

# Introduction à l'analyse de séquence et illustration de son application en sciences sociales à partir de patrons de transitions de l'école au travail

É. Thouin, C. Courdi, E. Olivier, V. Dupéré, A.-S. Denault et É. Lacourse

2022-10-27

## Contents

<b>Introduction</b>	<b>2</b>
<b>1. Analyse de séquence</b>	<b>3</b>
1.1 Préparation des données . . . . .	3
1.2 Représentations visuelles et analyses descriptives . . . . .	5
1.3 Création de la typologie . . . . .	8
<b>Licence</b>	<b>15</b>

**Résumé:** Plusieurs champs de recherche en psychoéducation, en psychologie développementale et en sociologie visent à examiner les changements individuels et au sein de groupes dans la population. On cherche, par exemple, à identifier les parcours typiques lors de périodes développementales précises, comme la délinquance à l'adolescence et la transition de l'école au travail au début de l'âge adulte, afin de comprendre celles qui paraissent plus ou moins adaptatives ou optimales. Cet article a comme objectif de présenter une approche algorithmique permettant de tracer de tels parcours sous forme de séquences à partir de variables catégorielles/nominales, comme dont des statuts (p. ex.: occupationnels, maritaux), des états (p. ex. : de santé) ou la présence de comportements (p. ex. : consommation ou pas). Cette approche, l'analyse de séquences, est abondamment utilisée par les chercheurs européens, mais demeure peu connue en Amérique du Nord. L'article présente les fondements et l'application de cette approche analytique, en décrivant chacune des étapes de l'analyse à partir d'un exemple fictif tiré de banques de données portant sur le passage de l'adolescence à l'âge adulte. L'article conclut par une discussion rapportant les forces et les limites de l'analyse de séquences dite algorithmiques en sciences sociales. Le script utilisé pour réaliser les analyses de cet article est également fourni en ligne pour les lecteurs intéressés par cette technique analytique.

**Mots-clés:** analyses de séquence; appariement optimal; trajectoires et parcours longitudinaux; approche algorithmique

# Introduction

Ce document présente le code utilisé pour la création du modèle présenté dans l'article et explique en détail le code de l'analyse de séquence.

Tout d'abord, il faut évidemment télécharger les packages nécessaires à l'analyse.

```
#Téléchargement des packages nécessaire à l'analyse (si vous installez ces packages  
#pour la première fois, retirez # au début de chaque ligne "install.packages")
```

```
#install.packages("readr")  
#install.packages("ggplot2")  
#install.packages("poLCA")  
#install.packages("haven")  
#install.packages("dplyr")  
#install.packages("TraMineR")  
#install.packages("WeightedCluster")  
#install.packages("fpc")  
#install.packages("descr")  
#install.packages("flexclust")  
#install.packages("nnet")  
#install.packages("glmnet")  
#install.packages("lmtest")
```

```
library(readr)  
library(ggplot2)  
library(poLCA)  
library(haven)  
library(dplyr)  
library(TraMineR)  
library(WeightedCluster)  
library(fpc)  
library(descr)  
library(flexclust)  
library(nnet)  
library(glmnet)  
library(lmtest)  
library(latex2exp)  
library(knitr)
```

```
opts_chunk$set(echo = TRUE, prompt = TRUE, comment = "", cache = TRUE)  
options(xtable.comment = FALSE)
```

Il faut également télécharger la base de données (disponible sur Github au <https://github.com/labo-lacourse/Analyse-sequence>) à partir de laquelle les analyses seront réalisées.

**Attention:** ce lien sera mis à jour une fois que le Github sera public.

```
> #Télécharger le fichier de données à partir de github
> library(RCurl)
```

Attaching package: 'RCurl'

The following object is masked from 'package:lmtest':

reset

The following object is masked from 'package:modeltools':

clone

```
> data.csv <- getURL("https://raw.githubusercontent.com/clementine-courdi/analyse-sequence-transition-e
>
> #Lire le fichier de données téléchargé depuis github (vérification du téléchargement)
> library(readr)
> df <- read.csv(text=data.csv)
> ls(df)
```

```
[1] "M10r" "M11r" "M12r" "M13r" "M14r" "M15r" "M16r" "M17r" "M18r" "M19r"
[11] "M1r" "M20r" "M21r" "M22r" "M23r" "M24r" "M25r" "M26r" "M27r" "M28r"
[21] "M29r" "M2r" "M30r" "M31r" "M32r" "M33r" "M34r" "M35r" "M36r" "M37r"
[31] "M38r" "M39r" "M3r" "M40r" "M41r" "M42r" "M43r" "M44r" "M45r" "M46r"
[41] "M47r" "M48r" "M4r" "M5r" "M6r" "M7r" "M8r" "M9r"
```

## 1. Analyse de séquence

### 1.1 Préparation des données

Pour simplifier l'interprétation des résultats, nous commençons par attribuer des étiquettes (*labels*), ainsi qu'une version abrégée de ces étiquettes, à chaque modalité des variables. Ainsi, à chaque temps, un individu se voit attribuer l'un de ces quatre statuts: Emploi (E), Éducation secondaire (ES), Éducation postsecondaire (EP) ou NEET (N).

```
> #Étiquetter les données
> df.lab <- c("NEET", "Emploi", "Education secondaire",
+            "Education postsecondaire")
> df.shortlab <- c("N", "T", "ES", "EP")
>
> #Attacher la base de données avec les nouvelles étiquettes
> attach(df)
```

