R/qtl & R/qtlbim Tutorials

- R statistical graphics & language system
- R/qtl tutorial
  - R/qtl web site: www.rqtl.org
  - Tutorial: www.rqtl.org/tutorials/rqtltour.pdf
  - R code: www.stat.wisc.edu/~yandell/qtlbim/rqtltour.R
  - url.show("http://www.stat.wisc.edu/~yandell/qtlbim/rqtltour.R")
- R/qtlbim tutorial
  - R/qtlbim web site: www.qtlbim.org
  - Tutorial and R code:
    - www.stat.wisc.edu/~yandell/qtlbim/rqtlbimtour.pdf
    - www.stat.wisc.edu/~yandell/qtlbim/rqtlbimtour.R
R/qtl tutorial (www.rqtl.org)

```r
> library(qtl)
> data(hyper)
> summary(hyper)

Backcross

No. individuals: 250
No. phenotypes: 2
Percent phenotyped: 100 100
No. chromosomes: 20
  Autosomes: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
  X chr: X
Total markers: 174
No. markers: 22 8 6 20 14 11 7 6 5 5 14 5 5 5 11 6 12 4 4 4
Percent genotyped: 47.7
Genotypes (%): AA:50.2 AB:49.8
```

> plot(hyper)
> plot.missing(hyper, reorder = TRUE)

Sysgen Tutorial Seattle SISG: Yandell © 2010
R/qtl: find genotyping errors

```r
> hyper <- calc.errorlod(hyper, error.prob=0.01)
> top.errorlod(hyper)

<table>
<thead>
<tr>
<th>chr</th>
<th>id</th>
<th>marker</th>
<th>errorlod</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>118</td>
<td>D1Mit14</td>
<td>8.372794</td>
</tr>
<tr>
<td>2</td>
<td>162</td>
<td>D1Mit14</td>
<td>8.372794</td>
</tr>
<tr>
<td>3</td>
<td>170</td>
<td>D1Mit14</td>
<td>8.372794</td>
</tr>
<tr>
<td>4</td>
<td>159</td>
<td>D1Mit14</td>
<td>8.350341</td>
</tr>
<tr>
<td>5</td>
<td>73</td>
<td>D1Mit14</td>
<td>6.165395</td>
</tr>
<tr>
<td>6</td>
<td>65</td>
<td>D1Mit14</td>
<td>6.165395</td>
</tr>
<tr>
<td>7</td>
<td>88</td>
<td>D1Mit14</td>
<td>6.165395</td>
</tr>
<tr>
<td>8</td>
<td>184</td>
<td>D1Mit14</td>
<td>6.151606</td>
</tr>
<tr>
<td>9</td>
<td>241</td>
<td>D1Mit14</td>
<td>6.151606</td>
</tr>
<tr>
<td>10</td>
<td>215</td>
<td>D1Mit267</td>
<td>5.822192</td>
</tr>
<tr>
<td>11</td>
<td>108</td>
<td>D1Mit267</td>
<td>5.822192</td>
</tr>
<tr>
<td>12</td>
<td>138</td>
<td>D1Mit267</td>
<td>5.822192</td>
</tr>
<tr>
<td>13</td>
<td>226</td>
<td>D1Mit267</td>
<td>5.822192</td>
</tr>
<tr>
<td>14</td>
<td>199</td>
<td>D1Mit267</td>
<td>5.819250</td>
</tr>
<tr>
<td>15</td>
<td>84</td>
<td>D1Mit267</td>
<td>5.808400</td>
</tr>
</tbody>
</table>
```

> plot.geno(hyper, chr=1, ind=c(117:119,137:139,157:184))

Sysgen Tutorial Seattle SISG: Yandell © 2010
R/qtl: 1 QTL interval mapping

> hyper <- calc.genoprob(hyper, step=1, error.prob=0.01)
> out.em <- scanone(hyper)
> out.hk <- scanone(hyper, method="hk")
> summary(out.em, threshold=3)
  chr  pos  lod
  c1.loc45   1 48.3 3.52
  D4Mit164   4 29.5 8.02

> summary(out.hk, threshold=3)
  chr  pos  lod
  c1.loc45   1 48.3 3.55
  D4Mit164   4 29.5 8.09

> plot(out.em, chr = c(1,4,6,15))
> plot(out.hk, chr = c(1,4,6,15), add = TRUE, lty = 2)
R/qtl: permutation threshold

> operm.hk <- scanone(hyper, method="hk",
  n.perm=1000)
Doing permutation in batch mode ...
> summary(operm.hk, alpha=c(0.01,0.05))
LOD thresholds (1000 permutations)
  lod
1% 3.79
5% 2.78

