基于R的动态非点源磷污染指数模型

苏静君

2017年7月7日

## 磷指数简介

流域非点源磷污染指数(Phosphorus Index,PI)通过对影响磷养分流失的因子(如源因子、迁移因子)及其相互作用进行评估以表征养分流失至水体的潜在风险，并以此风险的高低为依据判定养分流失的关键源区。和其它经验型非点源模型如输出系数模型相比，指数模型考虑了影响养分流失的关键因子和过程及其在空间上的差异性，可以更加真实的反映流失过程及流失程度在空间分布的差异性；而和复杂的过程模型相比，指数模型多利用现有数据，受到数据缺乏的制约较小。在流域管理和区域规划层面，养分指数被用于识别区域、流域乃至田间最大潜在养分输出区或关键源区，帮助管理者和农民确定最佳管理措施的具体实施位置。养分指数已广泛应用于欧美部分国家的养分管理体系，而我国在流域养分指数方面的研究和应用工作尚少。本文的磷指数基于R语言平台建立，利用R超强的数据处理能力及其在空间分析上的拓展，构建了一个融水文、空间、数学运算于一体的非点源磷污染指数模型，模型输出既能提供非点源磷污染负荷值，又能展示其在空间上的分布，有助于识别非点源磷污染的关键源区。

## 磷指数结构

### 结构

指数模型由三个模块组成，分别估算来自土壤、肥料、畜禽养殖及农村生活排放非点源磷污染的负荷。在每个模块中包括多个输入数据及参数，分别表征影响该种磷流失的源因子及迁移因子。

### 流程

### 模型当前运行环境

软件版本：R version 3.3.3 (2017-03-06) Platform: i386-w64-mingw32/i386 (32-bit)

RStudio-1.0.136

电脑软硬件： 系统：Windows 7 32位操作系统 CPU：Inter(R) Core(TM) i7-4770 CPU @ 3.40GHz

模型工作空间

path <- "D:\\SpatialR\\Paper2"  
  
setwd(path)

### 模型所需调用的包

模型运行涉及空间数据的读取、处理等，以及水文数据的处理。需要安装并加载以下包。

library(sp)  
library(raster)  
library(rgdal)  
library(gstat) #反距离插值  
library(rgeos) #欧式距离计算  
library(dplyr)  
library(EcoHydRology) #基流分割

### 模型输入数据及参数

#### (1)肥料

肥料数据主要来自各县市统计年鉴，土地利用在本案例用的是2010年30m\*30m土地利用栅格。

PferFile <- paste(path,"\\Pfer\_kgha.csv",sep="") ###以kg/ha为单位的单位面积农田施肥量  
luFile <- paste(path,"\\landsue.tif",sep="") ###土地利用源文件30\*30m  
countybd <- paste(path,"\\county1.shp",sep="") ###研究区内各行政单元边界  
countyname <- c("Xinglong","Yingzi") ###研究区内各行政单元名称

(2)畜禽养殖及人类生活

畜禽养殖及农村人口数据来自各县市统计年鉴，污染物排放系数参考相关研究及调查报告。

Livestock <- paste(path,"\\Livestock.csv",sep="")   
head(read.csv(Livestock,header=T,sep=","))

## County Year Cattle Poultry Rural Rabbits Sheep Swine  
## 1 Xinglong 2000 150 14915 2773 460 1729 3549  
## 2 Xinglong 2001 123 19879 2778 503 1906 3674  
## 3 Xinglong 2002 121 19599 2782 142 1867 3381  
## 4 Xinglong 2003 150 23036 2546 102 2194 3600  
## 5 Xinglong 2004 162 21882 2791 245 2399 3771  
## 6 Xinglong 2005 175 24400 2599 203 2418 3878

ECC <- paste(path,"\\ECC.csv",sep="")   
read.csv(ECC,header=T,sep=",")

## Type TP  
## 1 Cattle 7.56  
## 2 Poultry 0.24  
## 3 Rural 0.52  
## 4 Rabbits 0.12  
## 5 Sheep 2.26  
## 6 Swine 5.14

countyarea <- paste(path,"\\countyarea.csv",sep="") ###各行政单元面积  
read.csv(countyarea,header=T,sep=",")

## County SQKM\_COUN  
## 1 Xinglong 1968  
## 2 Yingzi 150

perarea <- paste(path,"\\perc.area.csv",sep="") ###研究区内各行政单元面积占总行政单元比例  
read.csv(perarea,header=T,sep=",")

