Supplementary File 1

```
################################################################################
#
# name:    LISA.R
#
# purpose:  calculate LISA based on OLS night-lights at pixel level.
#
# paper:    Exploring regional and urban clusters and patterns in Europe
#           using satellite observed lighting
#
#################################################################################
# required libraries
library(raster)
library(beepr)

cpath       <- getwd()

# read one OLS raster saved locally.
ols         <- raster('FIL1992.tif')

# read NUTS I polygons saved locally
countries    <- shapefile("NUTSI_moll.shp")
country_code <- 'FR1'                    # select a specific NUTS I code
country      <- countries[countries$NUTS_ID == country_code,]
r_tmp        <- crop(ols, extent(country)) # to reduce masking time (next step)
r_tmp        <- mask(r_tmp, country)       # mask lights to polygon boundaries.

# convert to pixels to points and compute neighbor information
k_size       <- 3                          # specify Moran's I kernel size
p_tmp        <- rasterToPoints(r_tmp, fun=NULL, spatial=TRUE)
```

```r
OLS_points  <- knearneigh(p_tmp, k=k_size^2)
nb          <- knn2nb(OLS_points, row.names = NULL, sym = FALSE)
lw          <- nb2listw(nb, style="W", zero.policy=FALSE)

# initialize output
nn_out      <- matrix(nrow=OLS_points$np,ncol=8)
fn_out      <- matrix(nrow=OLS_points$np,ncol=3)

# compute Morans'I
# NOTE: FIL1992  is data specific
moran_out    <- localmoran(p_tmp$FIL1992, lw, zero.policy=FALSE)

# write output
nn_out[,1:5] <- moran_out[,1:5]
nn_out[,7:8] <- OLS_points$x

# if significance is low (p-value > 0.1) then set observed Moran's I to null due to low significance
nn_out[,1]   <- ifelse(nn_out[,5] >= 0.1, NA, nn_out[,1])

# create points with Moran's I observation
fn_out[,1]   <- nn_out[,7]  # x-coord
fn_out[,2]   <- nn_out[,8]  # y-coord
fn_out[,3]   <- nn_out[,1]  # I observed
colnames(fn_out) <- c("x","y", "I-observed")

# convert points back to raster with Moran's  I values
r_I   <- rasterFromXYZ(fn_out)

# compute the average Moran's I in the kernel.
f    <- matrix(1,k_size,k_size)
f[((k_size - 1)/2) + 1,((k_size - 1)/2) + 1] <- 0  # focal center does not count in average
```

```
r_mo  <- focal(r_I, w=f, fun=mean)


# normalize Moran's I raster based on min/max values
min_I <- cellStats(r_I, "min")
max_I <- cellStats(r_I, "max")
r_In  <- (2*(r_I-min_I)/(max_I - min_I)) - 1


# normalize the average Moran's I kernel raster based on min/max values
min_M <- cellStats(r_mo, "min")
max_M <- cellStats(r_mo, "max")
r_mN  <- (2*(r_mo-min_M)/(max_M - min_M)) - 1


# make a plot
plot(values(r_In), values(r_mN), xlab = "Local Moran's I", ylab = "Average Local Moran's I", xaxs="i",
yaxs="i", xlim=c(-1, 1), ylim=c(-1, 1))
abline(h = 0:1, v = 0:1, col = "black", lty = 3)


# compare each Moran"s value to the average Moran's I value of the kernel and
# classify output values in four categories based on the quadrant (LISA).
r_In[r_In >  0] <- 100
r_In[r_In <= 0] <- 200


r_mN[r_mN >  0] <- 10
r_mN[r_mN <= 0] <- 20


r_lisa <- r_In  + r_mN
r_lisa[r_lisa ==  110] <- 1
r_lisa[r_lisa ==  120] <- 2
r_lisa[r_lisa ==  220] <- 3
r_lisa[r_lisa ==  210] <- 4
```

```
# write output LISA raster and data

setwd(cpath)

crs(r_lisa) <- "+proj=moll +lon_0=0 +x_0=0 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m
+no_defs"

writeRaster(r_lisa, filename='LISA', format='GTiff', datatype='INT1U', overwrite=TRUE)

# csv output format:

# column 1: Moran's I observed

# column 5: p-value

# column 7: x-coord

# column 8: y-coord

write.csv2(nn_out,'local_morans_nn.csv')

print(paste("Done...", Sys.time()))

beep()

#end
```