> summary(out.hk, perms=operm.hk, alpha=0.05,
pvalues=TRUE)
  chr pos lod pval
1 1 48.3 3.55 0.015
2 4 29.5 8.09 0.000
R/qtl: 2 QTL scan

```r
> hyper <- calc.genoprob(hyper, step=5, error.prob=0.01)
> out2.hk <- scantwo(hyper, method="hk")
--Running scanone
--Running scantwo
(1,1)
(1,2)
(19,19)
(19,X)
(X,X)
> summary(out2.hk, thresholds=c(6.0, 4.7, 4.4, 4.7, 2.6))
```

```r
c1 : c4   68.3 30.0 14.13 6.51 0.225      68.3 30.0 13.90 6.288
c2 : c19  47.7  0.0  6.71  5.01 3.458      52.7  0.0  3.25 1.552

c3 : c3   37.2 42.2  6.10  5.08 0.226      37.2 42.2  5.87 4.853

c6 : c15  60.0 20.5  7.17  5.22 3.237      25.0 20.5  3.93 1.984

c9 : c18  67.0 37.2  6.31  4.79 4.083      67.0 12.2  2.23 0.708

c12 : c19  1.1 40.0  6.48  4.79 4.090      1.1  0.0  2.39 0.697
```

```r
> plot(out2.hk, chr=c(1,4,6,15))
```
## Effect & Interaction Plots

### Effect plots and interaction plot.

```r
hyper <- sim.geno(hyper, step=2, n.draws=16, error.prob=0.01)
effectplot(hyper, pheno.col = 1, mname1 = "D1Mit334")
effectplot(hyper, pheno.col = 1, mname1 = "D4Mit164")
markers <- find.marker(hyper, chr = c(6,15), pos = c(70,20))
effectplot(hyper, pheno.col = 1, mname1 = markers[1], mname2 = markers[2])
effectplot(hyper, pheno.col = 1, mname1 = markers[2], mname2 = markers[1])
```

### Strip plot of data (phenotype by genotype).

```r
plot.pwg(hyper, "D1Mit334")
plot.pwg(hyper, "D4Mit164")
plot.pwg(hyper, markers)
```
R/qtl: ANOVA imputation at QTL

```r
> hyper <- sim.geno(hyper, step=2, n.draws=16, error.prob=0.01)
> qtl <- makeqtl(hyper, chr = c(1, 1, 4, 6, 15), pos = c(50, 76, 30, 70, 20))
> my.formula <- y ~ Q1 + Q2 + Q3 + Q4 + Q5 + Q4:Q5
> out.fitqtl <- fitqtl(hyper, pheno.col = 1, qtl, formula = my.formula)
> summary(out.fitqtl)
```

### Full model result

```
---
Model formula is:  y ~ Q1 + Q2 + Q3 + Q4 + Q5 + Q4:Q5

<table>
<thead>
<tr>
<th></th>
<th>df</th>
<th>SS</th>
<th>MS</th>
<th>LOD</th>
<th>%var</th>
<th>Pvalue(Chi2)</th>
<th>Pvalue(F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>6</td>
<td>5789.089</td>
<td>964.84822</td>
<td>21.54994</td>
<td>32.76422</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Error</td>
<td>243</td>
<td>11879.847</td>
<td>48.88826</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>249</td>
<td>17668.936</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Drop one QTL at a time ANOVA table:

```
---
<table>
<thead>
<tr>
<th>Chr &amp; pos</th>
<th>df</th>
<th>Type III SS</th>
<th>LOD</th>
<th>%var</th>
<th>F value</th>
<th>Pvalue(F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chr1@50</td>
<td>1</td>
<td>297.149</td>
<td>1.341</td>
<td>1.682</td>
<td>6.078</td>
<td>0.01438 *</td>
</tr>
<tr>
<td>Chr1@76</td>
<td>1</td>
<td>520.664</td>
<td>2.329</td>
<td>2.947</td>
<td>10.650</td>
<td>0.00126 **</td>
</tr>
<tr>
<td>Chr4@30</td>
<td>1</td>
<td>2842.089</td>
<td>11.644</td>
<td>16.085</td>
<td>58.134</td>
<td>5.50e-13 ***</td>
</tr>
<tr>
<td>Chr6@70</td>
<td>2</td>
<td>1435.721</td>
<td>6.194</td>
<td>8.126</td>
<td>14.684</td>
<td>9.55e-07 ***</td>
</tr>
<tr>
<td>Chr15@20</td>
<td>2</td>
<td>1083.842</td>
<td>4.740</td>
<td>6.134</td>
<td>11.085</td>
<td>2.47e-05 ***</td>
</tr>
<tr>
<td>Chr670:Chr1520</td>
<td>1</td>
<td>955.268</td>
<td>4.199</td>
<td>5.406</td>
<td>19.540</td>
<td>1.49e-05 ***</td>
</tr>
</tbody>
</table>
```

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Sysgen Tutorial Seattle SISG: Yandell © 2010

---

selected R/qtl publications

www.stat.wisc.edu/~yandell/statgen

- www.rqtl.org
- tutorials and code at web site
  - www.rqtl.org/tutorials
- Broman et al. (2003 *Bioinformatics*)
  - R/qtl introduction
- Broman (2001 *Lab Animal*)
  - nice overview of QTL issues
- Broman & Sen 2009 book (*Springer*)
R/qtlbim (www.qtlbim.org)

- cross-compatible with R/qtl
- model selection for genetic architecture
  - epistasis, fixed & random covariates, GxE
  - samples multiple genetic architectures
  - examines summaries over nested models
- extensive graphics

