File S1: Code for re-projection and extraction of sub-datasets for remote sensing images in batches, and calculating the monthly maximum ground temperature.

The supporting information is organized in three sections. In the first section the code for re-project remote sensing images in batches is provided. In the second section the code for extracting sub-dataset of remote sensing images in batches is provided. And the code needs to run with MRT (MODIS Reprojection Tool) software. In the third section the code for extracting the maximum ground temperature per month from the daily land surface temperature of the MOD11A1 data is provided.

Section 1. The code for re-project remote sensing images in batches.

```
rem set the MRTDATADIR environmental var to the MRT data directory.
# define the path to MRTDATADIR
set MRTDATADIR=D:\MRT\bin\data
# re-projection using saved parameter file
for %%i in (*.hdf) do D:\MRT\bin\resample -p my.prm -i %%i -o %%iout.tif
# keep bat run window for error reminder
pause
```

Section 2. The code for extraction of sub-datasets for remote sensing images in batches.

```
# define the path to MRTDATADIR
```

```
set MRTDATADIR=c:\Modis\data
# put the file name of all data within a month in a TXT file
set /a DAY=start data
set /a DEADLINE=end data
:start
if %DAY% leq %DEADLINE% (goto ORDER) else exit
:ORDER
dir *%DAY%.*.hdf/a/b/s > MOSAICINPUT.TXT
# extract the LST_Day_1km band sub-dataset from HDF dataset
c:/Modis/bin/mrtmosaic.exe -i MOSAICINPUT.TXT -s "1 0 0 0 0 0 0 0 0 0 0 0" -o
MOSAIC_TMP_%DAY%.hdf
# copy the extracted data to the built R folder and delete the original daily data
copy MOSAIC_TMP_%DAY%.hdf    R& del MOSAIC_TMP_%DAY%.hdf
del *%DAY%.*.hdf
# extract the data of the next phase
set /a DAY= %DAY% + 1
goto start
```

Section 3. The code for calculating the maximum ground temperature per month (the example for August 2009).

```
# install the necessary package
install.packages("sp")
```

```
install.packages("raster")
install.packages("rgdal")
# load the necessary package
library(sp)
library(raster)
# path for the daily land surface temperature of the MOD11A1 data
path<-"E://2009/8"
# extracting the maximum ground temperature per month
fileNames<-dir(path)
filePath<-sapply(fileNames,function(x) {paste(path,x,sep="/")})
b<-array(1:43650,dim=c(194,225, length(filePath)))
for(n in 1: length(filePath)) {
R<-raster(filePath[n])
a <- as(extent(338000,546000,3750000,3930000), 'SpatialPolygons')
crs(a) <- crs(R)
A <- crop(R,a)
for (i in 1:nrow(b)){
   for (j in 1:ncol(b)) {
                  t<-getValuesBlock(A,row=i,nrow=1,col=j,ncol=1)
                  T<-0.02*t-273.15
                  b[i,j,n]<-T
      }
```

```
}
}
B<-matrix(1:43650,nrow=194,ncol=225)
for (I in 1:nrow(B)){
   for (J in 1:ncol(B)) {
B[I,J]<-max(b[I,J,])
}
}
# outputting monthly maximum ground temperature data in CSV format
write.csv(B,file = "E://R/200908.csv")
# outputting monthly maximum ground temperature data in TIF format
re<-raster(xmn= 337630.3,xmx= 546121,ymn= 3749957,ymx=
3929722,ncols=225,nrows=194)
for(N in 1: nrow(re)) {
            for(M in 1:ncol(re)) {
                        re[N,M]<-B[N,M]
                  if(re[N,M]==-273.15) {
                                  re[N,M]<-NA
}
}
}
writeRaster(re,filename="E://R/200908.tif",overwrite=TRUE)
```

```
# outputting monthly maximum ground temperature data in JPEG format
cols <- rev(heat.colors(50))
setwd("E://R")
jpeg(file="200908.jpeg")
plot(re,col=cols)
dev.off()
```

File S2: R code for calculating normalized spectral entropy of the study region

The supporting information is organized in two sections. In the first section the code for the procedure “sim.Hsn” is provided. In the second section the code for calculating the Normalized Spectral Entropy (Hs_n) based on the monthly maximum ground temperature and the predefined “sim.hsn” function is provided.

