUI (ProjectionBasedClustering and VarSelLCM) to the best of our knowledge. In addition, VarSelLCM does not work, while the ProjectionBasedClustering library does not allow the comparison of algorithms with quality measures.

The structure of this contribution is as follows: firstly, in Sect. 2 a library presentation is given together with the definition of the architecture and functionalities; in Sect. 3, an example of the use of the library is described; and finally, Section 4 outlines the conclusions reached.

2 Software description

The *Clustering* library is implemented in R [11]. R can work with practically any type of data, begin multiplatform, and include advanced graphics capabilities. It is also constantly being improved by its fairly large community, the development of new functionalities and bug fixes. Among the properties of the *Clustering* library, it is worth highlighting that it uses:

- *5 Libraries* amap [7], apcluster [8], cluster [9], ClusterR [19], and pvclust [10].
- *16 Algorithms* aggExCluster [8], agnes [13], apcluster [8], clara [14], daisy [15], diana [16], fanny [17], gmm [19], hcluster [18], KMeans_arma [19], KMeans_rcpp [19], MiniBatchKmeans [19], mona [20], pam [21], pvpick [10], and pvclust [10].
- *6 External measurements* Entropy [23], Variation Information [24], Precision [27–29], Recall [27–29], F-measure [25], and Fowlkes mallows [26].
- *3 Internal measurements* Silhouette [30], Dunn [6], and Connectivity [31].

However, the library can be easily extended with new algorithms and user-specified metrics making use of the *clustering* object. In R, it is possible to implement an object-oriented programming style that implements a number of generic methods. This style is called S3. The methods implements are print(), summary(), and plot(). In addition, another essential functionality incorporated concerning existing solutions in CRAN[2] is the possibility of sorting, filtering, and exporting the results for further analysis. Regarding the source of the data, it is important to remark that the library accepts different formats of input data sets, such as CSV, KEEL, ARFF (Weka), and `data.frame` objects, highlighting the possibility to work with directories containing different data sets instead of working with a single data set. This allows the execution of multiple data sets with a single configuration, saving a lot of time and effort. Finally, the results can be easily exported to LaTeX to facilitate their incorporation into reports and documents. Note the non-inclusion of the RWeka and RKeel libraries in the package for not to increase the number of dependencies since the code needed to read files of these formats is easy to develop.

2.1 Software architecture

The Clustering library imports a series of libraries that are used internally for processing. These libraries are represented in Fig. 1. In the image, the grey squares represent all the imports of the Clustering library. The red database represents the dependency with the R library. And the orange box represents the Clustering library. The dependency on R is required.

As mentioned previously, the library includes 16 clustering algorithms from different libraries. Specifically, the algorithms included in the *Clustering* library are given in Table 2.

All these algorithms are wrapped up in the *Clustering* library and can be executed through of the `clustering()` method, which is the core of this library. This method is in charge of several aspects:

1. To properly handle the parameters of each method.
2. To run in parallel the chosen algorithms and to collect their results for each data set.
3. To assess the quality of the clusters extracted and to perform a ranking concerning them.
4. To display and allow easy management of all this information in a user-friendly way.

When the `clustering()` method is executed, it returns an object called `clustering`. This object contains information about: what algorithms have been executed, metrics used, flags to indicate if it has internal and external measurements, and the results of the execution. The library provides several helper functions to evaluate and rank the results extracted, plot, and export them, to perform further analysis. These functions can be evaluated using the `clustering` object. Finally, all this functionality is accessible through the GUI. To run the GUI the library has a method called

[2] https://cran.r-project.org/web/packages/index.html.

