

PROJECT PE117519 *Herramientas para la enseñanza de la Geomática con programas de código abierto*

SUPPLEMENTARY MATERIAL

Analysis of High Temporal Resolution Land Use/Land Cover Trajectories

J.F. Mas



Dirección General de Asuntos
del Personal Académico

Universidad Nacional Autónoma de México
Centro de Investigación en Geografía Ambiental
Campus Morelia, Michoacán.

2019

Mas, J-F., Analysis of High Temporal Resolution Land Use/Land Cover Trajectories - Supplementary material, Centro de Investigaciones en Geografía Ambiental-Universidad Nacional Autónoma de México, 2019, 38 p.

Universidad Nacional Autónoma de México.
Ciudad Universitaria, Coyoacán, C.P. 04510, México, D.F.

Centro de Investigaciones en Geografía Ambiental.
Antigua carretera a Pátzcuaro 8701, Col. Exhacienda de San José de la Huerta, C.P. 58190,
Morelia, Michoacán, México.

Contents

1	Introduction	1
2	Materials and preprocessing	1
2.1	Importing cartographic data into R	1
2.2	Creation of tabular data	4
2.3	Preliminary sequence analysis	6
2.4	Creation of a state sequence object	7
3	Visualization of the sequences	8
4	Dissimilarity analysis	17
4.1	Computing of dissimilarities	17
4.2	Plotting the clusters	28
5	Assessing the effect of covariates	32
6	Acknowledgments	37

1 Introduction

In this document, a R script to carry out an analysis of land use / cover sequences similar to the one reported in the article *Analysis of High Temporal Resolution Land Use/Land Cover Trajectories* is presented. Methods, originally developed to assess life course trajectories and implemented in the R package **TraMineR** (<http://traminer.unige.ch/>), are explored in order to evaluate land change through the analysis of sequences of land use / cover.

We will process a very small portion of the maps from the MapBiomas project (mapbiomas.org/) used in the paper along with covariates maps (distance from roads, elevation and slope). The maps show the distribution of land use/cover in a small region of Northeastern Brazil for eleven dates from circa 1986 to 2016. The sequences analysis is mainly conducted using the R package **TraMineR** (Gabadinho et al., 2011).

2 Materials and preprocessing

2.1 Importing cartographic data into R

The maps (raster tif files) are:

1. Time series of land use / cover maps (serie.tif),
2. Distance from roads (dist_roads.tif),
3. Elevation (dem.tif),

4. Slope (slope.tif).

It is also necessary to install the packages **TraMineR**, **raster** and **fastcluster**. First, the tif files are imported as raster objects into R.

```
# Working directory and libraries
setwd("/home/jf/Dropbox/sequences/SupplementaryMaterials")
library(TraMineR)
library(raster)
library(fastcluster)

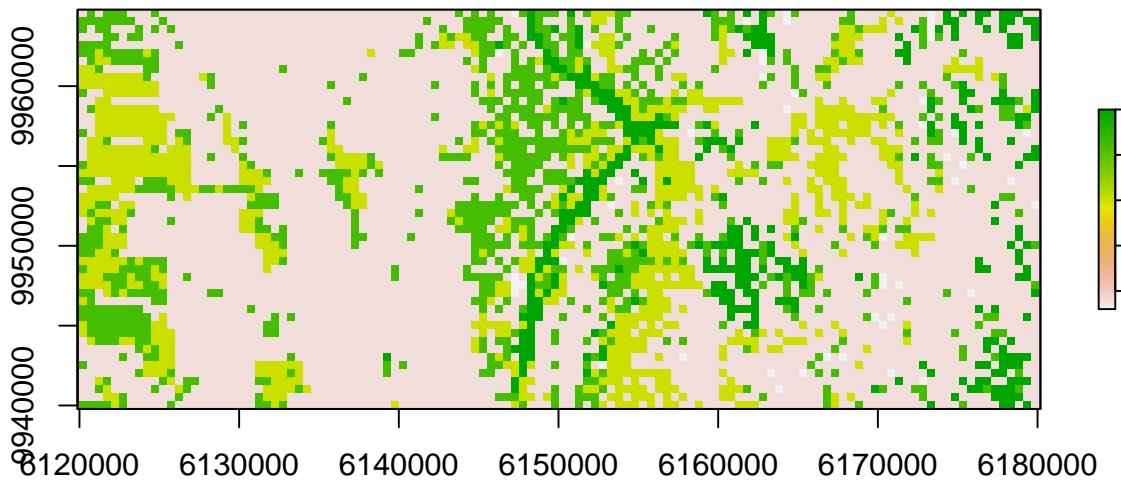
## Read rasters of time serie of land cover and covariates

## LUC maps for eleven dates
serie <- stack("serie.tif")
summary(serie)

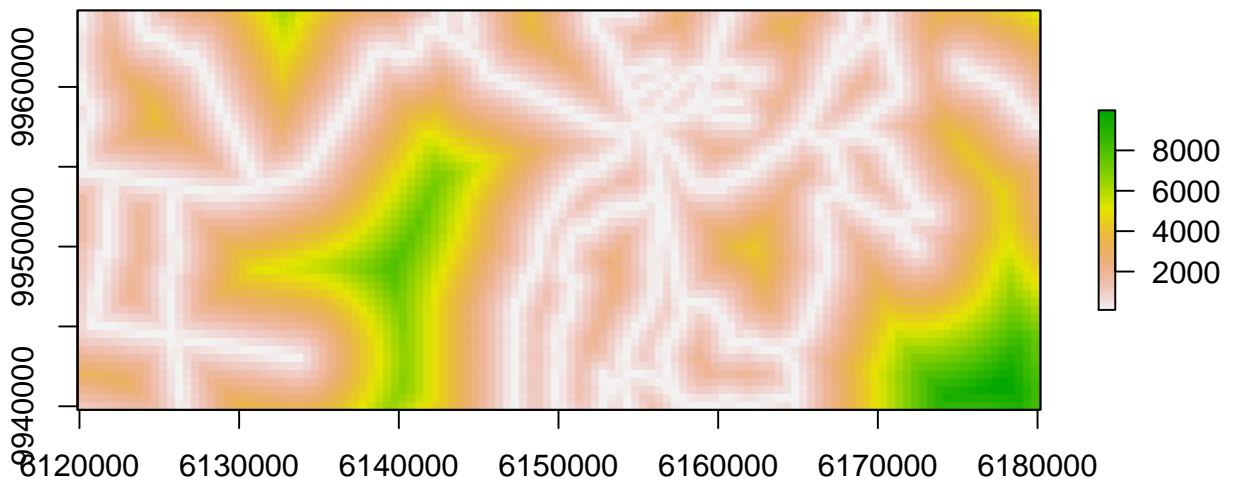
##          serie.1 serie.2 serie.3 serie.4 serie.5 serie.6 serie.7 serie.8
## Min.      3       3       3       3       3       3       3       3
## 1st Qu.   4       4       4       4       4       4       4       4
## Median    4       4       4       4       4       4       4       4
## 3rd Qu.   15      15      15      15      15      15      15      15
## Max.     25      25      25      25      25      25      25      25
## NA's      0       0       0       0       0       0       0       0
##          serie.9 serie.10 serie.11
## Min.      3       3       3
## 1st Qu.   4       4       4
## Median    4       4       4
## 3rd Qu.   15      15      15
## Max.     25      25      25
## NA's      0       0       0

# Distance from roads
dist <- raster("dist_roads.tif")
# Elevation
dem <- raster("dem.tif")
# Slope
slope <- raster("slope.tif")

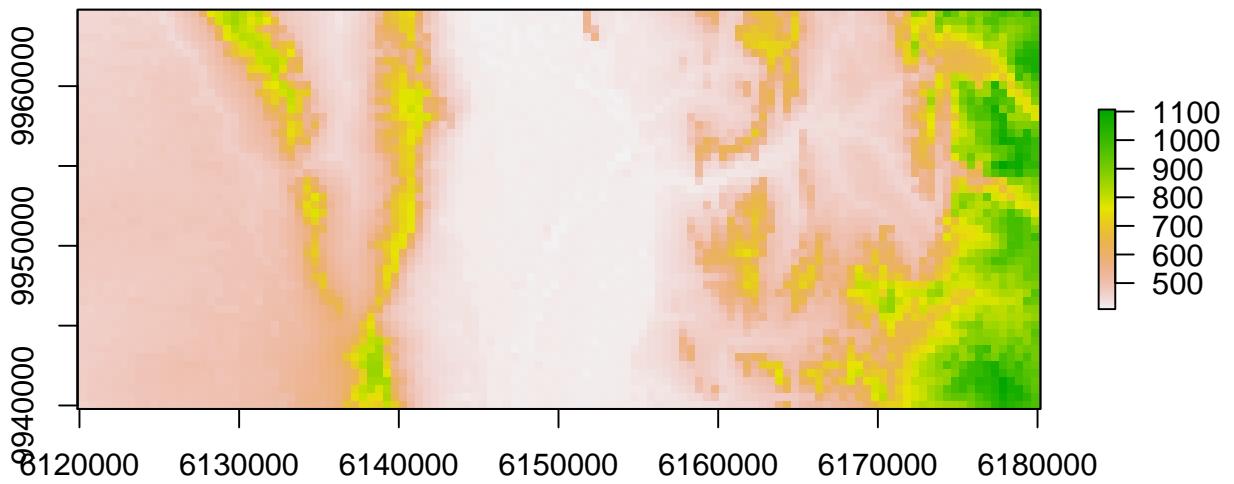
## Plot the maps
plot(serie[[1]])
```



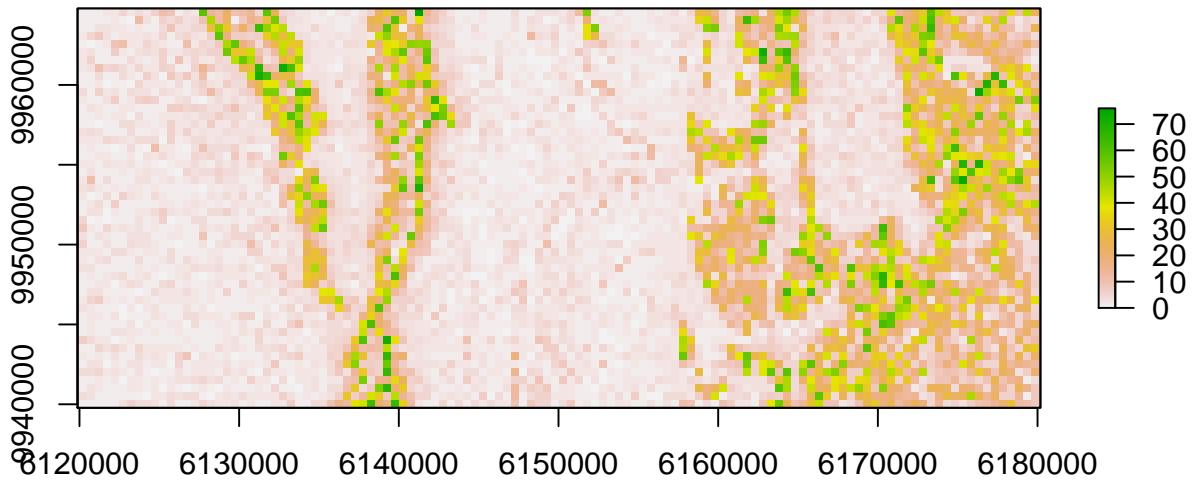
```
plot(dist)
```



```
plot(dem)
```



```
plot(slope)
```



