Package 'genomic.autocorr'

October 13, 2022

Title Models Dealing with Spatial Dependency in Genomic Data

Version 1.0-1

Description Local structure in genomic data often induces dependence between observations taken at different genomic locations. Ignoring this dependence leads to underestimation of the standard error of parameter estimates. This package uses block bootstrapping to estimate asymptotically correct standard errors of parameters from any standard generalised linear model that may be fit by the glm() function.

Date 2017-10-17

Depends R (>= 3.2.2)

License GPL-2

LazyData true

RoxygenNote 6.0.1

Imports data.table, magrittr, reshape, parallel

Suggests testthat

URL https://github.com/chr1swallace/genomic.autocorr

NeedsCompilation no

Author Chris Wallace [aut, cre],
Oliver Burren [aut]

Maintainer Chris Wallace <cew54@cam.ac.uk>

Repository CRAN

Date/Publication 2017-10-20 09:17:08 UTC

R topics documented:

.sim.data *internal function to simulate data for examples*

Description

internal function to simulate data for examples

Usage

```
.sim.data(n = 500, m = 10, beta = 0.2)
```

Arguments

n	number of independent observations
m	group size - number of times each independent observation is repeated
beta	Y1 ~ N(beta * X, 1)

Value

data.table with Chr (always 1, possibly needed for bootstrap), x (explanatory variable), y1 (response variable related to x), y0 (response variable unrelated to x) name (unique name for each independent observation)

Author(s)

Chris Wallace

acf.summary *acf.summary*

Description

summarize the autocorrelation in

Usage

```
acf.summary(data, variables, order.by = NULL, lag.max = 100)
```

Arguments

data	data.table containing variables named in ‘variables‘ and ‘order.by‘
variables	character vector listing columns of ‘data‘ to be explored for autocorrelation
order.by	optionally, order ‘data‘ by variables in character vector ‘order.by‘
lag.max	maximum block size to explore (default=100)

Examples

```
## simulate data with 10 repeated observations in a row - ie there
## should be autocorrelation only within windows <= 10
library(data.table)
data <- genomic.autocorr:::.sim.data()
summ <- acf.summary(data,c("x","y0","y1"),lag.max=20)

## plot it
df <- melt(summ,c("lag","variable"),variable.name="acf")
par(mfrow=c(2,1))
matplot(matrix(df[acf=="full",]$value,ncol=3),
        main="full",
        pch=c("x","o","+"),
        type="b")
abline(h=0,lty=2)
legend("bottomright",
       c("x","y0","y1"),
       pch = "xo+", col = 1:3)
matplot(matrix(df[acf=="partial",]$value,ncol=3),
        main="partial",
        pch=c("x","o","+"),
        type="b")
abline(h=0,lty=2)
legend("bottomright",
       c("x","y0","y1"),
       pch = "xo+", col = 1:3)
```

`block.glm` *block.glm*

Description

Regression models for genomic data often assume there is independence between neighbouring genomic elements when, in reality, there is spatial dependence. This function implements a block bootstrap method for estimating correct variances of parameter estimates.

Usage

```
block.glm(f.lhs, f.rhs, data, order.by = NULL, strat.by = NULL,
  block.size = 20, B = 200, ...)
```

Arguments

`f.lhs`	character vector, left hand side of a formula, the model(s) to be fit will be defined by ‘paste(f.lhs, f.rhs, sep=" ~ ")‘
`f.rhs`	character string, right hand side of a formula
`data`	data.table containing the columns referred to in f.lhs and f.rhs
`order.by`	if not ‘NULL‘, the name of a column in ‘data‘ on which it should be sorted

strat.by	if not 'NULL', the name of a column in 'data' on which it should be stratified before block sampling. Eg, if you are considering genomic data, you should stratify by chromosome as there should be no spatial correlation between chromosomes
block.size	size of blocks of contiguous observations that will be sampled for bootstrap estimation of variance of parameter estimates
B	number of bootstrap estimates
...	other arguments passed to 'glm()' (eg 'family="binomial"')

Details

Note that this function uses 'mclapply' to parallelise the bootstrapping. Please set 'mc.cores' to something sensible, eg `options(mc.cores=10)` if you have 10 cores.

Value

data.table giving the estimated effect ("beta") of each item in f.rhs on each item in f.lhs, together with block bootstrap estimates of confidence interval (beta.025, beta.975) and standard error (se.beta) and the number of bootstraps on which those estimates are based.

Author(s)

Chris Wallace and Oliver Burren

Examples

```
## simulate data with 10 repeated observations in a row - ie there
## should be autocorrelation only within windows <= 10
library(data.table)
data <- genomic.autocorr:::.sim.data(beta=0.2)

## suppose we ignored the autocorrelation and look at the
## confidence interval for the effect of x on y1
r1<-summary(glm(y1 ~ x, data=data))$coefficients
r1

## if we know the block structure, as here, we can see the
## confidence interval is (inappropriately) much tighter than
## if we used just independent observations
r2<-summary(glm(y1 ~ x, data=data[!duplicated(name),]))$coefficients
r2

## use block bootstrap - x should only have a significant effect
## on y1 and the confidence interval around its effect should be
## closer to r2, above
r <- block.glm(f.lhs=c("y0","y1"), f.rhs="x",data=data,block.size=20,B=200)
r

## compare the block bootstrap and model based confidence intervals for x on y1
results <- rbind(c(r1[2,1], r1[2,1]-1.96*r1[2,2], r1[2,1]+1.96*r1[2,2]),
```

```
c(r2[2,1], r2[2,1]-1.96*r2[2,2], r2[2,1]+1.96*r2[2,2]),
as.numeric(r[4,.(beta,beta.025,beta.975)]))
dimnames(results) <- list(c("standard, ignore blocked","standard, independent obs","bootstrap"),
c("beta","LCI","UCI"))
results

with(as.data.frame(results), {
plot(1:nrow(results), beta,ylim=c(min(c(-0.01,LCI)),max(UCI)),axes=FALSE,xlab="Method",
main="Comparison of confidence intervals around coefficient estimates")
segments(x0=1:nrow(results),y0=LCI,y1=UCI)
abline(h=c(0,0.2),lty="dotted")
axis(1,1:nrow(results),rownames(results))
axis(2)
text(x=c(3,3),y=c(0,0.2),labels=c("null","true"),adj=c(1.1,0))
box()
})
```

Index