```r
> url.show("http://www.stat.wisc.edu/~yandell/qtlbim/rqtlbimtour.R")
```

R/qtlbim: tutorial
(www.stat.wisc.edu/~yandell/qtlbim)

```r
> data(hyper)
## Drop X chromosome (for now).
> hyper <- subset(hyper, chr=1:19)
> hyper <- qb.genoprob(hyper, step=2)
## This is the time-consuming step:
> qbHyper <- qb.mcmc(hyper, pheno.col = 1)
## Here we get stored samples.
> data(qbHyper)
> summary(qbHyper)
```
R/qtlbim: initial summaries

```r
> summary(qbHyper)

Bayesian model selection QTL mapping object qbHyper on cross object hyper
had 3000 iterations recorded at each 40 steps with 1200 burn-in steps.

Diagnostic summaries:

<table>
<thead>
<tr>
<th></th>
<th>nqtl</th>
<th>mean</th>
<th>envvar</th>
<th>varadd</th>
<th>varaa</th>
<th>var</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>2.000</td>
<td>97.42</td>
<td>28.07</td>
<td>5.112</td>
<td>0.000</td>
<td>5.112</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>5.000</td>
<td>101.00</td>
<td>44.33</td>
<td>17.010</td>
<td>1.639</td>
<td>20.180</td>
</tr>
<tr>
<td>Median</td>
<td>7.000</td>
<td>101.30</td>
<td>48.57</td>
<td>20.060</td>
<td>4.580</td>
<td>25.160</td>
</tr>
<tr>
<td>Mean</td>
<td>6.543</td>
<td>101.30</td>
<td>48.80</td>
<td>20.310</td>
<td>5.321</td>
<td>25.630</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>8.000</td>
<td>103.70</td>
<td>53.11</td>
<td>23.480</td>
<td>7.862</td>
<td>30.370</td>
</tr>
<tr>
<td>Max.</td>
<td>13.000</td>
<td>103.90</td>
<td>74.03</td>
<td>51.730</td>
<td>34.940</td>
<td>65.220</td>
</tr>
</tbody>
</table>

Percentages for number of QTL detected:
2 3 4 5 6 7 8 9 10 11 12 13
2 3 9 14 21 19 17 10  4  1  0  0

Percentages for number of epistatic pairs detected:

<table>
<thead>
<tr>
<th>pairs</th>
<th>6 3 2 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>123456</td>
<td>29 31 23 11 5 1</td>
</tr>
</tbody>
</table>

Percentages for common epistatic pairs:

<table>
<thead>
<tr>
<th></th>
<th>6.15 4.15 4.6 1.7 15.15 1.4 1.6 4.9 1.15 1.17 1.5 5.11 1.2 7.15 1.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>63 18 10 6 6 5 4 4 3 3 3 3 2 2 2 2</td>
<td></td>
</tr>
</tbody>
</table>

> plot(qb.diag(qbHyper, items = c("herit", "envvar")))
```

Sysgen Tutorial Seattle SISG: Yandell © 2010

23

diagnostic summaries
R/qtlbim: 1-D (not 1-QTL!) scan

> one <- qb.scanone(qbHyper, chr = c(1,4,6,15), type = "LPD")
> summary(one)

LPD of bp for main, epistasis, sum

<table>
<thead>
<tr>
<th>n.qtl</th>
<th>pos</th>
<th>m.pos</th>
<th>e.pos</th>
<th>main</th>
<th>epistasis</th>
<th>sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>1.331</td>
<td>64.5</td>
<td>64.5</td>
<td>67.8</td>
<td>6.10</td>
<td>0.442</td>
</tr>
<tr>
<td>c4</td>
<td>1.377</td>
<td>29.5</td>
<td>29.5</td>
<td>29.5</td>
<td>11.49</td>
<td>0.375</td>
</tr>
<tr>
<td>c6</td>
<td>0.838</td>
<td>59.0</td>
<td>59.0</td>
<td>59.0</td>
<td>3.99</td>
<td>6.265</td>
</tr>
<tr>
<td>c15</td>
<td>0.961</td>
<td>17.5</td>
<td>17.5</td>
<td>17.5</td>
<td>1.30</td>
<td>6.325</td>
</tr>
</tbody>
</table>

> plot(one, scan = "main")
> plot(out.em, chr=c(1,4,6,15), add = TRUE, lty = 2)
> plot(one, scan = "epistasis")
most probable patterns

> summary(qb.BayesFactor(qbHyper, item = "pattern"))

