User manual for the the R package Spatial Dependence Random Forest (SDRF)

How to use the SDRF function

The next example uses the iris dataset, see Murphy (2012) for details, which is widely used to test predictive models and is included in the R program. In this case we simulate that the 150 different flowers were collected in 5 different training areas. We run SDRF function to calibrate the random forest model splitting the cases by areas, and we keep the in-bag information. Firstly we install the package from the bash terminal. You must be superuser. For example in Ubuntu:

```
sudo R CMD INSTALL SDRF_0.0-1.tar.gz
```

Once the package has been installed, we can run R and load the library

```
library(SDRF)
```

Now, we simulate the 5 training plots stored in the variable area. The first 50 cases belong to *Iris setosa*, the cases 51 to 100 to *Iris versicolor* and the last 50 cases to *Iris virginica*. Ten cases of each species will correspond to each area.

```
area=rep(rep(1:5,each=10),3)
```

We then tun the function SDRF to calibrate a random forest model keeping training and validation areas separated:

```
model=SDRF(Species~.,data=iris,areas=area,keep.inbag=TRUE)
```

We can then plot the distribution of the cases in the in-bag and the out-of-bag for the first 100 trees:

```
plot(0,xlim=c(0,151),ylim=c(0,101),type="n",xlab="Cases",ylab="Trees")

for (i in 1:100) points(1:150,rep(i,150),col=model$inbag[,i],pch=16,cex=0.2)
```

When the in-bag information of the first 100 trees is ploted (Figure 1), it is clear that the function have randomly selected areas for each tree, and included all the cases belonging to one area in the in-bag or the out-of-bag, but they are not mixed.

Finally we check that, at least in this toy example, we have not loss prediction capability. First we obtain the confusion matrix for our model and finally for the conventional random forest model:

```
model$confusion
```

```
 ##             setosa versicolor   virginica  class.error

 ##   setosa        50           0          0         0.00

 ##   versicolor     0          47          3         0.06

 ##   virginica      0           2         48         0.04
```

```
randomForest(Species~.,data=iris)$confusion
```

```
##              setosa versicolor   virginica  class.error
##   setosa         50          0           0         0.00
##   versicolor      0         47           3         0.06
##   virginica       0          4          46         0.08
```