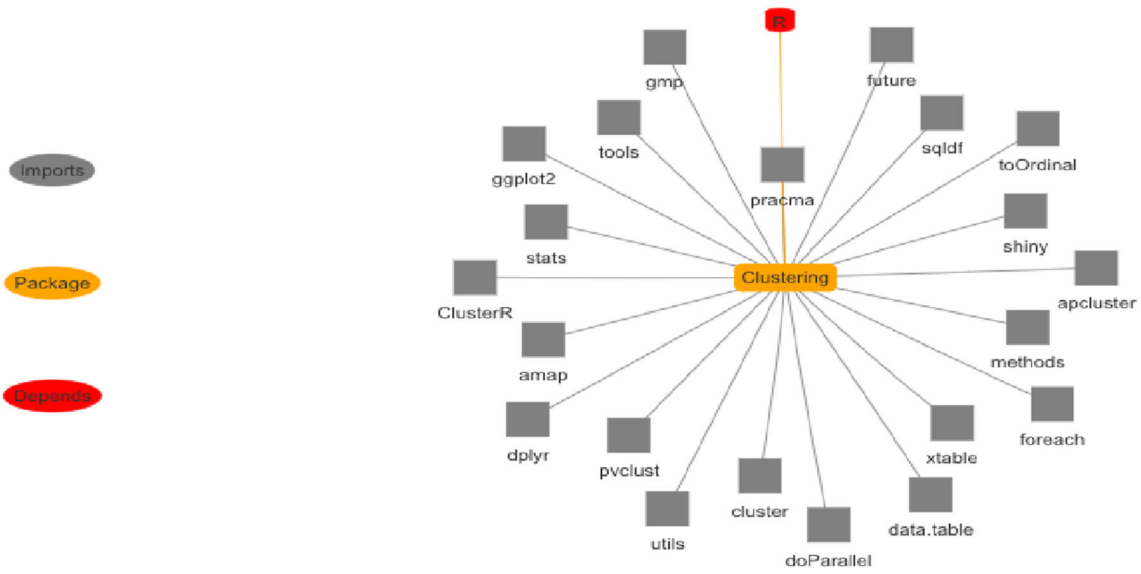

Fig. 1 Graphic with dependencies and imports of the Clustering library

`appClustering()`. The library is available in the Cran directory[3].

2.2 Software functionalities

The *Clustering* library provides several functionalities to handle all the previously described components:

- `clustering()`: It is the core function of the library. The parameters of the method are as follows:
  - *Path* The file path. It is only allowed to use path or df but not both at the same time. The directory must contain a list of files with the data sets to be uploaded. Allowed formats are CSV, KEEL, ARFF (Weka), and data.frame.
  - *df* Data matrix or data frame, or similarity matrix. It is only allowed to use path or df but not both at the same time. Through this parameter, it is possible to load a data set. R has several utility packages for reading data sets. The best known is utils [18].
  - *Packages* String array with the libraries that import algorithms. The imported libraries are: pvclust, cluster, apcluster, ClusterR, and amap. By default the system runs all packages. It is only allowed to use packages or algorithms but not both at the same time.
  - *Algorithm* An array of strings with the algorithms implemented in the imported libraries. It is only allowed to use packages or algorithms but not both at the same time. The algorithms are: aggExCluster, agnes, apcluster, clara, daisy, diana, fanny, gmm, hcluster, kmeans_arma, kmeans_rcpp, mini_kmeans, mona, pam, pvpick, and pvclust.
  - *Min* An integer with the minimum number of clusters. The default value is 3.
  - *Max* An integer with the maximum number of clusters. The default value is 4.
  - *Metrics* Array of strings with quality measures. The measures are: Connectivity, Dunn, Entropy, F-measure, Fowlkes mallows, Precision, Recall, Silhouette, and Variation Information. It is required to indicate an external measure.

  As a result, it generates the `clustering` object. The library allows sorting and filtering operations for further processing of the results. The '[' operator makes use of the filter method of the `dplyr` library [22].
- External metrics. These methods are responsible for assessing the quality of the extracted clusters using the attributes in the data set as target. The following methods receive the `clustering` object as an input parameter. The methods return the best algorithms, distance measures, and the number of clusters based on the quality measures. For the methods evaluate_best_validation_external_by_metrics and result_external_algorithm_by_metric in addition to the clustering object, it is necessary to indicate the external quality measure.
  - `best_ranked_external_metrics()`: The execution of this method allows to obtain the attributes with better behaviour by algorithm, measure of distance, and number of clusters in a ranking way.

[3] https://cran.r-project.org/web/packages/Clustering/index.html.

Table 2 Clustering algorithms integrated into the *Clustering* library

Library	Algorithm	Distance	Description
amap	hcluster [18]	Euclidean	Hierarchical cluster based in Kmeans
apcluster	aggExCluster [8]	Euclidean	Exemplar-based agglomerative clustering
apcluster	apclusterK [8]	Euclidean, Manhattan, Minkowski	Affinity propagation for pre-defined number of clusters
cluster	Agnes [13]	Euclidean, Manhattan	Agglomerative nesting
cluster	Clara [14]	Euclidean, Manhattan	Clustering large applications
cluster	Daisy [15]	Euclidean, Manhattan, Gower	The main feature of daisy is its ability to handle other variable types as well e.g. nominal, ordinal, (a)symmetric binary
cluster	Diana [16]	Euclidean	Divisive analysis clustering
cluster	Fanny [17]	Euclidean, Manhattan	Fuzzy analysis clustering
cluster	Mona [20]	–	Monothetic Analysis Clustering of binary variables
cluster	pam [21]	Euclidean, Manhattan	Partitioning around medoids
ClusterR	gmm [19]	Euclidean, Manhattan	Gaussian mixture model clustering
ClusterR	KMeans_arma [19]	–	*k*-means using the Armadillo library
ClusterR	KMeans_rcpp [19]	–	*k*-means using RcppArmadillo
ClusterR	MiniBatchKmeans [19]	–	Mini-batch-*k*-means using RcppArmadillo
pvclust	pvclust [10]	Correlation	Function pvclust conducts multiscale bootstrap
pvclust	pvpick [10]	Euclidean, Correlation	Function pvclust conducts multiscale bootstrap