```
   nqtl posterior  prior   bf  bfse
1,4,6,15,6:15    5  0.03400 2.71e-05 24.30 2.360
1,4,6,15,6:15    6  0.00467 5.22e-06 17.40 4.630
1,1,4,6,15,6:15  6  0.00600 9.05e-06 12.80 3.020
1,1,4,5,6,15,6:15 7  0.00267 4.11e-06 12.60 4.450
1,4,6,15,6:15    6  0.00300 4.96e-06 11.70 3.910
1,4,6,15,6:15    6  0.00300 5.81e-06 10.00 3.330
1,2,4,6,15,6:15  6  0.00767 1.54e-05  9.66 2.010
1,4,5,6,15,6:15  6  0.00500 1.28e-05  7.56 1.950
1,2,4,5,6,15,6:15 7  0.00267 6.98e-06  7.41 2.620
1,4                  2  0.01430 1.51e-04  1.84 0.279
1,1,2,4              4  0.00300 3.66e-05  1.59 0.529
1,2,4                3  0.00733 1.03e-04  1.38 0.294
1,1,4                3  0.00400 6.05e-05  1.28 0.370
1,4,19               3  0.00300 5.82e-05  1.00 0.333
```

> plot(qb.BayesFactor(qbHyper, item = "nqtl"))

hyper: number of QTL
posterior, prior, Bayes factors

```
Prior strength of evidence:

- weak
- moderate
- strong

MCMC error

Bayes factor ratios:

- weak
- moderate
- strong

QTL posterior

prior

QTL posterior

prior

number of QTL

number of QTL

posterior / prior

strength of evidence

number of QTL
```

Sysgen Tutorial Seattle SISG: Yandell © 2010 28
what is best estimate of QTL?

- find most probable pattern
  - 1,4,6,15,6:15 has posterior of 3.4%
- estimate locus across all nested patterns
  - Exact pattern seen ~100/3000 samples
  - Nested pattern seen ~2000/3000 samples
- estimate 95% confidence interval using quantiles

```r
> best <- qb.best(qbHyper)
> summary(best)$best

<table>
<thead>
<tr>
<th>chrom</th>
<th>locus</th>
<th>locus.LCL</th>
<th>locus.UCL</th>
<th>n.qtl</th>
</tr>
</thead>
<tbody>
<tr>
<td>247</td>
<td>1</td>
<td>69.9</td>
<td>24.44875</td>
<td>95.7985</td>
</tr>
<tr>
<td>245</td>
<td>4</td>
<td>29.5</td>
<td>14.20000</td>
<td>74.3000</td>
</tr>
<tr>
<td>248</td>
<td>6</td>
<td>59.0</td>
<td>13.83333</td>
<td>66.7000</td>
</tr>
<tr>
<td>246</td>
<td>15</td>
<td>19.5</td>
<td>13.10000</td>
<td>55.7000</td>
</tr>
</tbody>
</table>
```

> plot(best)

what patterns are “near” the best?

- size & shade ~ posterior
- distance between patterns
  - sum of squared attenuation
  - match loci between patterns
  - squared attenuation = \((1-2r)^2\)
  - sq.atten in scale of LOD & LPD
- multidimensional scaling
  - MDS projects distance onto 2-D
  - think mileage between cities
how close are other patterns?

```r
> target <- qb.best(qbHyper)$model[[1]]
> summary(qb.close(qbHyper, target))

score by sample number of qtl

<table>
<thead>
<tr>
<th>score</th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.437</td>
<td>1.735</td>
<td>1.919</td>
<td>1.834</td>
<td>1.919</td>
<td>2.000</td>
</tr>
<tr>
<td>3</td>
<td>1.352</td>
<td>1.735</td>
<td>1.916</td>
<td>1.900</td>
<td>1.919</td>
<td>2.016</td>
</tr>
<tr>
<td>4</td>
<td>1.270</td>
<td>1.916</td>
<td>2.437</td>
<td>2.648</td>
<td>3.574</td>
<td>4.000</td>
</tr>
<tr>
<td>5</td>
<td>1.295</td>
<td>1.919</td>
<td>2.835</td>
<td>2.798</td>
<td>3.611</td>
<td>4.000</td>
</tr>
<tr>
<td>6</td>
<td>1.257</td>
<td>2.254</td>
<td>3.451</td>
<td>3.029</td>
<td>3.648</td>
<td>4.000</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>3.694</td>
<td>3.694</td>
<td>3.694</td>
<td>3.694</td>
<td>3.694</td>
<td>3.694</td>
</tr>
</tbody>
</table>

