

Package ‘ParallelDSM’

May 21, 2021

Type Package

Title Parallel Digital Soil Mapping using Machine Learning

Version 0.2.1

Description Parallel computing, multi-core CPU is used to efficiently compute and process multi-dimensional soil data. This package includes the parallelized Quantile Regression Forests algorithm for Digital Soil Mapping and is mainly dependent on the package 'quantregForest' and 'snowfall'. Detailed references to the R package and the web site are described in the methods, as detailed in the method documentation.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0), snowfall, raster

Imports methods, pryr, utils, caret, geoR, gstat, quantregForest, randomForest, rgdal, stats, sp,

RoxygenNote 7.1.0

NeedsCompilation no

Author Xiaodong Song [aut],
Gaoqiang Ge [aut, cre],
Jun Zhu [aut],
Ganlin Zhang [aut]

Maintainer Gaoqiang Ge <gaoqiangge2020@163.com>

Repository CRAN

Date/Publication 2021-05-21 09:00:05 UTC

R topics documented:

CVfunction	2
df.dem	3
df.input	3
df.mrrtf	4

df.plancur	4
df.procur	5
df.twi	5
DsmParallel	6
GetPredictorSubset	7
InsepectionVariable	8
MergingTiles	8
Parallelinit	9
Preprocess	10
smalltesttoy	11
Index	13

CVfunction	<i>For the gap between the predicted value and expected value of the model, the model validates the function</i>
------------	------------------------------------------------------------------------------------------------------------------

Description

For the gap between the predicted value and expected value of the model, the model validates the function

Usage

```
CVfunction(pred,actual)
```

Arguments

pred : Value predicted by the model
actual : The real value

Examples

```
test.pred <- c(2,4,5,7,2,4)
test.real <- c(1,2,3,4,5,6)
myres <- CVfunction(test.pred,test.real)
print(myres)
```

df.dem	<i>Sampling test data of the dem</i>
--------	--------------------------------------

Description

A dataset containing the df.dem and other attributes of almost 212000 df.dem The variables are as follows:

Usage

df.dem

Format

A data frame with 211415 rows and 3 variables:

dem data variable DEM

x The coordinate variable x

y The coordinate variable y

df.input	<i>Sampling test data</i>
----------	---------------------------

Description

A dataset containing the testdata and other attributes of almost 110 socd030 The variables are as follows:

Usage

df.input

Format

A data frame with 109 rows and 6 variables:

socd030 data variable socd030

dem data variable DEM

plancur data variable plancur

procur data variable procur

mrrtf data variable mrrtf

twi data variable twi

df.mrrtf	<i>Sampling test data of the mrrtf</i>
----------	----------------------------------------

Description

A dataset containing the df.mrrtf and other attributes of almost 212000 df.mrrtf The variables are as follows:

Usage

df.mrrtf

Format

A data frame with 211415 rows and 3 variables:

mrrtf data variable MRRTF

x The coordinate variable x

y The coordinate variable y

df.plancur	<i>Sampling test data of the plancur</i>
------------	------------------------------------------

Description

A dataset containing the df.plancur and other attributes of almost 212000 df.plancur The variables are as follows:

Usage

df.plancur

Format

A data frame with 211415 rows and 3 variables:

plancur data variable PLANCUR

x The coordinate variable x

y The coordinate variable y

df.procur	<i>Sampling test data of the procur</i>
-----------	-----------------------------------------

Description

A dataset containing the df.procur and other attributes of almost 212000 df.procur The variables are as follows:

Usage

df.procur

Format

A data frame with 211415 rows and 3 variables:

procur data variable PROCUR

x The coordinate variable x

y The coordinate variable y

df.twi	<i>Sampling test data of the twi</i>
--------	--------------------------------------

Description

A dataset containing the df.twi and other attributes of almost 212000 df.twi The variables are as follows:

Usage

df.twi

Format

A data frame with 211415 rows and 3 variables:

twi data variable TWI

x The coordinate variable x

y The coordinate variable y

DsmParallel

*Dsmparallel computings***Description**

Dsmparallel computings

Usage

DsmParallel(outpath,mymodels,from=NULL,to=NULL)

Arguments

outpath : Output path of the result of the prediction file. The default is "output".

mymodels : The models were selected, including QRF,RF and MLR.