Section 1. The “sim.hsn” procedure

```
# load the necessary package
library(TSA)
# define “sim.hsn” function
sim.hsn <-function(tseries, mw=3, bootreplicates=9999, demean=TRUE,
detrend=TRUE,conf = 0.95, plot.hist = TRUE)
    {
# check the package dependence and parameters
    check = suppressPackageStartupMessages(require(TSA))
    if (check != TRUE) stop("Please install the package TSA from CRAN")
    mw = as.integer(mw)
    if ((mw <=0) ||(mw > 6)) stop("Please check the mw parameter, wrong data type or
value. Do not use windows bigger then 6.")
    bootreplicates= as.integer(bootreplicates)
    if (bootreplicates <= 0) stop("Please check the number of replicates.")
```

```
    if (demean %in% c(TRUE,FALSE)==FALSE) stop("Please check the value of demean.Must be TRUE or FALSE")
    if (detrend %in% c(TRUE,FALSE)==FALSE) stop("Please check the value of detrend.Must be TRUE or FALSE")
    conf = as.numeric(conf)
    if ((conf <=0) || (conf >= 1)) stop("Please check the conf parameter, wrong data type or value. Must be within zero and one.")
    if (plot.hist %in% c(TRUE,FALSE)==FALSE) stop("Please check the value of plot.hist. Must be TRUE or FALSE")
# calculate the power spectrum for the different frequencies
    periodog = spec(tseries, kernel=kernel('daniell', m=(mw-1)), log='no', plot=FALSE, demean=demean, detrend=detrend)
    periodog = periodog$spec
    kermw = seq(from = -mw, to = mw, by = 1)
    lengthts = length(periodog)
# build main resampling matrix
    MC = array(sample(kermw, (bootreplicates * (lengthts - 2* mw-2)),prob=rep(1, (mw*2 +1)), replace=TRUE), dim=c(bootreplicates,(lengthts - 2*mw-2)))
    for (i in 1:mw) {
        if (i == 1) Ms = sample(seq(from = 0, to = mw, by=1), bootreplicates,prob=rep(1, (mw + i)), replace=TRUE)
```

```
      Ms = cbind(Ms, sample(seq(from = (-i), to = mw, by=1), bootreplicates,prob=rep(1, (mw +i+1)), replace=TRUE))
   }
   for (i in 1:mw) {
      if (i == 1) Md = sample(seq(from = -mw, to = mw, by=1), bootreplicates,prob=rep(1, (2*mw + 1)), replace=TRUE)
      Md = cbind(Md, sample(seq(from = -mw, to = (mw-i), by=1), bootreplicates,prob=rep(1, (2*mw +1 -i)), replace=TRUE))
   }
   MC = cbind(Ms, MC, Md)
   MC = MC + t(array(rep(seq(1:lengthts), bootreplicates), dim=c(lengthts, bootreplicates)))
# calculate the value of Hsn for the provided time series
   N = length(periodog)
   periodog[periodog == 0] <- 1e-07
   periodogn = periodog/sum(periodog)
   Hsn = -sum(periodogn * log(periodogn)) / log(N)
   for (i in 1:bootreplicates){
      nperiodog = periodog[MC[i, ]]
      nperiodog[nperiodog == 0] <- 1e-07
      nperiodogn = nperiodog/sum(nperiodog)
      Hsn = rbind(Hsn, -sum(nperiodogn * log(nperiodogn)) / log(N))
```

```
    }
    Hsn = as.vector(Hsn)
# output the results
    cum.distr = ecdf(Hsn)
    pvalue= cum.distr(Hsn[1])
    cat("\n The value of Hsn is: ")
    cat(as.numeric(Hsn[1]), "\n")
    cat("\n The p-value is: ")
    cat(pvalue, "\n")
    th0 = Hsn[1]
    th = Hsn
    n <- length(periodog) #observations in rows
    N <- 1:n
    alpha <- (1 + c(-conf, conf))/2
    zalpha <- qnorm(alpha)
    z0 <- qnorm(sum(th < th0) / length(th))
    th.jack <- numeric(n)
    for (i in 1:n) {
        serie2 = periodog[-i]
        serie2[serie2 == 0] <- 1e-07
        serien = serie2/sum(serie2)
        th.jack[i] = -sum(serien * log(serien)) / log(n-1)
```

```
  }
  L <- mean(th.jack) - th.jack
  a <- sum(L^3)/(6 * sum(L^2)^1.5)
  adj.alpha <- pnorm(z0 + (z0+zalpha)/(1-a*(z0+zalpha)))
  limits <- quantile(th, adj.alpha, type=6)
  Ris = list("est"=th0, pvalue = pvalue, "BCa"=limits, bootstrap = Hsn)
  class(Ris) = "Hsn"
  cat("\n The BCa confidence limits are: ")
  cat(as.numeric(limits), "\n")
  cat("\n The p-values are: ")
  cat(adj.alpha, "\n")
# plot results
  if (plot.hist == TRUE) {
    hist(Hsn, xlab="Hsn", main = "Bootstrap distribution of Hsn")
    abline(v=limits,col="red")
    abline(v=Hsn[1], col="blue")
  }
  return(Ris)}
```