2.2 Creation of tabular data

For further analysis, information from the raster data should be presented as tabular data. In this table, each row represents a cell, and each column its value in the maps (category number and

value of the covariates). Additional columns are created to hold the information of the covariates as categorical data. A column with the entire concatenated sequence is also created. The function **rasterToPoints** enables us to obtain the coordinates of each cell, this information will permit the spatial visualization of the results.

```
## Elaboration of a table which describes for each pixel the land category
# at each time step + value of covariates

## using rasterToPoints(r) to get coordinates
tab <- as.data.frame(rasterToPoints(stack(serie,dem,slope,dist)))
names(tab) <- c("x","y","c1986","c1989","c1992","c1995","c1998","c2001","c2004",
               "c2007","c2010","c2013","c2016","dem","slope","dist")
# A look at the first rows of the table:
head(tab)

##           x         y c1986 c1989 c1992 c1995 c1998 c2001 c2004 c2007 c2010
## 1 6120280 9964535     4    21    21    21    15    15    15    15    15
## 2 6120780 9964535    21    21    21    21    15    21    15    15    15
## 3 6121280 9964535     4    21    21    21    21    21    15    15    15
## 4 6121780 9964535    21    21    21    15    15    15    15    15    15
## 5 6122280 9964535    21    21    15    15    15    15    15    15    15
## 6 6122780 9964535    21    21    15    15    15    15    15    15    15
##   c2013 c2016 dem slope dist
## 1    15    15 451     1 100
## 2    21    15 452     1 600
## 3    15    15 453     1 1100
## 4    15    15 457     2 1600
## 5    21    21 454     2 1264
## 6    21    15 458     5 806

# and a summary:
summary(tab)

##           x                  y          c1986          c1989
##  Min.   :6120280   Min.   :9940035   Min.   : 3.000   Min.   : 3.000
##  1st Qu.:6135155   1st Qu.:9946035   1st Qu.: 4.000   1st Qu.: 4.000
##  Median :6150030   Median :9952285   Median : 4.000   Median : 4.000
##  Mean   :6150030   Mean   :9952285   Mean   : 9.148   Mean   : 9.778
##  3rd Qu.:6164905   3rd Qu.:9958535   3rd Qu.:15.000   3rd Qu.:15.000
##  Max.   :6179780   Max.   :9964535   Max.   :25.000   Max.   :25.000
##           c1992          c1995          c1998          c2001
##  Min.   : 3.000   Min.   : 3.000   Min.   : 3.000   Min.   : 3.000
##  1st Qu.: 4.000   1st Qu.: 4.000   1st Qu.: 4.000   1st Qu.: 4.000
##  Median : 4.000   Median : 4.000   Median : 4.000   Median : 4.000
##  Mean   : 9.591   Mean   : 9.357   Mean   : 9.075   Mean   : 9.399
##  3rd Qu.:15.000   3rd Qu.:15.000   3rd Qu.:15.000   3rd Qu.:15.000
##  Max.   :25.000   Max.   :25.000   Max.   :25.000   Max.   :25.000
```

```

##      c2004          c2007          c2010          c2013
##  Min.   : 3.000   Min.   : 3.000   Min.   : 3.000   Min.   : 3.000
##  1st Qu.: 4.000  1st Qu.: 4.000  1st Qu.: 4.000  1st Qu.: 4.000
##  Median : 4.000  Median : 4.000  Median : 4.000  Median : 4.000
##  Mean   : 9.373  Mean   : 9.063  Mean   : 9.112  Mean   : 8.493
##  3rd Qu.:15.000 3rd Qu.:15.000 3rd Qu.:15.000 3rd Qu.:15.000
##  Max.   :25.000  Max.   :25.000  Max.   :25.000  Max.   :25.000
##      c2016          dem          slope          dist
##  Min.   : 3.000   Min.   :409.0   Min.   : 0.00   Min.   : 100
##  1st Qu.: 4.000   1st Qu.:442.0   1st Qu.: 2.00   1st Qu.: 608
##  Median : 4.000   Median :482.0   Median : 4.00   Median :1486
##  Mean   : 8.673   Mean   :539.5   Mean   :10.08   Mean   :2136
##  3rd Qu.:15.000  3rd Qu.:578.0   3rd Qu.:13.00   3rd Qu.:3041
##  Max.   :25.000  Max.   :1107.0  Max.   :76.00   Max.   :9985

## Create categorical data from covariates
tab$demK<-cut(tab$dem, c(400,700,1000,1200))
tab$slopeK<-cut(tab$slope, c(-1,2,6,102))
tab$distK<-cut(tab$dist, c(0,500,1000,2000,5000,10000))
## Two categories for distance and slope (binary variable)
tab$distK2<-cut(tab$dist, c(0,1000,10000))
tab$slopeK2<-cut(tab$slope, c(-1,6,102))

# Put an extra column with the concatenated sequence
tab$sec <- as.vector(paste(tab[,3],tab[,4],tab[,5],tab[,6],tab[,7],
                           tab[,8],tab[,9],tab[,10],tab[,11],tab[,12],tab[,13],sep="-"))

```

2.3 Preliminar sequence analysis

As a first step, the categories which appear in the entire time serie are determined. There are six land cover categories (3, 4, 15, 18, 21, 25) which are 1) Forest, 2) Savanna, 3) Grasslands, 4) Mosaic of agriculture and pastures and 5) others. Then, existing sequences, and their respective frequency, are computed.

```

# Verify all the categories for all the time steps
clases <- c()
for (year in 3:13){
  clases <- unique(c(clases,tab[,year]))
}
print(sort(clases))

## [1] 3 4 15 18 21 25

# [1] 3 4 15 18 21 25

### Determine the number of different sequences

```

```

unique_traj <- sort(unique(tab$sec))
length(unique_traj) # 1651 different trajectories

## [1] 1651

# Determine the frequency of these trajectories
tab_freq <- as.data.frame(table(tab$sec))

# Sort the sequences from more to less frequent
tab_freq2 <- tab_freq[order(tab_freq$Freq, decreasing = TRUE),]
head(tab_freq2)

##                               Var1 Freq
## 1651          4-4-4-4-4-4-4-4-4-4 2515
## 1      15-15-15-15-15-15-15-15-15  271
## 867    25-25-25-25-25-25-25-25-25  142
## 551    21-21-21-21-21-21-21-21-21  49
## 695    21-21-4-4-4-4-4-4-4-4-4  44
## 795    21-4-4-4-4-4-4-4-4-4-4  36

```

2.4 Creation of a state sequence object

To carry out the analysis with **TraMineR**, a state sequence object should be created. It comprises information as the names and the colors assigned to each one of the land cover category ("states"). In sequence analysis, the alphabet is the finite set of possible states.

```

# Creation of a state sequence object to which names are assigned
# short state names (for printed output) and long state labels
# (for the graphics' legend).
# Also define the alphabet (be sure that the order of the names and labels
# is the same of the alphabet and a palette of colors for display.

## State sequence object
alphabet <- c(3,4,15,18,21,25)
labels <- c("Forest", "Savanna", "Pasture", "Agriculture", "Mosaic Agri/Past", "Others")
short_labels <- c("F", "Sav", "Past", "Agri", "Mosaic", "Others")
palette <- c("darkolivegreen", "green", "yellow", "red", "orange", "grey")
tab.seq <- seqdef(tab, 3:13, alphabet = alphabet, states = short_labels,
                  cpal = palette, labels = labels)

## [>] state coding:
##      [alphabet] [label] [long label]
##      1          3F      Forest
##      2          4Sav    Savanna
##      3          15Past  Pasture
##      4          18Agri  Agriculture

```

```

##      5          21Mosaic  Mosaic Agri/Past
##      6          25Others   Others
## [>] 6000 sequences in the data set
## [>] min/max sequence length: 11/11

```

3 Visualization of the sequences

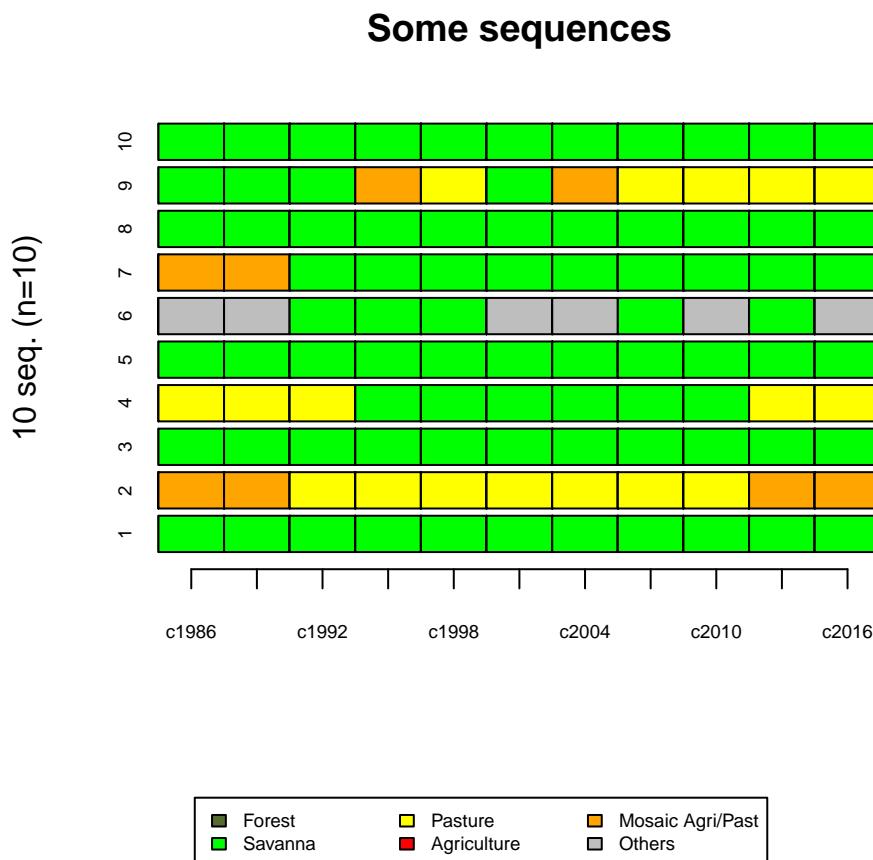
Figures which describe the observed sequences are elaborated: graph of some representative or random sequences, graph of all the observed sequences, most common sequences, land cover (states) proportion over time, landscape entropy over time, most frequent (dominant) category, average time of permanance of each land cover category.

```

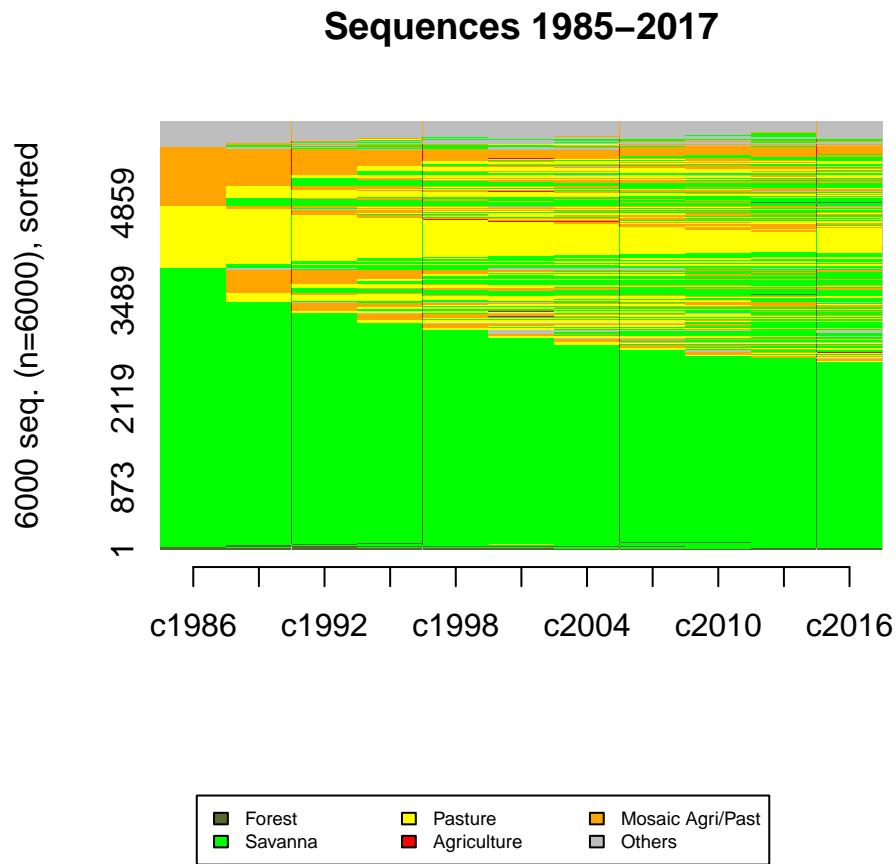
## Visualize the sequence data set

# Plot 10 sequences in the tab.seq sequence object (chosen to show diversity)
some_seq <- tab.seq[c(19,5,973,976,34,84,930,3893,993,995),]
seqiplot(some_seq, with.legend = T, cex.legend = 0.6, border = T,
         cex.axis=0.6, main = "Some sequences")