  - `evaluate_best_validation_external_by_metrics()`: This method groups the data by algorithm and distance measure, instead of obtaining the best attributes from the data set.
  - `evaluate_validation_external_by_metrics()`: It groups the results by algorithms.
  - `result_external_algorithm_by_metric()`: It is used for obtaining the results of a given algorithm grouped by number of clusters.
- Internal metrics. Incorporates the same set of methods and input parameters mentioned above for the external metrics:
  - `best_ranked_internal_metrics()`.
  - `evaluate_best_validation_internal_by_metrics()`.
  - `evaluate_validation_internal_by_metrics()`.
  - `result_internal_algorithm_by_metric()`.
- `plot_clustering()`: This method represents the results of clustering in a bar chart. The graph represents the distribution of the algorithms based on the number of partitions and the evaluation metrics employed, which can be internal or external.
- `export_external_file()`: It exports the results of external metrics in LaTeX format, for integration into documents with that format.
- `export_internal_file()`: This method is used in order to export in LaTeX format the results of the internal metrics.

3 Example of use of `Clustering` library

This section presents an illustrative example to show the performance of the *Clustering* library. At this point, it is time to work with the library to examine the real potential it has by evaluating algorithms, internal and external quality measures as well as being able to work with a range of clusters that allows us to select the best algorithm from the configured data. The operation of the library consists of performing parallel runs for each of the attributes of the data set. To execute the `clustering()` method, it is necessary to indicate to the package through external parameters the quality measures, the number of clusters, the algorithms, and the data set or set of data sets. For the simulation, it is used Precision [27–29] and Recall [27–29] as external quality measures and Silhouette [30] as internal quality measures. The Precision [27–29] is the ratio $\frac{t_p}{(t_p+f_p)}$ where t_p is the number of true positives and f_p the number of false positives. The Precision [27–29] is intuitively the ability of the classifier not to label as positive a sample that is negative. The best value is 1 and the worst value is 0. The Recall [27–29] is the ratio $\frac{tp}{(tp+fn)}$ where t_p is the number of true positives and f_n the number of false negatives. The best value is 1 and the worst value is 0. The value of the Silhouette [30] coefficient is between $[-1, 1]$. A score of 1 denotes the best meaning that the data point is very compact within the cluster to which it belongs and far away from the other clusters. The worst value is -1. Values near 0 denote overlapping clusters. The data set used is called Stock and is included in the library. This data set contains the daily stock price data of ten aerospace companies from January 1998 to October 1991. The algorithms used are clara [9] and kmeans_rcpp [19]. Finally, it is required indicate the cluster number. It is also possible to work with a range of clusters. The range used for this study is set up between 3 and 5.

Tables 3 and 4 show the results obtained after the execution of the `clustering()` method. In this table, *Algorithm* indicates the name of the algorithm, *Distance* represents the distance measurement employed (for methods with a single metric), *Clusters* is the number of clusters used in that execution, and *Data* is the data set analysed. The *Clustering* library tries to find the attribute that provides the best partitioning of the data about the external metrics used. It is mandatory to indicate an external measure in the clustering() method. To achieve this, it selects each attribute in the data set as a target and calculates its associated external metrics. This is given in Tables 3 and 4 in column *Var*, which reflects which attribute in the data set has been used as the target. The remaining columns presented below, i.e. *Time*, *Precision*, and *Recall*, show the value of the external metrics employed concerning using that attribute *Var* as the target.

Once the complete analysis is performed, the *Clustering* library is ready to summarise the data. The objective of this summary is twofold: on the one hand, it tries to determine the optimum number of clusters of each algorithm according to the results extracted; on the other hand, the attribute that shows the best influence on the results is also determined. The methods `best_ranked_external_metrics()` and `best_ranked_internal_metrics()` have been employed to achieve this. These methods need a clustering object as a parameter. This object is obtained with the output of the clustering() method. The results are given in Tables 5 and 6.

It is important to highlight that new columns ending with *Att* appear in Tables 5 and 6, showing the attribute of the data set with the greatest influence on the metrics analysed.