## County perc.area  
## 1 Xinglong 0.832  
## 2 Yingzi 0.153

TPmanfn <- paste(path,"\\TPman.csv",sep="") ###以kg/ha为单位的单位流域面积畜禽养殖P产生量  
head(read.csv(TPmanfn,header=T,sep=","))

## Year Xinglong Yingzi  
## 1 2000 11.98966 1.379305  
## 2 2001 12.85105 1.694873  
## 3 2002 12.12486 1.683326  
## 4 2003 13.30069 1.831961  
## 5 2004 13.85052 1.994508  
## 6 2005 14.35388 1.704318

(3)土壤

土壤数据包括有机质(OM,%)，速效磷(OlsenP,mg/kg)。全磷含量(TPsoi,mg/kg)由经典公式推算。

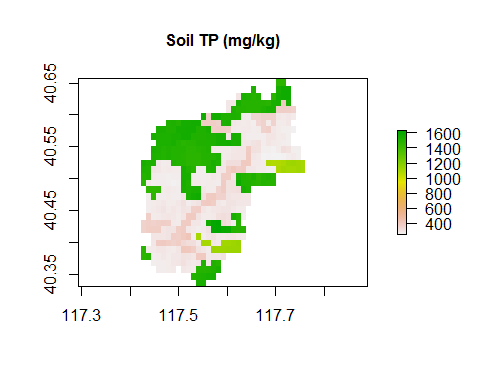
library(sp)  
library(raster)  
library(rgdal)  
  
crs.geo <- CRS("+proj=longlat +ellps=WGS84 +datum=WGS84") ####定义统一的地理坐标系  
mask <- raster(paste(path,"\\mask.c.grd",sep="")) ###定义了统一边界、分辨率及坐标系  
  
om <- raster(readGDAL(paste(path,"\\OM.tif",sep=""))) ###源文件90\*90M

## D:\SpatialR\Paper2\OM.tif has GDAL driver GTiff   
## and has 343 rows and 356 columns

oc <- om/1.724 ###有机质转化为有机碳  
om <- resample(om,mask,method="ngb")  
  
OlsenP <- raster(readGDAL(paste(path,"\\OlsenP1.tif",sep=""))) ###源文件

## D:\SpatialR\Paper2\OlsenP1.tif has GDAL driver GTiff   
## and has 1044 rows and 1086 columns

OlsenP <- resample(OlsenP,mask,method="ngb") ###转换为统一边界、分辨率及坐标系  
  
TPsoi <- (13+2.7\*om + 0.06\*OlsenP)^2 ###产生TP的图层  
  
plot(TPsoi,main="Soil TP (mg/kg)",cex.main=1)



(4)降雨、径流

Rday <- paste(path,"\\Rday.csv",sep="")   
TFts <- paste(path,"\\TF.csv",sep="")   
  
head(read.csv(Rday),header=T,sep=",") ###Flow代表流速，单位为m3/s

## Year Month Day Flow  
## 1 2000 1 1 0.665  
## 2 2000 1 2 0.697  
## 3 2000 1 3 0.678  
## 4 2000 1 4 0.649  
## 5 2000 1 5 0.614  
## 6 2000 1 6 0.592

head(read.csv(TFts),header=T,sep=",") ###TF代表年总径流深，单位为mm/year

## Year TF  
## 1 1980 129.5  
## 2 1981 46.8  
## 3 1982 232.1  
## 4 1983 141.6  
## 5 1984 82.1  
## 6 1985 232.1

(5)土壤侵蚀相关

library(sp)  
library(raster)  
library(rgdal)  
  
Boundfile <- paste(path,"\\boundLiu.shp",sep="") ###边界  
  
dem <- raster(readGDAL(paste(path,"\\dem30.tif",sep=""))) ###读取DEM数据

## D:\SpatialR\Paper2\dem30.tif has GDAL driver GTiff   
## and has 1168 rows and 1215 columns

clay <- raster(readGDAL(paste(path,"\\clay.tif",sep=""))) ###源文件90\*90M

## D:\SpatialR\Paper2\clay.tif has GDAL driver GTiff   
## and has 343 rows and 356 columns

sand<-raster(readGDAL(paste(path,"\\sand1.tif",sep=""))) ###源文件90\*90M

## D:\SpatialR\Paper2\sand1.tif has GDAL driver GTiff   
## and has 343 rows and 356 columns

silt <- raster(readGDAL(paste(path,"\\silt.tif",sep=""))) ###源文件90\*90M

## D:\SpatialR\Paper2\silt.tif has GDAL driver GTiff   
## and has 343 rows and 356 columns

Rainfile <- paste(path,"\\rainStation.csv",sep="") ###仅有日降雨量>12mm的降雨被定义为有效降雨，表格中为年总有效降雨量，单位mm.  
  
