SHORT TERM SCIENTIFIC MISSION (STSM) – SCIENTIFIC REPORT

The STSM applicant submits this report for approval to the STSM coordinator

Action number: ES1309
STSM title: The FFP Footprint online tool: automatization and enhancement of S2 landcover classification
STSM start and end date: 27/02/2018 to 18/03/2018
Grantee name: Enrico Tomelleri

PURPOSE OF THE STSM/

Remote and proximal sensing data (satellite, UAV or ground sensors) are increasingly being used for improved interpretation of eddy covariance (EC) flux measurements. The integration of such approaches gained popularity because they are capable of providing ancillary information for ecosystem functioning understanding and offer possibilities for upscaling ecosystem fluxes. The key challenge in this context is the mismatch between the flux footprint of EC measurements, i.e. the area from which the measured flux originates, and the minimum measured unit of area by remote sensing (pixel). The footprint of EC measurements is typically a few hundreds of meters and varies in time due to changing meteorological conditions, while the proximal/remote sensing pixel depends largely on the selected platform and sensors and is often constant in time. In addition to the spatial allocation issue, the two sets of measurements often represent different temporal scales.

At a COST OPTIMISE workshop held in Innsbruck from 13.02.17 to 16.02.17, the FFP online tool system was initiated. Currently, a user of can provide site information (coordinates) and upload meteorological and turbulence data. Then an orthophoto is automatically downloaded from Bing - if available - or Google maps services and a kmeans classification is applied to derive a simple land cover classification. The number of classes is fixed to five. The user receives statistics about the contribution per land cover classes to the measured flux, based on the FFP footprint model. Detailed information on the tool is available at a dedicated website[1]. The project status and suggested further developments are represented in Figure 1.

[1] http://footprint.kljun.net/ffp2d.html

Funded by the Horizon 2020 Framework Programme of the European Union

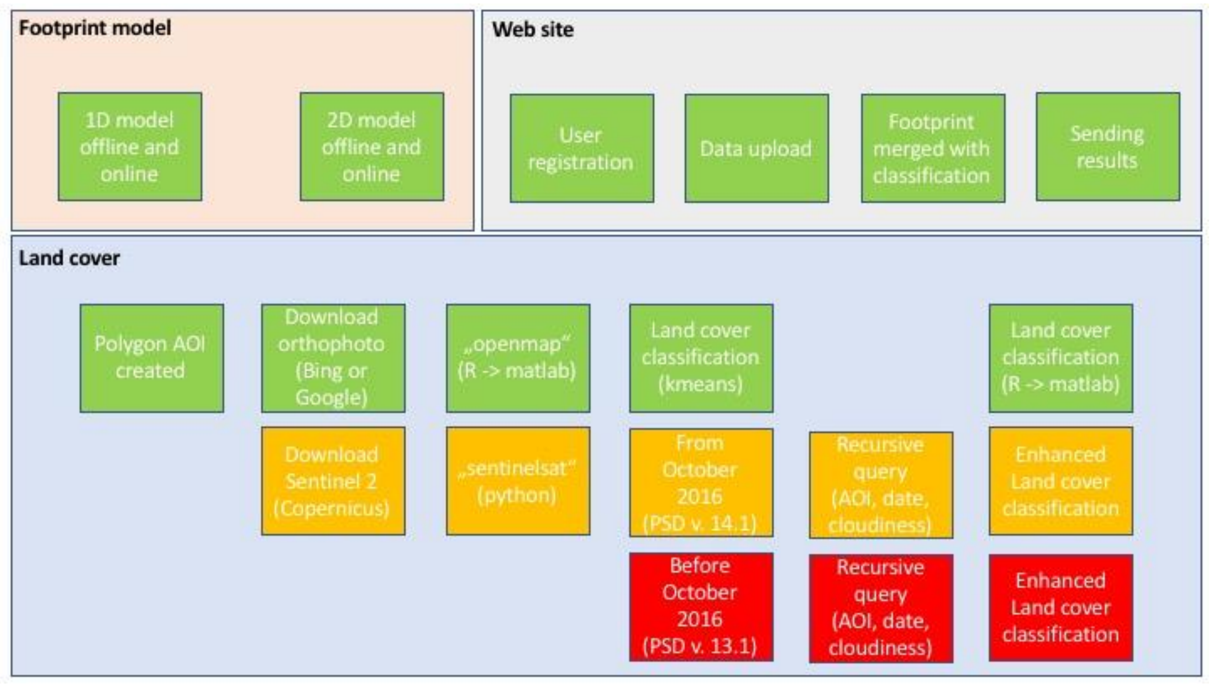
Footprint model
1D model offline and online
2D model offline and online
Web site
User registration
Data upload
Footprint merged with classification
Sending results
Land cover
Polygon AOI created
Download orthophoto (Bing or Google)
„openmap" (R -> matlab)
Land cover classification (kmeans)
Land cover classification (R -> matlab)
Download Sentinel 2 (Copernicus)
„sentinelsat" (python)
From October 2016 (PSD v. 14.1)
Recursive query (AOI, date, cloudiness)
Enhanced Land cover classification
Before October 2016 (PSD v. 13.1)
Recursive query (AOI, date, cloudiness)
Enhanced Land cover classification