```

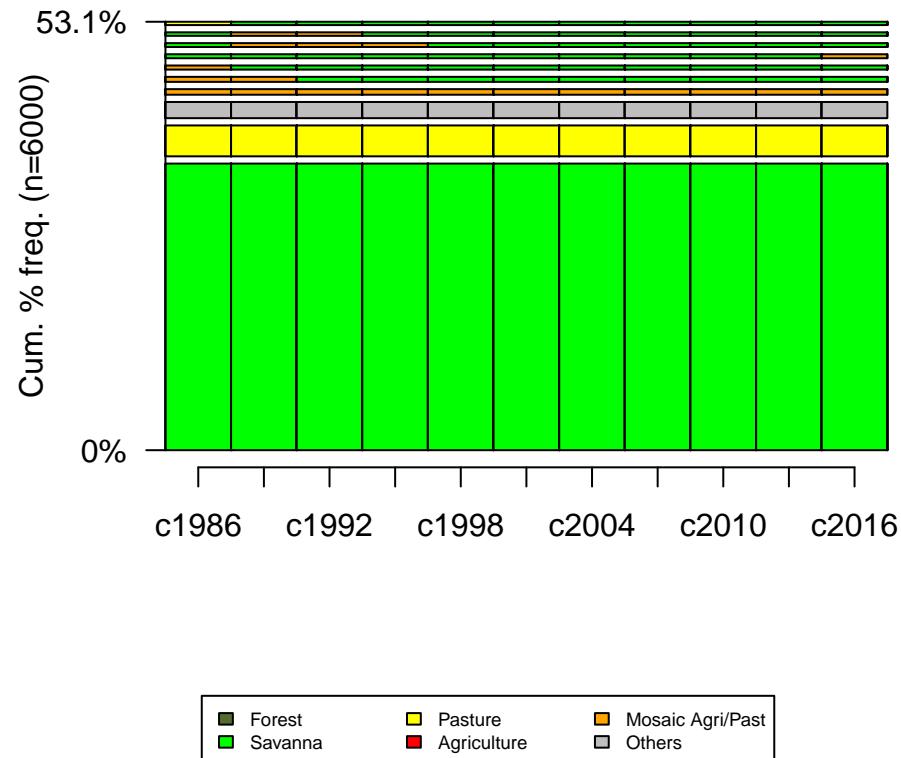


```
# Plot all the sequences in the data set, sorted by states from start.
seqIplot(tab.seq, sortv = "from.start", with.legend = T,
         cex.legend = 0.6, main = "Sequences 1985–2017")
```



```
# Plot the 10 most frequent sequences.
seqfplot(tab.seq, with.legend = T, cex.legend = 0.6,
         main="Most common sequences")
```

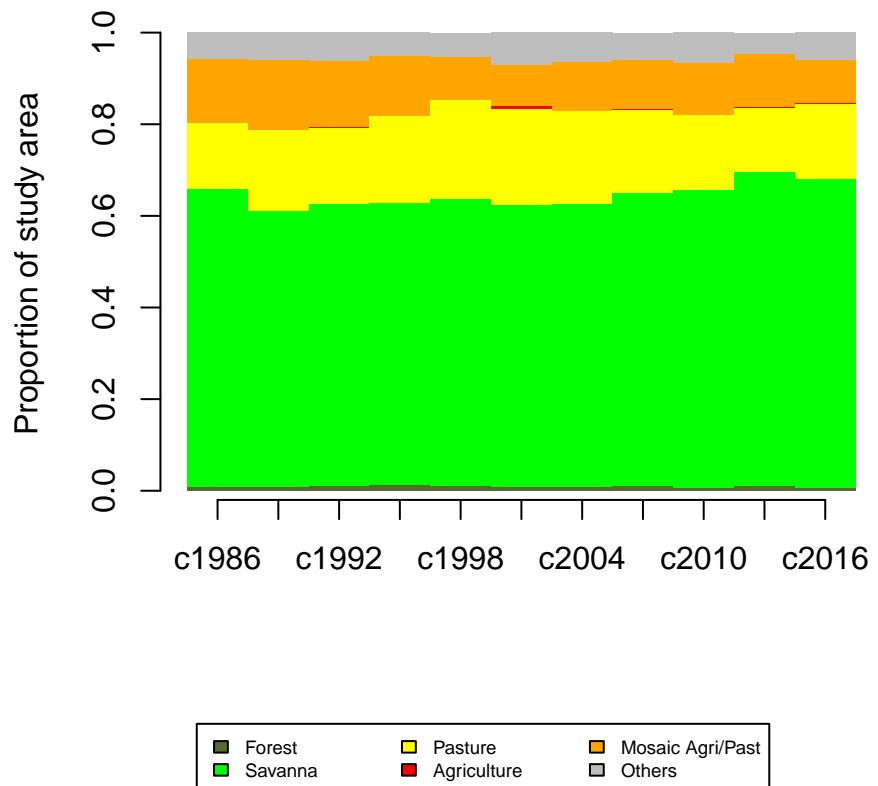
Most common sequences



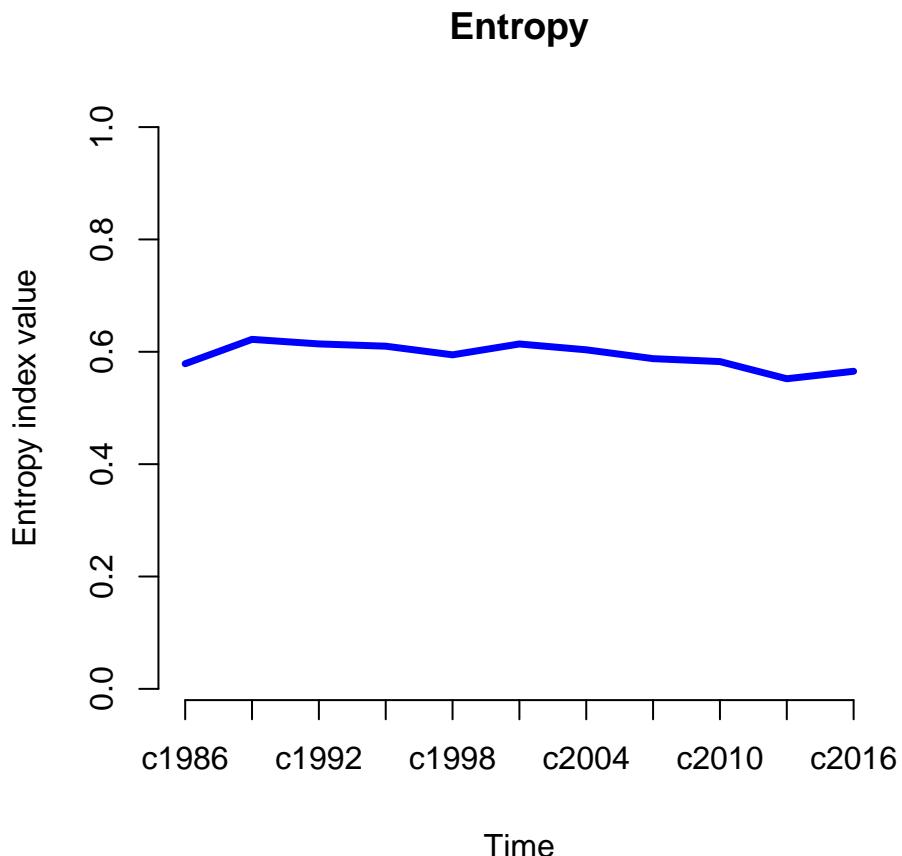
```
## Explore the sequence data set by computing and visualizing
# descriptive statistics

# Compute and plot the state distributions by time step.
# With border = NA, borders surrounding the bars are removed.
seqdplot(tab.seq, with.legend = T, border = NA,, cex.legend = 0.6,
main="Land cover (states) distribution", ylab="Proportion of study area")
```

Land cover (states) distribution

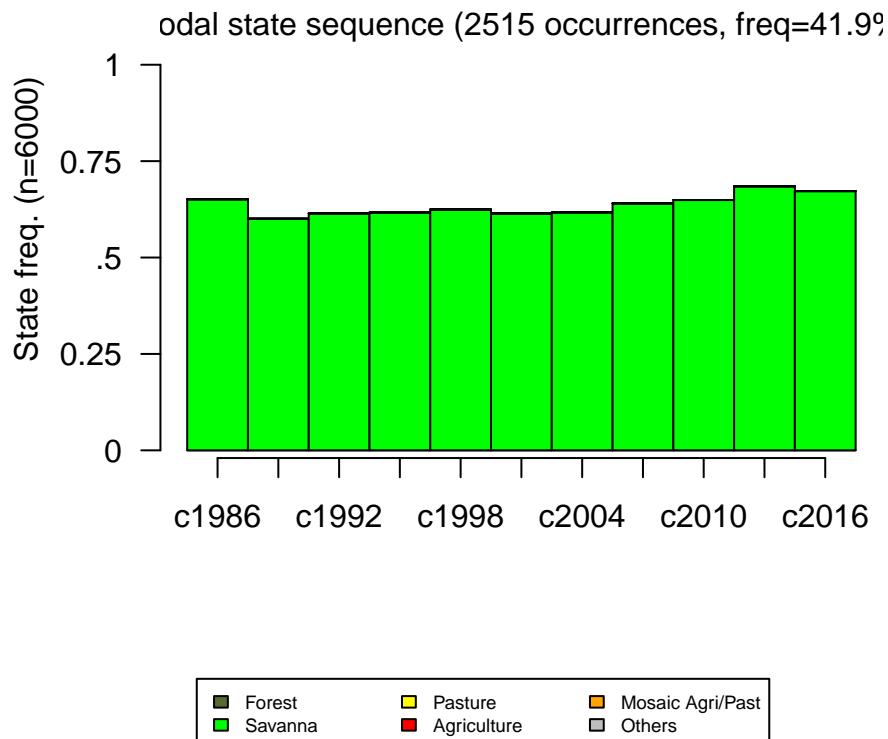


```
# Compute and plot the transversal entropy index (Landscape entropy over time)
seqHtplot(tab.seq, main = "Entropy", ylab="Entropy index value", xlab="Time")
```



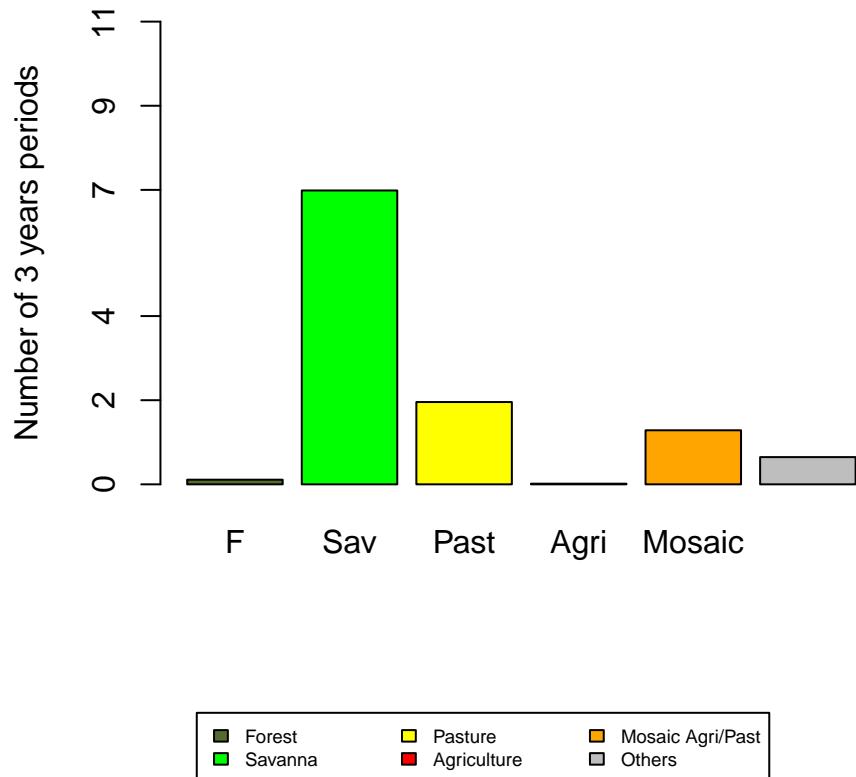
```
#Plot the sequence of modal states (dominant land cover)
# of the transversal state distributions.
seqmsplot(tab.seq, with.legend = T, main ="Most frequent land cover",
           cex.legend = 0.6)
```

Most frequent land cover



```
# Plot the mean time spent in each land cover category.  
seqmtpplot(tab.seq, with.legend = T, main = "Permanence", cex.legend = 0.6,  
            ylab="Number of 3 years periods")
```

Permanence



Spatial complexity is evaluated by computing and mapping the longitudinal entropy index (also known as Shannon index) and the turbulence index (Elzinga, 2007).

