Vignette moveNT

Guillaume Bastille-Rousseau

April 18, 2017

Simulating movement strategies - *sim_mov*

The function *sim_mov* generates movement trajectories including patches and movement between patches. Movement within patches can follow an ornstein uhlenbeck process (based on *simm.mou* function from package *adehabitatLT*) or two-states movement model (based on *simmData* function from package *moveHMM*). Movement between patches is following a brownian bridge movement model (based on *simm.bb* function from package *adehabitatLT*). Generated outputs are of the class *ltraj* from package *adehabitatlt*.

```
# Simulating migration with two-states model
mig<-sim_mov(type="2states", npatches=2, ratio=2, nswitch=25, ncore=150, grph=F)
```

```
## [1] "Packages are loaded correctly"
```

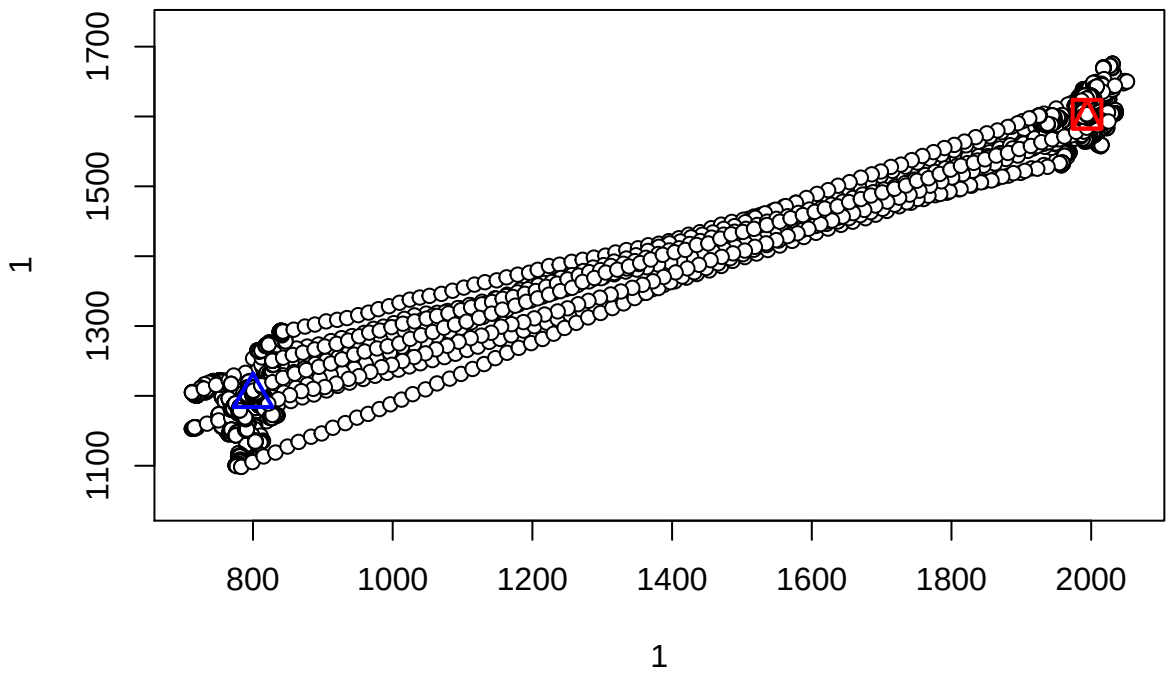
```
mig
```

```
##
## *********** List of class ltraj ***********
##
## Type of the traject: Type II (time recorded)
## * Time zone: GMT *
## Regular traject. Time lag between two locs: 1 seconds
##
## Characteristics of the bursts:
##   id burst nb.reloc NAs          date.begin            date.end
## 1 id    id     4425   0 1960-01-01 00:00:01 1960-01-01 01:13:45
##
##
##  infolocs provided. The following variables are available:
## [1] "out.Corri"
```

```
head(ld(mig))
```