Figure 1: project modules solved (green) to be addressed in this STMS (orange) and further developments (red). Download approaches need to be distinguished for different versions of "Sentinel-2 Products Specification Document" (PSD).

During this STSM the work initiated during the Innsbruck workshop got consolidated. Methods for automatically downloading Sentinel 2 scenes were tested and advanced classification methods for multispectral satellite imagery were implemented and compared at 3 test sites. Finally, an experiment for investigating the effects of spatial heterogeneity on artificial data was designed.

DESCRIPTION OF WORK CARRIED OUT DURING THE STSMS

WP1: automatization of S2 download

Sentinel-2 images are stored on the Open Access Data Hub and accessible to the public free of charge. Copernicus provides a GUI access to the dataset which, however, is not suitable for automated download. For this reason, a number of tools are available (e.g. Sentinelsat, AWSdownload, Sentinel-download, PEPS-download). Each of these tools provides different access modalities. I've checked the possibility to use the "S2 value adder"[2] hosted at Universität für Bodenkultur Wien. The bulk data access is possible through a R-package and processing on request is possible. This is suitable for our needs, but it has a cost. I found sentinelsat[3] to be the most suitable for the online FPP tool aims. Sentinelsat is a python API and command line interface for searching, downloading and retrieving the metadata of Sentinel satellite images from the Copernicus Open Access Hub. Sentinelsat is a community-based project available on GitHub with a GPLv3+ license. Sentinelsat - as other tools - allows querying dates. I found it more advantageous than other tools because it permits to perform also queries based on geographic information, cloudiness and level of data processing (i.e., ToA or BoA reflectance). In this work

[2] https://ivfl-arc.boku.ac.at/eodc/
[3] https://github.com/sentinelsat/sentinelsat

package, sentinelsat was integrated in the existing conceptual workflow. The newly developed approach allows for downloading Sentinel-2 images corresponding to the area of interest and to the very specific time frame of the user provided meteorological data. Compared to the previous approach this is a step forward because it allows for taking into account the closest in time available imagery with regards to the uploaded time stamp.

Considering that on 27 September 2016, the products distributed on the Open Access Data Hub begun to embed one single tile of the tiling grid (PSD version 14.1) opposed to a set of tiles as it was before (PSD version 13.1), I focused on the most recent data format (Fig. 1). Furthermore, Sentinel-2 Level-2A (BoA) pilot products are available for data acquired over Europe since 2 May 2017. If data are available, they are preferred. If not Level-1C (ToA) data are used.

WP2: enhancement of classification method

During the Innsbruck workshop we made use of an unsupervised kmeans classification coded in Matlab. The limitation of such an approach is the difficulty in the automatization of class number selection. For this reason, we need to implement a solution for automatic selection of this parameter. From a method review, I identified different potential approaches for classification with automated class number selection:

1) **kmeans**: standard as we did in the orthophoto classification, but the number of cluster is now selected by means of the elbow method (within cluster sums of squares).

