

Méthodologie de développement du package RollerClustR

Romain BUONO, Nico DENA, Mohamed Habib BAH

Master 2 SISE, Université Lyon 2

Novembre 2025

Abstract

Ce rapport présente la méthodologie de développement du package R `RollerClustR`, une bibliothèque dédiée au clustering de variables implémentée selon le paradigme de programmation orientée objet R6. Le package intègre trois algorithmes complémentaires : `VAR_CAH` (Classification Ascendante Hiérarchique de variables), `VAR_KMEANS` (K-Means pour variables) et `TandemVarClust` (clustering de modalités pour données mixtes). L'architecture logicielle repose sur les principes SOLID et le pattern Factory, garantissant extensibilité, maintenabilité et cohérence d'interface. Une documentation technique exhaustive pour chaque algorithme est disponible dans le répertoire `/doc` du package.

Contents

1	Introduction et Contexte	4
1.1	Problématique du Clustering de Variables	4
1.2	Choix du Paradigme R6	4
1.3	Objectifs du Package	5
2	Architecture Logicielle Globale	5
2.1	Hiérarchie de Classes	5
2.2	Pattern Factory	5
2.3	Documentation Technique	5
3	VAR_CAH : Classification Hiérarchique de Variables	6
3.1	Fondements Théoriques	6
3.1.1	Construction de la Matrice de Dissimilarité	6
3.1.2	Algorithme Hiérarchique avec Critère de Ward	6

3.1.3	Synthèse par Composante Principale	6
3.2	Innovations Méthodologiques	6
3.2.1	Reconfiguration Dynamique	6
3.2.2	Méthodes de Prédiction Duales	7
3.3	Complexité Algorithmique	7
4	VAR_KMEANS : K-Means pour Variables	7
4.1	Principe Général	7
4.2	Architecture Itérative	7
4.3	Stratégie Multi-Initialisation	7
4.4	Avantages Computationnels	8
5	TandemVarClust : Clustering de Modalités	8
5.1	Distinction Fondamentale	8
5.2	Architecture Tandem Séquentielle	8
5.3	Prétraitement des Données Mixtes	8
5.4	Analyse Factorielle : AFDM	8
5.5	Méthode <code>predict_variable()</code> : Projection AFDM	9
5.5.1	Principe de la Projection	9
5.5.2	Métriques d'Association Complémentaires	9
5.6	Assignation des Observations : Coefficient de Dice	10
6	Interface Shiny : Exploration Interactive	10
6.1	Architecture de l'Application	10
6.2	Fonctionnalités Principales	10
6.2.1	Gestion des Données	10
6.2.2	Configuration et Clustering	11
6.2.3	Visualisation des Résultats	11
6.2.4	Prédiction de Nouvelles Variables	12
6.2.5	Historique et Sauvegarde	12
6.2.6	Export de Résultats	13
6.3	Aspects Techniques	13
6.3.1	Gestion de l'État Réactif	13
6.3.2	Adaptateur R6-Shiny	13
6.3.3	Robustesse et UX	13
7	Principes SOLID et Bonnes Pratiques	14
7.1	Single Responsibility Principle	14
7.2	Open/Closed Principle	14
7.3	Dependency Inversion Principle	14

8 Applications et Perspectives	14
8.1 Cas d'Usage	14
8.2 Extensions Envisagées	14
9 Conclusion	15

1 Introduction et Contexte

1.1 Problématique du Clustering de Variables

Le clustering de variables constitue une approche complémentaire au clustering d'observations traditionnel. Plutôt que de regrouper des individus similaires, l'objectif consiste à identifier des ensembles de variables présentant des structures de dépendance similaires. Cette démarche trouve des applications majeures en :

- **Réduction de dimensionnalité** : Synthèse de p variables corrélées en k composantes latentes ($k \ll p$)
- **Feature engineering** : Construction de variables synthétiques non redondantes
- **Détection de multicollinéarité** : Identification de groupes de variables explicatives redondantes
- **Analyse exploratoire** : Compréhension de la structure de corrélation dans des jeux de données de haute dimension

Dans des contextes de haute dimensionnalité (génomique, chimiométrie, analyse sensorielle), où le nombre de variables p peut excéder plusieurs milliers, les approches de clustering de variables offrent une alternative computationnellement efficace aux méthodes de régularisation classiques.