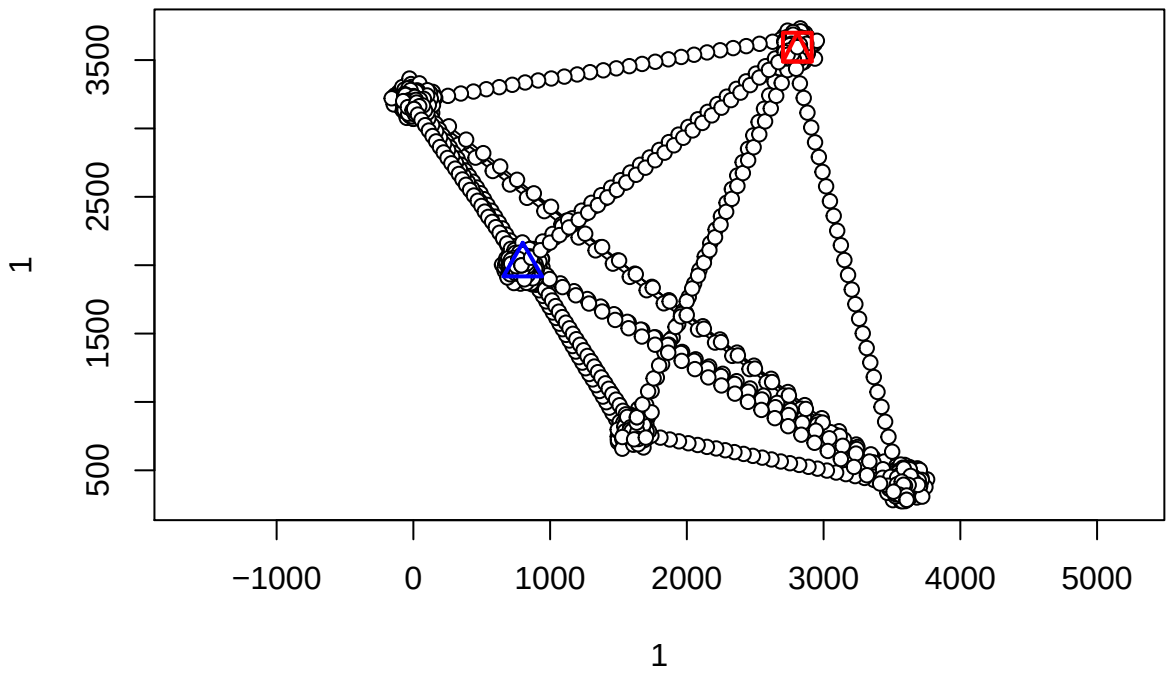
```
##          x        y                date           dx           dy
## 1 800.0000 1200.000 1960-01-01 00:00:01  0.112144927 -0.037390709
## 2 800.1121 1199.963 1960-01-01 00:00:02  0.569195137 -0.229896626
## 3 800.6813 1199.733 1960-01-01 00:00:03  0.004129487 -0.003410731
## 4 800.6855 1199.729 1960-01-01 00:00:04 -0.393493080  0.043177994
## 5 800.2920 1199.772 1960-01-01 00:00:05  1.165473051  2.018292704
## 6 801.4574 1201.791 1960-01-01 00:00:06  0.089369831  0.042546565
##          dist dt        R2n  abs.angle   rel.angle id burst out.Corri
## 1 0.118214000  1 0.00000000 -0.3218233          NA id    id         2
## 2 0.613869337  1 0.01397455 -0.3838620 -0.06203865 id    id         2
## 3 0.005355908  1 0.53566680 -0.6903623 -0.30650036 id    id         2
## 4 0.395854952  1 0.54314595  3.0322999 -2.56052307 id    id         2
## 5 2.330629287  1 0.13701564  1.0471191 -1.98518084 id    id         2
## 6 0.098980690  1 5.33102573  0.4443235 -0.60279561 id    id         2
```

```
plot(mig)
```



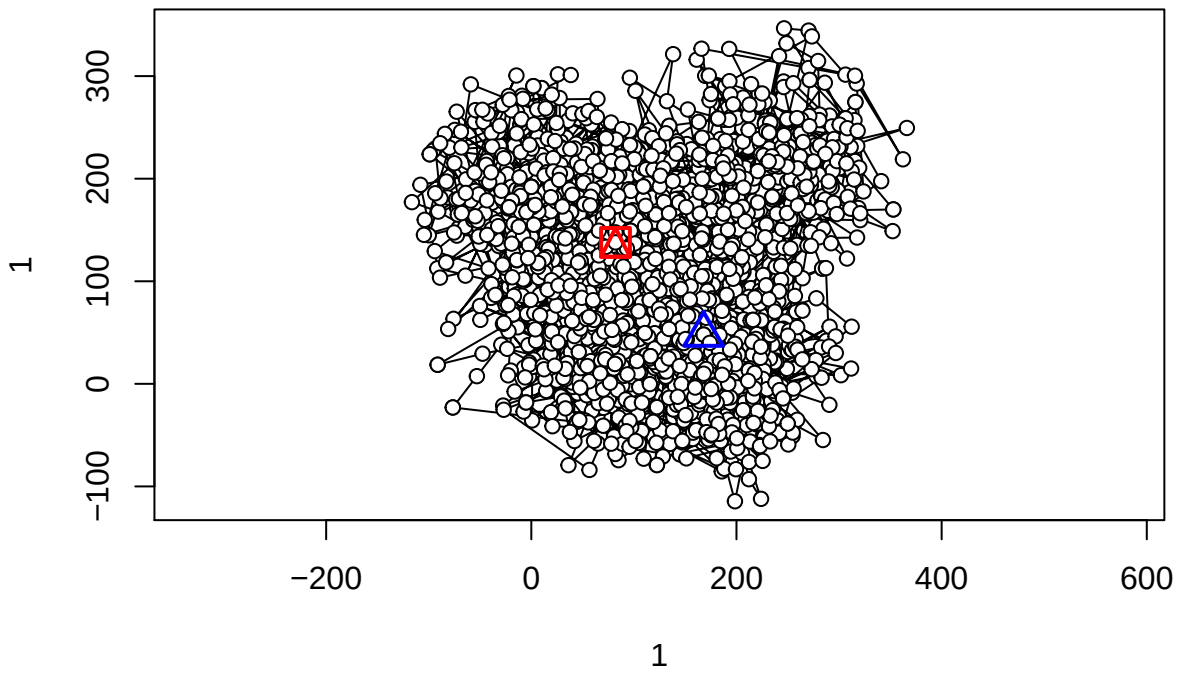

```
# Simulating multi-patches movement with ornstein uhlenbeck process
patches<-sim_mov(nswitch=25, ncore=150, ratio=5, type="OU", npatches=5, grph=T)
```

```
## [1] "Packages are loaded correctly"
```




```
# Simulating sedentary movement
seden<-sim_mov(type="OU", npatches=10, spacecore=12, ratio=3, nswitch=150, ncore=20, grph=T)
```

```
## [1] "Packages are loaded correctly"
```