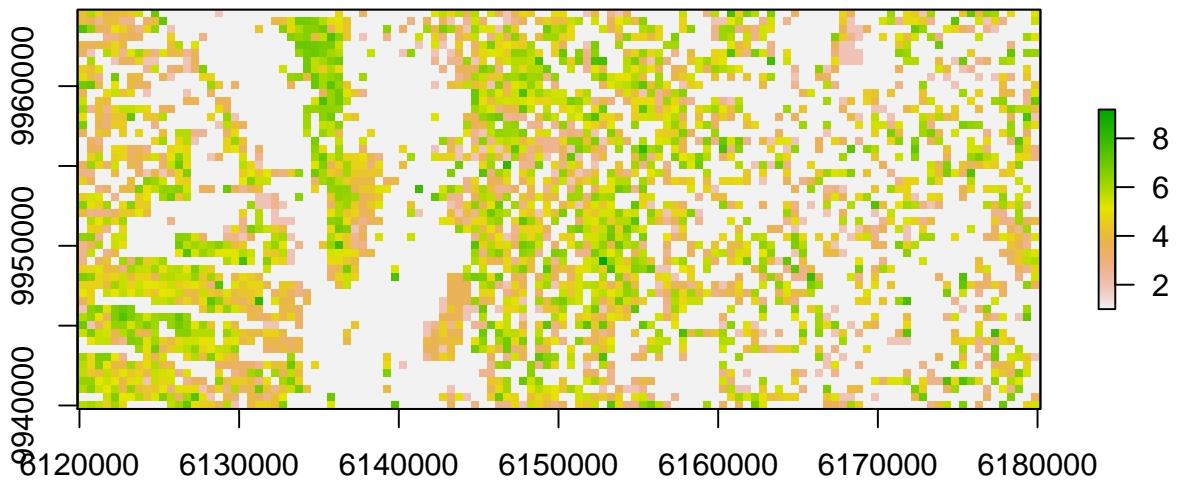
```
### Longitudinal turbulence and entropy indices
# Computed for each pixel over the time
tab$Turb <- seqST(tab.seq, norm=FALSE)

## [>] extracting symbols and durations ...
## [>] computing turbulence for 6000 sequence(s) ...

tab$Entrop <- seqient(tab.seq, norm=TRUE, base=exp(1))

## [>] computing entropy for 6000 sequences ...
## [>] computing state distribution for 6000 sequences ...

# Generate rasters which represent these indices
xyt <- as.data.frame(cbind(tab$x, tab$y, tab$Turb))
names(xyt) <- c("x", "y", "t")
coordinates(xyt) <- ~ x + y
gridded(xyt) <- TRUE
raster_turb <- raster(xyt)
plot(raster_turb)
```



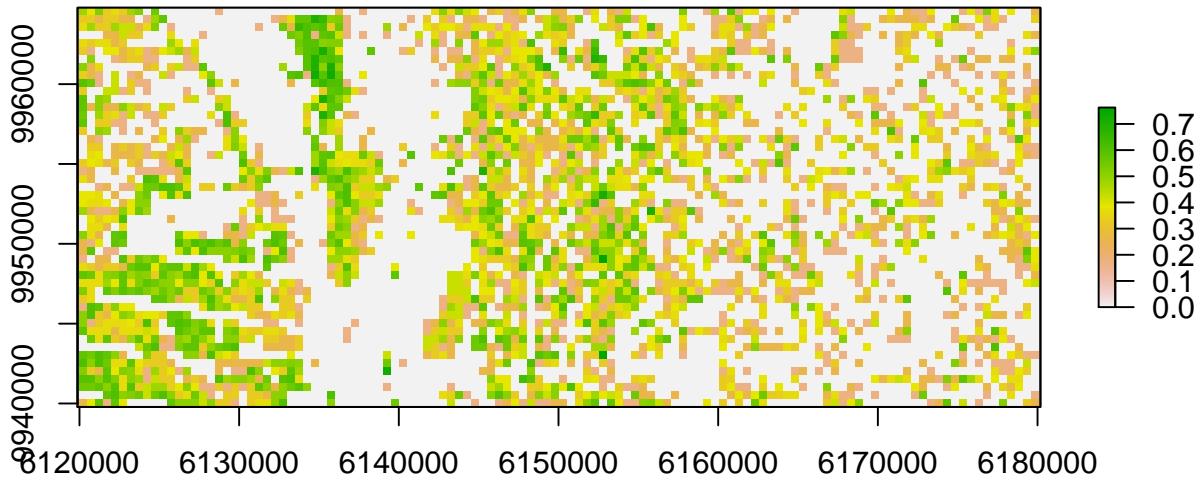
```

xye <- as.data.frame(cbind(tab$x,tab$y,tab$Entrop))
names(xye) <- c("x", "y", "e")
head(xye)

##           x         y         e
## 1 6120280 9964535 0.4799569
## 2 6120780 9964535 0.3845434
## 3 6121280 9964535 0.5217050
## 4 6121780 9964535 0.3270264
## 5 6122280 9964535 0.3658313
## 6 6122780 9964535 0.3270264

coordinates(xye) <- ~ x + y
gridded(xye) <- TRUE
raster_entrop <- raster(xye)
plot(raster_entrop)

```



```
# Calculate the correlation between both indices
cor(tab$Turb,tab$Entrop) # 0.9394874

##           Entropy
## Turbulence 0.9404563
```

Finally, the transition rates between land cover categories can be easily computed.

```
## Computes the transition rates
tr_rates <- seqtrate(tab.seq)

## [>] computing transition probabilities for states F/Sav/Past/Agri/Mosaic/Others
...
print(tr_rates)

##          [-> F]    [-> Sav]    [-> Past]    [-> Agri]    [-> Mosaic]
## [F ->] 7.594108e-01 0.22913257 0.001636661 0.0000000000 0.003273322
## [Sav ->] 3.669192e-03 0.93392815 0.016155004 0.0002375736 0.033973022
## [Past ->] 9.294544e-05 0.06022865 0.832326424 0.0033460359 0.099823404
## [Agri ->] 0.000000e+00 0.01449275 0.492753623 0.2898550725 0.202898551
## [Mosaic ->] 5.592059e-04 0.19795890 0.172095624 0.0039144415 0.600587166
## [Others ->] 5.673759e-04 0.12056738 0.011347518 0.0000000000 0.057872340
##          [-> Others]
## [F ->] 0.006546645
## [Sav ->] 0.012037061
## [Past ->] 0.004182545
```

```

## [Agri ->] 0.000000000
## [Mosaic ->] 0.024884664
## [Others ->] 0.809645390

```

4 Dissimilarity analysis

4.1 Computing of dissimilarities

Dissimilarity between sequences can be calculated using different methods. In this document, we used the longest common subsequence (LCS) distance, the longest common prefix (LCP) and the optimal matching (OM) based on transition rates (OM1) and on features (OM2) (Elzinga, 2007; Gabadinho et al., 2011; Studer and Ritschard, 2016). Then, a cluster hierarchical analysis is applied to identify similar sequences.

```

# Compute distances between sequences using different dissimilarity indices

## OM with substitution costs based on transition
## probabilities and indel set as half the maximum
## substitution cost
costs.tr <- seqcost(tab.seq, method = "TRATE", with.missing = FALSE)

## [>] creating substitution-cost matrix using transition rates ...
## [>] computing transition probabilities for states F/Sav/Past/Agri/Mosaic/Others
...
print(costs.tr)

## $indel
## [1] 1
##
## $sm
##          F->    Sav->   Past->   Agri-> Mosaic-> Others->
## F->      0.000000 1.767198 1.998270 2.000000 1.996167 1.992886
## Sav->    1.767198 0.000000 1.923616 1.985270 1.768068 1.867396
## Past->   1.998270 1.923616 0.000000 1.503900 1.728081 1.984470
## Agri->   2.000000 1.985270 1.503900 0.000000 1.793187 2.000000
## Mosaic-> 1.996167 1.768068 1.728081 1.793187 0.000000 1.917243
## Others-> 1.992886 1.867396 1.984470 2.000000 1.917243 0.000000

dist.om1 <- seqdist(tab.seq, method = "OM", indel = costs.tr$indel,
                     sm = costs.tr$sm, with.missing = F)

## [>] 6000 sequences with 6 distinct states
## [>] checking 'sm' (one value for each state, triangle inequality)
## [>] 1651 distinct sequences
## [>] min/max sequence length: 11/11
## [>] computing distances using the OM metric
## [>] elapsed time: 1.376 secs

```

```

dim(dist.om1)

## [1] 6000 6000

### OM based on features
tab_state_features <- data.frame(state=c(10,5,3,3,3,1))
costs.gower <- seqcost(tab.seq, method = "FEATURES", with.missing = FALSE,
                      state.features = tab_state_features)
print(costs.gower)

## $indel
## [1] 0.5
##
## $sm
##          F->    Sav->   Past->   Agri->  Mosaic->  Others->
## F->      0.0000000 0.5555556 0.7777778 0.7777778 0.7777778 1.0000000
## Sav->    0.5555556 0.0000000 0.2222222 0.2222222 0.2222222 0.4444444
## Past->   0.7777778 0.2222222 0.0000000 0.0000000 0.0000000 0.2222222
## Agri->   0.7777778 0.2222222 0.0000000 0.0000000 0.0000000 0.2222222
## Mosaic-> 0.7777778 0.2222222 0.0000000 0.0000000 0.0000000 0.2222222
## Others-> 1.0000000 0.4444444 0.2222222 0.2222222 0.2222222 0.0000000

dist.om2 <- seqdist(tab.seq, method = "OM", indel = costs.gower$indel,
                     sm = costs.gower$sm, with.missing = F)

## [>] 6000 sequences with 6 distinct states
## [>] checking 'sm' (one value for each state, triangle inequality)
## [>] 1651 distinct sequences
## [>] min/max sequence length: 11/11
## [>] computing distances using the OM metric
## [>] elapsed time: 1.076 secs

dim(dist.om2)

## [1] 6000 6000

## LCS
dist.lcs <- seqdist(tab.seq, method = "LCS")

## [>] 6000 sequences with 6 distinct states
## [>] creating a 'sm' with a substitution cost of 2
## [>] creating 6x6 substitution-cost matrix using 2 as constant value
## [>] 1651 distinct sequences
## [>] min/max sequence length: 11/11
## [>] computing distances using the LCS metric
## [>] elapsed time: 1.052 secs

## LCP
dist.lcp <- seqdist(tab.seq, method = "LCP")

```

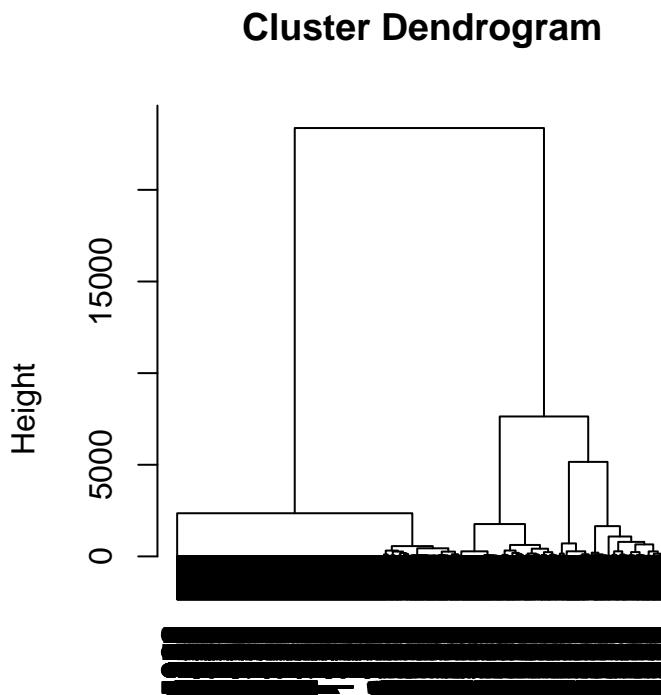
```

## [>] 6000 sequences with 6 distinct states
## [>] 1651 distinct sequences
## [>] min/max sequence length: 11/11
## [>] computing distances using the LCP metric
## [>] elapsed time: 0.664 secs

# Elaboration of a typology of the trajectories by hierarchical clustering
# of the sequences from the different distances and retrieve for each cell
# sequence the cluster membership of the 4 class solution.

## Cluster based on OM transition rates
clusterward_om1 <- hclust(as.dist(dist.om1), method="ward.D")
plot(clusterward_om1)