head(read.csv(Rainfile),header=T,sep=",",tidy=TRUE)

## Name lon lat X2000 X2001 X2002 X2003 X2004 X2005 X2006  
## 1 Liying 117.4833 40.41208 321.5 522.7 203.1 462.2 427.3 420.8 384.1  
## 2 Xinglong 117.7379 40.60508 434.9 571.9 251.7 494.8 433.4 441.0 290.0  
## X2007 X2008 X2009 X2010  
## 1 334.5 383.5 376.4 397.8  
## 2 418.4 426.9 446.2 367.1

(6)模拟时段

Year <- 2000:2010  
nyear <- length(Year)

(7)输入参数

Conv.coef <- 0.004  
DPDR <- 0.4  
PDR <- 0.5  
Perc.WSP <- 0.6  
Ploss.man <- 0.35  
Ploss.fer <- 0.65  
param <- as.data.frame(cbind (Conv.coef,DPDR,PDR,Perc.WSP,Ploss.man,Ploss.fer))

### 数据前处理

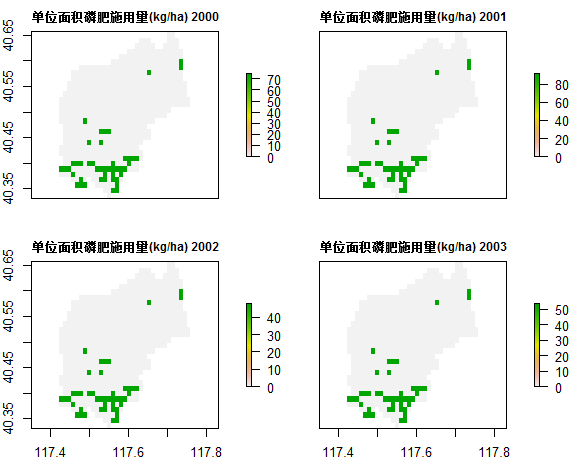
#### 肥料磷

肥料磷主要基于各县市统计年鉴磷肥施用量及农田面积，计算单位面积农田磷肥施用量，其他土地利用类型上磷肥施用量为0，对其进行数字化以用于模型运算。

f.Pfer <- function(PferFile,luFile,countybd,countyname){  
   
 Pfer\_kgha <- read.csv(PferFile,header=T,sep=",")  
   
 land30<-raster(readGDAL(luFile))###源文件30\*30m  
   
 ###提取出耕地  
 AgriLand <- land30  
 AgriLand[AgriLand != 412] <- 0   
 AgriLand[AgriLand == 412] <- 1  
   
 ###提取出县市区  
 county.r <- readOGR(countybd)  
 county.r <- rasterize (county.r,AgriLand,field ="CountyCode") ###countycode 1,2,3等分别代指研究区内各行政单元  
   
 ###intersect 县市及耕地  
 inter.r <- AgriLand\*county.r  
   
 ###生成施肥量的空间分布时间序列文件  
 DPfer.ls <- list()  
 for (i in 1:length(Year))  
 {  
 temp <- inter.r  
 for (j in 1:length(countyname))  
 {   
 temp [temp == j] <- Pfer\_kgha[i,j+1]  
 temp <- resample(temp,mask,method="ngb")   
 }  
 DPfer.ls [[i]] <- temp  
 }   
   
return(DPfer.ls)  
}  
  
TPfer <- stack(f.Pfer(PferFile,luFile,countybd,countyname))

## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\county1.shp", layer: "county1"  
## with 2 features  
## It has 3 fields

plot(TPfer[[1:4]],main=paste("单位面积磷肥施用量(kg/ha)",Year[1:4],sep=" "),cex.main=0.9)



#### 畜禽养殖及农村生活排放磷

(1)畜禽养殖及农村生活磷污染的产生量计算

f.excrPman <- function (Livestock,ECC,countyarea,perarea){  
  
 ###读取数据   
 livestock <- read.csv(Livestock,head=T,sep=",")  
 ecc.TP <- read.csv(ECC,header=T,sep=",")  
 area <- read.csv(countyarea,header=T,sep=",")  
 perc.area <- read.csv(perarea,header=T,sep=",")  
   