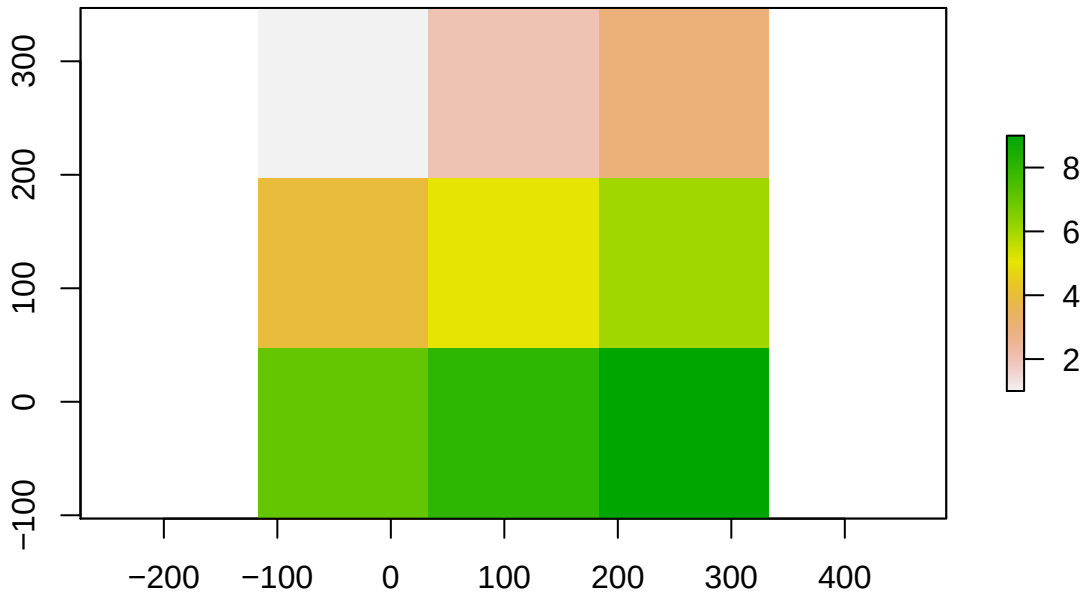
Converting movement to adjancency matrix - *traj2adj*

The function *traj2adj* convert a trajectory object of class *ltraj* to an adjacency matrix. This is done by overlapping a grid over the relocation data and tallying the number of transitions between each pixels. User needs to specify the grid size, which can be based on distance travelled. The function *quant* is a wrapper that allows to sample a quantile of step length distribution from a ltraj object. Output produced by *traj2adj* is a list containing the adjacency matrix, the grid used (raster format), and a raster indicating pixel numbers that are occupied. These rasters are used by other functions such as *adj2stack* and *clustnet*.

```
# Using sedentary movement and user specific grid-size
adj_seden<-traj2adj(seden, res=150) #Pixel size of 100m
adj_seden[[1]] # Adjency matrix
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
##  [1,]   75   16    0   51   10    0    0    0    0
##  [2,]   14   52   22   10   63    3    0    0    0
##  [3,]    0   17  161    0    7   53    0    0    0
##  [4,]   54   14    0  251  114    0    7   13    0
##  [5,]    9   58   10  124  995   64   11  150   19
##  [6,]    0    6   47    0   63  140    0   14   38
##  [7,]    0    0    0    7   10    0   16   19    0
##  [8,]    0    0    0   10  159   17   18  371   68
##  [9,]    0    1    0    0   19   29    0   77  174
```