2) **kmeans + random forests**: subsetting the image, kmeans classification and upscaling by random forest. The number of cluster is now selected by means of the elbow method.

3) **clara:** Clustering for Large Applications (makes use of the PAM algorithm, numbers of clusters selected by the average silhouette width criterion)

4) **xmeans**[4]: an extension of kmeans with automatic estimation of clusters by optimizing the Bayesian Information Criterion (BIC) or the Akaike Information Criterion (AIC).

5) **cmeans**[5]: the fuzzy version of kmeans clustering algorithm

In this WP, I tested the suitability of these approaches in order propose a solution to be implemented in the online footprint tool.

WP3: synthetic experiment

In this WG, I looked into synthetic data, to understand the effects of different spatial variability patterns and spatial resolution. For this purpose, I multiple images with different types of spatial structure (gradient, hot-spots, random patches). These will be combined with different footprint types (e.g. stable and instable conditions). This should help to list guidelines on optical sampling strategies based on ecosystem heterogeneity.

[4] D. Pelleg and A. W. Moore (2006). X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In: *Seventeenth International Conference on Machine Learning*, 727–734.
[5] Nikhil R. Pal, James C. Bezdek, and Richard J. Hathaway (1996). Sequential competitive learning and the fuzzy c-means clustering algorithms.Neural Networks, 9(5), 787--796.

DESCRIPTION OF THE MAIN RESULTS OBTAINED

WP1: automatization of S2 download

```
#-------------------------------------------------------------------------

# NAME: download_sentinel2.R

# PROJECT: OPTIMISE

# PURPOSE: download S2 image matching the time stamp of meteo data

# CATEGORY: R-script

# CALLING SEQUENCE: write.geojson(46.683575, 10.585611, 0.001)

downloadS2(46.683575, 10.585611, 20170601, 20170730, "your_id", "your_pwd")

# INPUTS: coordinates, start_day, end_day

# OUTPUTS: S2 images (all bands)

# NOTES: -

# CREATION DATE: 18-03-2018

# AUTHOR: Enrico Tomelleri, enrico.tomelleri@posteo.de

# UPDATE:

#------------------------------------------------------------------------


home of the sentinelsat tool:

#http://sentinelsat.readthedocs.io/en/stable/index.html


#load libraries

library(lubridate)

library(rgdal)

library(mgrs)


#write geojson file

write.geojson <- function(lat,lon,d)#call example write.geojson(60,17,0.001)

{

  sink("outfile.geojson")

  cat("{

\"type\": \"FeatureCollection\",

\"features\": [

{

\"type\": \"Feature\",
\"properties\": {},
\"geometry\": {
\"type\": \"Polygon\",
\"coordinates\": [
  [
  [")
  cat(
    lon-d,
    ",",
    lat-d
  )
  cat("],
      [")
  cat(
    lon-d,
    ",",
    lat+d
  )
  cat("],
      [")
  cat(
    lon+d,
    ",",
    lat+d
  )
  cat("],
      [")
  cat(
    lon+d,
    ",",
    lat-d
  )
  cat("],
      [")
  cat(
    lon-d,
    ",",
    lat-d
  )
  cat("]
]
]
}
}
]
}")
  sink()
}



#the following function makes use of sentinelsat
#https://github.com/sentinelsat/sentinelsat

#call example downloadS2(20160601, 20160830, "your_id", "your_pwd")
downloadS2 <- function(lat, lon, date1, date2, id, pwd) #date format yyyymmdd
{

  #k <- 0 #switch for processing level
  for (k in 0:1)
  {
    date1_p <- ymd(date1)
    date2_p <- ymd(date2)

    daystart <- as.numeric(format(as.Date(date1_p), '%Y%m%d'))
    #print("daystart=",daystart)
    dayend <- as.numeric(format(as.Date(date1_p+3), '%Y%m%d'))
    #print("dayend=",dayend)
    cloudiness <- 5

    j <- 1 #switch for cloudiness

    for (i in 1:100)
    {
      #sentinel -s 20160501 -e 20160530 -f -c 40 --sentinel 2 "id" "pwd" outfile.geojson
      print(daystart)


      #select sensor
      if (k == 0){proc_lev <- "*2A*"}
      if (k == 1){proc_lev <- "*1C*"}
      #select tile
      tile_id <- latlng_to_mgrs(lat, lon, degrees = TRUE, precision = 0)



      command <- paste("sentinelsat -s", daystart, "-e", dayend, " -d -c", cloudiness,
"--sentinel 2 -u et_eurac -p Sentinel2 -g outfile.geojson --query 'filename=*",tile_id ,
"*, producttype=*", proc_lev, "*'")
      system(command)


      #download the first image with less than 5% clouds ... recursive 10-20-30%
      file_in <- list.files(path = ".", pattern = "zip")
      if (length(file_in)==1){break}
      daystart <- as.numeric(format(as.Date(date1_p)+3*j+1, '%Y%m%d'))
      dayend <- as.numeric(format(as.Date(date1_p)+3*j+3, '%Y%m%d'))
      if (dayend >= date2)
      {
        daystart <- as.numeric(format(as.Date(date1_p), '%Y%m%d'))
        #print("daystart=",daystart)
        dayend <- as.numeric(format(as.Date(date1_p+3), '%Y%m%d'))
        #print("dayend=",dayend)
        cloudiness <- cloudiness+5
        print("increase cloudiness!")
        j <- 1
      }
      j <- j+1
    }


  }
}
#------------------------------------------------------------------------
```