 #####计算总磷的产生量  
 livestck <- stack (livestock[,3:8])  
 livestck$County <-rep(rep(countyname,each=nyear),6)  
 livestck$Year <- rep(Year,2\*6)  
   
 livestck <- merge(livestck,ecc.TP,by.x="ind",by.y="Type",all=T)  
 livestck <- merge(livestck,area,by="County",all=F)  
 livestck <- merge(livestck,perc.area,by="County",all=F)  
   
 livestck$excr.TP <-livestck$values \* livestck$TP \*livestck$perc.area / livestck$SQKM\_COUN #######各行政区TP kg/ha  
   
 result <- aggregate (livestck$excr.TP, by=list(livestck$Year,livestck$County),FUN = sum)  
 colnames(result)<-c("year","County","excr.Pman")  
 result <- cbind(Year,unstack (result,excr.Pman~County))  
   
 return(result)  
   
}  
  
TPman <- f.excrPman(Livestock,ECC,countyarea,perarea)  
head(TPman)

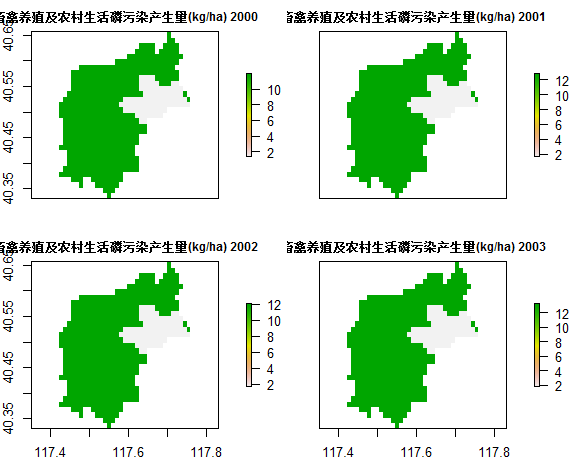
## Year Xinglong Yingzi  
## 1 2000 11.98966 1.379305  
## 2 2001 12.85105 1.694873  
## 3 2002 12.12486 1.683326  
## 4 2003 13.30069 1.831961  
## 5 2004 13.85052 1.994508  
## 6 2005 14.35388 1.704318

(2)畜禽养殖及农村生活磷污染产生量时间序列的数字化

f.Pman <- function(TPman,countybd,countyname){  
   
 county.r <- readOGR(countybd)  
 county.r <- rasterize (county.r,mask,field ="CountyCode") ###countycode 1,2分别指的是兴隆县和营子区  
  
 DPman.ls <- list()  
 for (i in 1:nyear){  
 temp <- county.r  
 for (j in 1:length(countyname)){  
 temp [temp == j] <- TPman[i,j+1]}  
 DPman.ls [[i]] <- temp  
 }  
   
 return(DPman.ls)  
   
}  
  
TPman <- stack(f.Pman(TPman,countybd,countyname))

## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\county1.shp", layer: "county1"  
## with 2 features  
## It has 3 fields

plot(TPman[[1:4]],main=paste("畜禽养殖及农村生活磷污染产生量(kg/ha)",Year[1:4],sep=" "),cex.main=0.9)



## 土壤侵蚀模数

土壤侵蚀模数由经典土壤侵蚀方程计算所得。输入数据需要土壤、土地利用、降雨等。

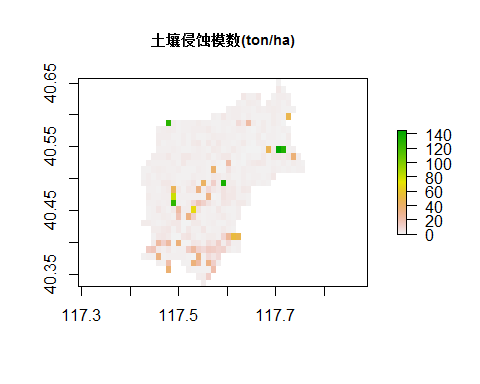
### 所需函数

library(sp)  
library(raster)  
library(rgdal)  
library(gstat)  
  