from : Which row to start cutting the matrix

to : Where does the last row of the cut matrix go

References

Breiman, L. (2001). Random forests. Mach. Learn. 45, 5–32. Meinshausen, N. (2006) "Quantile Regression Forests", Journal of Machine Learning Research 7, 983-999 <http://jmlr.csail.mit.edu/papers/v7/>

Song, X.D., Ge, G.Q., Zhang, G.L. and Wu, H.Y. ParallelDSM: A R package for parallel soil mapping. Computers & Geosciences (to be available in 2021)

Examples

```
#####
## Review the documentation for the Preprocess function before using this function ##
#####

#####
## This function is the main function that performs parallel computations ##
## The outpath field refers to the filename of the data output ##
## The mymodels field has three modes to choose from: QRF,RF and MLR ##
## 'QRF' stands for Random Forest Model Prediction Method ##
## 'RF' stands for Machine Learning Model Prediction Method ##
## 'MLR' stands for Multiple Linear Regression Prediction Model ##
## 'from' and 'to' are reserved fields that can be left unused by the user##
#####
DsmParallel(outpath = "myoutputs",mymodels = "MLR",from=1,to=200)
```

GetPredictorSubset	<i>calculation function for cutting spatial data (tool function,Not as an open function, only for function calls)</i>
--------------------	-----------------------------------------------------------------------------------------------------------------------

Description

calculation function for cutting spatial data (tool function,Not as an open function, only for function calls)

Usage

```
GetPredictorSubset(  
  predictor.name,  
  iblock,  
  nblock,  
  fn,  
  nr,  
  nc,  
  resolutions,  
  pro,  
  from,  
  to  
)
```

Arguments

predictor.name : the name of the predictor variable
iblock : sequence code of parallel computing
nblock : number of target blocks (integer)
fn : The passed value of a global variable
nr : The passed value of a global variable
nc : The passed value of a global variable
resolutions : The passed value of a global variable
pro : The passed value of a global variable
from : Which row to start cutting the matrix
to : Where does the last row of the cut matrix go

Value

Parallel calculation of the cut part of the data box data

References

Breiman, L. (2001). Random forests. *Mach. Learn.* 45, 5–32. Meinshausen, N. (2006) "Quantile Regression Forests", *Journal of Machine Learning Research* 7, 983-999 <http://jmlr.csail.mit.edu/papers/v7/>
Song, X.D., Ge, G.Q., Zhang, G.L. and Wu, H.Y. ParallelDSM: A R package for parallel soil mapping. *Computers & Geosciences* (to be available in 2021)

Examples

```
GetPredictorSubset("dem", 4, 10, "covariate", 486, 777, NULL, NULL, 1, 10)
```

InsepectionVariable *A function that checks the parallel computation for missing data*

Description

A function that checks the parallel computation for missing data

Usage

```
InsepectionVariable(myblock)
```

Arguments

myblock : the number of blocks for data cutting

Examples

```
InsepectionVariable(myblock = 10)
```

MergingTiles *A function that combines the results of parallel cutting into a single file*

Description

A function that combines the results of parallel cutting into a single file

Usage

```
MergingTiles(df_dem, f.i.d, f.iblock, n.block, f.o.d, f.suffix)
```


Arguments

df_dem : The predicted source file before merging
 f.i.d : Enter the absolute path to the file
 f.iblock : The filename prefix of the resulting result
 n.block : The number of blocks cut is calculated in parallel
 f.o.d : The absolute output path of the file
 f.suffix : The suffix for the output of the file

Examples

```
# you must have a file, which is name "myres"
# Merging files, for example:
# f.input.directory <- c("e:/3_20190603_R/results/mapping/test_merging/")
# f.input.iblock <- c("sics030_")
# n.block <- 100
# f.output.directory <- c("e:/3_20190603_R/results/mapping/erpu.sics030_fuse/")
# f.output.suffix <- c("sics030_together.tif")
# Naming rules: file.name.directory + file.name.iblock + ".tif"

rmap_dem <- raster("E:/12_Parallel_Test_Paper_R/covariate/250m/dem.tif")
spdf_dem <- as(rmap_dem,"SpatialPointsDataFrame")
df_dem <- as.data.frame(spdf_dem)
# mergeing results together
n.block <- 100
f.i.d <- c("E:/12_Parallel_Test_Paper_R/results/mapping_250m/")
f.o.d <- c("E:/12_Parallel_Test_Paper_R/results/mapping_250m_merge/")
f.iblock <- c("mlr.ak05.")
f.suffix <- c("mlr.ak05.tif")
MergingTiles(df_dem, f.i.d, f.iblock, n.block, f.o.d, f.suffix)
```

 Parallelinit

Parallel computing initialization preparation

Description

Parallel computing initialization preparation

Usage

```
Parallelinit(mymodel)
```

Arguments

mymodel : The models were selected, including QRF,RF and MLR.