WP2: enhancement of classification method

By means of comparing the results of the land cover classification at the site ES-LMa with a land cover classification from a Compact Airborne Spectrographic Imager (CASI) acquisition we can state that in this specific case the classification with the different methods (Figure 2) is consistent with an independent approach (Figure 3). Therefore, in the operational implementation of the FPP tool any of these methods can be used. Consistently, also in other test sites (AT-Neu, SE-Nor) we found that different land cover classification approaches give similar results. Furthermore, the use of 10 meters or 20 meters S2 bands is a trade-off between spatial and spectral information. Given that the script from WP1 will download all the S2 bands, the operational updated online tool will provide results based on the land cover classification with both the options (10m and 20m).

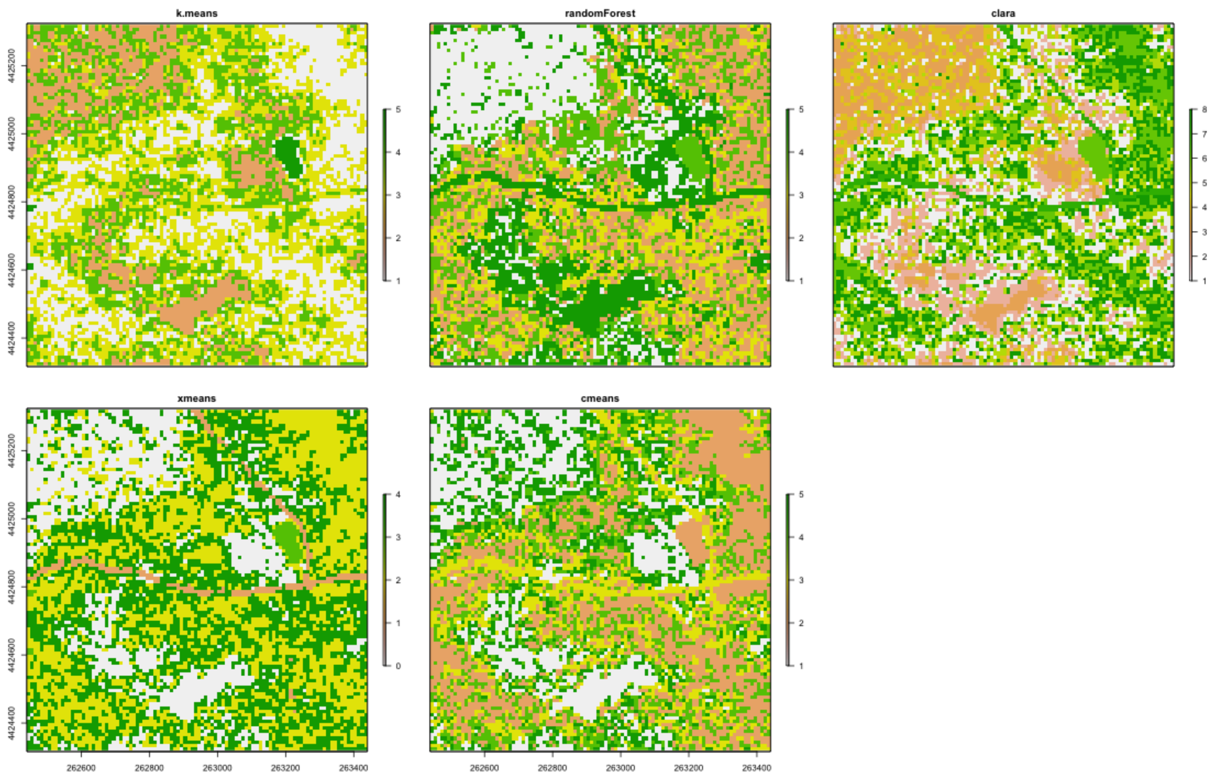

Figure 2: land cover classification with the 5 methods selected of the 10m S2 bands at the study site ES-LMa (Acquisition 02/04/2017).

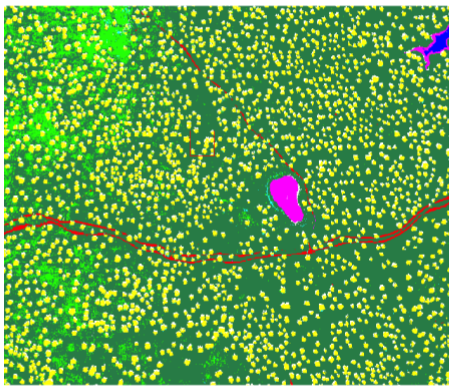

Figure 3: land cover classification of the CASI acquisition at the study site ES-LMa (19/05/2017).