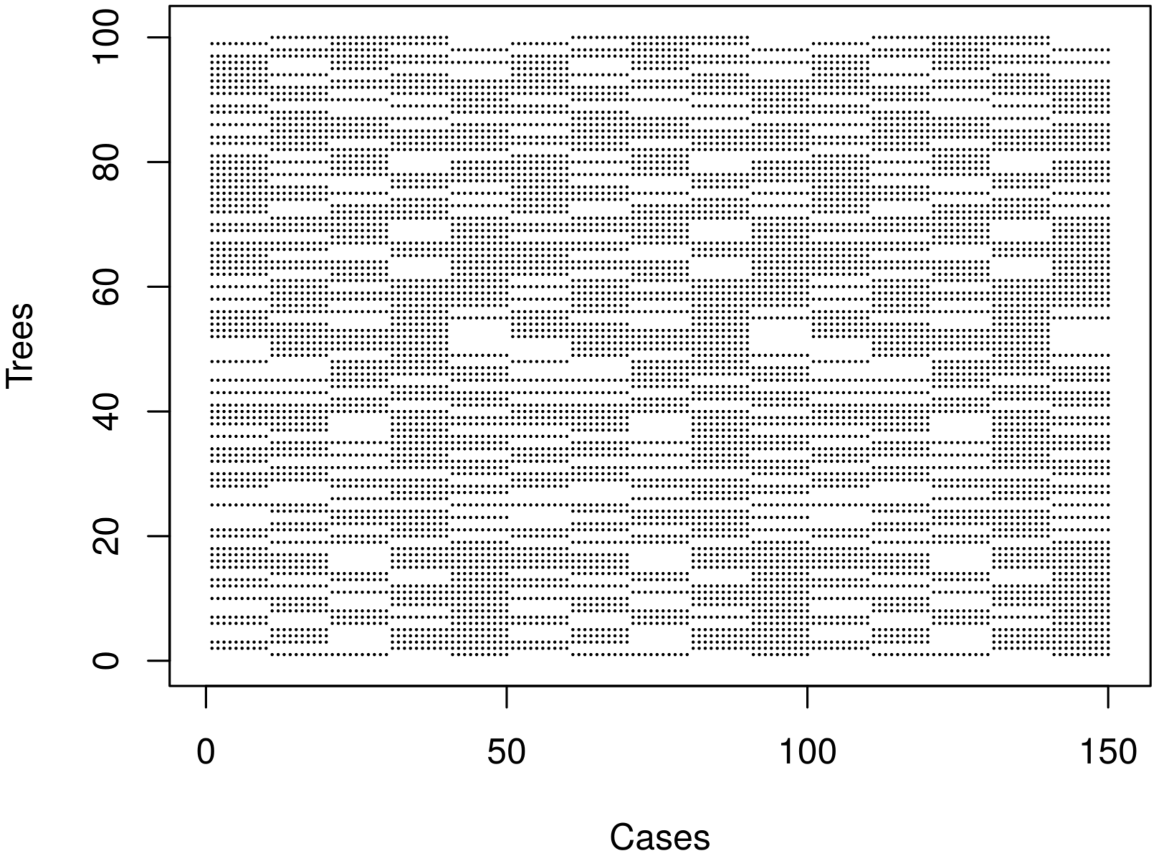
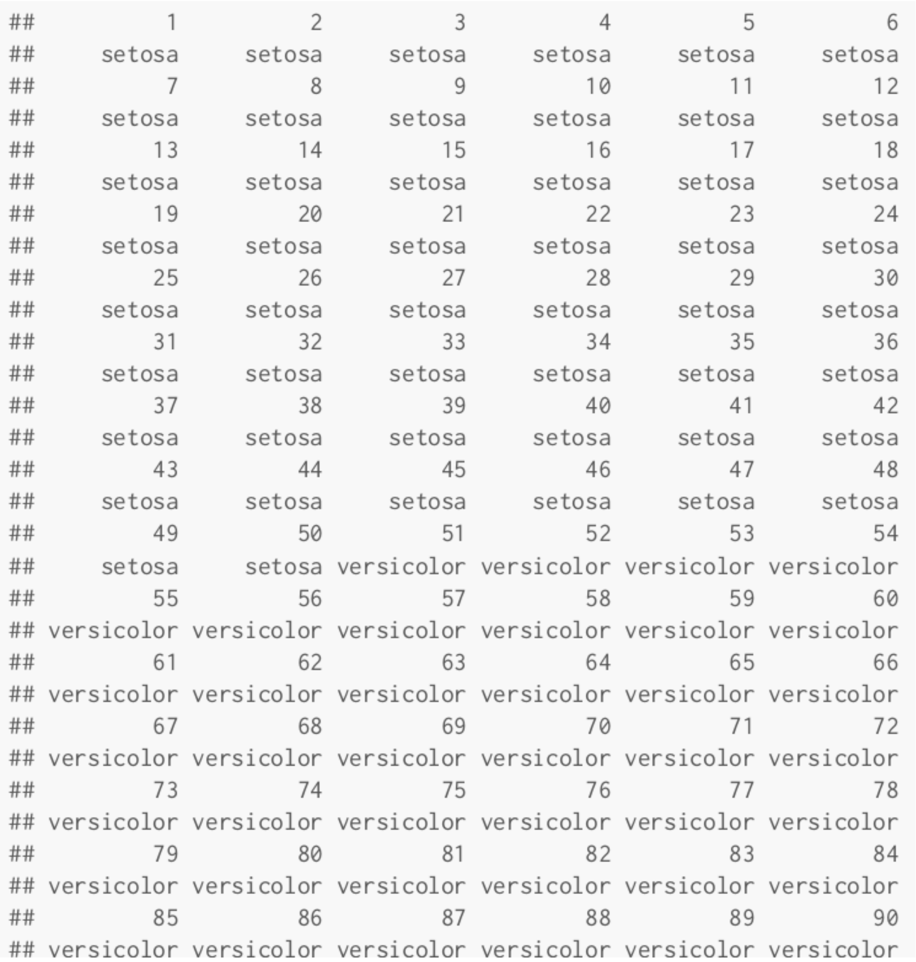


Figure 1. Cases in the in-bag of each tree

The object model is a randomForest object and can be used to predict (Figure 2):

```
predict(model,iris)
```



```
##          1          2          3          4          5          6
##     setosa     setosa     setosa     setosa     setosa     setosa
##          7          8          9         10         11         12
##     setosa     setosa     setosa     setosa     setosa     setosa
##         13         14         15         16         17         18
##     setosa     setosa     setosa     setosa     setosa     setosa
##         19         20         21         22         23         24
##     setosa     setosa     setosa     setosa     setosa     setosa
##         25         26         27         28         29         30
##     setosa     setosa     setosa     setosa     setosa     setosa
##         31         32         33         34         35         36
##     setosa     setosa     setosa     setosa     setosa     setosa
##         37         38         39         40         41         42
##     setosa     setosa     setosa     setosa     setosa     setosa
##         43         44         45         46         47         48
##     setosa     setosa     setosa     setosa     setosa     setosa
##         49         50         51         52         53         54
##     setosa     setosa versicolor versicolor versicolor versicolor
##         55         56         57         58         59         60
## versicolor versicolor versicolor versicolor versicolor versicolor
##         61         62         63         64         65         66
## versicolor versicolor versicolor versicolor versicolor versicolor
##         67         68         69         70         71         72
## versicolor versicolor versicolor versicolor versicolor versicolor
##         73         74         75         76         77         78
## versicolor versicolor versicolor versicolor versicolor versicolor
##         79         80         81         82         83         84
## versicolor versicolor versicolor versicolor versicolor versicolor
##         85         86         87         88         89         90
## versicolor versicolor versicolor versicolor versicolor versicolor
```

Figure 2. Prediction of the SDRF model

References

- Murphy, K.P. (2012) Machine Learning: A Probabilistic Perspective. The MIT Press.