score by sample chromosome pattern

<table>
<thead>
<tr>
<th>chromosome</th>
<th>score</th>
<th>Percent</th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>4@1,4,6,15,6:15</td>
<td>3.4</td>
<td>2.946</td>
<td>3.500</td>
<td>3.630</td>
<td>3.613</td>
<td>3.758</td>
<td>4.000</td>
<td></td>
</tr>
<tr>
<td>281,4</td>
<td>1.4</td>
<td>1.437</td>
<td>1.735</td>
<td>1.919</td>
<td>1.832</td>
<td>1.919</td>
<td>2.000</td>
<td></td>
</tr>
<tr>
<td>581,2,4,6,15,6:15</td>
<td>0.8</td>
<td>3.137</td>
<td>3.536</td>
<td>3.622</td>
<td>3.611</td>
<td>3.777</td>
<td>3.923</td>
<td></td>
</tr>
<tr>
<td>381,2,4</td>
<td>0.7</td>
<td>1.352</td>
<td>1.700</td>
<td>1.821</td>
<td>1.808</td>
<td>1.919</td>
<td>2.000</td>
<td></td>
</tr>
<tr>
<td>581,1,4,6,15,6:15</td>
<td>0.6</td>
<td>3.257</td>
<td>3.484</td>
<td>3.563</td>
<td>3.575</td>
<td>3.698</td>
<td>3.916</td>
<td></td>
</tr>
<tr>
<td>581,4,5,6,15,6:15</td>
<td>0.5</td>
<td>3.237</td>
<td>3.515</td>
<td>3.595</td>
<td>3.622</td>
<td>3.777</td>
<td>3.923</td>
<td></td>
</tr>
<tr>
<td>581,4,6,6,15,6:15</td>
<td>0.5</td>
<td>3.203</td>
<td>3.541</td>
<td>3.646</td>
<td>3.631</td>
<td>3.757</td>
<td>3.835</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

> plot(close)
> plot(close, category = "nqtl")
```

Sysgen Tutorial Seattle SISG: Yandell © 2010 32
R/qtlbim: automated QTL selection

```r
> hpd <- qb.hpdone(qbHyper, profile = "2logBF")
> summary(hpd)

<table>
<thead>
<tr>
<th>chr</th>
<th>n.qtl</th>
<th>pos</th>
<th>lo.50%</th>
<th>hi.50%</th>
<th>2logBF</th>
<th>A</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.829</td>
<td>64.5</td>
<td>72.1</td>
<td>6.692</td>
<td>103.611</td>
<td>99.090</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3.228</td>
<td>29.5</td>
<td>31.7</td>
<td>11.169</td>
<td>104.584</td>
<td>98.020</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1.033</td>
<td>59.0</td>
<td>66.7</td>
<td>6.054</td>
<td>99.637</td>
<td>102.965</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>0.159</td>
<td>17.5</td>
<td>17.5</td>
<td>5.837</td>
<td>101.972</td>
<td>100.702</td>
</tr>
</tbody>
</table>

> plot(hpd)
```

2log(BF) scan with 50% HPD region
R/qtlbim: 2-D (*not* 2-QTL) scans

```r
> two <- qb.scantwo(qbHyper, chr = c(6,15),
                    type = "2logBF")
> plot(two)

> plot(two, chr = 6, slice = 15)
> plot(two, chr = 15, slice = 6)

> two.lpd <- qb.scantwo(qbHyper, chr = c(6,15),
                       type = "LPD")
> plot(two.lpd, chr = 6, slice = 15)
> plot(two.lpd, chr = 15, slice = 6)
```

2-D plot of 2logBF: chr 6 & 15
1-D Slices of 2-D scans: chr 6 & 15

R/qtlbim: slice of epistasis

```r
> slice <- qb.slicetwo(qbHyper, c(6,15), c(59,19.5))
> summary(slice)

2logBF of bp for epistasis

<table>
<thead>
<tr>
<th>n.qtl</th>
<th>pos</th>
<th>m.pos</th>
<th>e.pos</th>
<th>epistasis</th>
<th>slice</th>
</tr>
</thead>
<tbody>
<tr>
<td>c6</td>
<td>0.838</td>
<td>59.0</td>
<td>59.0</td>
<td>66.7</td>
<td>15.8</td>
</tr>
<tr>
<td>c15</td>
<td>0.961</td>
<td>17.5</td>
<td>17.5</td>
<td>17.5</td>
<td>15.5</td>
</tr>
</tbody>
</table>

cellmean of bp for AA, HA, AH, HH

<table>
<thead>
<tr>
<th>n.qtl</th>
<th>pos</th>
<th>m.pos</th>
<th>AA</th>
<th>HA</th>
<th>AH</th>
<th>HH</th>
<th>slice</th>
</tr>
</thead>
<tbody>
<tr>
<td>c6</td>
<td>0.838</td>
<td>59.0</td>
<td>97.4</td>
<td>105</td>
<td>102</td>
<td>100.8</td>
<td>18.1</td>
</tr>
<tr>
<td>c15</td>
<td>0.961</td>
<td>17.5</td>
<td>99.8</td>
<td>103</td>
<td>104</td>
<td>98.5</td>
<td>60.6</td>
</tr>
</tbody>
</table>

estimate of bp for epistasis

<table>
<thead>
<tr>
<th>n.qtl</th>
<th>pos</th>
<th>m.pos</th>
<th>e.pos</th>
<th>epistasis</th>
<th>slice</th>
</tr>
</thead>
<tbody>
<tr>
<td>c6</td>
<td>0.838</td>
<td>59.0</td>
<td>66.7</td>
<td>-7.86</td>
<td>18.1</td>
</tr>
<tr>
<td>c15</td>
<td>0.961</td>
<td>17.5</td>
<td>17.5</td>
<td>-8.72</td>
<td>60.6</td>
</tr>
</tbody>
</table>