Script for S2 classification:

```
#-------------------------------------------------------------------------

# NAME: classify_sentinel2.R

# PROJECT: OPTIMISE

# PURPOSE: classify a S2 scene subset

# CATEGORY: R-script

# CALLING SEQUENCE: source(“classify_sentinel2.R”)

# INPUTS: -

# OUTPUTS: plots with classifications

# NOTES: -

# CREATION DATE: 18-03-2018

# AUTHOR: Enrico Tomelleri, enrico.tomelleri@posteo.de

# UPDATE:

#------------------------------------------------------------------------


library(raster)

library(RStoolbox)

library(rgdal)

library(cluster)

library(GMD)

library(randomForest)

library(fpc)

#other libraries are loaded in the specific sections


#read in data


b2 <- raster("*_B02.jp2")

b3 <- raster("*_B03.jp2")

b4 <- raster("*_B04.jp2")

b5 <- raster("*_B05.jp2")

b6 <- raster("*_B06.jp2")

b7 <- raster("*_B07.jp2")


b8 <- raster("*_B08.jp2")

b8a <- raster("*_B8a.jp2")

b11 <- raster("*_B11.jp2")

b12 <- raster("*_B12.jp2")
```

```
#create stack

S2_stack_20 <- stack(b2,b3,b4,b5,b6,b7,b8,b8a,b11,b12)#20m

S2_stack_10 <- stack(b2,b3,b4,b8)#10m


#crop

#central coordinats

lat <- 39.94035

lon <- -5.77456

xy <- cbind(lat, lon) #AT-Neu


utm_zone <-  latlng_to_mgrs(lat, lon, degrees = TRUE, precision = 0)


xy <- data.frame(ID = 1, X = lon, Y = lat)

coordinates(xy) <- c("X", "Y")

proj4string(xy) <- CRS("+proj=longlat +datum=WGS84")

res <- spTransform(xy, CRS(paste("+proj=utm +zone=",utm_zone, "+ellps=WGS84",sep="")))

#e <- extent(674506,  677506, 5219197, 5222197)

e <- extent(res$X-500,  res$X+500, res$Y-500, res$Y+500)

s2_crop <- crop(S2_stack_10,e)


## returns the values of the raster dataset and write them in a matrix.

v <- getValues(s2_crop)

i <- which(!is.na(v))

v <- na.omit(v)

#subsample the crop

vx <- v[sample(nrow(v), 500),]


#elbow method for deciding the number of clusters

elbow.k <- function(mydata){

  dist.obj <- dist(mydata)

  hclust.obj <- hclust(dist.obj)

  css.obj <- css.hclust(dist.obj,hclust.obj)

  elbow.obj <- elbow.batch(css.obj)
  k <- elbow.obj$k
  return(k)
}

nclus_elb <- elbow.k(vx)

mydata <- vx
wss <- (nrow(mydata)-1)*sum(apply(mydata,2,var))
for (j in 2:15) wss[j] <- sum(kmeans(mydata,
                                     centers=j)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares")


#------------------------------------------------------------------------
# kmeans classification
set.seed(42)
E <- kmeans(v, nclus_elb, iter.max = 10000, nstart = 100, algorithm="MacQueen")
kmeans_raster <- raster(s2_crop)
kmeans_raster[i] <- E$cluster
plot(kmeans_raster)

#------------------------------------------------------------------------
# clara classification
set.seed(42)
pk1 <- pamk(v,krange=5:15,criterion="asw",critout=TRUE,usepam=FALSE)