1.2 Choix du Paradigme R6

Le développement de `RollerClustR` repose sur la classe R6, un système de programmation orientée objet pour R offrant :

1. **Encapsulation** : Distinction claire entre méthodes publiques (API) et attributs privés (implémentation)
2. **Héritage** : Factorisation du code via une classe abstraite `ClusterAnalysis`
3. **Persistance** : Stockage d'état interne permettant la réutilisation de modèles ajustés
4. **Performance** : Modification en place évitant les copies mémoire coûteuses

Contrairement aux systèmes S3 et S4 natifs de R, R6 implémente une approche par référence similaire aux langages orientés objet classiques (Java, Python), facilitant la conception d'architectures complexes.

1.3 Objectifs du Package

Le package `RollerClustR` vise à fournir :

- Une **interface unifiée** pour trois algorithmes de clustering distincts
- Des **fonctionnalités de prédiction** permettant l'assignation incrémentale de nouvelles variables
- Une **architecture extensible** facilitant l'intégration de futurs algorithmes
- Une **documentation technique détaillée** dans `/doc` pour chaque algorithme
- Une **interface Shiny** pour l'exploration interactive

2 Architecture Logicielle Globale

2.1 Hiérarchie de Classes

L'architecture de `RollerClustR` s'articule autour d'une classe abstraite `ClusterAnalysis` dont héritent les trois implémentations algorithmiques. Cette hiérarchie garantit une interface homogène via les méthodes abstraites : `fit(X)`, `predict(newdata)`, et `summary()`.

2.2 Pattern Factory

Le module `ClusteringFactory` implémente le pattern de conception Factory, encapsulant la logique d'instanciation des algorithmes. Cette approche offre plusieurs avantages architecturaux :

1. **Découplage** : Le code client n'a pas besoin de connaître les détails d'instanciation
2. **Cohérence** : Paramétrage uniforme malgré les différences d'implémentation
3. **Extensibilité** : Ajout de nouveaux algorithmes sans modification du code existant

2.3 Documentation Technique

Le répertoire `/doc` contient des notices méthodologiques complètes pour chaque algorithme, incluant les fondements théoriques, les détails d'implémentation, les formules mathématiques et les cas d'usage. Ces documents constituent une référence exhaustive pour comprendre le comportement précis de chaque méthode.

3 VAR_CAH : Classification Hiérarchique de Variables

3.1 Fondements Théoriques

L'algorithme VAR_CAH réalise un clustering hiérarchique ascendant (agglomératif) de variables quantitatives. Le processus repose sur trois étapes fondamentales :

3.1.1 Construction de la Matrice de Dissimilarité

La dissimilarité entre variables est définie par la transformation de la corrélation de Pearson :

$$d(X_i, X_j) = 1 - |\rho(X_i, X_j)| \quad (1)$$

où $\rho(X_i, X_j)$ désigne le coefficient de corrélation entre les variables X_i et X_j . L'utilisation de la valeur absolue garantit que des variables fortement corrélées négativement sont considérées comme similaires.

3.1.2 Algorithme Hiérarchique avec Critère de Ward

La classification ascendante est réalisée selon le critère de Ward, qui minimise l'inertie intra-classe à chaque étape de fusion. On agrège les clusters C_i et C_j qui minimisent :

$$\Delta(C_i, C_j) = \frac{n_i n_j}{n_i + n_j} \|m_i - m_j\|^2 \quad (2)$$

où n_i et n_j sont les effectifs (nombre de variables) des clusters.

3.1.3 Synthèse par Composante Principale

Pour chaque cluster C_k obtenu, une Analyse en Composantes Principales est effectuée. La première composante principale est retenue comme variable synthétique S_k :

$$S_k = \sum_{j \in C_k} w_j X_j \quad (3)$$

où les poids w_j correspondent au premier vecteur propre de la matrice de corrélation.