> plot(slice, figs = c("effects", "cellmean", "effectplot"))
```
selected publications
www.stat.wisc.edu/~yandell/statgen

- www.qtlbim.org
- vignettes in R/qtlbim package
  - overview/comparison of QTL methods
- Yandell et al. (2007 *Bioinformatics*)
  - R/qtlbim introduction
- Yi et al. (2005 *Genetics, 2007 Genetics*)
  - methodology of R/qtlbim
Simulate a “null” cross

Start by simulating a “null backcross” composed of 1,000 phenotypes, 204 genetic markers equally spaced across 4 chr, and 100 ind. The latent.eff parameter controls the amount of correlation among the phenotypes.

```r
> library(qtlhot)
> ncross1 <- sim.null.cross(chr.len = rep(100, 4),
+ n.mar = 51,
+ n.ind = 100,
+ type = "bc",
+ n.pheno = 1000,
+ latent.eff = 3,
+ res.var = 1,
+ init.seed = 123457)
```
Include hotspots into null cross

The function `include.hotspots` takes the "null cross" as an input and includes 3 hotspots of size `hsize` at position `hpos` of chromosome `hchr` into it.

```r
> cross1 <- include.hotspots(cross = ncross1,
+   hchr = c(2, 3, 4),
+   hpos = c(25, 75, 50),
+   hsize = c(100, 50, 20),
+   Q.eff = 2,
+   latent.eff = 3,
+   lod.range.1 = c(2.5, 2.5),
+   lod.range.2 = c(5, 8),
+   lod.range.3 = c(10, 15),
+   res.var = 1,
+   nT = 1000,
+   init.seed = 12345)
```

Check correlation among phenotypes

By choosing `latent.eff` we generate highly correlated phenotype data.

```r
> nphe1 <- as.matrix(cross1$pheno)
> ncor1 <- cor(nphe1)
> ncor1 <- ncor1[lower.tri(ncor1)]
> summary(ncor1)
    Min. 1st Qu. Median  Mean 3rd Qu.    Max.
 0.4145  0.8517  0.8929  0.8649  0.9063  0.9691
```
Single trait QTL mapping permutation threshold
Obtain permutation thresholds for the sequence alphas of GWER levels.

```r
> set.seed(123)
> pt <- scanone(ncross1, method = "hk", n.perm = 1000)
> alphas <- seq(0.01, 0.10, by=0.01)
> spt <- summary(pt, alphas)
> spt

LOD thresholds (1000 permutations)
  lod
1% 3.11
2% 2.89
3% 2.68
4% 2.57
5% 2.44
6% 2.34
7% 2.26
8% 2.20
9% 2.15
10% 2.11
> lod.thrs <- as.vector(spt)
```

QTL mapping and LOD profile processing

Perform QTL mapping analysis using H-K regression, and processing of the LOD profiles by setting to zero LOD values outside the 1.5 LOD support interval around the peak at each chromosome (as well as LOD values below the single trait mapping threshold, thr).

```r
> scan1 <- scanone(cross1, pheno.col = 1:1000, method = "hk")
> scandrop1 <- set.to.zero.beyond.drop.int(chr = scan1[,1],
+   scanmat = as.matrix(scan1[,,-c(1,2)]),
+   thr = min(lod.thrs),
+   drop = 1.5)

By setting to zero the LOD scores outside the LOD support interval we can considerably decrease the spread of the hotspot.
```
Hotspot architecture at varying thresholds

For each genomic position, we count the number of traits with \( \text{LOD} \geq \text{lod.thrs} \).

The counts1 object is a matrix with 204 rows (genetic markers) and 10 columns (thresholds).

```r
> counts1 <- t(count.thr(scandrop1, lod.thrs, droptwo = FALSE))
> counts1[52:56,]
D2M1  0   0   0   0   0   0   0   0   0   0
D2M2  0   0   0   0   0   0   0   0   0   0
D2M3  0   1   2   3   4   5   6   8  13  15
D2M4  2   2   3   5   6  14  17  21  24  27
D2M5  0   2   3   3   4   6   8  11  13  14
```

The first column gives the counts for threshold of 3.11. The last one shows the counts for threshold 2.11. Note how the counts increase as the QTL mapping thresholds decrease.

**Hotspot architecture for LOD thr 2.44 (\( \alpha = 0.05 \))**

```r
> out1 <- data.frame(scan1[, 1:2], counts1)
> class(out1) <- c("scanone", "data.frame")
> par(mar=c(4.1,4.1,0.1,0.1))
> plot(out1, lodcolumn = 5, ylab = "counts", cex.lab = 1.5,
> +     cex.axis = 1.5)
```

Note the spurious hotspots on chr 1.
**Q-method**

The `WW.perm` function implements the Q-method's permutation scheme.