  
##(1) 求LS因子  
f.S<-function(x){  
 x[is.na(x)]<-NA  
 x[x >= 0 & x < 0.08726646] <- 10.8 \* sin(x[x >= 0 & x < 0.08726646]) +0.03  
 x[x >= 0.08726646 & x < 0.17453293]<- 16.8 \* sin(x[x >= 0.08726646 & x < 0.17453293]) - 0.5  
 x[x >= 0.17453293] <- 21.91 \* sin(x[x >= 0.17453293]) - 0.96  
 return(x)  
 } #########0.08726646等为将原公式的临界值转化为弧度aa\*pi/180  
   
  
f.li<-function(x,y){  
   
 flowdir <- terrain(x, opt='flowdir', neighbors=8)  
 Di <- flowdir  
 Di [Di==2 | Di==8 | Di==32 | Di==128 ] <- sqrt(2)\*30  
 Di [Di!=2 & Di!=8 & Di!=32 & Di!=128 ] <- 30  
   
 return(Di/cos(y))  
 } ############像元坡长函数  
   
   
f.m <- function(x){  
 m <- tan(x)  
 m[m >= 0.05] <- 0.5  
 m[m >= 0.03 & m < 0.05] <- 0.4  
 m[m >= 0.01 & m < 0.03] <- 0.3  
 m[m < 0.01] <- 0.2  
 return(m)  
 } #########m值,公式中使用百度比坡度  
   
   
##（2）求K因子  
f.k<-function(x,y,z,v){  
 temp1<-0.2+0.3\*exp(-0.0256\*x\*(1-y/100))  
 temp2<-(y/(z+y))^0.3  
 temp3<-1-0.25\*v/(v+exp(3.72-2.95\*v))  
 temp4<-1-0.7\*(1-x/100)/(1-x/100+exp(-5.51+22.9\*(1-x/100)))  
 k<- 0.0726\*temp1\*temp2\*temp3\*temp4-0.0102  
 return(k)  
} ####北京土壤侵蚀因子估算公式  
  
   
##（4）求R因子  
f.R<-function(Rainfile,Boundfile,crs,n,MASK){  
   
 Rain <- read.csv(Rainfile)  
 Rain[,4:(3+nyear)] <- 0.44 \* (Rain[,4:(3+nyear)]) ^ 1.463 #####将降雨转化成降雨侵蚀力  
   
 coordinates(Rain) <- c("lon","lat")  
 proj4string(Rain) <- crs ###  
   
 bound <- readOGR(Boundfile) ####  
 bound <- spTransform(bound,crs)  
   
 require(gstat)  
   
 idw.out <- idw(Rain[[n+1]]~1,Rain,bound,idp=1)####降雨侵蚀力反距离插值  
 R <- rasterize(idw.out,MASK,field ="var1.pred") ######降雨侵蚀力栅格化  
   
 return(R)  
}  
  
  
##（5）求CP因子  
   
f.c<-function(x){  
 x[x == 112 | x == 113 | x == 114 | x == 122] <- 0.008  
 x[x == 213] <- 0.10  
 x[x == 322 | x == 323 | x == 511] <- 0  
 x[x == 412] <- 0.31  
 x[x == 613] <- 1  
 return(x)  
 }  
   
   
f.p<-function(x){  
 x[x == 112 | x == 113 | x == 114 | x == 122 | x == 213 | x == 511 | x == 613 ] <- 1  
 x[x == 322 | x == 323] <- 0  
 x[x == 412] <- 0.35  
 return(x)  
}

### RUSLE模型

f.usle<-function (dem,sand,silt,clay,oc,luFile,Rainfile,Boundfile,crs,n,MASK){  
   
 slope.beta <- terrain(dem,opt='slope', unit='radians', neighbors=8)   
   
 ##################################（1）各因子计算  
 S <- f.S(slope.beta)###直接用弧度制计算  
 L <- (f.li(dem,slope.beta)/22.13)^ f.m(slope.beta)###########L为坡长因子  
 k <- f.k(sand,silt,clay,oc)  
 P <- f.p(raster(readGDAL(luFile)))  
 C <- f.c(raster(readGDAL(luFile)))  
   
   
 ##################################（2）统一范围、坐标系等  
   
 S.resamp <- resample(S,mask,method="ngb")  
 L.resamp <- resample(L,mask,method="ngb")  
 k.resamp <- resample(k,mask,method="ngb")  
 C.resamp <- resample(C,mask,method="ngb")  
 P.resamp <- resample(P,mask,method="ngb")  
   
 R.resamp <- f.R(Rainfile,Boundfile,crs,n,MASK)  
  
 ##################################（3）计算土壤侵蚀模数  
 mydata<-stack(S.resamp,L.resamp,k.resamp,C.resamp,P.resamp,R.resamp)  
   
 f.A <- function(x) {prod(x)}   
   
 return(f.A(mydata)) #######单位t•hm•a-1  
}  
  
  
plot(f.usle(dem,sand,silt,clay,oc,luFile,Rainfile,Boundfile,crs=crs.geo,n=1,MASK=mask),main="土壤侵蚀模数(ton/ha)",cex.main=0.9)