```



`as.dist(dist.om1)`
`hclust (*, "ward.D")`

```

cl_om1 <- cutree(clusterward_om1, k = 4)
tab$clusterom1 <- cl_om1
head(tab)

##           x      y c1986 c1989 c1992 c1995 c1998 c2001 c2004 c2007 c2010
## 1 6120280 9964535     4    21    21    21    15    15    15    15    15
## 2 6120780 9964535    21    21    21    21    15    21    15    15    15

```

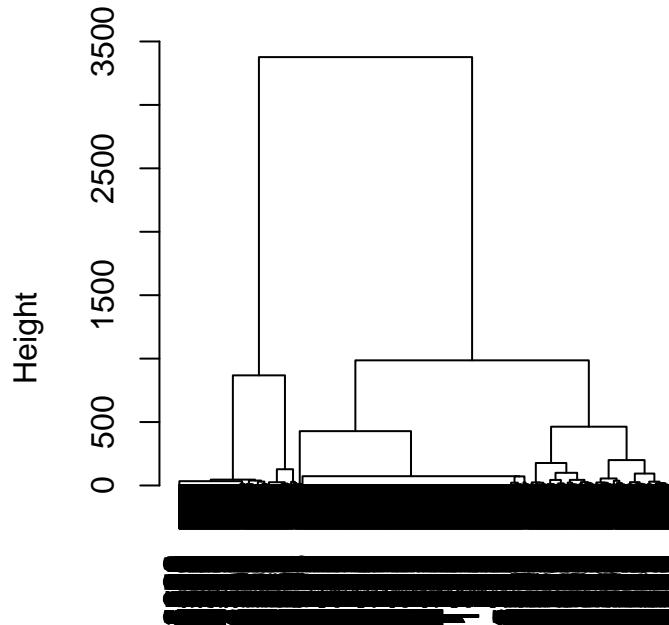
```

## 3 6121280 9964535      4   21   21   21   21   21   15   15   15
## 4 6121780 9964535      21   21   21   15   15   15   15   15   15
## 5 6122280 9964535      21   21   15   15   15   15   15   15   15
## 6 6122780 9964535      21   21   15   15   15   15   15   15   15
##   c2013 c2016 dem slope dist      demK slopeK          distK          distK2
## 1    15    15 451      1  100 (400,700] (-1,2]          (0,500] (0,1e+03]
## 2    21    15 452      1  600 (400,700] (-1,2] (500,1e+03] (0,1e+03]
## 3    15    15 453      1 1100 (400,700] (-1,2] (1e+03,2e+03] (1e+03,1e+04]
## 4    15    15 457      2 1600 (400,700] (-1,2] (1e+03,2e+03] (1e+03,1e+04]
## 5    21    21 454      2 1264 (400,700] (-1,2] (1e+03,2e+03] (1e+03,1e+04]
## 6    21    15 458      5  806 (400,700] (2,6] (500,1e+03] (0,1e+03]
##   slopeK2                      sec Turbulence Entropy clusterom1
## 1 (-1,6] 4-21-21-21-15-15-15-15-15-15 4.075664 0.4799569 1
## 2 (-1,6] 21-21-21-21-15-21-15-15-21-15 5.899578 0.3845434 1
## 3 (-1,6] 4-21-21-21-21-21-15-15-15-15 4.740480 0.5217050 1
## 4 (-1,6] 21-21-21-15-15-15-15-15-15-15 3.551410 0.3270264 1
## 5 (-1,6] 21-21-15-15-15-15-15-15-21-21 4.022744 0.3658313 1
## 6 (-1,6] 21-21-15-15-15-15-15-15-21-15 4.088201 0.3270264 1

## Cluster based on OM features
clusterward_om2 <- hclust(as.dist(dist.om2),method="ward.D")
plot(clusterward_om2)

```

Cluster Dendrogram



`as.dist(dist.om2)`
`hclust (*, "ward.D")`

```

cl_om2 <- cutree(clusterward_om2, k = 4)
tab$clusterom2 <- cl_om2
head(tab)

##           x      y c1986 c1989 c1992 c1995 c1998 c2001 c2004 c2007 c2010
## 1 6120280 9964535     4    21    21    21    15    15    15    15    15    15
## 2 6120780 9964535    21    21    21    21    15    21    15    15    15    15
## 3 6121280 9964535     4    21    21    21    21    21    21    15    15    15
## 4 6121780 9964535    21    21    21    15    15    15    15    15    15    15
## 5 6122280 9964535    21    21    15    15    15    15    15    15    15    15
## 6 6122780 9964535    21    21    15    15    15    15    15    15    15    15
##   c2013 c2016 dem slope dist      demK slopeK          distK          distK2
## 1    15    15 451     1  100 (400,700] (-1,2]        (0,500] (0,1e+03]
## 2    21    15 452     1  600 (400,700] (-1,2] (500,1e+03] (0,1e+03]
## 3    15    15 453     1 1100 (400,700] (-1,2] (1e+03,2e+03] (1e+03,1e+04]
## 4    15    15 457     2 1600 (400,700] (-1,2] (1e+03,2e+03] (1e+03,1e+04]
## 5    21    21 454     2 1264 (400,700] (-1,2] (1e+03,2e+03] (1e+03,1e+04]
## 6    21    15 458     5  806 (400,700] (2,6] (500,1e+03] (0,1e+03]
##   slopeK2          sec Turbulence Entropy clusterom1
## 1  (-1,6] 4-21-21-21-15-15-15-15-15-15  4.075664 0.4799569      1
## 2  (-1,6] 21-21-21-21-15-21-15-15-15-21-15  5.899578 0.3845434      1

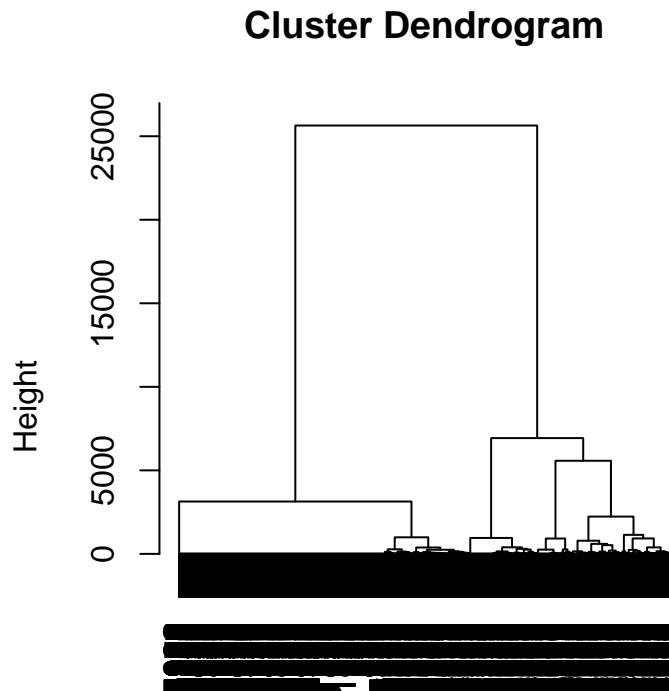
```

```

## 3  (-1,6]  4-21-21-21-21-15-15-15-15-15  4.740480 0.5217050 1
## 4  (-1,6]  21-21-21-15-15-15-15-15-15-15  3.551410 0.3270264 1
## 5  (-1,6]  21-21-15-15-15-15-15-15-21-21  4.022744 0.3658313 1
## 6  (-1,6]  21-21-15-15-15-15-15-15-21-15  4.088201 0.3270264 1
##   clusterom2
## 1          1
## 2          1
## 3          1
## 4          1
## 5          1
## 6          1

## Cluster based on LCS
clusterward_lcs <- hclust(as.dist(dist.lcs),method="ward.D")
plot(clusterward_lcs)

```



`as.dist(dist.lcs)`
`hclust (*, "ward.D")`

```

cl_lcs <- cutree(clusterward_lcs, k = 4)
tab$clusterlcs <- cl_lcs
head(tab)

##           x      y c1986 c1989 c1992 c1995 c1998 c2001 c2004 c2007 c2010

```

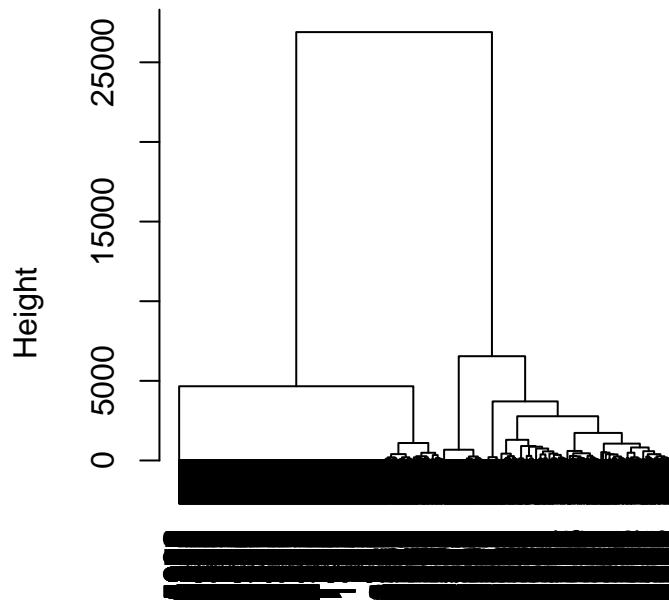
```

## 1 6120280 9964535      4   21   21   21   15   15   15   15   15
## 2 6120780 9964535      21  21   21   21   15   21   15   15   15
## 3 6121280 9964535      4   21   21   21   21   21   15   15   15
## 4 6121780 9964535      21  21   21   15   15   15   15   15   15
## 5 6122280 9964535      21  21   15   15   15   15   15   15   15
## 6 6122780 9964535      21  21   15   15   15   15   15   15   15
##    c2013 c2016 dem slope dist      demK slopeK          distK          distK2
## 1     15    15 451      1  100 (400,700] (-1,2]      (0,500] (0,1e+03]
## 2     21    15 452      1  600 (400,700] (-1,2] (500,1e+03] (0,1e+03]
## 3     15    15 453      1 1100 (400,700] (-1,2] (1e+03,2e+03] (1e+03,1e+04]
## 4     15    15 457      2 1600 (400,700] (-1,2] (1e+03,2e+03] (1e+03,1e+04]
## 5     21    21 454      2 1264 (400,700] (-1,2] (1e+03,2e+03] (1e+03,1e+04]
## 6     21    15 458      5  806 (400,700] (2,6]  (500,1e+03] (0,1e+03]
##    slopeK2          sec Turbulence Entropy clusterom1
## 1 (-1,6] 4-21-21-21-15-15-15-15-15-15-15-15 4.075664 0.4799569 1
## 2 (-1,6] 21-21-21-21-15-21-15-15-15-21-15 5.899578 0.3845434 1
## 3 (-1,6] 4-21-21-21-21-15-15-15-15-15-15 4.740480 0.5217050 1
## 4 (-1,6] 21-21-21-15-15-15-15-15-15-15-15 3.551410 0.3270264 1
## 5 (-1,6] 21-21-15-15-15-15-15-15-21-21 4.022744 0.3658313 1
## 6 (-1,6] 21-21-15-15-15-15-15-15-15-21-15 4.088201 0.3270264 1
##    clusterom2 clusterlcs
## 1           1           1
## 2           1           2
## 3           1           2
## 4           1           1
## 5           1           1
## 6           1           1

## Cluster based on LCP
clusterward_lcp <- hclust(as.dist(dist.lcp),method="ward.D")
plot(clusterward_lcp)

