Documents for MagicQTL R package

MagicQTL is an R package to perform QTL mapping in Multi-parent Advanced Generation Inter-cross (MAGIC) populations under both the fixed model and the random model methodology. The program also include two conventional QTL mapping methods, interval mapping (IM) and composite interval mapping (CIM). Users only need to call one function, **magicScan**. This user instruction has two parts: (1) how to install **MagicQTL** package in your computer; (2) an example to show the workflow using the **MagicQTL** package.

1. Install magicQTL package

In the Unix or Linux platform,

Just type the following command, R CMD INSTALL MagicQTL_1.0.tar.gz

Then complete installing the MagicQTL package!

In the windows platform,

The first step, download the Rtools from R CRAN (https://www.r-project.org/), then install the Rtools. Notes that you should add the "c:\program files\Rtools\bin", "c:\program files\Rtools\gcc-4.6.3\bin", "c:\program files\R\R.3.x.x\bin\i386" and "c:\program files\R\R.3.x.x\bin\x64" into the Path Variable on the Environment Variables panel.

The second step, in the search box, type "command prompt", then click.

In the command prompt, type the following command R CMD INSTALL MagicQTL_1.0.tar.gz.

Then install!

To use this package, Just type library(MagicQTL) and call the function magicScan()

2. Introduction of implementing the MagicQTL

Here we provide a test example to briefly introduce how to implement the MagicQTL package. Details can be obtained via help(magicScan) or ?magicScan.

The original data is *Arabidopsis thaliana* MAGIC population inherited from 19 founders obtained from the website (http://mus.well.ox.ac.uk/magic/). In consideration of file size, the test data is a subset, which is comprised of 65 markers distributed in the five chromosomes, 60 individuals with five traits. We can offer the original data applying to our program format if requested.

Demo code

First step-load data

```
library(MagicQTL)
> data(Ara)
> names(Ara)
[1] "gen"     "map"     "Ara.phe" "kk.eigen"
> gen<-Ara[[1]]
> map<-Ara[[2]]
> Ara.phe<-Ara[[3]]
> kk.eigen<-Ara[[4]]
> chrnum<-length(gen)
```

Data format Information

```
#gen, probability matrix
> dim(gen[[1]])
[1] 266  60
> class(gen[[1]])
[1] "matrix"
#map, marker information
> dim(map[[1]])
[1] 14  4
> class(map[[1]])
[1] "data.frame"
#Ara.phe, phenotype
> dim(Ara.phe)
[1] 60   6
#kk.eigen,including the kinship matrix, its eigendecomposition and
# the numeric
> names(kk.eigen)
[1] "kk" "qq" "cc"
```

#The probability matrix, like following

```
> gen[[1]][1:19,1:4]
             [,1]         [,2]         [,3]         [,4]
Bur   8.012807e-08 1.715535e-01 1.697059e-06 7.217872e-02
Can   1.718630e+00 3.708686e-08 6.840062e-02 1.108586e-07
Col   3.390267e-02 3.709196e-08 6.841290e-02 2.458504e-07
Ct    4.620957e-07 4.520535e-02 8.073149e-08 1.639100e+00
Edi   2.541389e-03 2.702748e-07 1.753971e-01 3.714008e-08
Hi    2.531023e-03 7.467149e-07 2.011564e-01 3.714148e-08
Kn    8.027694e-08 8.673264e-02 1.382932e-06 7.217847e-02
Ler   2.531022e-03 3.353259e-07 3.319853e-01 3.715066e-08
Mt    3.390267e-02 3.709195e-08 6.841290e-02 2.458504e-07
No    3.390267e-02 3.709196e-08 6.841290e-02 2.458504e-07
Oy    8.012806e-08 9.832535e-02 3.289830e-06 7.218036e-02
Po    8.012806e-08 9.832535e-02 3.289830e-06 7.218036e-02
Rsch  3.390267e-02 3.709195e-08 6.841290e-02 2.458504e-07
Sf    8.012807e-08 1.499856e+00 1.697417e-06 7.217872e-02
Tsu   2.531022e-03 3.353140e-07 6.757747e-01 3.715066e-08
Wil   3.390267e-02 3.709195e-08 6.841290e-02 2.458504e-07
Ws    3.390267e-02 3.709195e-08 6.840473e-02 1.197786e-06
Wu    3.391363e-02 3.712419e-08 6.840265e-02 1.287837e-07
Zu    3.390516e-02 3.716816e-08 6.840265e-02 1.286525e-07
> 
```

##The map format, like

```
> map[[1]][1:10,]
            markers chr       cm        bp
1         MN1_29291   1  0.12205     29291
2         MASC07424   1  6.26250   1502999
3       MN1_3229670   1 13.45770   3229846
4       MN1_4947324   1 20.61385   4947328
5           MFT_113   1 25.94783   6227484
6           GI_2186   1 33.59517   8062852
7   NMSNP1_10720273   1 44.66781  10720291
8       SGCSNP10165   1 55.00472  13201153
9       PERL0147872   1 72.81940  17474215
10      PERL0173191   1 85.47685  20510777
> 
```