3.2 Innovations Méthodologiques

3.2.1 Reconfiguration Dynamique

L'implémentation propose une méthode `refit_with_k(new_k)` permettant de modifier le nombre de clusters sans réexécuter l'algorithme complet. Le dendrogramme est simplement coupé à un nouveau niveau, puis les variables synthétiques sont recalculées.

3.2.2 Méthodes de Prédiction Duales

Deux stratégies d'assignation de nouvelles variables sont implémentées : la méthode "correlation" calcule la corrélation avec chaque variable synthétique S_k (complexité $O(Kn)$), tandis que la méthode "distance" calcule la corrélation moyenne avec toutes les variables du cluster.

3.3 Complexité Algorithmique

La complexité globale est $O(p^2n)$, rendant la méthode adaptée pour $p \leq 10^3$ variables.

4 VAR_KMEANS : K-Means pour Variables

4.1 Principe Général

L'algorithme VAR_KMEANS adapte le paradigme K-Means classique au contexte du clustering de variables, selon les travaux de Vigneau et Qannari (2003). Contrairement au K-Means standard, l'objet classé concerne des variables (colonnes) basées sur leurs profils, et le centre d'un cluster est la première composante principale du groupe.

4.2 Architecture Itérative

L'algorithme alterne entre deux étapes jusqu'à convergence : l'étape d'allocation affecte chaque variable x_j au cluster k maximisant le coefficient de corrélation au carré :

$$k^* = \arg \max_{k \in \{1, \dots, K\}} r^2(x_j, u_k) \quad (4)$$

où u_k est le centre (composante principale) du cluster k . L'étape de représentation recalcule pour chaque cluster C_k le nouveau centre u_k comme la première composante principale des variables du cluster.

4.3 Stratégie Multi-Initialisation

Le K-Means étant sensible aux optimums locaux, l'algorithme implémente une stratégie de redémarrage multiple (`n_init`). Le critère d'optimisation est l'inertie totale :

$$T = \sum_{k=1}^K \sum_{j \in C_k} r^2(x_j, u_k) \quad (5)$$

4.4 Avantages Computationnels

Contrairement à VAR_CAH qui nécessite le calcul d'une matrice de dissimilarité $p \times p$, VAR_KMEANS présente une complexité linéaire en nombre de variables : $O(I \cdot n \cdot p \cdot K)$ où I est le nombre d'itérations (typiquement < 100). Cette propriété rend l'algorithme particulièrement adapté aux jeux de données de très haute dimension ($p > 10^4$).

5 TandemVarClust : Clustering de Modalités

5.1 Distinction Fondamentale

TandemVarClust se distingue des deux approches précédentes : il opère au niveau des **modalités** plutôt que des variables. Chaque catégorie d'une variable qualitative ou chaque intervalle discrétisé d'une variable quantitative constitue une unité à classer. Cette granularité fine permet de capturer des structures que le clustering de variables masquerait.

5.2 Architecture Tandem Séquentielle

L'approche combine deux étapes indépendantes : (1) l'AFDM (Analyse Factorielle de Données Mixtes) pour la réduction de dimensionnalité du tableau disjonctif, et (2) la CAH pour la classification hiérarchique sur les coordonnées factorielles des modalités. Cette approche séquentielle offre l'avantage de la simplicité algorithmique et de l'interprétabilité : les axes factoriels peuvent être examinés indépendamment.

5.3 Prétraitement des Données Mixtes

Pour les variables qualitatives, le codage disjonctif complet (indicatrices) est appliqué. Pour une variable V à m modalités, on crée m colonnes binaires où $Z_{ij} = 1$ si l'observation i possède la modalité j . Les variables quantitatives sont discrétisées en `n_bins` intervalles (défaut : 5) par quantiles, puis codées par indicatrices. Le choix des quantiles garantit des effectifs équilibrés.