clus <- clara(v,pk1$nc,samples=1000,metric="manhattan",pamLike=T)
clara_raster <- raster(s2_crop)
clara_raster[i] <- clus$clustering
plot(clara_raster)


#------------------------------------------------------------------------
## unsupervised randomForest classification using kmeans
set.seed(42)
vx<-v[sample(nrow(v), 500),]
rf = randomForest(vx)
```

```
rf_prox <- randomForest(vx,ntree = 1000, proximity = TRUE)$proximity


E_rf <- kmeans(rf_prox, nclus_elb, iter.max = 100, nstart = 10)

rf <- randomForest(vx,as.factor(E_rf$cluster),ntree = 500)

rf_raster<- predict(s2_crop,rf)

plot(rf_raster)


#----------------------------------------------------------------------

#xmeans

library(RWeka)


# Print a list of available options for the X-Means algorithm

WOW("XMeans")


# Create a Weka_control object which will specify our parameters

weka_ctrl <- Weka_control(

  I = 1000,                          # max no. of overall iterations

  M = 1000,                          # max no. of iterations in the kMeans loop

  L = 4,                            # min no. of clusters

  H = 10,                           # max no. of clusters

  D = "weka.core.EuclideanDistance", # distance metric Euclidean

  C = 0.4,                           # cutoff factor ???

  S = 42                             # random number seed (for reproducibility)

)


# Run the algorithm

x_means <- XMeans(v, control = weka_ctrl)


# Assign cluster IDs to original data set

xmeans_raster <- raster(s2_crop)

xmeans_raster[i] <- x_means$class_ids

plot(xmeans_raster)


#----------------------------------------------------------------------

#cmeans, the fuzzy version of the known kmeans

library(e1071)

c_means <- cmeans (v, centers = nclus_elb, iter.max=100, verbose=FALSE, dist="euclidean",
                   method="ufcl")

# Assign cluster IDs to original data set

cmeans_raster <- raster(s2_crop)

cmeans_raster[i] <- c_means$cluster




#The five classifications are stacked into one layerstack and plotted for comparison

class_stack <- stack(kmeans_raster,rf_raster,clara_raster,xmeans_raster, cmeans_raster)

names(class_stack) <- c("k-means","randomForest","clara","xmeans", "cmeans")


png("../Figures/classifications_eslma2015_s2_20x20.png", width = 1100, height = 700)

plot(class_stack)

dev.off()

#------------------------------------------------------------------------
```

WP3: synthetic experiment

In this WP, I selected classes of spatial structure and heterogeneity (Figure 4). These classes aim at representing simplified situations in real ecosystems. The ingestion of such maps into the on-line footprint tool will support the developments of guidelines for spectral sensor placement for specific footprint cases over ecosystem with a particular spatial heterogeneity.