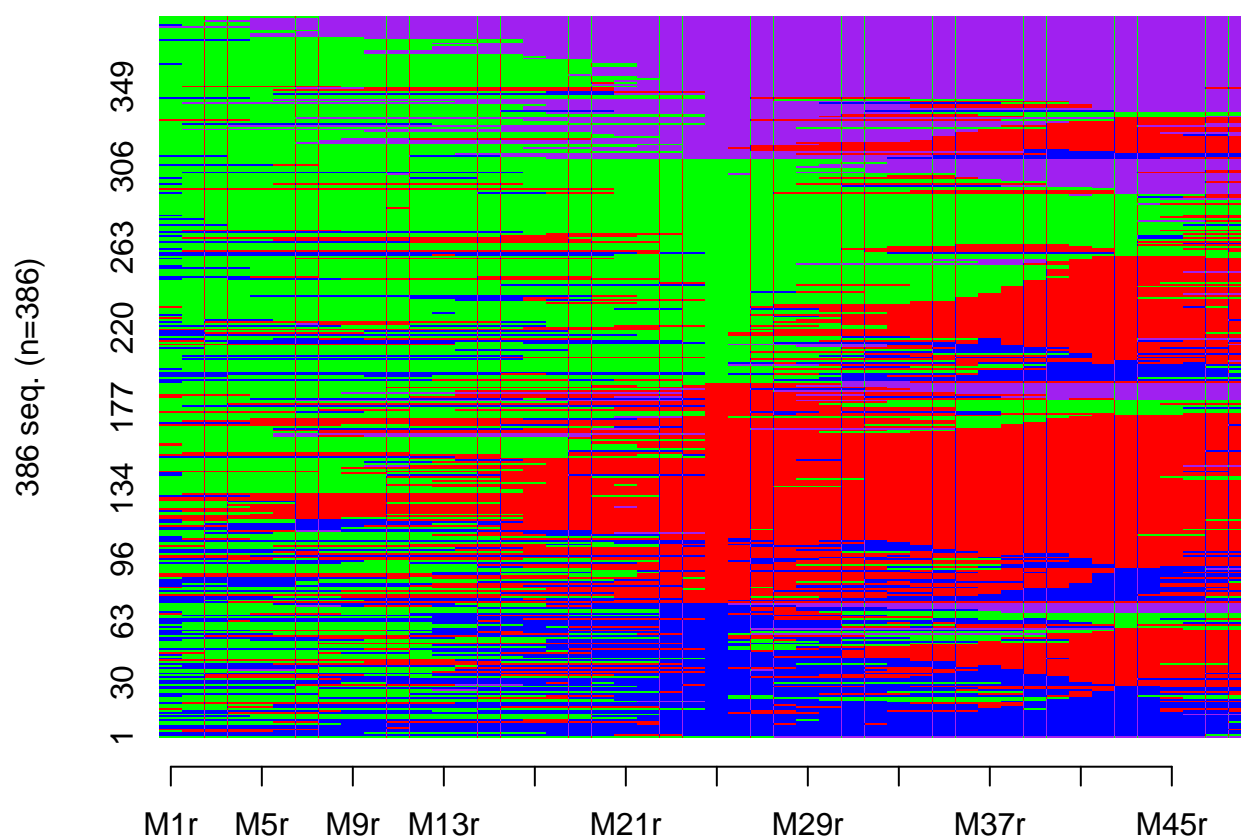
Ensuite, il faut indiquer au logiciel de traiter les variables visées comme des données d'une même séquence. Ainsi, les variables M1r à M48r, représentant le statut d'emploi/éducation de l'individu à chaque point de collecte de donnée, sont utilisées pour créer une séquence par individu, tout en conservant les étiquettes créées précédemment.

```
> #Analyse de séquence
> #Creation des donnees en sequence
> (df.alph <- seqstat1(df[, 1:48]))
> df.seq <- seqdef(df[, 1:48], alphabet = df.alph,
+                 labels = df.lab, states = df.shortlab,
+                 xtstep = 4)
```