Finally, it is possible to discover situations where it is necessary to know which distance measurement best suits the external and internal metrics. The main purpose is to reduce and facilitate the analysis and study of several algorithms for multiple data sets. In this way, the *Clustering* library incorpo-

Table 3 Results obtained by the *Clustering* library for the kmeans_rcpp algorithm

Algorithm	Distance	Clusters	Data	Var	Time	Precision	*Recall*
kmeans_rcpp	–	3	Stock	1st	0.0060	0.0269	0.7876
kmeans_rcpp	–	3	Stock	2nd	0.0060	0.0267	0.7642
kmeans_rcpp	–	3	Stock	3rd	0.0070	0.0189	0.7363
kmeans_rcpp	–	3	Stock	4th	0.0080	0.0149	0.6969
kmeans_rcpp	–	3	Stock	5th	0.0080	0.0132	0.6479
kmeans_rcpp	–	3	Stock	6th	0.0100	0.0120	0.6342
kmeans_rcpp	–	3	Stock	7th	0.0100	0.0115	0.6214
kmeans_rcpp	–	3	Stock	8th	0.0120	0.0093	0.6026
kmeans_rcpp	–	3	Stock	9th	0.0120	0.0063	0.5796
kmeans_rcpp	–	3	Stock	10th	0.0130	0.0058	0.5554
kmeans_rcpp	–	4	Stock	1st	0.0060	0.0293	0.7871
kmeans_rcpp	–	4	Stock	2nd	0.0070	0.0281	0.7076
kmeans_rcpp	–	4	Stock	3rd	0.0080	0.0185	0.6490
kmeans_rcpp	–	4	Stock	4th	0.0080	0.0172	0.6392
kmeans_rcpp	–	4	Stock	5th	0.0080	0.0139	0.5901
kmeans_rcpp	–	4	Stock	6th	0.0090	0.0133	0.5857
kmeans_rcpp	–	4	Stock	7th	0.0091	0.0123	0.5654
kmeans_rcpp	–	4	Stock	8th	0.0109	0.0105	0.5624
kmeans_rcpp	–	4	Stock	9th	0.0120	0.0066	0.5559
kmeans_rcpp	–	4	Stock	10th	0.0130	0.0065	0.5488
kmeans_rcpp	–	5	Stock	1st	0.0069	0.0331	0.8385
kmeans_rcpp	–	5	Stock	2nd	0.0070	0.0290	0.7092
kmeans_rcpp	–	5	Stock	3rd	0.0070	0.0222	0.6111
kmeans_rcpp	–	5	Stock	4th	0.0080	0.0189	0.5851
kmeans_rcpp	–	5	Stock	5th	0.0080	0.0187	0.5827
kmeans_rcpp	–	5	Stock	6th	0.0089	0.0152	0.4954
kmeans_rcpp	–	5	Stock	7th	0.0090	0.0145	0.4761
kmeans_rcpp	–	5	Stock	8th	0.0100	0.0139	0.4561
kmeans_rcpp	–	5	Stock	9th	0.0110	0.0087	0.4531
kmeans_rcpp	–	5	Stock	10th	0.0130	0.0085	0.4495

rates multiple methods, as given in Tables 7 and 8 for external metrics and Tables 9 and 10 for internal ones.

Clustering library incorporates other methods such as `plot()`. It shows a graphical representation of the distribution of the data by cluster and algorithm as shown in Fig. 2.

So far, this illustrative example has been performed at console level, but thanks to the `appClustering()` method, users can graphically interact with the library using its GUI. Specifically, a browser with the interface is available to facilitate the execution and analysis by any type of user (both novel and expert). There is a layout with a header, a side menu, and the main menu, as shown in Fig. 3. In the header, the user can choose to display results numerically or in plots in the same way as presented in Fig. 3. In the left menu, the user can see the different parameters with which can be run the algorithms. Finally, in the main menu, the result of the execution of the `clustering()` method is presented.

The operation of the application is simple, and Fig. 4 shows a step-by-step explanation. In more detail, Fig. 4 presents two ovals marked in red. The first one represents the header menu, while the second one represents the library configuration parameters. Rectangles are each of the configuration parameters. The parameters will explain as follows:

- Marked in red, the user can choose whether to work with test data sets or indicate a directory of data set files to be processed.
- In blue, the libraries that implement the clustering algorithms mentioned throughout the paper can be selected. It is possible to mark all the libraries or a subset of them. All the algorithms implemented within the selected library are marked when a library is marked.
- In yellow, the algorithms implemented by the libraries are shown. Multiple algorithms can be selected.