```

Cluster Dendrogram



`as.dist(dist.lcp)`
`hclust (*, "ward.D")`

```

cl_lcp <- cutree(clusterward_lcp, k = 4)
tab$clusterlcp <- cl_lcp
head(tab)

##          x      y c1986 c1989 c1992 c1995 c1998 c2001 c2004 c2007 c2010
## 1 6120280 9964535     4    21    21    21    15    15    15    15    15    15
## 2 6120780 9964535    21    21    21    21    15    21    15    15    15    15
## 3 6121280 9964535     4    21    21    21    21    21    21    15    15    15
## 4 6121780 9964535    21    21    21    15    15    15    15    15    15    15
## 5 6122280 9964535    21    21    15    15    15    15    15    15    15    15
## 6 6122780 9964535    21    21    15    15    15    15    15    15    15    15
##   c2013 c2016 dem slope dist      demK slopeK           distK           distK2
## 1    15    15 451     1  100 (400,700] (-1,2]       (0,500] (0,1e+03]
## 2    21    15 452     1  600 (400,700] (-1,2] (500,1e+03] (0,1e+03]
## 3    15    15 453     1 1100 (400,700] (-1,2] (1e+03,2e+03] (1e+03,1e+04]
## 4    15    15 457     2 1600 (400,700] (-1,2] (1e+03,2e+03] (1e+03,1e+04]
## 5    21    21 454     2 1264 (400,700] (-1,2] (1e+03,2e+03] (1e+03,1e+04]
## 6    21    15 458     5  806 (400,700] (2,6] (500,1e+03] (0,1e+03]
##   slopeK2           sec Turbulence Entropy clusterom1
## 1 (-1,6] 4-21-21-21-15-15-15-15-15-15 4.075664 0.4799569      1
## 2 (-1,6] 21-21-21-21-15-21-15-15-15-21-15 5.899578 0.3845434      1

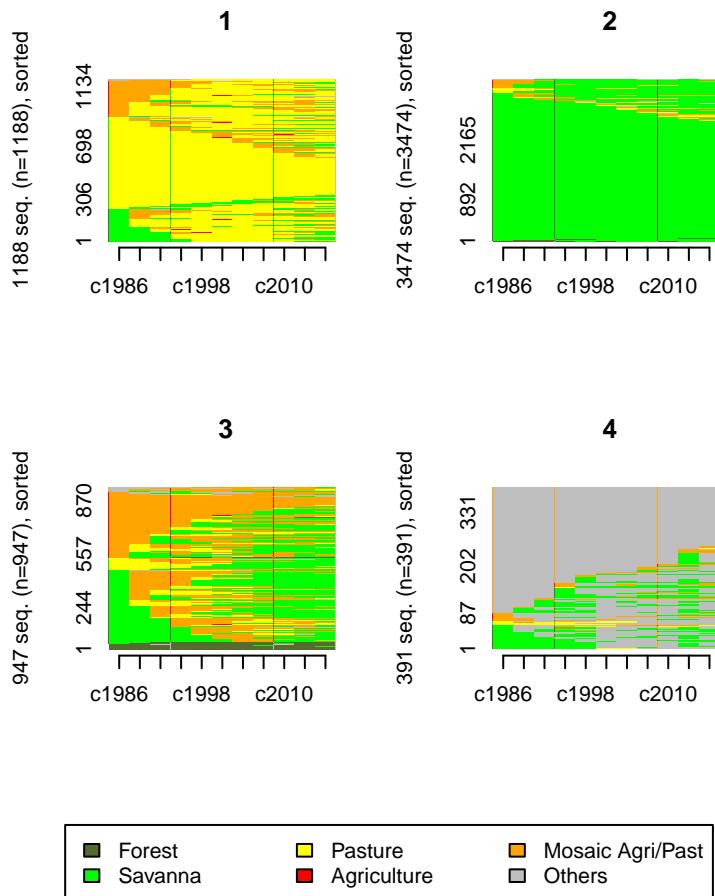
```

```

## 3  (-1,6] 4-21-21-21-21-15-15-15-15-15-15  4.740480 0.5217050 1
## 4  (-1,6] 21-21-21-15-15-15-15-15-15-15  3.551410 0.3270264 1
## 5  (-1,6] 21-21-15-15-15-15-15-15-21-21  4.022744 0.3658313 1
## 6  (-1,6] 21-21-15-15-15-15-15-15-15-21-15 4.088201 0.3270264 1
##   clusterom2 clusterlcs clusterlcp
## 1           1           1           1
## 2           1           2           1
## 3           1           2           1
## 4           1           1           1
## 5           1           1           1
## 6           1           1           1

# Plot all the sequences within each cluster para los 4 métodos
# OM1
seqIplot(tab.seq, group = tab$clusterom1, sortv = "from.start")

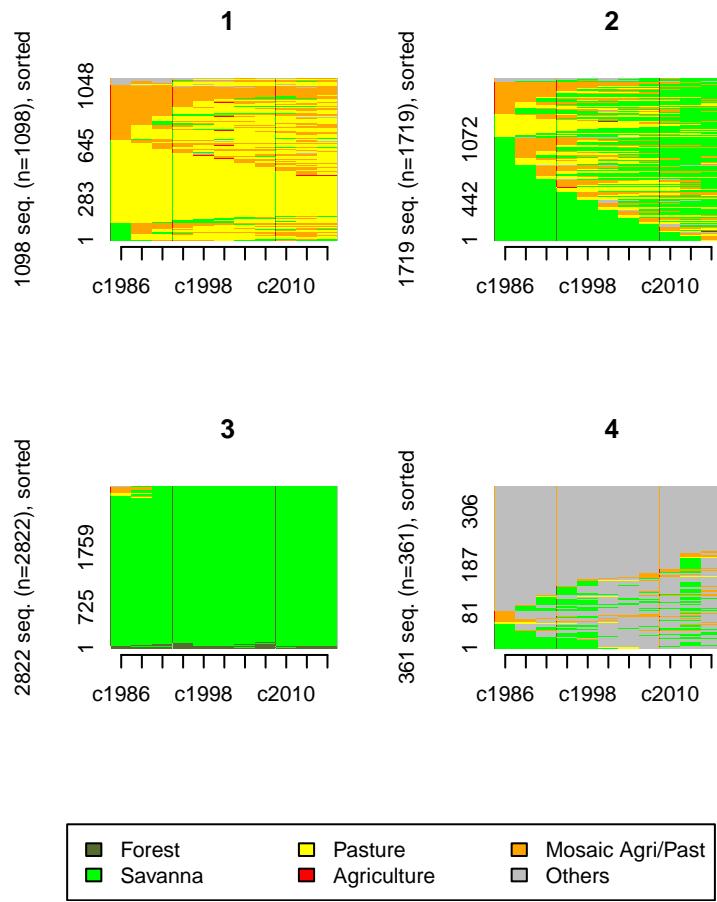
```



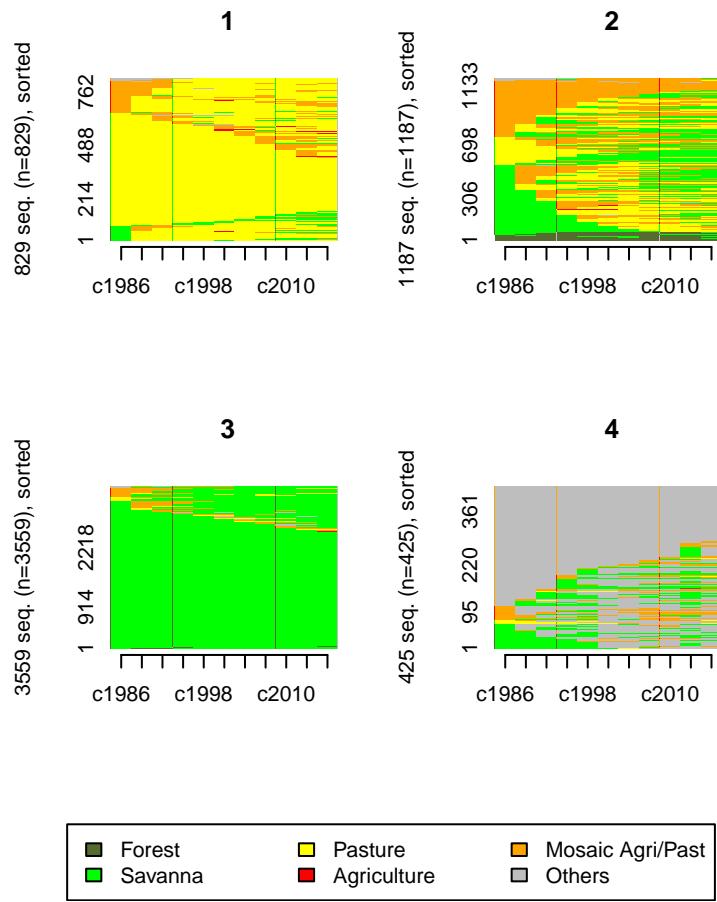
```

# OM2
seqIplot(tab.seq, group = tab$clusterom2, sortv = "from.start")

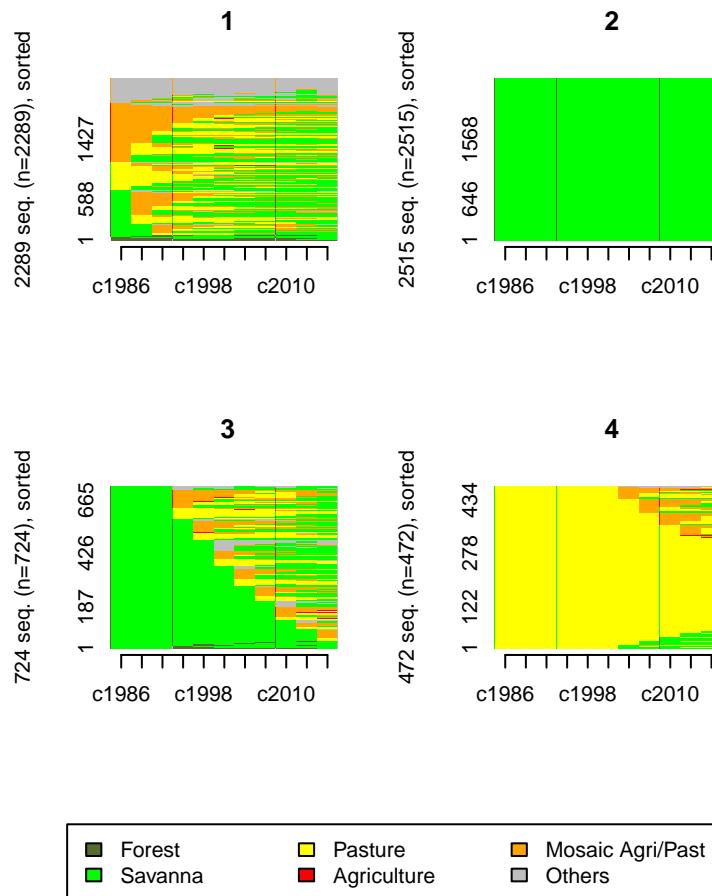
```



```
# LCS
seqIplot(tab.seq, group = tab$clusterlcs, sortv = "from.start")
```



```
# LCP
seqIplot(tab.seq, group = tab$clusterlcp, sortv = "from.start")
```