## 1.2 Représentations visuelles et analyses descriptives

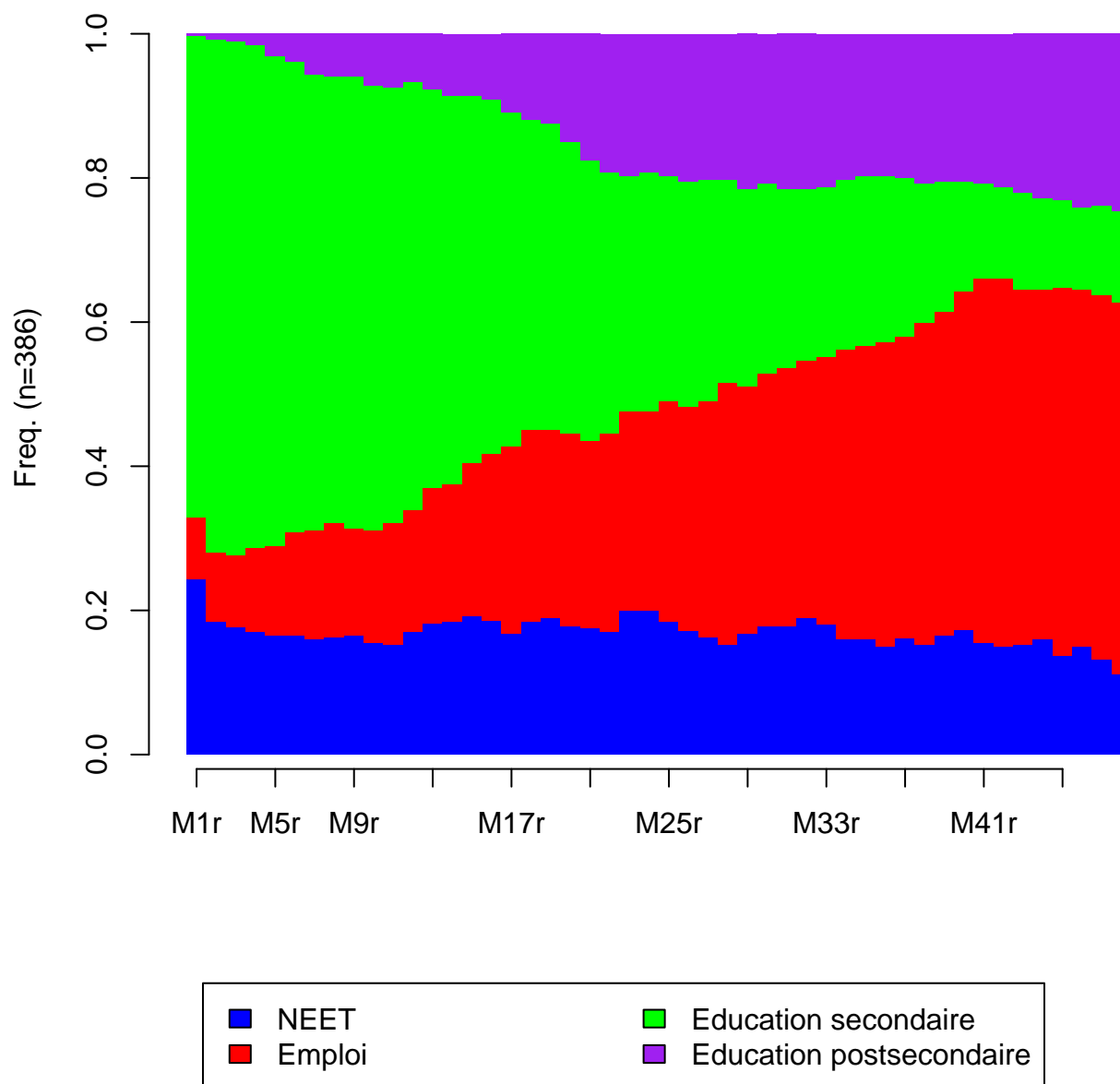
Comme les représentations graphiques sont essentielles à la compréhension de l'analyse de séquence, on attribue une couleur à chaque statut pour en faciliter l'interprétation. On crée ensuite un graphique représentant les séquences individuelles, superposées en suivant l'ordre des observations dans la base de données. L'axe des X indique le temps de mesure (de 1 à 48) et l'axe des Y le numéro de l'observation dont la séquence est illustrée.

```
> #Déterminer les couleurs#  
> cpal(df.seq)<- c("blue", "red", "green", "purple")  
>  
> #Faire graphique de toutes les séquences de l'échantillon##  
> #séquence individuelles  
> seqIplot (df.seq, border = NA)
```



Comme on le constate, ce premier graphique est plutôt difficile à interpréter. C'est pourquoi on crée également un deuxième graphique, celui-ci regroupant plutôt le nombre d'observations attribuées à chaque statut en fonction des mois, créant un graphique plus aisément interprétable. Encore un fois, l'axe des X indique le temps de mesure (de 1 à 48), mais l'axe des Y représente la proportion des observations auxquelles sont attribuées chaque statut. On peut concevoir ce graphique comme l'équivalent d'un graphique à barre empilées.

```
> #par mois, le nombre de personne dans chaque statut (plus lisible)
> seqdplot(df.seq, border = NA)
```