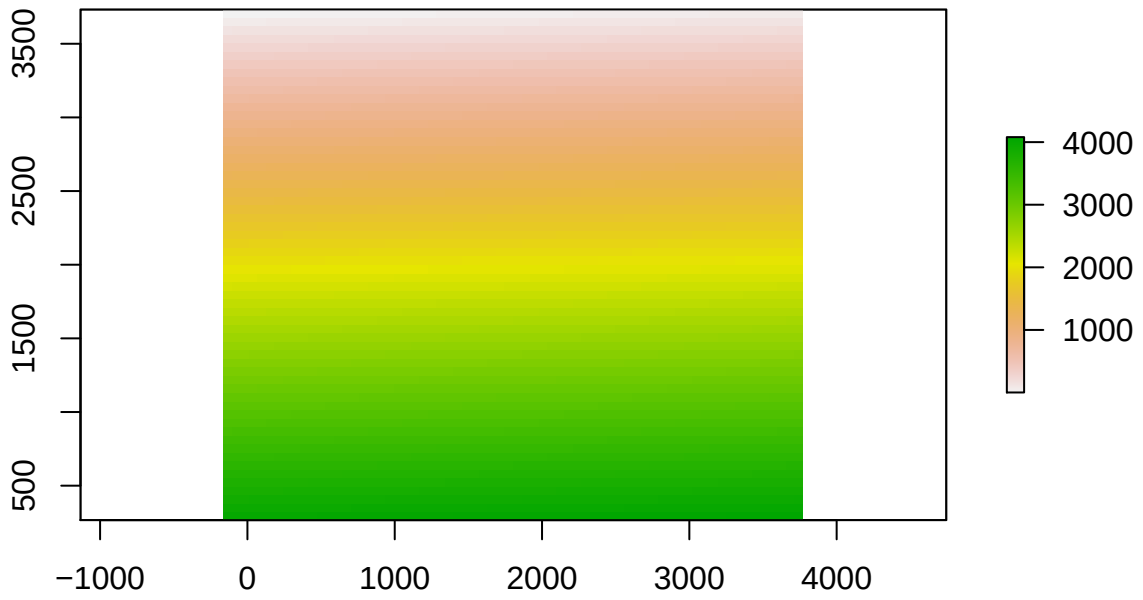
```r
plot(adj_seden[[2]]) #Plot grid used
```



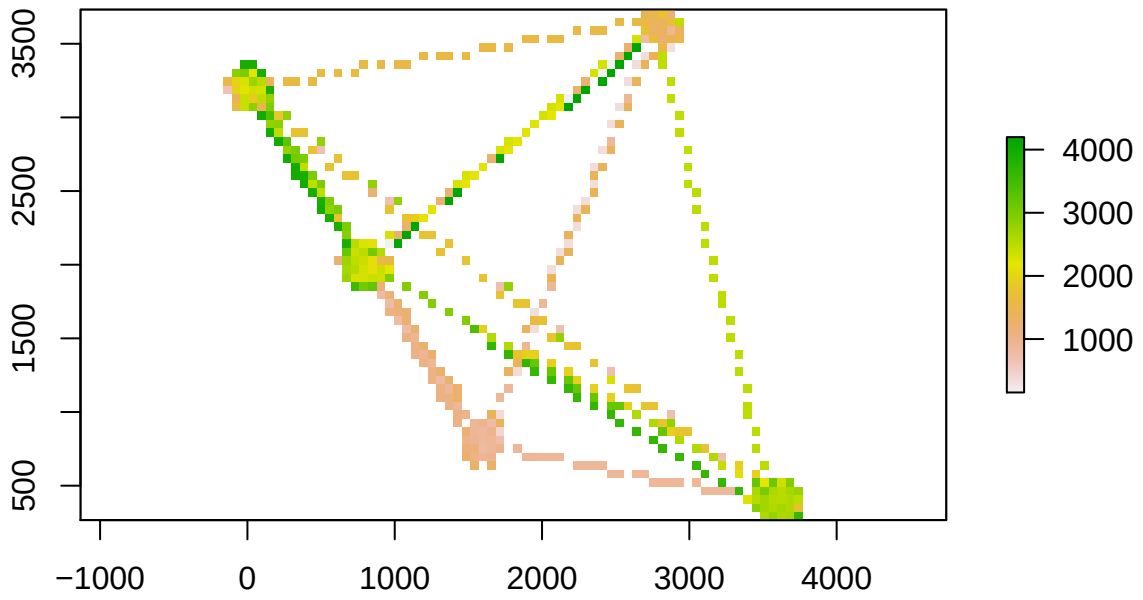

```r
# Using patches movement and median distance travel
adj_patches<-traj2adj(patches, res=quant(patches, p=0.5)) #Grid size based on median
dim(adj_patches[[1]]) # Adjency matrix
```

```
## [1] 4080 4080
```

```r
plot(adj_patches[[2]]) #Plot grid used
```



