Package 'ParallLogicReg'

May 7, 2011

Title Logic regression using parallel computing

Version 1.0-0

Author Unitsa Sangket

Description Support for parallel LogicReg package on R.

Maintainer Unitsa Sangket <usangket@yahoo.com>

License GPL

Depends R (>= 2.8), utils

R topics documented:

`ParallLogicReg` *Logic regression analyses using parallel computing*

Description

This function can be used in many situations, especially identification of SNP interactions

Usage

```
output = ParallLogicReg(infile,resp,begin,end,nperm,niter)
```

Arguments

`infile`	A file name that contains data, which will be analyzed.
`resp`	vector with the response variables. See more detail in LogicReg manual.
`begin`	The first gene id will be run.
`end`	The last gene id will be run.
`nperm`	number of permutations
`niter`	number of iterations

Details

An example of infile:

snp	chr	pos	gene-symbol	gene-id	V1	V2	V3	V4	V5
rs1	1	12691937	AADACL3	1	1	1	1	1	1
rs1	1	12691937	AADACL3	1	1	1	1	1	1
rs31	1	12717433	AADACL3	1	1	0	1	1	1
rs31	1	12717433	AADACL3	1	1	1	1	1	1
rs2	1	12668821	AADACL4	2	0	1	1	1	1
rs2	1	12668821	AADACL4	2	1	1	1	1	1
rs58	1	12633603	AADACL4	2	0	0	0	0	0
rs58	1	12633603	AADACL4	2	0	0	1	0	0
rs59	1	12651828	AADACL4	2	1	1	1	1	1

where,

snp = snp id
chr = chromosome number
pos = position of the snp on chromosome
gene-symbol = gene symbol or gene name
gene-id = gene id
V1,V2,V3,V4,V5,... = snp of control or case

Note: One snp has two data rows, and each field must be separated by tab.

Author(s)

Unitsa Sangket and Qi Liu

Examples

```
#--------------------------------------------------------------------------
#An example run on Sun Grid Engine.
#You have to create two files, which are a ".sh" file, and a ".R" file.
#for more information please visit http://math.acadiau.ca/ACMMaC/Rmpi/submitting.html
#--------------------------------------------------------------------------

#File 1 (R_script.sh):
##########################################################################
#!/bin/bash

# Run in the current directory
#$ -cwd

#$ -j y

#$ -V

# Run using bash
#$ -S /bin/bash

# Set the number of processors
# For example, 2 means one processor is master and slave1,
# and the rest of processors are slaves.
#$ -pe lam 2

# Run the job.
# lamrun -np 1 R  --slave CMD BATCH R_script.R R_script_sh.Rout
```

```
###########################################################################
#!!! remove "#" out from "lamrun" command when run the file

#File 2 (R_script.R):
###########################################################################
library(ParallLogicReg)

resp=c(rep(0,2935),rep(1,1745)) # number of controls = 2,935; number of cases = 1,745

nperm=20                        # number of permutations
niter=20                                # number of iterations
begin=1                         # the first gene id that will be run
end=10                          # the last gene id that will be run
in=("input.txt")        # data will be run

output = ParallLogicReg(infile=in,resp=resp,begin=begin,end=end,nperm=nperm,niter=niter)

mpi.quit(save="no")

##########################################################################

#--------------------------------------------------------------------------
#You can summit the job using this command:
#qsub R_script.sh

#You can see the progress on R_script_sh.Rout and progress files.
#--------------------------------------------------------------------------
```

Index