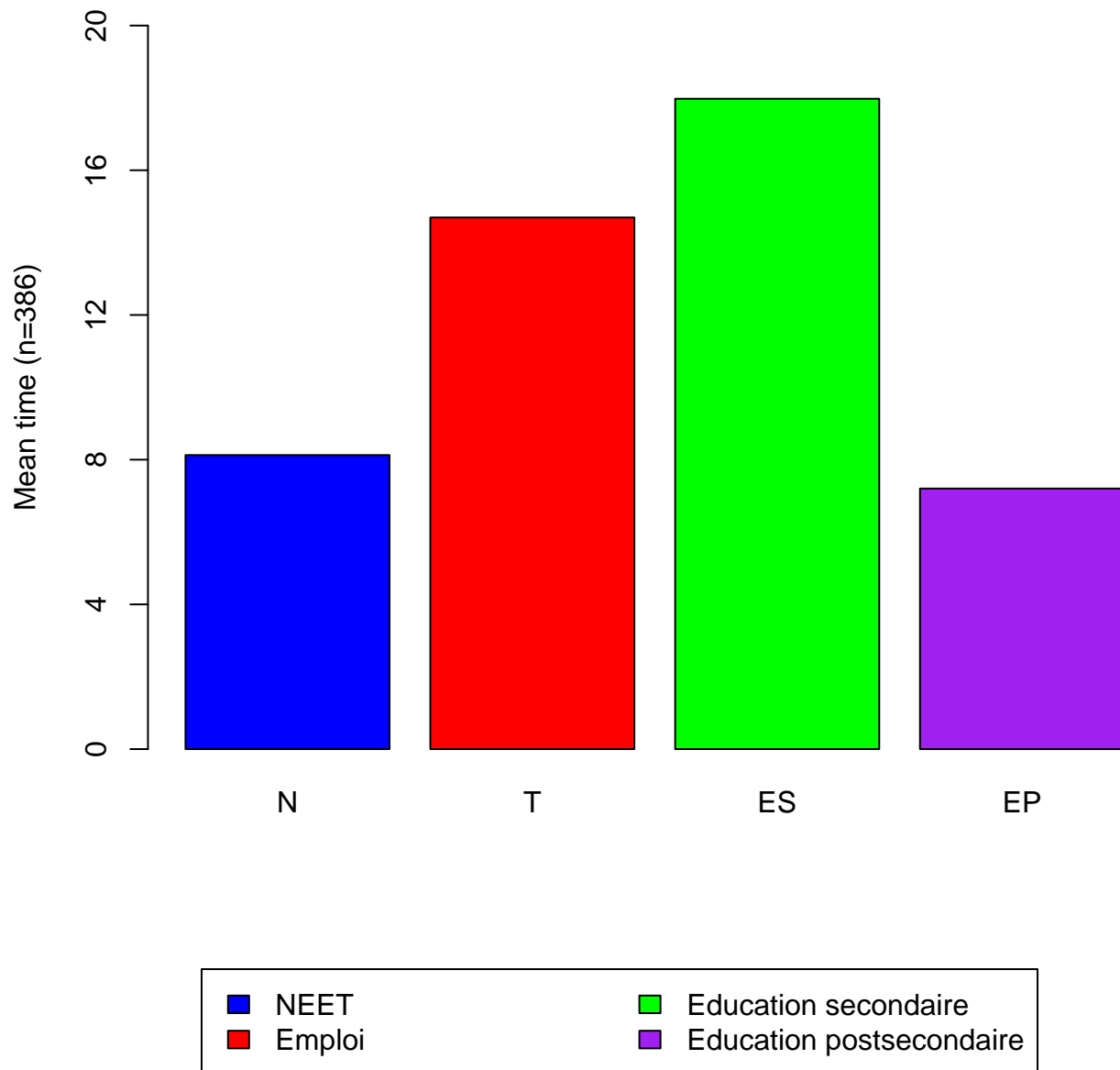


Afin de compléter l'analyse, on calcule également la moyenne de temps passé dans chacun des statuts (en nombre de mois), qu'on représente ensuite par un graphique. Ici, l'axe des Y représente le nombre de temps de mesure.

```
> ##Moyenne de temps passé dans chacun des statuts (en nombre de mois)
> seqmeant(df.seq)
```

```
      Mean
N      8.1
T     14.7
ES     18.0
EP      7.2
```

```
> #En graphique
> seqmtplot(df.seq, ylim = c(0, 20))
```



Cette mesure permet également de calculer un taux de transition, soit la proportion des fois où un statut X est suivi d'un statut Y. On remarque évidemment qu'il est particulièrement fréquent que le statut à un temps X soit suivi du même statut au temps X+1.

On peut aussi relever les séquences les plus fréquentes à travers l'échantillon. Ici-bas, on présente les 10 séquences les plus fréquentes. Finalement, on peut aussi calculer le nombre moyen de transitions par séquence, soit le nombre de changements de statut dans la trajectoire d'un individu.

```
> #Taux de transition
> round(seqtrate(df.seq), digits = 2)
```

```
[>] computing transition probabilities for states N/T/ES/EP ...
```

	[-> N]	[-> T]	[-> ES]	[-> EP]
[N ->]	0.87	0.08	0.04	0.00
[T ->]	0.04	0.93	0.02	0.01
[ES ->]	0.02	0.03	0.93	0.02
[EP ->]	0.00	0.02	0.00	0.97

```
> #Séquence la plus commune de l'échantillon
> seqtab(df.seq, idxs = 1:10)
```

	Freq	Percent
ES/48	9	2.33
ES/19-EP/29	7	1.81
ES/4-EP/44	5	1.30
ES/16-EP/32	4	1.04
T/48	4	1.04
ES/12-EP/36	3	0.78
ES/16-T/32	3	0.78
ES/39-T/9	3	0.78
ES/9-EP/39	3	0.78
ES/17-T/31	2	0.52

```
> #Donnees individuelles
> mean(seqtransn(df.seq)) ##Nombre de transition
```

```
[1] 3.463731
```