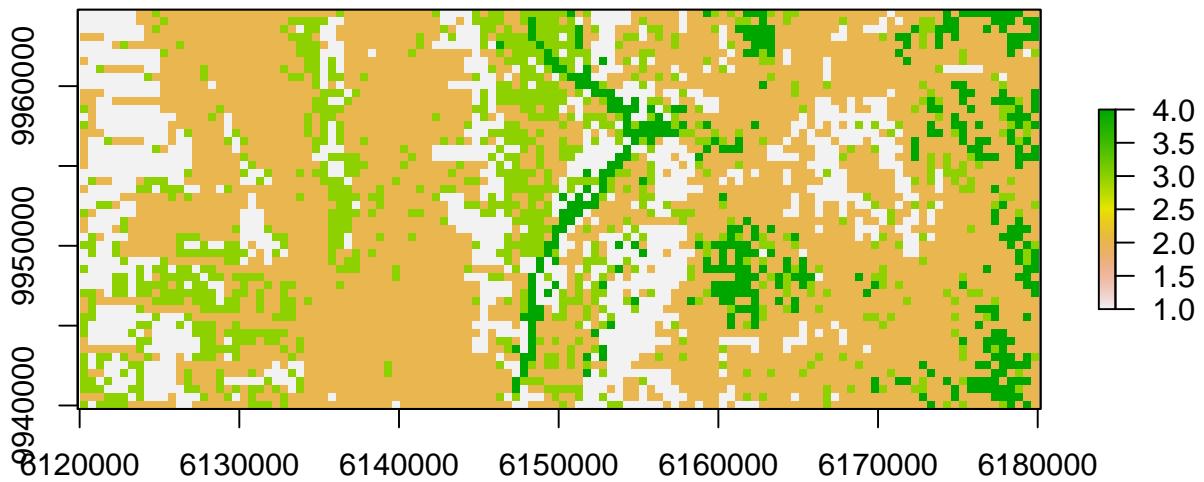


4.2 Plotting the clusters

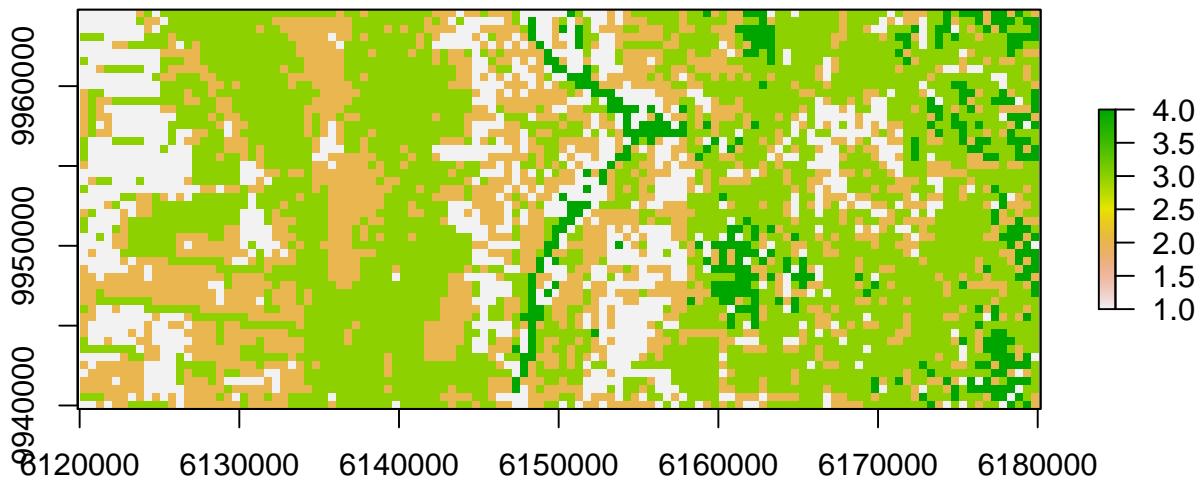
Rasters are created from the tabular data using the coordinates and the cluster number in order to assess the spatial distribution of the clusters.

```
##### Plot clusters' spatial distribution

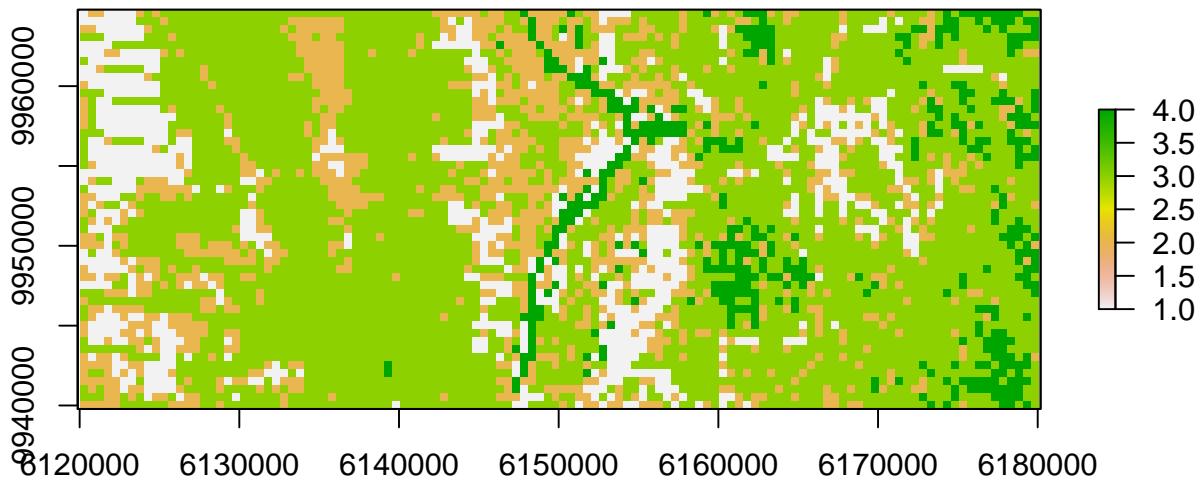
# Elaborate raster OM1
xyz <- as.data.frame(cbind(tab$x,tab$y,tab$clusterom1))
names(xyz) <- c("x", "y", "z")
xyz <- xyz[complete.cases(xyz), ]
coordinates(xyz) <- ~ x + y
gridded(xyz) <- TRUE
raster_com1 <- raster(xyz)
plot(raster_com1)
```



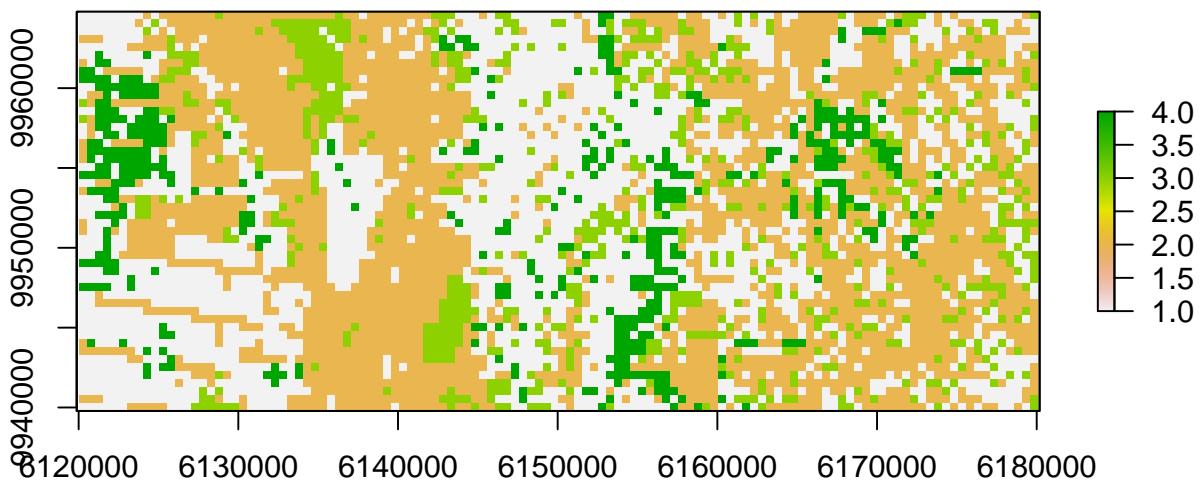
```
# Elaborate raster OM2
xyz <- as.data.frame(cbind(tab$x,tab$y,tab$clusterom2))
names(xyz) <- c("x", "y", "z")
xyz <- xyz[complete.cases(xyz), ]
coordinates(xyz) <- ~ x + y
gridded(xyz) <- TRUE
raster_com2 <- raster(xyz)
plot(raster_com2)
```



```
# Elaborate raster LCS
xyz <- as.data.frame(cbind(tab$x,tab$y,tab$clusterlcs))
names(xyz) <- c("x", "y", "z")
xyz <- xyz[complete.cases(xyz), ]
coordinates(xyz) <- ~ x + y
gridded(xyz) <- TRUE
raster_clcs <- raster(xyz)
plot(raster_clcs)
```



```
# Elaborate raster LCP
xyz <- as.data.frame(cbind(tab$x,tab$y,tab$clusterlcp))
names(xyz) <- c("x", "y", "z")
xyz <- xyz[complete.cases(xyz), ]
coordinates(xyz) <- ~ x + y
gridded(xyz) <- TRUE
raster_clcp <- raster(xyz)
plot(raster_clcp)
```