La matrice disjonctive finale \mathbf{Z} a pour dimension :

$$n \times p^* \quad \text{où} \quad p^* = \sum_{j \in \text{qual}} m_j + \sum_{j \in \text{quant}} n_{\text{bins},j} \quad (6)$$

5.4 Analyse Factorielle : AFDM

L'AFDM réalise une analyse factorielle sur \mathbf{Z} , combinant les principes de l'ACP et de l'ACM. Les modalités sont pondérées par l'inverse de leur fréquence marginale :

$$w_j = \frac{1}{f_j} \quad \text{où} \quad f_j = \frac{1}{n} \sum_{i=1}^n Z_{ij} \quad (7)$$

Cette pondération compense le déséquilibre d'effectifs entre modalités. La décomposition en valeurs singulières (SVD) de la matrice pondérée fournit les coordonnées factorielles des modalités, qui sont ensuite utilisées pour le clustering hiérarchique.

5.5 Méthode `predict_variable()` : Projection AFDM

Une innovation majeure de l'implémentation concerne la méthode de prédiction pour variables illustratives. Contrairement aux approches classiques, la nouvelle implémentation utilise une `**projection` dans l'espace AFDM suivie d'un calcul de distances aux centres de clusters.

5.5.1 Principe de la Projection

Pour une nouvelle variable illustrative, les étapes sont :

1. Transformation en tableau disjonctif (codage indicateur pour catégorielles, discrétisation puis codage pour numériques)
2. Projection des modalités dans l'espace factoriel établi lors de l'entraînement :

$$\mathbf{f}_{\text{new}} = (\mathbf{z}_{\text{new}} - \bar{\mathbf{z}}) \mathbf{D}_{\text{col}}^{1/2} \mathbf{V} \quad (8)$$

où \mathbf{V} est la matrice des vecteurs singuliers à droite de la SVD d'entraînement

3. Calcul des distances euclidiennes moyennes entre les modalités de la variable et les centres de chaque cluster
4. Assignment au cluster de distance minimale

5.5.2 Métriques d'Association Complémentaires

Au-delà de l'assignment basée sur les distances, la méthode calcule des métriques statistiques enrichies :

- **Table de contingence** : Croisement de la variable illustrative avec les clusters d'observations
- **Test du Chi²** : Évaluation de l'indépendance entre variable et clusters
- **V de Cramér** : Mesure d'association normalisée entre 0 et 1 :

$$V = \sqrt{\frac{\chi^2}{n \times \min(K-1, m-1)}} \quad (9)$$

- **Scores de Dice** : Matrice donnant pour chaque observation son indice de Dice avec chaque cluster

Cette approche garantit la cohérence méthodologique avec la structure factorielle établie lors du clustering et fournit une analyse multidimensionnelle de la relation entre variable illustrative et partition.

5.6 Assignment des Observations : Coefficient de Dice

Pour assigner les observations aux clusters de modalités, l'indice de similarité de Dice est utilisé :

$$\text{Dice}(O, C_k) = \frac{2|O \cap C_k|}{|O| + |C_k|} \quad (10)$$

où O est l'ensemble des modalités actives pour l'observation et C_k l'ensemble des modalités du cluster k . Le coefficient de Dice présente plusieurs avantages : traitement équilibré des tailles, interprétation naturelle comme proportion de chevauchement, et robustesse aux déséquilibres d'effectifs.

6 Interface Shiny : Exploration Interactive

6.1 Architecture de l'Application

L'application Shiny développée pour `RollerClustR` offre une interface utilisateur complète permettant l'exploration interactive des algorithmes de clustering de variables. L'architecture suit le modèle MVC (Model-View-Controller) avec trois composantes principales :

- **global.R** : Chargement des packages, des classes R6, et définition des fonctions utilitaires
- **ui.R** : Interface utilisateur construite avec `shinydashboard`
- **server.R** : Logique serveur gérant les valeurs réactives et les interactions

6.2 Fonctionnalités Principales

6.2.1 Gestion des Données

L'onglet "Données" permet :

- **Import de fichiers** : Support des formats CSV (virgule, point-virgule, tabulation) et Excel (.xlsx, .xls)