### 1.3 Création de la typologie

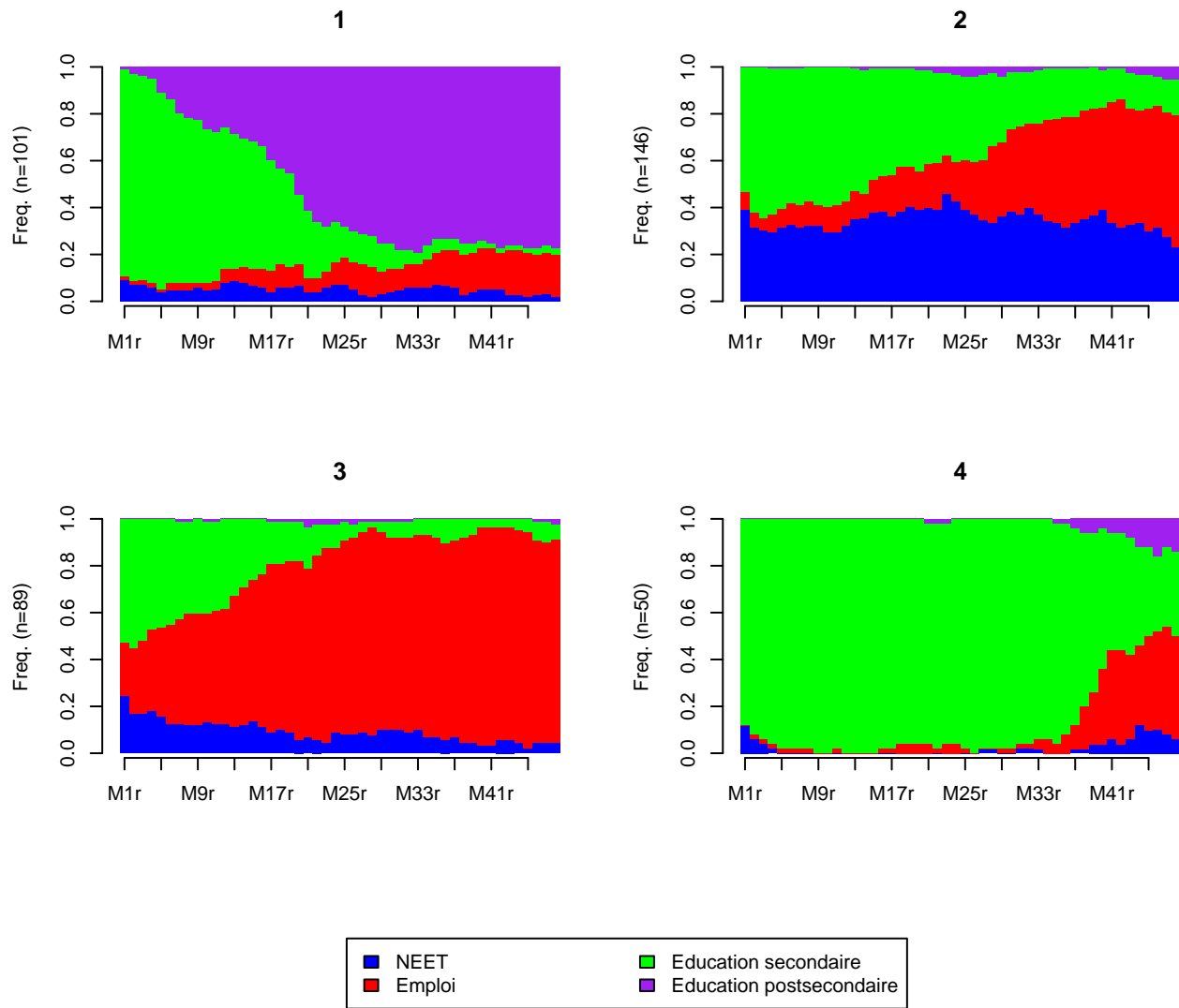
Nous passons maintenant à la création du modèle en soi. La première étape consiste à calculer les mesures de dissimilarité par la méthode d'appariement optimal [OMtrate] (avec les coûts de substitution correspondant aux taux de transition [TRATE] et ceux des indel établis à 1) entre chaque séquence. Cette opération crée également la matrice de dissimilarité. On indique ensuite le nombre de classifications que l'on veut examiner: ici, on examine les modèles comprenant de 1 à 8 catégories. La méthode du clustering hiérarchique a été retenue pour créer les classifications.

```
> ##Avec clustering hierarchique#
> OMtrate <- seqdist(df.seq, method = "OM", indel = 1, sm = "TRATE")
> hc.ward <- hclust(as.dist(OMtrate), method = "ward.D")
> df.clust <- as.clustrange(hc.ward, diss = OMtrate, ncluster = 8)
```



Par la suite, on examine donc en détail les différentes solutions possibles. Les solutions de 2 à 8 catégories ont été analysées afin de sélectionner la solution la mieux ajustée aux données. À des fins de concision, seuls les résultats des solutions à quatre et cinq catégories sont présentés en détail ici. On produit d'abord les graphiques représentant le nombre d'observations attribuées à chaque statut en fonction des mois, qui permet de comparer aisément la composition des différentes catégories. On calcule également la moyenne de temps passé dans chacun des statuts (en nombre de mois), pour produire le graphique représentant la séquence "moyenne" ou "typique" de chaque catégorie. Ces représentations graphiques permettent de comparer les différentes classifications.