Table 4 Results obtained by the *Clustering* library for the clara algorithm

Algorithm	Distance	Clusters	Data	Var	Time	Precision	Recall
Clara	Euclidean	3	Stock	1st	0.0050	0.0273	0.7945
Clara	Euclidean	3	Stock	2nd	0.0060	0.0261	0.7386
Clara	Euclidean	3	Stock	3rd	0.0060	0.0180	0.7267
Clara	Euclidean	3	Stock	4th	0.0060	0.0150	0.7089
Clara	Euclidean	3	Stock	5th	0.0080	0.0132	0.6456
Clara	Euclidean	3	Stock	6th	0.0080	0.0121	0.6421
Clara	Euclidean	3	Stock	7th	0.0087	0.0115	0.6378
Clara	Euclidean	3	Stock	8th	0.0090	0.0092	0.6092
Clara	Euclidean	3	Stock	9th	0.0100	0.0063	0.5776
Clara	Euclidean	3	Stock	10th	0.0120	0.0057	0.5674
Clara	Euclidean	4	Stock	1st	0.0050	0.0292	0.7841
Clara	Euclidean	4	Stock	2nd	0.0050	0.0280	0.7052
Clara	Euclidean	4	Stock	3rd	0.0070	0.0186	0.6536
Clara	Euclidean	4	Stock	4th	0.0079	0.0171	0.6315
Clara	Euclidean	4	Stock	5th	0.0080	0.0139	0.5873
Clara	Euclidean	4	Stock	6th	0.0080	0.0133	0.5832
Clara	Euclidean	4	Stock	7th	0.0080	0.0122	0.5702
Clara	Euclidean	4	Stock	8th	0.0100	0.0106	0.5573
Clara	Euclidean	4	Stock	9th	0.0110	0.0066	0.5547
Clara	Euclidean	4	Stock	10th	0.0230	0.0065	0.5473
Clara	Euclidean	5	Stock	1st	0.0060	0.0362	0.6554
Clara	Euclidean	5	Stock	2nd	0.0070	0.0299	0.6351
Clara	Euclidean	5	Stock	3rd	0.0070	0.0234	0.6090
Clara	Euclidean	5	Stock	4th	0.0070	0.0195	0.5784
Clara	Euclidean	5	Stock	5th	0.0070	0.0159	0.5725
Clara	Euclidean	5	Stock	6th	0.0080	0.0155	0.5033
Clara	Euclidean	5	Stock	7th	0.0080	0.0145	0.4399
Clara	Euclidean	5	Stock	8th	0.0090	0.0128	0.4375
Clara	Euclidean	5	Stock	9th	0.0119	0.0091	0.4092
Clara	Euclidean	5	Stock	10th	0.0220	0.0084	0.3983
Clara	Manhattan	3	Stock	1st	0.0050	0.0274	0.8299
Clara	Manhattan	3	Stock	2nd	0.0060	0.0260	0.7504
Clara	Manhattan	3	Stock	3rd	0.0060	0.0180	0.7291
Clara	Manhattan	3	Stock	4th	0.0061	0.0146	0.7032
Clara	Manhattan	3	Stock	5th	0.0070	0.0127	0.6855
Clara	Manhattan	3	Stock	6th	0.0070	0.0118	0.6292
Clara	Manhattan	3	Stock	7th	0.0070	0.0115	0.6179
Clara	Manhattan	3	Stock	8th	0.0090	0.0090	0.5960
Clara	Manhattan	3	Stock	9th	0.0100	0.0061	0.5765
Clara	Manhattan	3	Stock	10th	0.0100	0.0056	0.5620
Clara	Manhattan	4	Stock	1st	0.0050	0.0319	0.8174
Clara	Manhattan	4	Stock	2nd	0.0059	0.0291	0.7578
Clara	Manhattan	4	Stock	3rd	0.0070	0.0207	0.6455
Clara	Manhattan	4	Stock	4th	0.0070	0.0158	0.6378
Clara	Manhattan	4	Stock	5th	0.0080	0.0157	0.6375

Table 4 continued

Algorithm	Distance	Clusters	Data	Var	Time	Precision	Recall
Clara	Manhattan	4	Stock	6th	0.0085	0.0138	0.5615
Clara	Manhattan	4	Stock	7th	0.0090	0.0137	0.5373
Clara	Manhattan	4	Stock	8th	0.0090	0.0130	0.5229
Clara	Manhattan	4	Stock	9th	0.0109	0.0079	0.4768
Clara	Manhattan	4	Stock	10th	0.0120	0.0075	0.4491
Clara	Manhattan	5	Stock	1st	0.0070	0.0370	0.6744
Clara	Manhattan	5	Stock	2nd	0.0070	0.0285	0.6701
Clara	Manhattan	5	Stock	3rd	0.0070	0.0229	0.6111
Clara	Manhattan	5	Stock	4th	0.0070	0.0202	0.5624
Clara	Manhattan	5	Stock	5th	0.0080	0.0165	0.5548
Clara	Manhattan	5	Stock	6th	0.0090	0.0151	0.5117
Clara	Manhattan	5	Stock	7th	0.0090	0.0150	0.4515
Clara	Manhattan	5	Stock	8th	0.0100	0.0133	0.4223
Clara	Manhattan	5	Stock	9th	0.0120	0.0092	0.4142
Clara	Manhattan	5	Stock	10th	0.0230	0.0089	0.4081