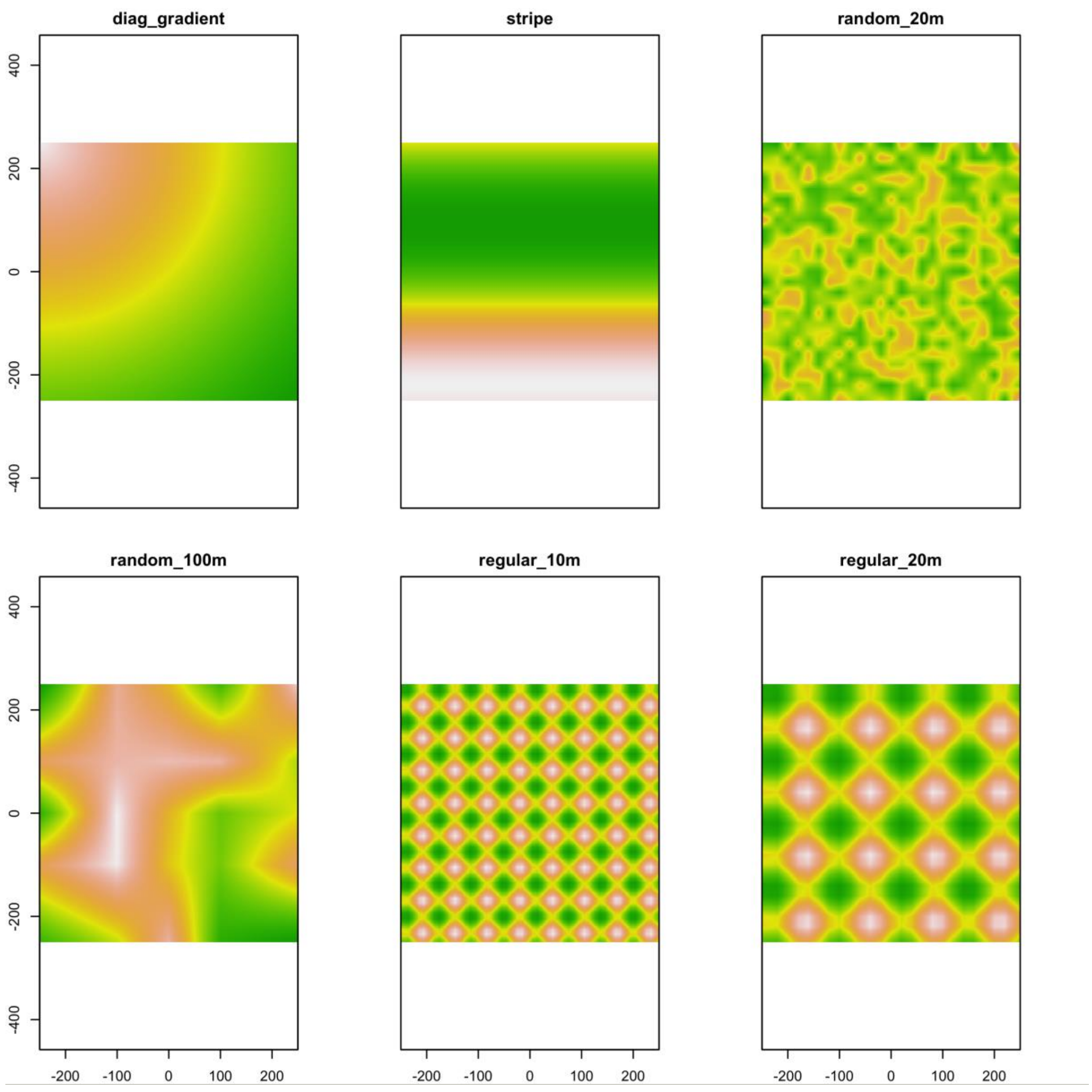

Figure 4: examples of artificial spatial heterogeneity and structure.

FUTURE COLLABORATIONS (if applicable)

The grantee will keep cooperating with the host for finalizing the implementation of the R scripts into the on-line version of the FFP footprint and for consolidating the results from WP3. A Manuscript is in preparation.