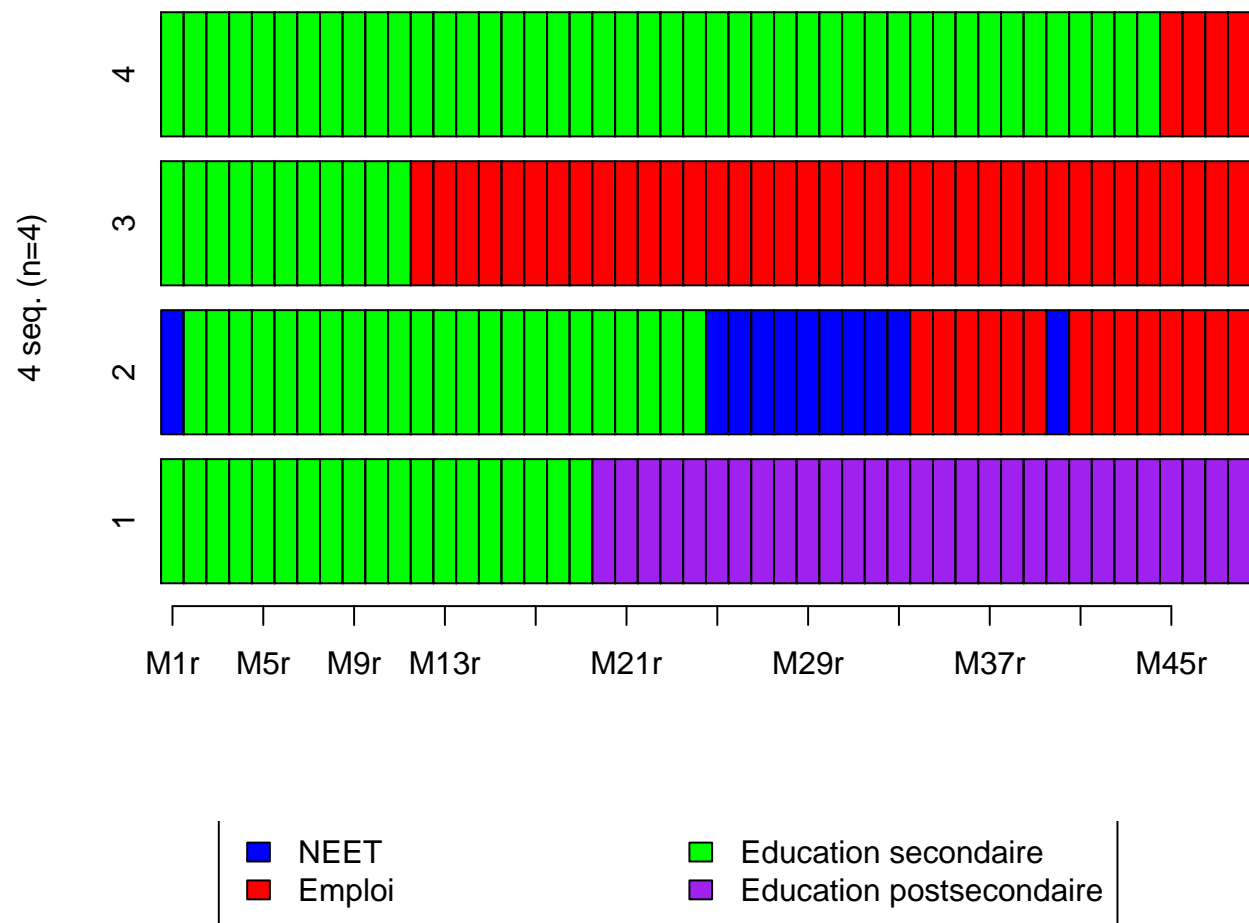
```
> #modèles à 4 et 5 catégories
>
> #4 solution
> #enregistrer la solution à 4 clusters
> clusterH4 <- df.clust$clustering$cluster4
>
> #graphique
> seqdplot(df.seq, group = df.clust$clustering$cluster4, border = NA)
```



```

> #graphique
> icenter <- disscenter(OMtrate, factor(clusterH4), medoids.index="first")
> seqipLOT (df.seq[icenter,])