Section 2. Calculating the Normalized Spectral Entropy (Hs_n) of the study region.

```
# input the maximum ground temperature data per month
path<- "E://CSV"
```

```
fileNames <-list.files(path,full.names = TRUE)
b<-array(1:43650,dim=c(194,225, length(fileNames)))
R<-matrix(1:43650,nrow=194,ncol=225)
for(n in 1: length(fileNames)) {
D<-read.csv(fileNames[n])
for (i in 1:194){
   for (j in 1:225) {
                        b[i,j,n]<-D[i,j+1]
            }
                        }
}
# call "sim.hsn" function
hsn<-array(1:43650,dim=c(194,225))
date<-c(1: length(fileNames))
for (I in 1:nrow(b)){
   for (J in 1:ncol(b)) {
            for(n in 1:length(fileNames)) {
                        date[n]<-b[I,J,n]
                        }
pippo = sim.hsn(date, bootreplicates=5000)
hsn[I,J]<-pippo$est
            }
```

```
}
# output the resulted Hsn and Hsn map
write.csv(hsn,"E://Hsn.csv")
re<-raster(xmn= 337630.3,xmx= 546121,ymn= 3749957,ymx=
3929722,ncols=225,nrows=194)
for(N in 1:nrow(re)) {
        for(M in 1:ncol(re)) {
                re[N,M]<-hsn[N,M]
                if(re[N,M]==1) {
                        re[N,M]<-NA
}
}
}
writeRaster(re,filename="E://Hsn.tif",overwrite=TRUE)
cols <- rev(heat.colors(50))
setwd("E:// ")
jpeg(file="Hsn.jpeg")
plot(re,col=cols)
dev.off()
# output the resulted Hsn and Hsn map for the study region
R<-raster("E://Hsn.tif")
a <- as(extent(338000,546000,3750000,3930000), 'SpatialPolygons')
```

```
crs(a) <- crs(R)
A <- crop(R,a)
x<- nrow(A)
y<- ncol(A)
for (i in 1:x){
    for (j in 1:y) {
                              t<-getValuesBlock(A,row=i,nrow=1,col=j,ncol=1)
                              b[i,j]<-t
                              }
                              }
write.csv(b,file = "E://Hsn2.csv")
re<-raster(xmn= 337630.3,xmx= 546121,ymn= 3749957,ymx=
3929722,ncols=y,nrows=x)
for(N in 1:nrow(re)) {
               for(M in 1:ncol(re)) {
                              re[N,M]<-hsn[N,M]
}
}
}
writeRaster(re,filename="E://Hsn2.tif",overwrite=TRUE)
cols <- rev(heat.colors(50))
setwd("E:// ")
```

```
jpeg(file="Hsn2.jpeg")
plot(re,col=cols)
dev.off().
```

Table S1. ANOVA of Hs_n of different roads and the area outside the road buffer with respect to land cover and elevation.