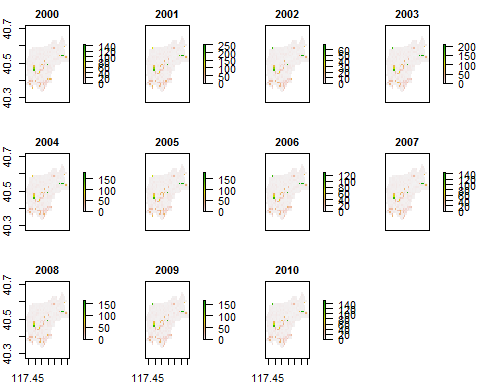
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]



a<-list()  
for (i in 1:length(Year)){  
a[[i]] <- f.usle(dem,sand,silt,clay,oc,luFile,Rainfile,Boundfile,crs=crs.geo,n=i,MASK=mask)  
}

## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]

plot(stack(a),main=Year,cex.main=1)



## 径流因子

library(EcoHydRology)

## Loading required package: operators

##   
## Attaching package: 'operators'

## The following objects are masked from 'package:base':  
##   
## options, strrep

## Loading required package: topmodel

## Loading required package: DEoptim

## Loading required package: parallel

##   
## DEoptim package  
## Differential Evolution algorithm in R  
## Authors: D. Ardia, K. Mullen, B. Peterson and J. Ulrich

## Loading required package: XML

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:operators':  
##   
## %>%

## The following objects are masked from 'package:raster':  
##   
## intersect, select, union

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

f.runoff<-function(Rday,TFts){  
   
 #####计算径流深，基径比等  
 mydata1<-read.csv(Rday,header=T,sep=",")  
   
 bf<-BaseflowSeparation(mydata1$Flow, filter\_parameter = 0.95, passes = 1)  
   
 Rday.bf<-cbind(mydata1[,1:4],bf$bt)  
 colnames(Rday.bf)[5]<-"bf"  
   
   
 Rday.bf <- mutate (Rday.bf,RDsub = bf\*3600\*24/620000,RDsur = (Flow-bf)\*3600\*24/620000)   
 Rday.bf <- mutate (Rday.bf,DRsub = RDsub / (RDsub+RDsur),DRsur = 1-RDsub / (RDsub+RDsur),TF = RDsub + RDsur)   
   
 RDsub<-tapply(Rday.bf$RDsub,Rday.bf$Year,FUN=sum,simplify=T)  
 RDsur<-tapply(Rday.bf$RDsur,Rday.bf$Year,FUN=sum,simplify=T)  
 DRsub<-tapply(Rday.bf$DRsub,Rday.bf$Year,FUN=mean,simplify=T)  
 DRsur<-tapply(Rday.bf$DRsur,Rday.bf$Year,FUN=mean,simplify=T)  
 TF<-tapply(Rday.bf$TF,Rday.bf$Year,FUN=sum,simplify=T)  
   
 ####计算TFranks.利用长时间序列径流生成年径流的经典累积分布曲线。任一年径流对应的频率可以通过经典累积分布曲线获得。  
 mydata2 <- read.csv(TFts,header=T,sep=",")  
 arrange(mydata2,TF)  
   
 f.ecdf<-ecdf(mydata2$TF)  
   
 TFranks <- f.ecdf(TF)  
   
 return(cbind(TF,RDsub,RDsur,DRsub,DRsur,TFranks))  
}  
  
Runoff <- as.data.frame(f.runoff(paste(path,"\\Rday.csv",sep=""),paste(path,"\\TF.csv",sep="")))  
head(Runoff)

## TF RDsub RDsur DRsub DRsur TFranks  
## 2000 52.47838 40.16583 12.312553 0.9064653 0.09353470 0.06250  
## 2001 155.08312 100.85079 54.232330 0.8912975 0.10870250 0.50000  
## 2002 38.17041 33.80643 4.363976 0.9114298 0.08857021 0.00000  
## 2003 59.58827 47.74385 11.844422 0.8542496 0.14575040 0.15625  
## 2004 104.69534 78.37737 26.317974 0.8587157 0.14128427 0.34375  
## 2005 146.00276 99.88393 46.118835 0.8762127 0.12378729 0.50000

## SDR因子

SDR是表征流域尺度侵蚀土壤产生量与流域出口泥沙量的比值，反映侵蚀土壤的输移效率。在此指数中，采用了基于研究区多年径流与泥沙数据得到的经典公式。公式中的回归系数可以因研究区的不同而进行调整。

f.SDR <- function(x) {0.00028 \* exp(2.95\*log10(x))}  
  
f.SDR(Runoff$RDsur)