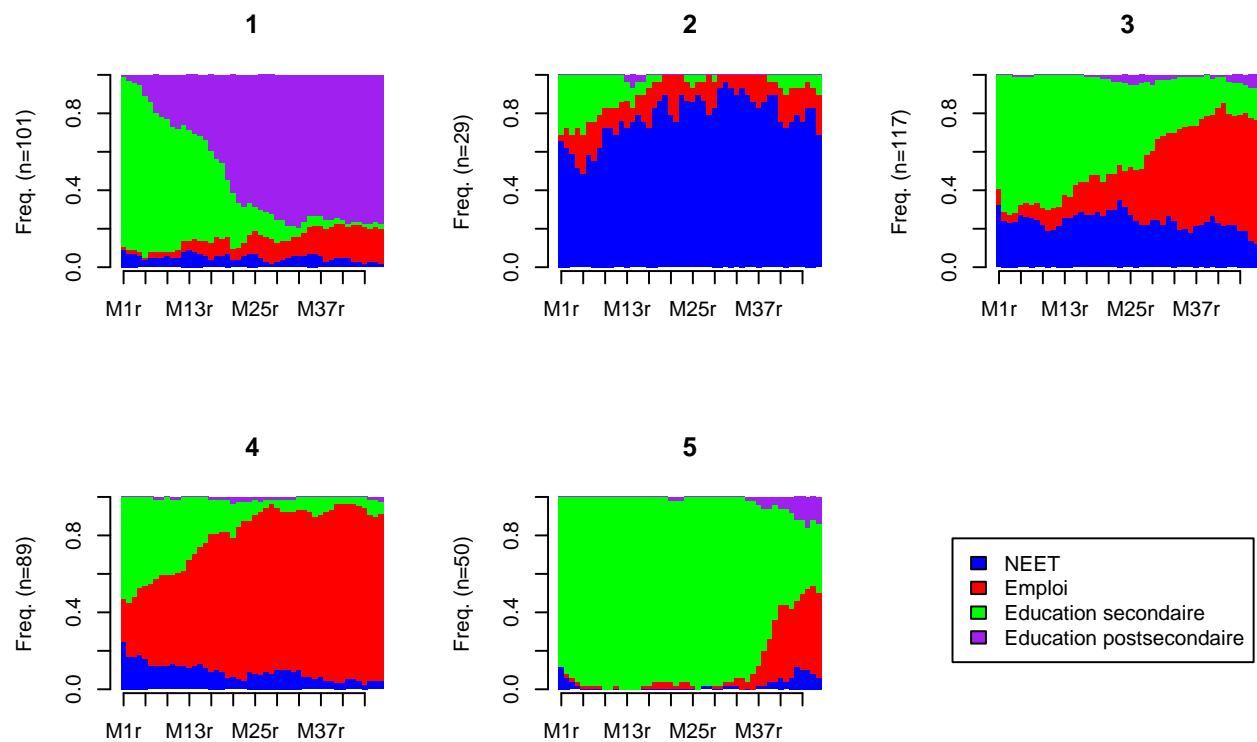
```



```

> #5 solution
> #enregistrer la solution à 5 clusters
> clusterH5 <- df.clust$clustering$cluster5
> #graphique
> seqdplot(df.seq, group = df.clust$clustering$cluster5, border = NA, cols = 3)

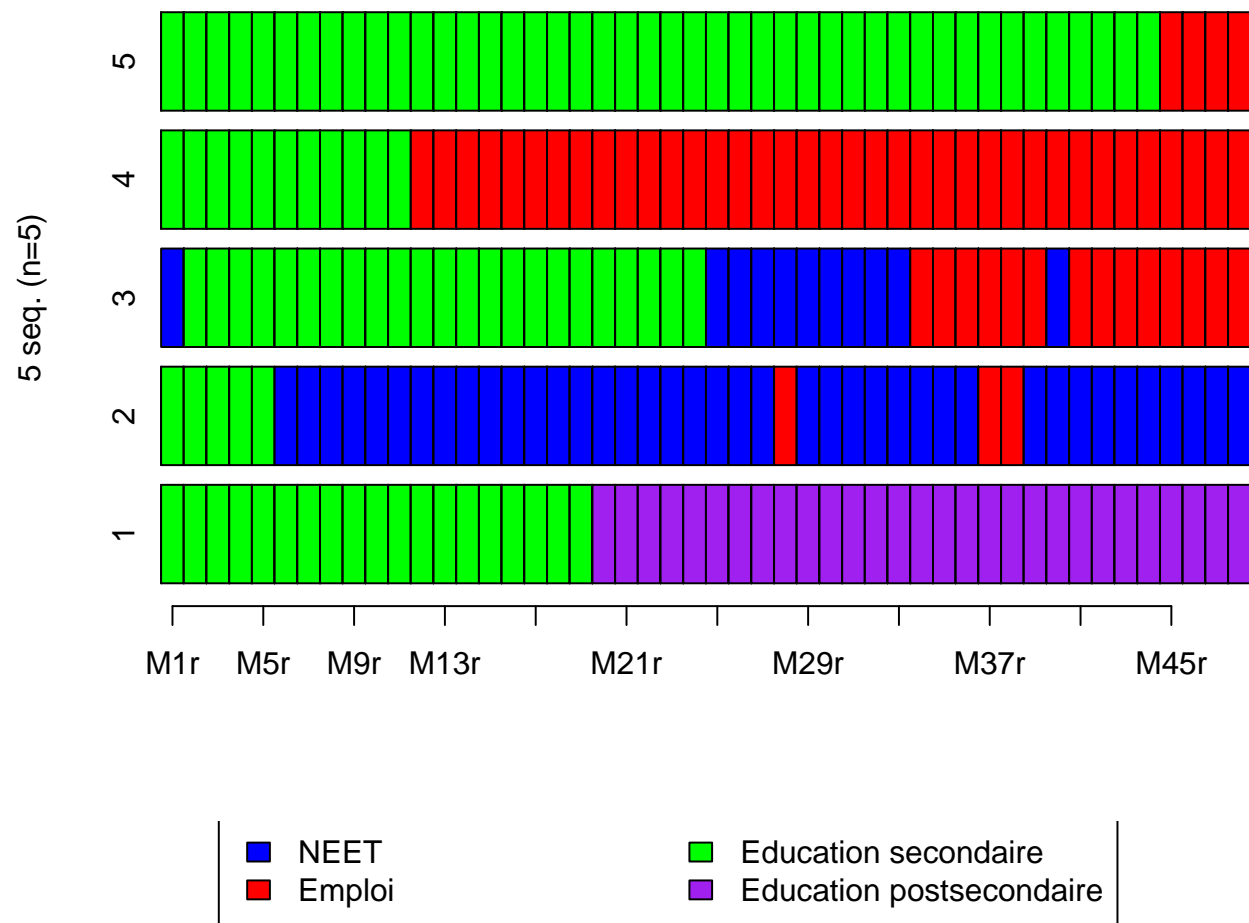
```



```

> #graphique
> icenter <- disscenter(OMtrate, factor(clusterH5), medoids.index="first")
> seqiplot (df.seq[icenter,])