```r
> set.seed(12345)
> Q.1 <- WW.perm(scanmat = scandrop1,
+     lod.thrs = lod.thrs,
+     n.perm = 100,
+     verbose = FALSE)
```

The output is a matrix with 100 rows (permutations), and 10 columns (thresholds). Each entry $ij$ represents the maximum number of significant linkages across the entire genome detected at permutation $i$, using the LOD threshold $j$.

---

## Q-method

The `WW.summary` function computes the hotspot size permutation thresholds.

```r
> Q.1.thr <- WW.summary(Q.1, alphas)
> Q.1.thr
   0.01  0.02  0.03  0.04  0.05  0.06  0.07  0.08  0.09  0.1
3.10508056313925 11.00 10.02 10.00 10.00 10.00 10.00 10.00 10.00 10.0
2.89135162173146 12.00 12.00 11.03 11.00 11.00 11.00 11.00 11.00 11.0
2.67690269000741 14.01 13.02 13.00 13.00 13.00 13.00 13.00 13.00 13.0
2.5743266994317  16.01 16.00 16.00 15.04 15.00 15.00 15.00 15.00 15.0
2.43869721183317  18.00 18.00 17.03 17.00 17.00 17.00 17.00 17.00 17.0
2.335067939838  21.01 21.00 20.03 20.00 20.00 20.00 20.00 20.00 20.0
2.257747088154  22.02 22.00 22.00 21.04 21.00 21.00 20.08 20.00 20.0
2.19884780562269  23.01 23.00 22.03 22.00 22.00 22.00 22.00 22.00 21.1
2.15023439516803  24.02 24.00 24.00 23.04 23.00 23.00 22.09 22.09 22.0
2.11039422475441  26.02 26.00 25.03 25.00 25.00 24.07 24.00 24.00 24.0
```
**Q-method**

In general, we are interested in using the same error rates for the QTL mapping and hotspot analysis.

Therefore, we are usually more interested on the diagonal of \( Q.1.\text{thr} \).

For the hotspots depicted in the previous figure, we adopted a GWER of 5%, and the corresponding Q-method’s permutation threshold is 17.

According to this threshold, all hotspots are significant.

```r
> diag(Q.1.thr)
[1]  11.00  12.00  13.00  15.04  17.00  20.00  21.00  22.00  22.09  24.00
```

**N- and NL-methods**

The `NL.N.perm` function implements the N- and NL-methods’ permutation schemes.

The argument `Nmax` sets the maximum hotspot size to be analyzed by the NL-method.

The argument `drop` controls the magnitude of the LOD support interval computation during the LOD profile processing step.

```r
> set.seed(12345)
> NL.N.1 <- NL.N.perm(cross = cross1,
+     Nmax = 300,
+     n.perm = 100,
+     lod.thrs = lod.thrs,
+     drop = 1.5,
+     verbose = TRUE)
> names(NL.N.1)
[1] "max.lod.quant" "max.N"
```

The function’s output is a list with two elements: `max.lod.quant` and `max.N`. 
**N- and NL-methods**

**max.lod.quant** stores the output of the *NL*-method’s perms. It is given by a matrix with 100 rows (permutations), and 300 columns (hotspot sizes analyzed).

Entry $ij$ stores the maximum genome wide $qLOD(n)$ computed at permutation $i$ using threshold $j$, where $qLOD(n)$ corresponds to the $n$th LOD score in a sample ordered from highest to lowest.

For instance, consider the first 3 lines and 6 columns of **max.lod.quant**. At the 3rd permutation, the maximum LOD score across the genome was 3.37, the second maximum across the genome was 3.36, and so on.

```r
> NL.N.1[[1]][1:3, 1:6]
[1,]  2.115918 1.903466 1.713409 1.649016 1.600378 1.594265
[2,]  2.464650 2.162832 1.932474 1.885934 1.878833 1.839507
[3,]  3.374947 3.358949 3.198482 3.195974 3.121577 3.105578
```

**N- and NL-methods**

**max.N** stores the output of the *N*-method’s perms. It is given by a matrix with 100 rows (permutations), and 10 columns (thresholds).

Entry $ij$ stores the maximum genome wide hotspot size detected at permutation $i$ when computed using threshold $j$ (note the output is transposed).

```r
> t(NL.N.1[[2]][1:6,])
[1,] 3.105080563 0.0060 1.99  2.8913516 2.6769027 2.5743267
[2,] 3.358949  0.00  2.62  2.6769027 2.5743267 2.4386972
[3,] 3.105080563 0.10  2.16  2.5743267 2.4386972 2.3350680
[4,] 3.121577  0.13  2.67  2.4386972 2.3350680 2.1988478
[5,] 3.105080563 0.25  2.16  2.3350680 2.1988478 2.1103942
[6,] 3.105080563 0.25  2.16  2.3350680 2.1988478 2.1103942
```
**N- and NL-methods**

The `NL.N.summary` function computes the N- and NL-method’s hotspot size permutation thresholds.