Table 5 A summary of the external measurements sorted by algorithm, distance measure, and number of clusters

Algorithm	Distance	Clusters	Data	Var	Time	Precision	Recall	TimeAtt	PrecisionAtt	RecallAtt
kmeans_rcpp	–	3	Stock	1st	0.0060	0.0269	0.7876	10th	8th	2nd
kmeans_rcpp	–	4	Stock	1st	0.0060	0.0293	0.7871	10th	8th	2nd
kmeans_rcpp	–	5	Stock	1st	0.0069	0.0331	0.8385	10th	8th	2nd
Clara	Euclidean	3	Stock	1st	0.0050	0.0273	0.7945	10th	8th	2nd
Clara	Euclidean	4	Stock	1st	0.0050	0.0292	0.7841	10th	8th	2nd
Clara	Euclidean	5	Stock	1st	0.0060	0.0362	0.6554	9th	8th	2nd
Clara	Manhattan	3	Stock	1st	0.0050	0.0274	0.8299	10th	3rd	2nd
Clara	Manhattan	4	Stock	1st	0.0050	0.0319	0.8174	10th	8th	2nd
Clara	Manhattan	5	Stock	1st	0.0070	0.0370	0.6744	10th	8th	2nd

Table 6 A summary of the internal measurements sorted by algorithm, distance measure, and number of clusters

Algorithm	Distance	Clusters	Data	Var	Time	Silhouette	TimeAtt	SilhouetteAtt
kmeans_rcpp	–	3	Stock	1st	0.0279	0.44	6th	1st
kmeans_rcpp	–	4	Stock	1st	0.0269	0.47	7th	1st
kmeans_rcpp	–	5	Stock	1st	0.0269	0.46	6th	1st
Clara	Euclidean	3	Stock	1st	0.0269	0.43	1st	1st
Clara	Euclidean	4	Stock	1st	0.0269	0.47	5th	1st
Clara	Euclidean	5	Stock	1st	0.0269	0.44	1st	1st
Clara	Manhattan	3	Stock	1st	0.0309	0.47	3rd	1st
Clara	Manhattan	4	Stock	1st	0.0309	0.44	1st	1st
Clara	Manhattan	5	Stock	1st	0.0309	0.43	1st	1st

Table 7 Classification of the result by algorithm and distance measures through the `evaluate_best_validation_external_by_metrics()` method

Algorithm	Distance	Clusters	Time	Precision	Recall	TimeAtt	PrecisionAtt	RecallAtt
kmeans_rcpp	–	5	0.0069	0.0331	0.8385	10th	8th	2nd
Clara	Euclidean	5	0.0060	0.0362	0.6554	9th	8th	2nd
Clara	Manhattan	5	0.0070	0.0370	0.6744	10th	8th	2nd

Table 8 Results of the algorithms on external metrics through the `result_external_algorithm_by_metric()` method

Algorithm	Distance	Clusters	Time	Precision	Recall	TimeAtt	PrecisionAtt	RecallAtt
kmeans_rcpp	–	5	0.0069	0.0331	0.8385	10th	8th	2nd
Clara	Manhattan	5	0.0070	0.0370	0.6744	10th	8th	2nd

Table 9 Classification of the result by algorithm and distance measures through the *evaluate_best_validation_internal_by_metrics*() method

Algorithm	Distance	Clusters	Time	Silhouette	TimeAtt	SilhouetteAtt
kmeans_rcpp	kmeans_rcpp	4	0.0269	0.47	7th	1st
Clara	Euclidean	4	0.0269	0.47	5th	1st
Clara	Manhattan	3	0.0309	0.47	3rd	1st

Table 10 Results of the algorithms on internal metrics through the *result_internal_algorithm_by_metric*() method

Algorithm	Distance	Clusters	Time	Silhouette	TimeAtt	SilhouetteAtt
kmeans_rcpp	kmeans_rcpp	4	0.0269	0.47	7th	1st
Clara	Euclidean	4	0.0269	0.47	5th	1st

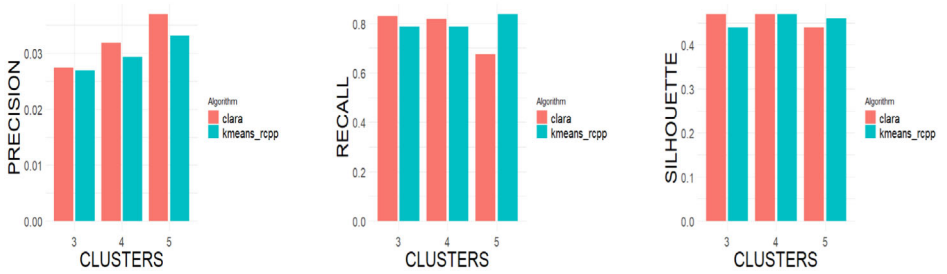

Fig. 2 Graphical representation of the external and internal metrics by a number of clusters for the algorithms indicated in the execution

(a) Precision by number of clusters for each algorithm.