## [1] 0.006983577 0.046669749 0.001849078 0.006645236 0.018481663  
## [6] 0.037919875 0.004089124 0.008285004 0.014762025 0.006643415  
## [11] 0.006345019

## 磷指数模型

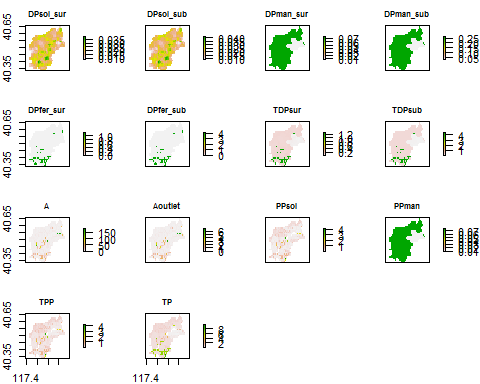
f.Pind <- function(year,param,Runoff,TPfer,TPman){  
  
 n <- which(Year==year)  
   
 ###计算土壤溶解态磷的流失，考虑转换系数及河道过程溶解态磷的输移率  
   
 DPsoi\_sur <- OlsenP\*prod(Runoff$RDsur[n],param$Conv.coef,param$DPDR,0.01)  
 DPsoi\_sub <- OlsenP\*prod(Runoff$RDsub[n],param$Conv.coef,param$DPDR,0.01,param$PDR)  
  
 ###计算Manure溶解态磷的流失，考虑溶解比例，陆水流失系数、河道过程溶解态磷的输移率、地表地下浓度比  
   
 DPman\_sur <- TPman[[n]]\*prod(Runoff$TFranks[n],Runoff$DRsur[n],param$Perc.WSP,param$DPDR,param$Ploss.man)  
 DPman\_sub <- TPman[[n]]\*prod(Runoff$TFranks[n],Runoff$DRsub[n],param$Perc.WSP,param$DPDR,param$Ploss.man,param$PDR)  
  
 ###计算FER溶解态磷的流失，考虑肥料利用率、河道过程溶解态磷的输移率、地表地下浓度比  
  
 DPfer\_sur <- TPfer[[n]]\*prod(Runoff$TFranks[n],Runoff$DRsur[n],param$Ploss.fer,param$DPDR)  
 DPfer\_sub <- TPfer[[n]]\*prod(Runoff$TFranks[n],Runoff$DRsub[n],param$Ploss.fer,param$DPDR,param$PDR)  
   
 ###计算DP组分  
 TDPsur <- sum(DPsoi\_sur,DPman\_sur,DPfer\_sur)  
 TDPsub <- sum(DPsoi\_sub,DPman\_sub,DPfer\_sub)  
   
 ################################PP计算  
 A <- f.usle(dem,sand,silt,clay,oc,luFile,Rainfile,Boundfile,crs.geo,n,mask)  
   
 PER <- calc(A,fun=function(x){exp( 2 - 0.16 \* log(1000 \* x))})   
   
 SDR <-f.SDR(Runoff$RDsur[n])  
   
 Aoutlet <- A\*SDR###单位ton/ha  
   
 PPsoi <- overlay(Aoutlet,TPsoi,PER,fun=function(x,y,z){return(x\*y\*z\*0.001)})####Soil产生的PP,单位kg/ha  
   
 PPman <- TPman[[n]]\*(1-param$Perc.WSP)\*param$Ploss.man\*SDR  
   
 TPP <- sum(PPsoi, PPman)  
   
 TP <- sum(TDPsur,TDPsub,TPP)  
   
 f.summary <- function (x){return(cellStats(x,"mean",na.rm=T))}  
   
 plot.ls <- list (DPsoi\_sur,DPsoi\_sub,DPman\_sur,DPman\_sub,DPfer\_sur,DPfer\_sub,  
 TDPsur,TDPsub,A,Aoutlet,PPsoi,PPman,TPP,TP)  
   
 Ploss <- sapply(plot.ls,FUN=f.summary)  
   
 return(list(plot.ls,Ploss))  
   
}  
  
### 以2005年为例，模拟当年的非点源磷负荷  
result <- f.Pind(year=2005,param,Runoff,TPfer,TPman)

## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]