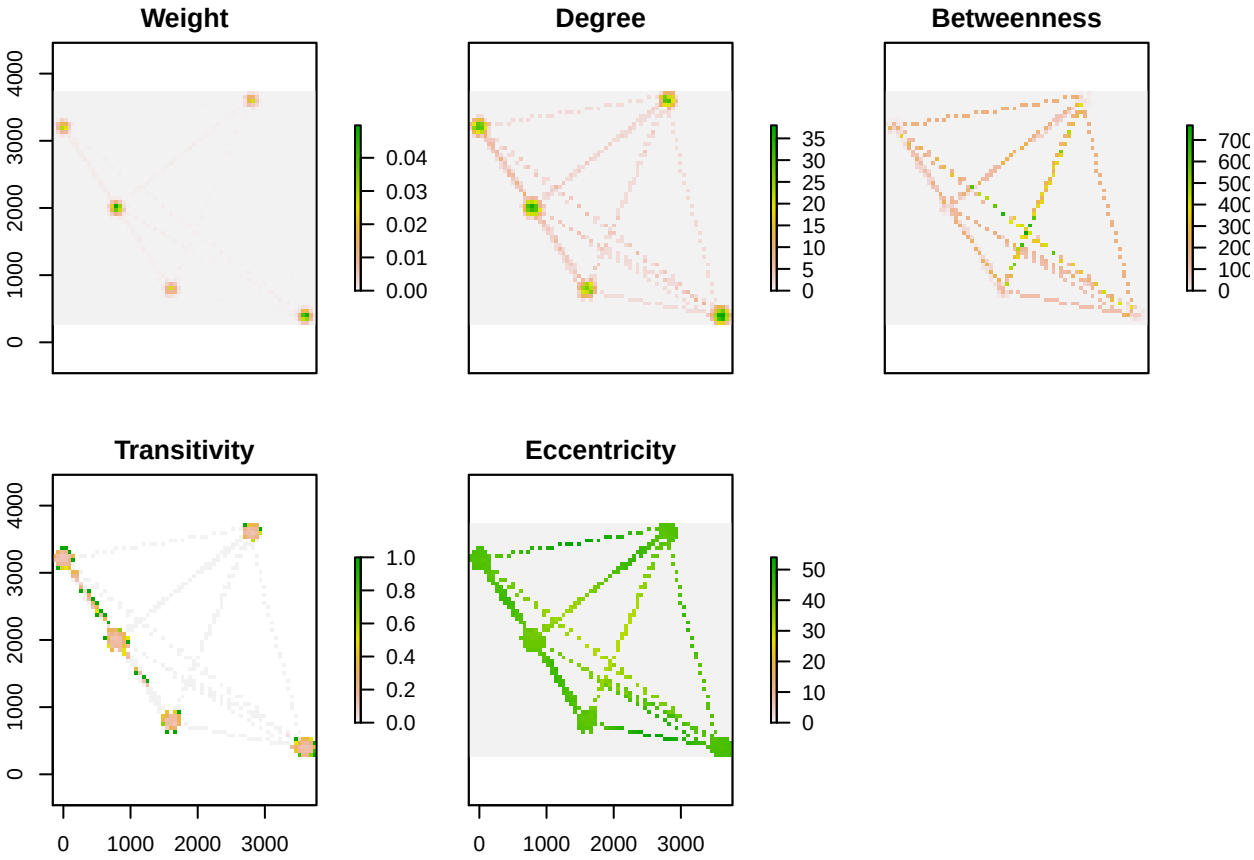

```
plot(adj_patches[[3]]) #Plot occupied pixels
```



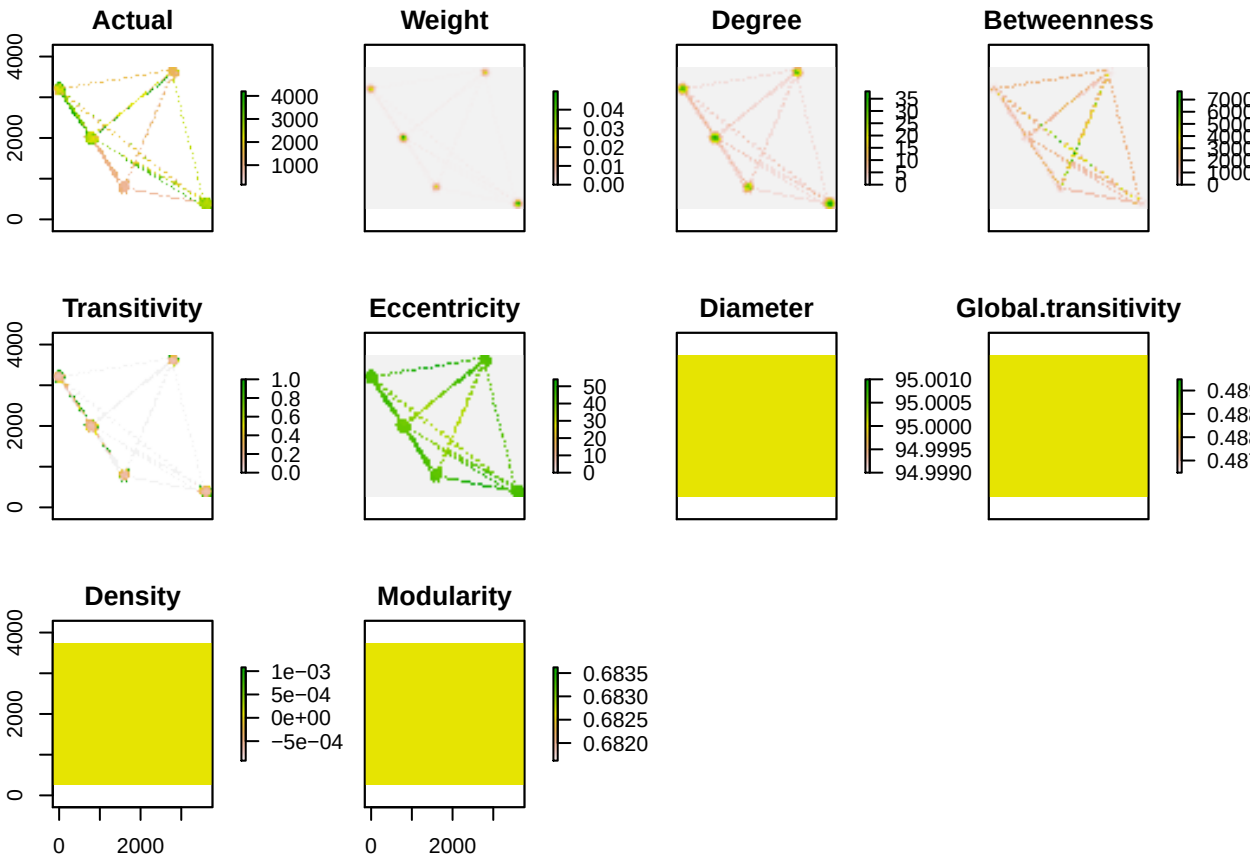
Calculation of network metrics - *adj2stack*