(b) Recall by number of clusters for each algorithm.

(c) Silhouette by number of clusters for each algorithm.

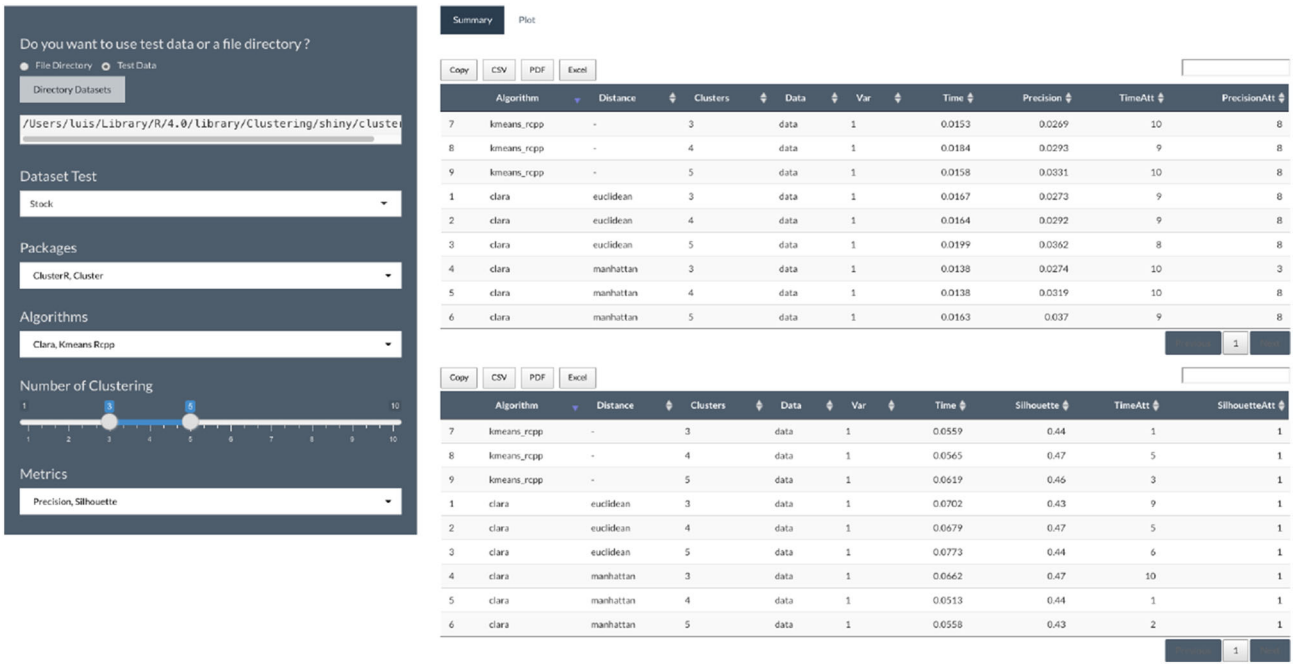

Fig. 3 Clustering app user interface

Fig. 4 Clustering app user interface

- In green, it is possible to choose the desired number of clusters. It is possible to indicate ranges or select only one cluster by positioning the maximum and minimum on the same value.
- Finally, in violet the evaluation metrics used when validating the clusters are chosen.

4 Conclusions

This paper presents a novelty library for R to facilitate the execution and analysis of clustering algorithms available in CRAN. Specifically, the *Clustering* library is emphasised on the metrics to measure the quality of clusters. In addition, *Clustering* library offers the following advantages: it allows to analyse one or multiple data sets simultaneously using different algorithms, to use multiple distance measures in the executions, to work with a range of clusters, to incorporate quality metrics to analyse the most relevant attributes for the data set, as well as to be able to use user-friendly graphical interface that facilitates the use of the library with no need of in-depth knowledge of R. As future work, the quality of the clusters is being improved using classification techniques such as hyperrectangle with genetics (CHC), to reduce the number of clusters and improve quality measures.

Acknowledgements This work was supported by the Spanish Ministry of Economy and Competitiveness under the project PID2019-107793GB-I00.