- **Datasets pré-chargés** : Accès direct à 6 datasets R classiques (iris, mtcars, US-Arrests, swiss, state.x77, airquality)
- **Génération de données synthétiques** : Création de jeux de données artificiels avec paramètres contrôlables (nombre d'observations, bruit, graine aléatoire)
- **Traitement des valeurs manquantes** : Trois stratégies (suppression, imputation par médiane/mode, imputation par moyenne)
- **Prévisualisation interactive** : Affichage tabulaire avec `DT::datatable`

6.2.2 Configuration et Clustering

L'onglet "Configuration" propose :

- **Sélection d'algorithme** : Choix entre VAR_CAH, VAR_KMEANS, et TandemVarClust avec descriptions contextuelles
- **Sélection de variables** : Interface interactive pour choisir les variables actives
- **Paramétrage algorithmique** : Nombre de clusters (K), standardisation, critères d'agrégation (Ward, complete, average, single)
- **Paramètres spécifiques** : Nombre d'initialisations pour VAR_KMEANS, nombre de bins pour TandemVarClust
- **Détection automatique** : Option de sélection automatique du nombre de clusters

L'onglet "Clustering" lance l'analyse avec barre de progression et gestion d'erreurs robuste.

6.2.3 Visualisation des Résultats

L'onglet "Résultats" offre une analyse multi-facettes :

- **Résumé textuel** : Statistiques globales (nombre de clusters, distribution des variables, métriques de qualité)
- **Dendrogramme interactif** : Visualisation hiérarchique avec `plotly` pour VAR_CAH et TandemVarClust
- **Graphique de Silhouette** : Évaluation de la qualité du clustering par variable
- **Plans factoriels** : Projection 2D des variables dans l'espace factoriel pour TandemVarClust
- **Matrice de corrélation** : Heatmap interactive réorganisée selon les clusters

- **Tables détaillées** : Assignment cluster par cluster avec métriques individuelles

L'onglet "Diagnostics" fournit :

- **Métriques globales** : Silhouette moyen, Davies-Bouldin, Dunn Index, Calinski-Harabasz
- **Analyse de stabilité Bootstrap** : Évaluation de la robustesse du clustering par rééchantillonnage
- **Variables discriminantes** : Identification des variables les plus contributives à la structure

6.2.4 Prédiction de Nouvelles Variables

L'onglet "Prédiction" implémente :

- **Import de variables illustratives** : Chargement de nouvelles variables à classer
- **Génération de variables test** : Création de variables aléatoires pour démonstration
- **Assignment automatique** : Application de la méthode `predict()` de l'objet R6
- **Résultats enrichis** :
 - Pour VAR_CAH et VAR_KMEANS : Scores de corrélation par cluster
 - Pour TandemVarClust : Table de contingence, test du χ^2 , V de Cramér, scores de Dice
- **Visualisation comparative** : Positionnement des nouvelles variables dans l'espace des clusters existants

6.2.5 Historique et Sauvegarde

L'onglet "Historique" permet :

- **Sauvegarde de sessions** : Enregistrement automatique de chaque analyse avec métadonnées (timestamp, algorithme, paramètres, métriques)
- **Gestion multi-sessions** : Conservation de plusieurs analyses pour comparaison
- **Annotations** : Ajout de notes descriptives pour chaque session
- **Rechargement** : Restauration complète d'une session antérieure (données, modèle, configuration)
- **Export/Import** : Sauvegarde des sessions au format RDS pour partage ou archivage

6.2.6 Export de Résultats

L'onglet "Export" offre :

- **Export tabulaire** : Tables de résultats au format CSV ou Excel
- **Export graphique** : Dendrogrammes et plans factoriels en PNG ou PDF haute résolution
- **Rapport HTML** : Génération automatique d'un rapport complet avec tous les résultats et visualisations
- **Export du modèle** : Sauvegarde de l'objet R6 complet pour réutilisation programmatique

6.3 Aspects Techniques

6.3.1 Gestion de l'État Réactif

L'application utilise `reactiveValues` pour gérer l'état global :

- `rv$data` : Données brutes importées
- `rv$data_cleaned` : Données après prétraitement
- `rv$model_r6` : Objet R6 complet du modèle
- `rv$model` : Structure adaptée pour Shiny
- `rv$sessions_history` : Liste des sessions sauvegardées
- `rv$prediction_results` : Résultats de prédiction

6.3.2 Adaptateur R6-Shiny

Une fonction `model_to_shiny()` convertit les objets R6 en structures compatibles avec l'interface Shiny, extrayant automatiquement les propriétés actives et calculant les métriques de qualité via `.compute_metrics()`.