The function *adj2stack* takes the output of function *traj2adj* and calculate a series of node and graph-level metrics. Each metric is stored as a individual raster and the output is a raster stack combining each metric. Graph-level metric are also stored as a raster, containing an unique value. The function *graphmet* extracts graph level metrics. The function *val* extracts only the occupied cell (remove NA) in a raster and allow the calculation of statistics from node-level metrics.

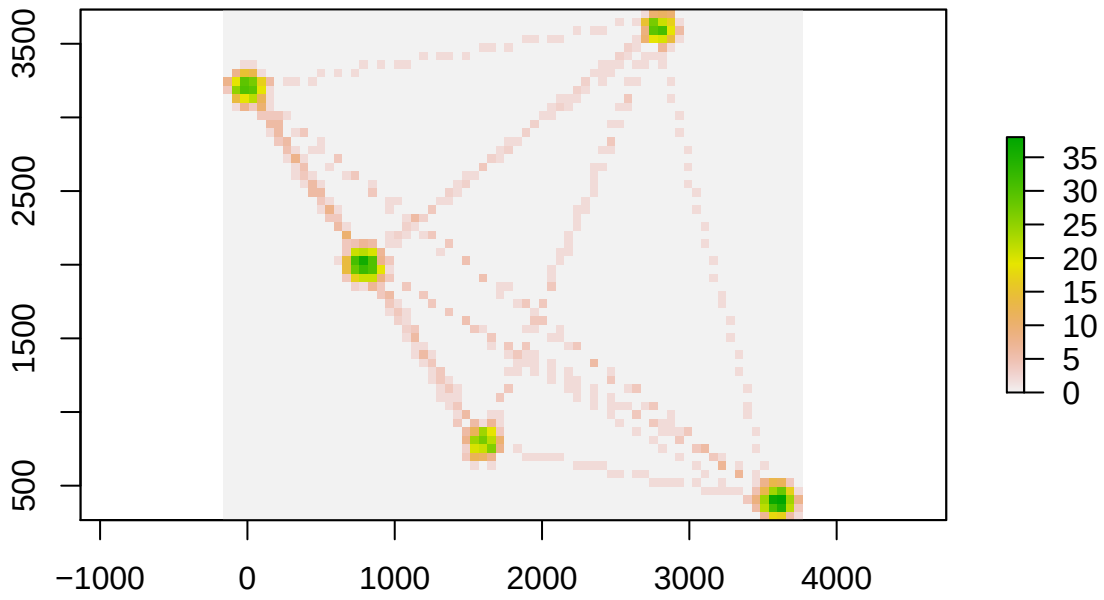
```
# Using patches movement and median distance travel
stck<-adj2stack(adj_patches,grph=T) #Plot the node-level metrics at the same time
```




```
plot(stck) #Plot also the graph-level metrics (not really useful)
```




```
plot(stck[[3]]) #Plot only one metric (degree)
```




```
graphmet(stck) # Extract graph-level metrics
```

```
##            Diameter Global.transitivity             Density
##        9.500000e+01        4.882399e-01        8.165929e-05
##          Modularity
##        6.826224e-01
```