Funding Funding for open access publishing: Universidad de Jaén/CBUA

References

1. Kaur, M., Garg, S.: Survey on clustering techniques in data mining for software engineering. Int. J. Adv. Innov. Res. **5**(3), 238–243 (2014)
2. Steinbach, M., Karypis, G., Kumar, V.: A comparison of document clustering techniques. In: Proceedings of the International KDD Workshop on Text Mining, p. 6 (2000)
3. Dolnicar, S.: A Review of Data-Driven Market Segmentation in Tourism. J. Travel Tour. Mark. **12**, 1–22 (2002)
4. Garg, N., Rani, R.: Analysis and visualization of Twitter data using k-means clustering. In: 2017 International Conference On Intelligent Computing And Control Systems (ICICCS), pp. 670–675 (2017)
5. Pandeeswari, N., Kumar, G.: Anomaly detection system in cloud environment using fuzzy clustering based ANN. Mob. Netw. Appl. **21**, 494–505 (2016)
6. Rendón, E., Abundez, I., Arizmendi, A., Quiroz, E.: Internal versus external cluster validation indexes. Int. J. Comput. Commun. **1**(5), 27–34 (2011)

7. Lucas, A.: amap: Another Multidimensional Analysis Package (2019)
8. Bodenhofer, U., Kothmeier, A., Hochreiter, S.: APCluster: an R package for affinity propagation clustering. Bioinformatics **27**(17), 2463–2464 (2011)
9. Maechler M.: Finding groups in data: cluster analysis extended Rousseeuw et al. R Package Version **2** (2019)
10. Suzuki, R., Shimodaira, H.: Pvclust: an R package for assessing the uncertainty in hierarchical clustering. Bioinformatics **22**(12), 1540–1542 (2006)
11. Fox, J., Leanage, A.: R and the journal of statistical software. J. Stat. Softw. **9**(73), 1–13 (2016)
12. Shirkhorshidi, A.S., Aghabozorgi, S., Wah, T.Y.: Comparison study on similarity and dissimilarity measures in clustering continuous data. PLoS ONE **12**(10), 1–20 (2015). https://doi.org/10.1371/journal.pone.0144059
13. Lance, G., Williams, W.: A generalized sorting strategy for computer classifications. Nature **212**, 218–218 (1966)
14. Ramprasanth, H., Devi, A.: Outlier analysis of medical dataset using clustering algorithms. J. Anal. Comput. **15**, 1–9 (2019)
15. Kaufman, L., Rousseeuw, P.J.: Introduction. Find. Groups Data 1–67 (1990)
16. Kaufman, L., Rousseeuw, P.J.: TDivisive analysis (Program DIANA). Find. Groups Data 253–279 (1990)
17. Kaufman, L., Rousseeuw, P.J.: Fuzzy analysis (Program FANNY). Find. Groups Data 164–198 (1990)
18. R Core Team. R: A Language and Environment for Statistical Computing (R Foundation for Statistical Computing, 2021). https://www.R-project.org/
19. Struyf, A., Hubert, M., Rousseeuw, P.: Clustering in an object-oriented environment. J. Stat. Softw. **1**(4), 1–30 (1997)
20. Kaufman, L., Rousseeuw, P.J.: Monothetic analysis (Program MONA). *Find. Groups Data* 280–311 (1990)
21. Kaufman, L., Rousseeuw, P.J.: Partitioning around medoids (Program PAM). Find. Groups Data 68–125 (1990)
22. Wickham, H., François, R., Henry, L., Müller, K.: dplyr: a grammar of data manipulation (2021). https://CRAN.R-project.org/package=dplyr, R package version 1.0.5
23. Sripada, S.C., Rao, M.S.: Comparison of purity and entropy of k-means clustering and fuzzy c means clustering. Indian J. Comput. Sci. Eng. **2**(3) (2011)
24. Meilă, M.: Comparing Clusterings by the Variation of Information, pp. 173–187. Springer, Berlin Heidelberg, Berlin, Heidelberg (2003)
25. Wu, J., Xiong, H., Chen, J.: Towards understanding hierarchical clustering: a data distribution perspective. Neurocomputing **72**(10–12), 2319–2330 (2009)
26. Nemec, A.F.L., Brinkhurst, R.O.: The Fowlkes-Mallows statistic and the comparison of two independently determined dendrograms. Can. J. Fish. Aquat. Sci. **45**(6), 971–975 (1988)
27. Hanczar, B., Nadif, M.: Precision-recall space to correct external indices for biclustering. In: Proceedings of the 30th International Conference on Machine Learning, vol. 28, pp. 136–144 (2013)
28. Palacio-Niño, J.O., Berzal, F.: Evaluation metrics for unsupervised learning algorithms. arxiv:1905.05667 (2019)
29. Rezaei, M., Fränti, P.: Set matching measures for external cluster validity. IEEE Trans. Knowl. Data Eng. **28**(8), 2173–2186 (2016)
30. Starczewski, A., Krzyżak, A.: Performance evaluation of the silhouette index. In: Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science), vol. 9120, pp. 49–58 (2015)
31. Saha, S., Bandyopadhyay, S.: A validity index based on connectivity. In: 2009 Seventh International Conference on Advances in Pattern Recognition, pp. 91–94 (2009)