	Land covers	Source of difference	Sum of Squares	df	Mean Square	F	P-value	F crit
The Hs_n of road buffer and outside the buffer	**High coverage grassland**	Between Groups	0.068	1	0.068	5.724	0.017	3.846
		Within Groups	25.222	2132	0.012			
		Total	25.290	2133				
	Medium coverage grassland	Between Groups	0.200	1	0.200	23.144	<0.001	3.842
		Within Groups	137.264	15876	0.009			
		Total	137.464	15877				
	Low coverage grassland	Between Groups	0.090	1	0.090	9.622	0.002	3.842
		Within Groups	139.736	14895	0.009			
		Total	139.826	14896				
	Shrubbery	Between Groups	0.000	1	0.000	0.047	0.829	3.861
		Within Groups	2.477	484	0.005			
		Total	2.478	485				
	Unused land	Between Groups	0.004	1	0.004	0.213	0.644	3.844
		Within Groups	64.734	3178	0.020			
		Total	64.739	3179				
The Hs_n of different roads buffer and outside the buffer	**High coverage grassland**	Between Groups	0.193	3	0.064	5.457	0.001	2.609
		Within Groups	25.097	2130	0.012			
		Total	25.290	2133				
	Medium coverage grassland	Between Groups	1.189	4	0.297	34.626	<0.001	2.372
		Within Groups	136.275	15873	0.009			
		Total	137.464	15877				
	Low coverage grassland	Between Groups	1.388	4	0.347	37.330	<0.001	2.373
		Within Groups	138.438	14892	0.009			

		Total	139.826	14896				
	Shrubbery	Between Groups	0.025	4	0.006	1.209	0.306	2.390
		Within Groups	2.453	481	0.005			
		Total	2.478	485				
	Unused land	Between Groups	0.483	4	0.121	5.969	<0.001	2.375
		Within Groups	64.286	3179	0.020			
		Total	64.768	3183				
The Hs_n of road buffer and outside the buffer	**3700-3900m**	Between Groups	0.000	1	0.000	1.766	0.187	3.924
		Within Groups	0.025	114	0.000			
		Total	0.025	115				
	3900-4100m	Between Groups	0.013	1	0.013	0.768	0.381	3.851
		Within Groups	17.053	994	0.017			
		Total	17.066	995				
	4100-4300m	Between Groups	0.064	1	0.064	9.367	0.002	3.842
		Within Groups	86.726	12704	0.007			
		Total	86.790	12705				
	4300-4500m	Between Groups	0.043	1	0.043	8.004	0.005	3.842
		Within Groups	90.820	16742	0.005			
		Total	90.864	16743				
	4500-4700m	Between Groups	0.221	1	0.221	17.510	<0.001	3.842
		Within Groups	134.824	10690	0.013			
		Total	135.045	10691				
	4700-4900m	Between Groups	0.017	1	0.017	0.801	0.371	3.844
		Within Groups	91.671	4200	0.022			
		Total	91.688	4201				

	4900-5100m	Between Groups	0.092	1	0.092	3.199	0.074	3.854
		Within Groups	21.789	758	0.029			
		Total	21.881	759				
The *Hs*$_n$ of different roads buffer and outside the buffer	**3700-3900m**	Between Groups	0.001	3	0.000	0.880	0.454	2.686
		Within Groups	0.025	112	0.000			
		Total	0.025	115				
	3900-4100m	Between Groups	0.103	4	0.026	1.499	0.200	2.381
		Within Groups	16.964	991	0.017			
		Total	17.066	995				
	4100-4300m	Between Groups	1.517	4	0.379	56.474	<0.001	2.373
		Within Groups	85.273	12701	0.007			
		Total	86.790	12705				
	4300-4500m	Between Groups	0.261	4	0.065	12.033	<0.001	2.372
		Within Groups	90.603	16739	0.005			
		Total	90.864	16743				
	4500-4700m	Between Groups	0.333	4	0.083	6.608	<0.001	2.606
		Within Groups	134.712	10687	0.013			
		Total	135.045	10691				
	4700-4900m	Between Groups	0.294	3	0.098	4.501	0.004	2.607
		Within Groups	91.394	4198	0.022			
		Total	91.688	4201				
	4900-5100m	Between Groups	0.178	2	0.089	3.106	0.045	3.008
		Within Groups	21.702	757	0.029			
		Total	21.881	759				

Difference is significant at the 0.05 level.