##Ara.phe, the phenotype, including the five traits.

```
> Ara.phe[1:5,]
        ID bolt.to.flower days.to.bolt days.to.germ total.cm growth rate
1 MAGIC.10           10.0         40.0          6.0   37.060  -2.8677181
2 MAGIC.94           10.0         24.6          6.0   41.520   3.1500000
3 MAGIC.95            7.0         20.2          6.0   31.900  -2.8500000
4 MAGIC.96           10.0         36.0          6.0   48.200  -0.2500000
5 MAGIC.97           10.2         25.2          9.4   51.225  -0.8633877
> 
```

#Second step-scan the markers

```
> indi<-nrow(Ara.phe)

> x<-rep(1,indi)

> y<-Ara.phe[,5] # Phenotype,total length, that is height of the Arabdopsis

> d<-data.frame(y=y,x=x)

> scans<-
magicScan(dataframe=d,gen=gen,map=map,kk.eigen=kk.eigen,nfounders=19,
model="Random-A")

lambda: 1.125352e-07 Residual error: 70.79299 Model: Random-A

Data of chr have been completed 0

Data of chr have been completed 0

Data of chr have been completed 0

Data of chr have been completed 0

Data of chr have been completed 0
```

#output the result after scanning

```
#Output

> parms<-lapply(1:chrnum, function(i){ return(scans[[i]][[1]]) })

> parms<-do.call(rbind,parms)

> write.csv(parms,file="Ara.parm.csv",row.names=FALSE)

> #

> blupp<-lapply(1:chrnum, function(i){ return(scans[[i]][[2]]) })

> blupp<-do.call(rbind,blupp)

> write.csv(blupp,file="Ara.blupp.csv",row.names=FALSE)
```

#Output information

#parms format, like following

```
> parms[1:5,]
  Num chr       ccM         lrt      lrt.p  lrt.logp       wald     wald.p
1   1   1   0.12205 0.212660545 0.3223450 0.4916790 2.5934390 0.9999910
2   2   1   6.26250 0.734221490 0.1957591 0.7082780 5.5863602 0.9976058
3   3   1  13.45770 0.038718467 0.4220037 0.3746837 1.2147610 1.0000000
4   4   1  20.61385 0.002749201 0.4790919 0.3195812 0.3154737 1.0000000
5   5   1  25.94783 0.685325117 0.2038795 0.6906264 5.3563576 0.9981869
      wald.logp     tau_k    sigma2       lam_k conv
1 3.890128e-06 1.6091917 67.88415 0.023704971    0
2 1.041022e-03 2.0943954 64.58632 0.032427845    0
3 7.806182e-09 0.4470102 69.36368 0.006444441    0
4 6.277772e-14 0.1225779 70.41699 0.001740743    0
5 7.881258e-04 1.7812334 64.92988 0.027433185    0
```

#blupp format, like following

```
> blupp[1:2,]
        Gamma1     Gamma2       Gamma3     Gamma4      Gamma5    Gamma6      Gamma7     Gamma8
blup 0.8591198 -0.4682053 -0.022309143 0.05509254 -0.02871033 0.7041129 -0.8938757  0.1415811
blup 1.0667156 -1.5576318  0.007362878 0.24340898 -0.23955166 1.1250004 -1.1860768 -0.1792806
         Gamma9    Gamma10   Gamma11   Gamma12   Gamma13     Gamma14    Gamma15    Gamma16
blup -0.24231020 0.4035547 0.1130185 0.1132037 0.1534979  0.01137074 -0.4144860 -0.1187667
blup  0.04603282 0.7335408 0.1129925 0.1133913 0.4658266 -0.15829037 -0.2452165 -0.2201461
         Gamma17    Gamma18   Gamma19   stderr1   stderr2   stderr3   stderr4   stderr5   stderr6
blup 0.06213689 -0.6817075 0.2536821 1.213400 1.187375 1.202695 1.194544 1.265652 1.132944
blup 0.09724851 -0.7794576 0.5541311 1.367922 1.166058 1.207533 1.236783 1.397190 1.185177
      stderr7   stderr8   stderr9 stderr10 stderr11 stderr12 stderr13 stderr14 stderr15 stderr16
blup 0.9927385 1.127535 1.237929 1.233045 1.232292 1.232277 1.160970 1.122480 1.243798 1.238761
blup 1.1331874 1.101253 1.270078 1.330291 1.385856 1.385814 1.196191 1.138972 1.442503 1.416296
     stderr17 stderr18 stderr19
blup 1.189109 1.157115 1.211248
blup 1.287618 1.190256 1.305421
```