Value

Represents whether the loading of the required variables and dependent packages is complete

Examples

```
Parallelinit(mymodel = "QRF")
```

Preprocess	<i>As a data preprocessing function, sets some global variables that are not visible to the user</i>
------------	------------------------------------------------------------------------------------------------------

Description

As a data preprocessing function, sets some global variables that are not visible to the user

Usage

```
Preprocess(Fpath="", fn="", tname="", mc=NULL, icsv=NULL, itif=NULL, nblock=6, ncore=2, Fc=1)
```

Arguments

<code>Fpath</code>	: the path of file
<code>fn</code>	: Name of the folder in which the soil data is stored
<code>tname</code>	: Standard soil files, which can be used as sample files (under in the Folder-Name)
<code>mc</code>	: Read file mode (using data sets, or read yourself)
<code>icsv</code>	: Use <code>df.input</code> from the built-in dataset)
<code>itif</code>	: Use <code>df.dem</code> in the built-in data set)
<code>nblock</code>	: the number of blocks for data cutting
<code>ncore</code>	: Computes the CPU's kernel in parallel(fill in according to the computer configuration)
<code>Fc</code>	: the encoding of file

References

Breiman, L. (2001). Random forests. *Mach. Learn.* 45, 5–32. Meinshausen, N. (2006) "Quantile Regression Forests", *Journal of Machine Learning Research* 7, 983-999 <http://jmlr.csail.mit.edu/papers/v7/>
 Song, X.D., Ge, G.Q., Zhang, G.L. and Wu, H.Y. ParallelDSM: A R package for parallel soil mapping. *Computers & Geosciences* (to be available in 2021)

Examples

```
#####
## Example code 1                                     ##
## Select your own reading method, as shown below   ##
#####
mydatas <- system.file("extdata", "all.input.csv", package = "ParallelDSM")
sampledatas <- system.file("extdata", "covariate", package = "ParallelDSM")
Preprocess(mydatas,sampledatas,"twi.tif")

#####
## Example code 2 (It is highly recommended)        ##
## If you want to use test cases, load the relevant data sets ##
#####
# Select the data set that comes with this package

# data("df.input")
# data("df.dem")

#####
## Use the data file references that come with this package ##
#####
# sampledatas <- system.file("extdata", "covariate", package = "ParallelDSM")

#####
## Use preprocessor functions to process the data that is loaded in##
#####
# Preprocess(fn = sampledatas,mc = TRUE,icsv = df.input,itif = df.dem)

#####
## This function is the main function that performs parallel computations ##
## The outpath field refers to the filename of the data output ##
## The mymodels field has three modes to choose from: QRF,RF and MLR ##
## 'QRF' stands for Random Forest Model Prediction Method ##
## 'RF' stands for Machine Learning Model Prediction Method ##
## 'MLR' stands for Multiple Linear Regression Prediction Model ##
## 'from' and 'to' are reserved fields that can be left unused by the user##
#####

#DsmParallel(outpath = "myoutputs",mymodels = "MLR",from=1,to=200)
```

smallesttoy

Black box test function to test whether R package was installed successfully

Description

Black box test function to test whether R package was installed successfully

Usage

```
smalltesttoy(myflag)
```

Arguments

myflag : The black box tests the successful entry mark

Examples

```
smalltesttoy(myflag = "1")
```

Index

* datasets

- df.dem, 3
- df.input, 3
- df.mrrtf, 4
- df.plancur, 4
- df.procur, 5
- df.twi, 5

CVfunction, 2

- df.dem, 3
- df.input, 3
- df.mrrtf, 4
- df.plancur, 4
- df.procur, 5
- df.twi, 5
- DsmParallel, 6

GetPredictorSubset, 7

InsepectionVariable, 8

MergingTiles, 8

Parallelinit, 9

Preprocess, 10

smallesttoy, 11