5 Assessing the effect of covariates

Understanding the role of drivers is an important topic in land science. Multi-factor analysis of variance (ANOVA) based on the dissimilarity matrix allows to evaluate the part of discrepancy explained by covariates. The chi-square test enables identifying subsequences that discriminate significantly groups based on covariates. These analysis can be focussed on a subset of sequences which contain specific subsequences.

```
## Run discrepancy analyses to study how sequences are related to covariates

# Compute and test the share of discrepancy explained by different categories
# on covariates
da1 <- dissassoc(dist.om1, group = tab$slopeK, R = 50)
print(da1$stat)

##                      t0 p.value
## Pseudo F    290.97565167   0.02
## Pseudo Fbf  292.87606297   0.02
## Pseudo R2    0.08845655   0.02
## Bartlett    162.58157351   0.02
## Levene      255.53957691   0.02

da2 <- dissassoc(dist.om1, group = tab$distK, R = 50)
print(da2$stat)

##                      t0 p.value
```

```

## Pseudo F    65.8847395   0.02
## Pseudo Fbf 68.3539356   0.02
## Pseudo R2   0.0421087   0.02
## Bartlett   78.7893707   0.02
## Levene     61.0976289   0.02

da3 <- dissasssoc(dist.om1, group = tab$distK2, R = 50)
print(da3$stat)

##                      t0 p.value
## Pseudo F    182.78595649   0.02
## Pseudo Fbf 172.82498629   0.02
## Pseudo R2   0.02957325   0.02
## Bartlett   35.10210286   0.02
## Levene     130.17397698   0.02

# Selecting event subsequences:
# The analysis was restricted to sequences that exhibit the state Mosaic

tabe.seq <- seqecreate(tab.seq, use.labels = FALSE)
mosaic <- seqecontain(tabe.seq, event.list = c("Mosaic"))
mosaic_tab <- tab[mosaic,]
mosaic.seq <- tab.seq <- seqdef(mosaic_tab, 3:13, alphabet = alphabet,
                                 states = short_labels, cpa1 = palette, labels = labels)

## [>] state coding:
##      [alphabet] [label] [long label]
##      1          3F      Forest
##      2          4Sav    Savanna
##      3          15Past  Pasture
##      4          18Agri  Agriculture
##      5          21Mosaic Mosaic Agri/Past
##      6          25Others Others
## [>] 834 sequences in the data set
## [>] min/max sequence length: 11/11

mosaic.seqe <- seqecreate(mosaic.seq, use.labels = FALSE)

# Look for frequent event subsequences and plot the 10 most frequent ones.
fsubseq <- seqefsub(mosaic.seqe, pmin.support = 0.05)
head(fsubseq)

##           Subsequence   Support Count
## 1          (Mosaic) 1.0000000 834
## 2  (Mosaic)-(Mosaic>Sav) 0.5539568 462
## 3          (Mosaic>Sav) 0.5539568 462
## 4  (Mosaic)-(Mosaic>Past) 0.4892086 408
## 5          (Mosaic>Past) 0.4892086 408

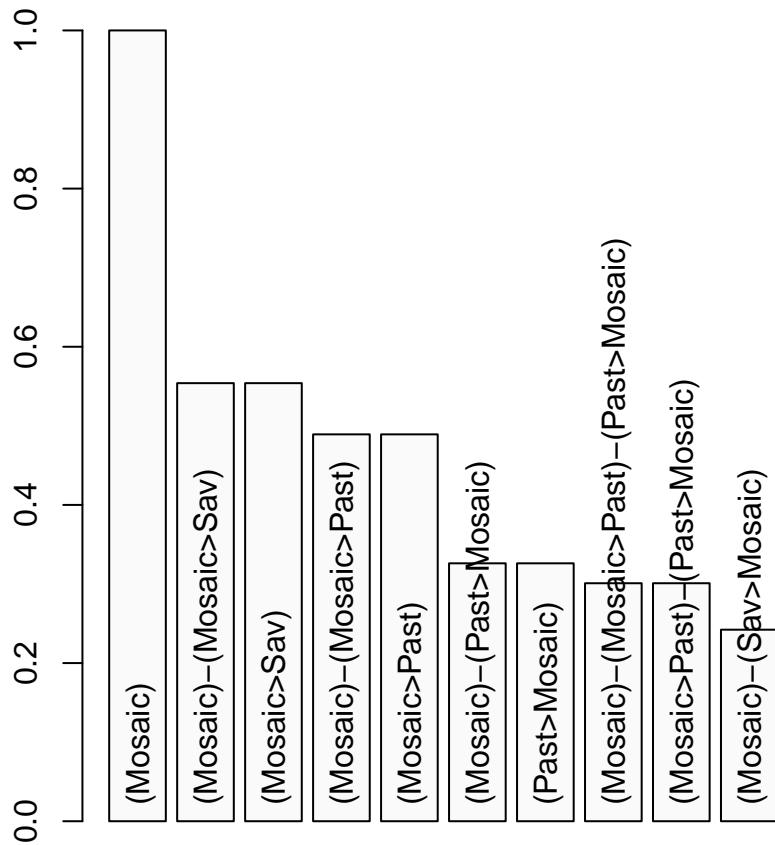
```

```

## Computed on 834 event sequences
## Constraint Value
## count.method COBJ

# 10 Most common subsequences
plot(fsubseq[1:10], col = "grey98")

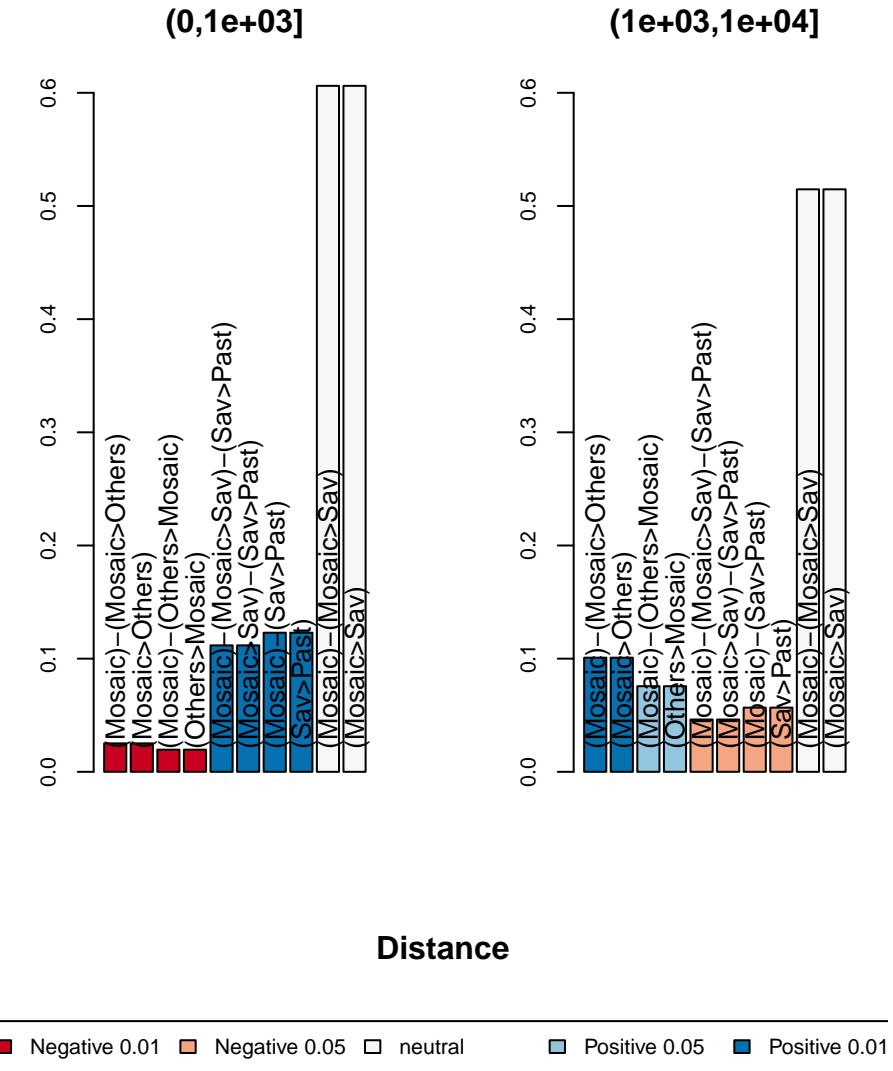
```



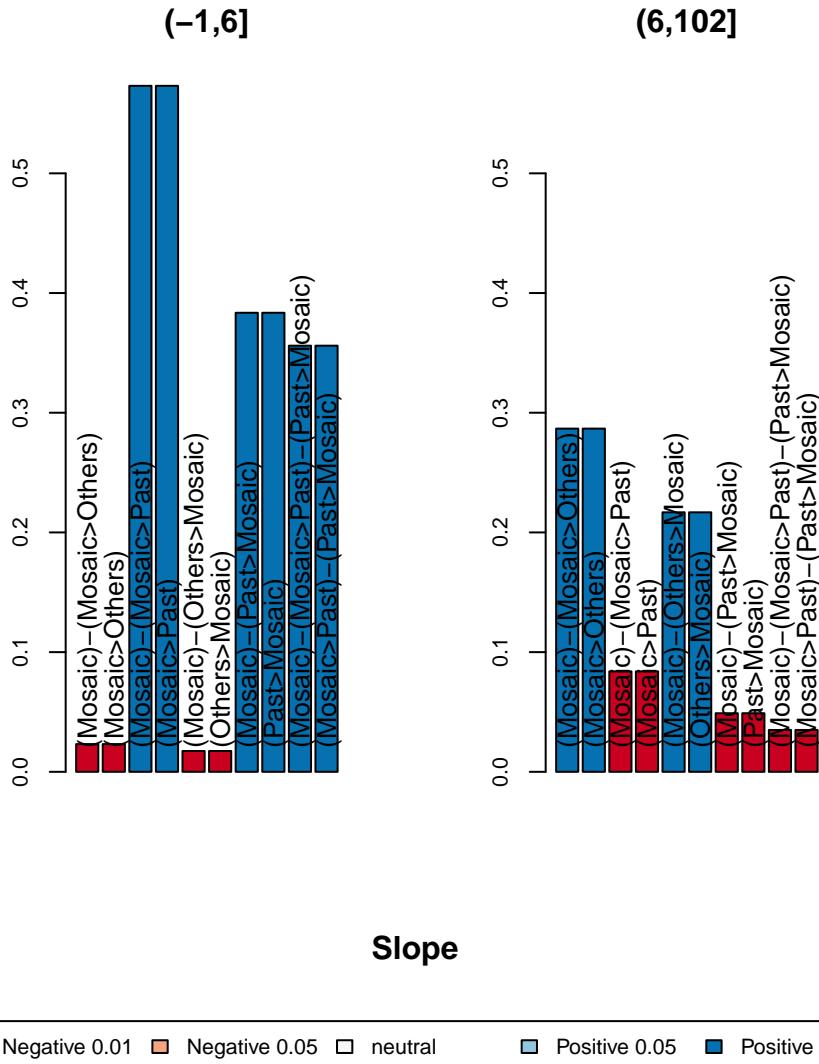
```

# Determine the subsequences of transitions which best discriminate the groups as
# areas close and faraway from roads
discr1 <- seqecmpgroup(fsubseq, group = mosaic_tab$distK2)
plot(discr1[1:10], cex=1, cex.legend=0.8, legend.title="Distance", cex.lab=0.8,
     cex.axis = 0.8)

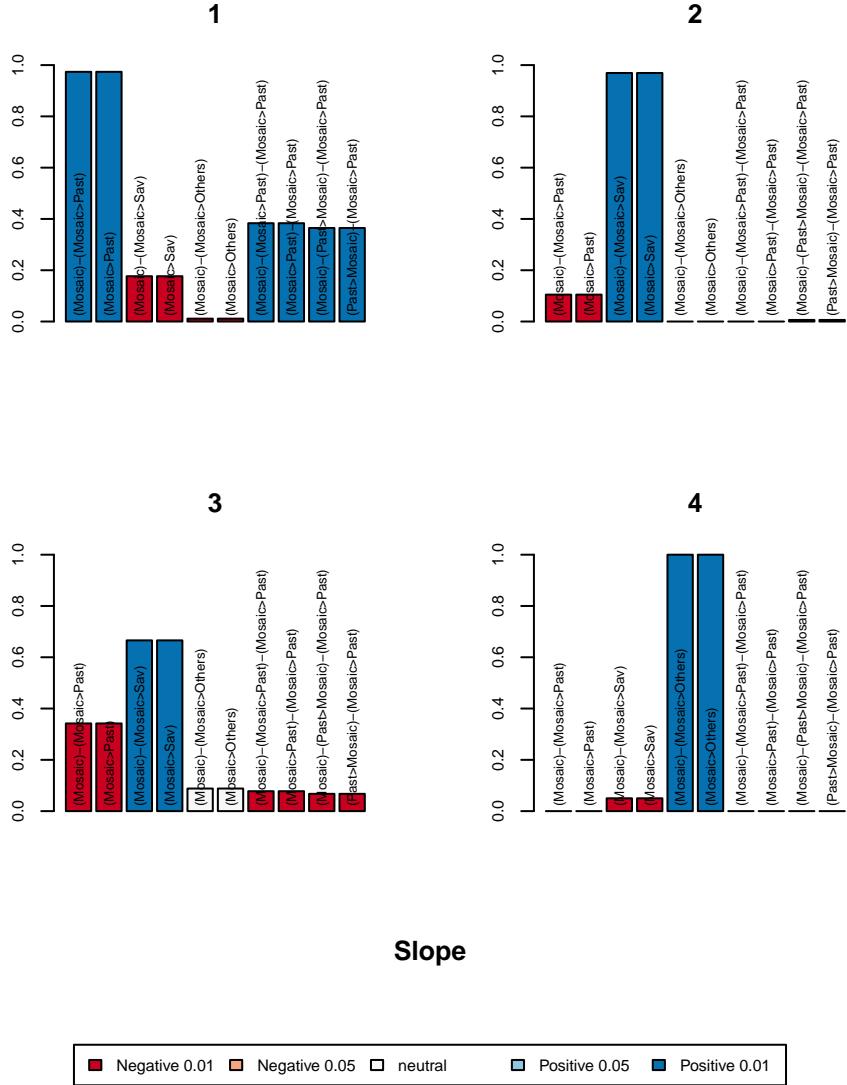
```



```
# areas with moderate vs steep slope
discr2 <- seqecmpgroup(fsubseq, group = mosaic_tab$slopeK2)
plot(discr2[1:10], cex=1, cex.legend=0.8, legend.title="Slope", cex.lab=0.8,
     cex.axis = 0.8)
```



```
# Clusters
discr3 <- seqecmpgroup(fsubseq, group = mosaic_tab$clusterom1)
plot(discr3[1:10], cex=0.6, cex.legend=0.8, legend.title="Slope", cex.lab=0.8,
     cex.axis = 0.8)
```



In this last example, the chi-square test identifies subsequences that discriminate significantly the sequences of each group obtained from the cluster analysis.

6 Acknowledgments

This material was elaborated with the support of the *Programa de Apoyo a Proyectos para la Innovación y Mejoramiento de la Enseñanza - PAPIME-DGAPA* at the *Universidad Nacional Autónoma de México* (Project PE117519 *Herramientas para la enseñanza de la Geomática con programas de código abierto*). I thank the authors of the package **TraMineR** for developing and sharing this excellent and useful program.

Bibliography

- Elzinga, C. (2007). Sequence analysis: Metric representations of categorical time series. Technical report, Department of Social Science Research Methods Vrije Universiteit Amsterdam, Amsterdam.
- Gabadinho, A., Ritschard, G., Müller, N., and Studer, M. (2011). Analyzing and Visualizing State Sequences in R with TraMineR. *Journal of Statistical Software, Articles*, 40(4):1–37.
- Studer, M. and Ritschard, G. (2016). What matters in differences between life trajectories: A comparative review of sequence dissimilarity measures. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 179:481–511.