```

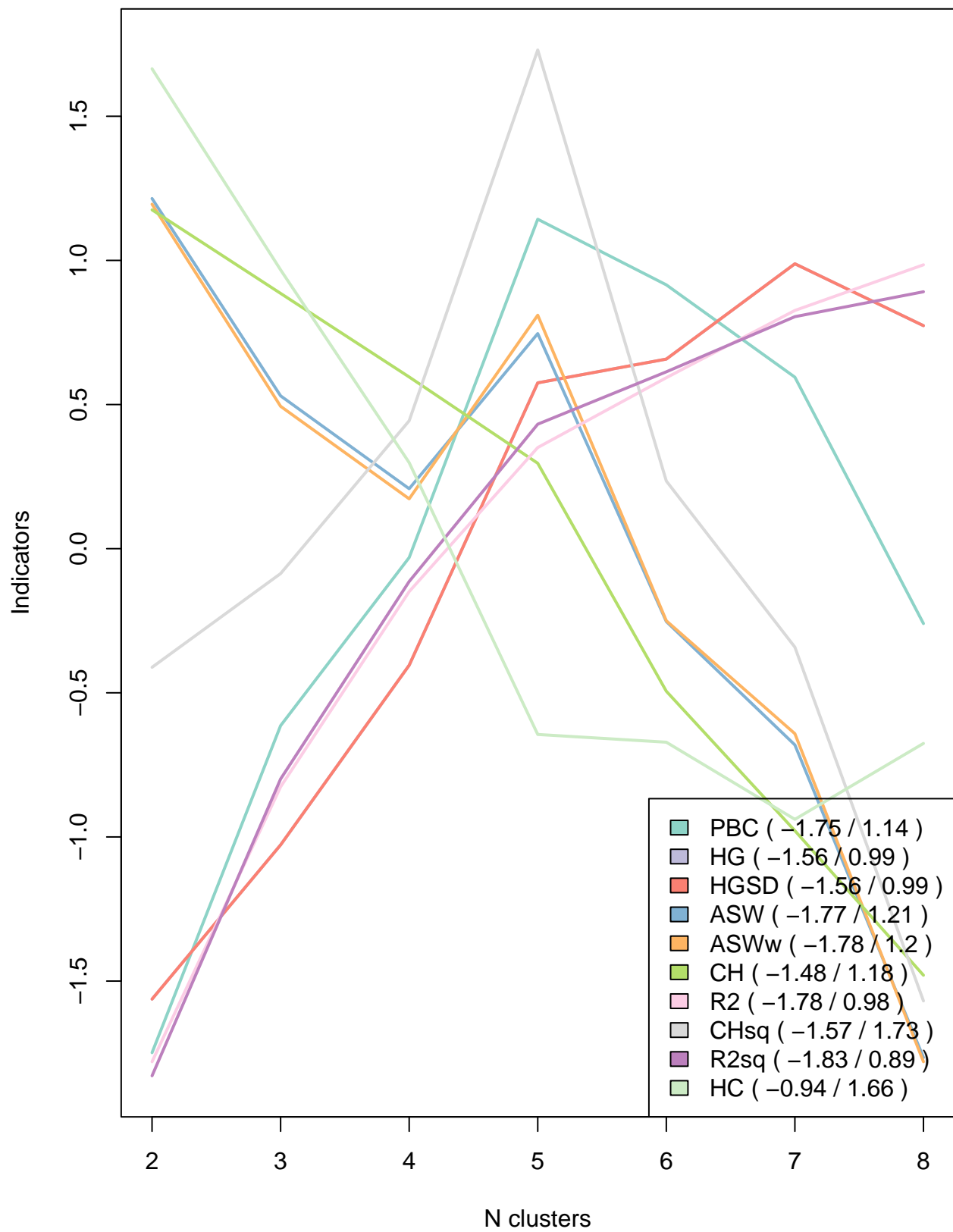


Ensuite, on examine les indicateurs d'ajustement des différentes classifications pour confirmer le choix du modèle. Un ensemble de tests statistiques permet de comparer les typologies estimées afin de déterminer celle qui se révèle la plus adéquate. Parmi ceux-ci, le point biserial et le HG (Hubert's Gamma) peuvent être utilisés, et indiquent à quel point la classification élaborée (nombre de catégories) parvient à réassigner de manière constante les séquences dans la bonne catégorie. De leur côté, les critères de ASW et de ASWw (Average Silhouette Width; Average Silhouette Width weighted) examine le degré d'homogénéité des catégories et si elles parviennent à se distinguer significativement les unes des autres. Enfin, l'index de Hubert (Hubert's C; HC) relate l'écart entre la classification testée et la meilleure classification théoriquement possible effectuée avec le même nombre de catégories et les mêmes propriétés de séquences. Ces indicées d'ajustement sont présentées ici pour les solutions de 2 à 8 catégories: on voit que la majorité des indices pointe vers la sélection du modèle à 5 catégories.

```
> #indices d'ajustement
> df.clust
```

	PBC	HG	HGSD	ASW	ASWw	CH	R2	CHsq	R2sq	HC
cluster2	0.43	0.52	0.52	0.31	0.31	80.60	0.17	156.94	0.29	0.23
cluster3	0.48	0.57	0.57	0.28	0.29	77.91	0.29	160.45	0.46	0.20
cluster4	0.51	0.64	0.64	0.27	0.28	75.22	0.37	166.16	0.57	0.17
cluster5	0.56	0.75	0.75	0.29	0.30	72.44	0.43	180.03	0.65	0.12
cluster6	0.55	0.76	0.76	0.26	0.27	65.10	0.46	163.91	0.68	0.12
cluster7	0.53	0.79	0.79	0.24	0.25	60.61	0.49	157.70	0.71	0.10
cluster8	0.50	0.77	0.77	0.21	0.22	55.95	0.51	144.47	0.73	0.12

```
> plot(df.clust, stat = 'all', norm = 'zscore', lwd = 2)
```



Afin d'examiner plus précisément les solutions à 4 et 5 catégories, on peut également produire les ASW et ASWw pour chaque catégorie de ces deux modèles.

```
> #ASW par catégorie
> cl4qual <- wcClusterQuality(OMtrate,df.clust$clustering$cluster4)
> cl4qual$ASW
```

	ASW	ASWw
1	0.32055324	0.32668322
2	-0.02325361	-0.01759512
3	0.46316417	0.46919603
4	0.69747693	0.70352740

```
> cl5qual <- wcClusterQuality(OMtrate,df.clust$clustering$cluster5)
> cl5qual$ASW
```

	ASW	ASWw
1	0.31534632	0.32152629
2	0.58591763	0.60019633
3	-0.05348231	-0.04607252
4	0.41341589	0.42000672
5	0.65550986	0.66239966

Finalement, on ajoute la classification choisie (5 catégories) comme variable dans la base de données, identifiant la catégorie d'appartenance de chaque observation, afin de pouvoir l'utiliser dans de futures analyses.

```
> #ajout d'une variable dans la base de données
> df$SA5class <- as.numeric(clusterH5)
> attach(df)
```

## Licence

Dans le respect des principes de la science ouverte, le code présenté ici est protégé par une licence CC-BY permettant la reproduction et la modification libre du contenu tant et aussi longtemps que la source est dûment citée. Pour plus de détails sur cette licence et les conditions qu'elle entraîne, consulter <https://creativecommons.org/licenses/by/2.0/>.