6.3.3 Robustesse et UX

L'application intègre :

- **Gestion d'erreurs** : Blocs `tryCatch` avec notifications contextuelles
- **Validation d'entrées** : Vérification des types de variables selon l'algorithme
- **Barres de progression** : Feedback visuel durant les calculs longs

- **Aide contextuelle** : Onglet "Aide" avec guide pas-à-pas et documentation des métriques

7 Principes SOLID et Bonnes Pratiques

7.1 Single Responsibility Principle

Chaque classe possède une responsabilité unique : `ClusteringFactory` pour la création, `ClusteringComparator` pour la comparaison, `ClusteringEvaluator` pour la sélection de k , et `ClusteringHelper` pour la génération de rapports.

7.2 Open/Closed Principle

L'architecture est ouverte à l'extension mais fermée à la modification. L'ajout d'un nouvel algorithme nécessite uniquement la création d'une classe héritant de `ClusterAnalysis` et l'ajout d'une méthode `factory`.

7.3 Dependency Inversion Principle

Les classes de haut niveau dépendent d'abstractions (interface `ClusterAnalysis`) plutôt que d'implémentations concrètes, garantissant la flexibilité architecturale.

8 Applications et Perspectives

8.1 Cas d'Usage

Le package trouve des applications en génomique (regroupement de gènes avec profils d'expression corrélés), analyse sensorielle (synthèse de descripteurs en dimensions perceptives), et marketing (segmentation basée sur données socio-démographiques mixtes avec `TandemVarClust`).

8.2 Extensions Envisagées

Les extensions futures incluent le support natif des variables ordinales, le clustering flou avec assignation probabiliste, la sélection automatique de k via Gap statistic, et l'optimisation par parallélisation avec `future` et implémentation de boucles critiques en C++ avec `Rcpp`.

9 Conclusion

Le package `RollerClustR` propose une architecture logicielle cohérente et extensible pour le clustering de variables en R. L'implémentation de trois algorithmes complémentaires (`VAR_CAH`, `VAR_KMEANS`, `TandemVarClust`) offre une couverture méthodologique large, adaptée à divers contextes (variables numériques, mixtes, haute dimension).

Les principes de conception appliqués garantissent maintenabilité, extensibilité et reproductibilité. Les innovations méthodologiques, notamment la projection AFDM pour `TandemVarClust` et les méthodes de prédiction duales, enrichissent les implémentations de référence. L'interface Shiny développée rend ces méthodes accessibles aux utilisateurs non-programmeurs tout en préservant la puissance analytique du package.

La documentation technique exhaustive dans `/doc`, combinée à l'interface interactive, positionne `RollerClustR` comme un outil complet pour l'analyse de structure de variables dans des données complexes.

References

- [1] Chavent, M., Kuentz-Simonet, V., Liquet, B., & Saracco, J. (2012). `ClustOfVar`: An R Package for the Clustering of Variables. *Journal of Statistical Software*, 50(13), 1-16.
- [2] Vigneau, E., & Qannari, E. M. (2003). Clustering of variables around latent components. *Communications in Statistics-Simulation and Computation*, 32(4), 1131-1150.
- [3] Ward, J. H., Jr. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301), 236-244.
- [4] Escofier, B. (1979). Traitement simultané de variables quantitatives et qualitatives. *Les Cahiers de l'Analyse des Données*, IV(2), 137-146.
- [5] Pagès, J. (2004). Analyse factorielle de données mixtes. *Revue de Statistique Appliquée*, 52(4), 93-111.
- [6] Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3), 297-302.
- [7] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- [8] Wickham, H. (2015). *R Packages: Organize, Test, Document, and Share Your Code*. O'Reilly Media.