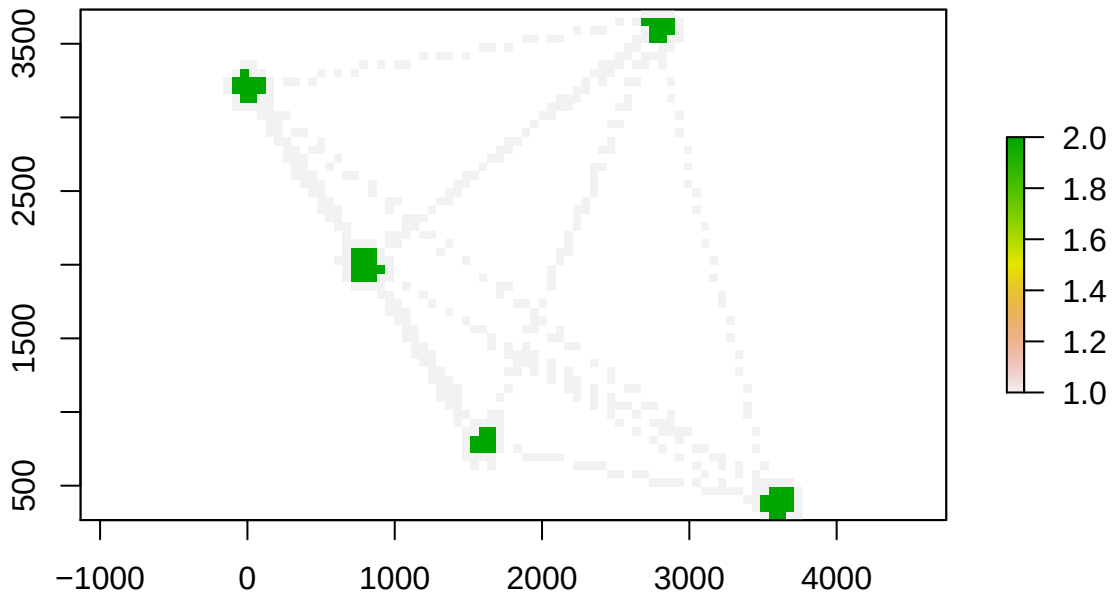
```
cv(val(stck, 4)) #Extract coefficient of variation of node-level betweenness.
```

```
## [1] 91.36045
```

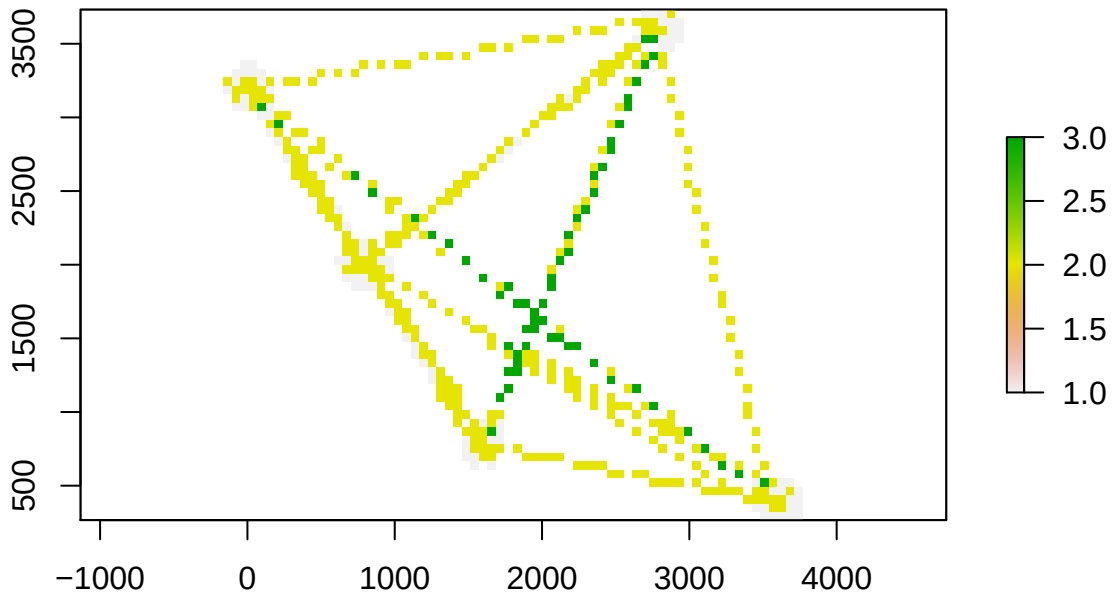
Clustering of node level metrics - *clustnet*

The function *clustnet* applies a normal mixture model to node-level metrics in order to cluster them into separate groups (default = 2). The function takes the output of function *adj2stack* with the user specifying the metric to cluster and the number of groups. Return a list containing output of function *Mclust* from package *mclust* and a raster displaying classification.

```
# Using patches movement and median distance travel
clust2<-clustnet(stck, id=3, nclust=2) # Clustering of degree in two groups
```