Ploss <- as.matrix(round(result[[2]],digits=2)) ###流域平均非点源磷流失量(kg/ha)  
  
  
rownames(Ploss) <- c("DPsoi\_sur","DPsoi\_sub","DPman\_sur","DPman\_sub","DPfer\_sur","DPfer\_sub",  
 "TDPsur","TDPsub","A","Aoutlet","PPsoi","PPman","TPP","TP")  
  
Ploss

## [,1]  
## DPsoi\_sur 0.02  
## DPsoi\_sub 0.02  
## DPman\_sur 0.06  
## DPman\_sub 0.23  
## DPfer\_sur 0.09  
## DPfer\_sub 0.31  
## TDPsur 0.17  
## TDPsub 0.54  
## A 6.00  
## Aoutlet 0.23  
## PPsoi 0.19  
## PPman 0.07  
## TPP 0.26  
## TP 0.98

###磷指数计算的流域非点源磷流失量(kg/ha)及其组分的空间分布图  
plot(stack(result[[1]]),main=rownames(Ploss),cex.main=0.8)



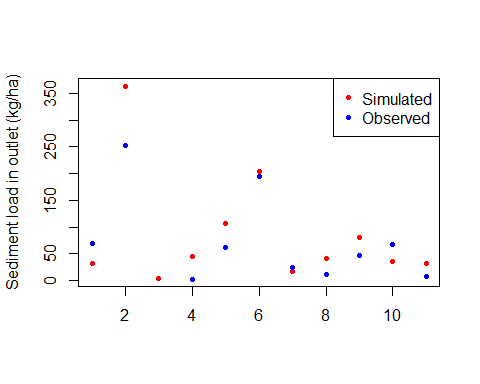
## 磷指数验证

这里采用Nash系数判断磷指数模拟的效果。

#### 利用磷指数模拟2000-2010年间的非点源磷污染负荷。  
  
A.sim <- vector()  
P.sim <- vector()  
  
for (i in 1:length(Year))  
{   
 result <- f.Pind(year=Year[i],param,Runoff,TPfer,TPman)  
 A.sim[i] <- result[[2]][10]   
 P.sim[i] <- result[[2]][14]   
}

## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## D:\SpatialR\Paper2\landsue.tif has GDAL driver GTiff   
## and has 1021 rows and 1062 columns  
## OGR data source with driver: ESRI Shapefile   
## Source: "D:\SpatialR\Paper2\boundLiu.shp", layer: "boundLiu"  
## with 1 features  
## It has 3 fields  
## [inverse distance weighted interpolation]

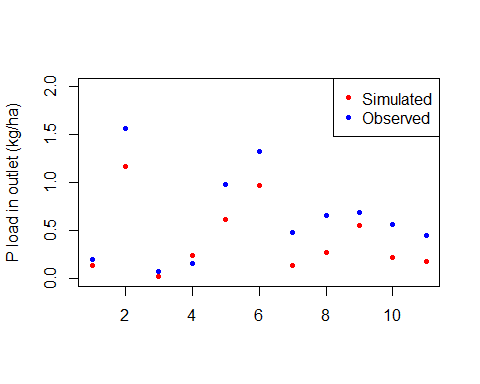
########## Sediment模拟效果  
  
### 观测值  
A.obs <- c(6.9, 25.3,NA, 0.2,6.3,19.4, 2.5,1.2,4.8,6.7,0.8) ### ton/km2  
A.obs <- A.obs/100### ton/ha  
  
### 模拟值与观测值的比较图  
plot(A.sim\*1000,pch=20,col="red",ylab="Sediment load in outlet (kg/ha)",xlab="")  
points(A.obs\*1000,pch=20,col="blue")  
legend("topright",c("Simulated","Observed"),pch=c(20,20),col=c("red","blue"))



### NASH系数  
NSH.f<-function(x,y){1-sum((x[!is.na(y)]-y[!is.na(y)])^2)/sum((y[!is.na(y)]-mean(y[!is.na(y)]))^2)}  
NSH.f(A.sim,A.obs)

## [1] 0.6678089

######### TP模拟效果  
### 观测值  
P.obs <- c(0.193,1.558,0.070,0.158,0.982,1.321,0.480,0.660,0.682,0.561,0.451)  
  
### 模拟值与观测值的比较图  
plot(P.sim,ylim=c(0,2),pch=20,col="red",ylab="P load in outlet (kg/ha)",xlab="")  
points(P.obs,pch=20,col="blue")  
legend("topright",c("Simulated","Observed"),pch=c(20,20),col=c("red","blue"))



NSH.f(P.sim,P.obs)

## [1] 0.5987813