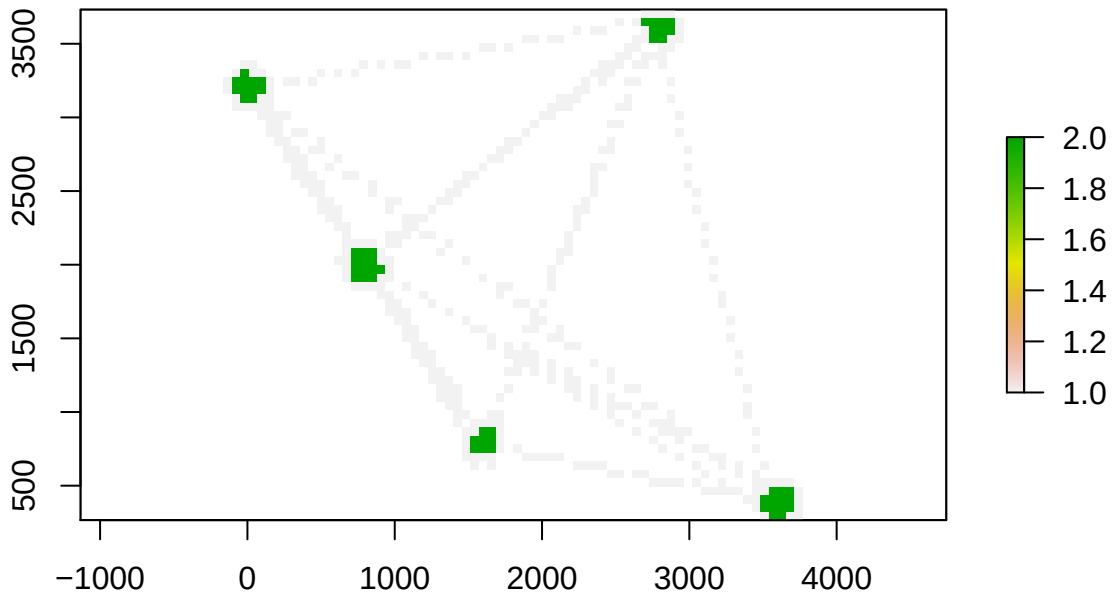

```
clust3<-clustnet(stck, id=4, nclust=3) #Clustering of betweenness in three groups
```




```
summary(clust2[[1]])
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust E (univariate, equal variance) model with 2 components:
##
##  log.likelihood   n df      BIC       ICL
##       -1371.635 477  4 -2767.94 -2771.508
##
## Clustering table:
##   1   2
## 423  54
```

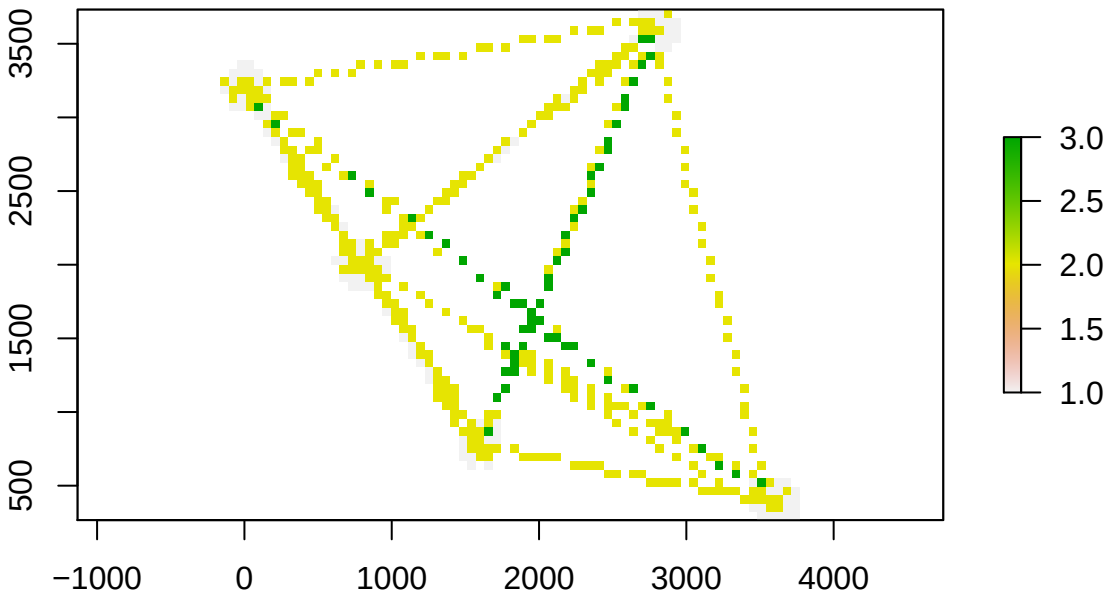
```
plot(clust2[[2]])
```




```
summary(clust3[[1]])
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust V (univariate, unequal variance) model with 3 components:
##
##  log.likelihood   n df       BIC       ICL
##       -5027.196 477  8 -10103.73 -10208.39
##
## Clustering table:
##   1   2   3
##  87 329  61
```

```
plot(clust3[[2]])
```


3500
2500
1500
500
−1000
0
1000
2000
3000
4000
3.0
2.5
2.0
1.5
1.0