```r
> NL.N.1.thrs <- NL.N.summary(NL.N.1[[1]], NL.N.1[[2]], alphas)
> NL.1.thr <- NL.N.1.thrs[[1]]
> N.1.thr <- NL.N.1.thrs[[2]]
```

**N- and NL-methods**

`N.1.thr` is a 10 by 10 matrix with rows indexing the QTL mapping thr and columns indexing the target GWER.

Each entry $ij$ shows the hotspot size above which a hotspot is considered significant at a GWER $j$ using the QTL mapping threshold $i$.

The $N$-method’s threshold that controls the hotspot GWER at a 5% level when the QTL mapping was controlled at a GWER of 5% is 195.75.

```r
> N.1.thr[1:3,]
        0.01  0.02  0.03  0.04  0.05  0.06  0.07  0.08  0.09
3.10508056313925 52.23 46.08 35.33 32.12 25.35 25.00 20.35 19.08 18.09
2.89135162173146 95.06 86.12 83.09 53.24 39.65 39.00 31.56 30.08 29.09
2.67690269000741 191.59 180.16 157.69 103.24 86.75 65.32 59.35 51.64 46.45
```

According to the $N$-method, none of the hotspots in the previous figure is significant.
The `NL.1.thr` object is a matrix with 300 rows (spurious hotspot sizes analyzed), and 10 columns (target GWER).

Each entry \( ij \) represents the LOD threshold at which a hotspot of size greater or equal than \( i \) is significant at a GWER less or equal to \( j \).

```
> round(NL.1.thr[1:3,:), 4)
   0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1
```
**N- and NL-methods**

For each genomic location this figure shows the hotspot sizes at which the hotspot was significant, that is, at which the hotspot locus had more traits than the hotspot size threshold on the left mapping to it with a LOD score higher than the threshold on the right than expected by chance.
Simulate data

We first use the `SimCrossCausal` function to simulate a cross object with 3 phenotypes, $y_1$, $y_2$ and $y_3$, where $y_1$ has a causal effect on both $y_2$ and $y_3$.

```r
> set.seed(987654321)
> Cross <- SimCrossCausal(n.ind = 100,
+   len = rep(100, 3),
+   n.mar = 101,
+   beta = rep(0.5, 2),
+   add.eff = 1,
+   dom.eff = 0,
+   sig2.1 = 0.4,
+   sig2.2 = 0.1,
+   eqspacing = FALSE,
+   cross.type = "bc",
+   normalize = TRUE)
```
QTL mapping

Compute the genotype conditional probabilities setting the maximum distance between positions at which genotype probabilities were calculated to 1cM.

```r
> Cross <- calc.genoprob(Cross, step = 1)
```

Perform QTL mapping using Haley-Knott regression.

```r
> Scan <- scanone(Cross, pheno.col = 1:3, method = "hk")
> plot(Scan, lodcolumn = 1:3, ylab = "LOD")
```

Black, blue and red curves represent phenos $y_1$, $y_2$ and $y_3$, respectively.

QTL mapping

Summarize the results for the 3 phenotypes.

```r
> summary(Scan[, c(1, 2, 3)], thr = 3)
  chr pos   y1
  c1.loc55 1  55 12.6
> summary(Scan[, c(1, 2, 4)], thr = 3)
  chr pos   y2
  c1.loc55 1  55  5.27
> summary(Scan[, c(1, 2, 5)], thr = 3)
  chr pos   y3
  D1M50 1 55.5  7.58
```

$y_1$ and $y_2$ map to the same QTL at position 55 cM on chr 1, $y_3$ maps to a distinct position.

Which QTL should we use as causal anchor?
QTL mapping

Our approach is to compute the joint LOD profile of both phenos and use the QTL detected by this joint approach as the causal anchor.

```r
> commqtls <- GetCommonQtls(Cross,
+   pheno1 = "y1",
+   pheno2 = "y3",
+   thr = 3,
+   peak.dist = 5,
+   addcov1 = NULL,
+   addcov2 = NULL,
+   intcov1 = NULL,
+   intcov2 = NULL)
> commqtls
       Q   Q.chr  Q.pos
1 c1.loc55   1    55
```

CMST tests

Fit the CMST tests.

```r
> nms <- names(Cross$pheno)
> out1 <- CMSTtests(Cross,
+   phen01 = nms[1],
+   phen02 = nms[2],
+   Q.chr = 1,
+   Q.pos = 55,
+   addcov1 = NULL,
+   addcov2 = NULL,
+   intcov1 = NULL,
+   intcov2 = NULL,
+   cross.type = "bc",
+   method = "all",
+   penalty = "both")
```
CMST tests - output

> out1[1:6]
$pheno1
[1] "y1"

$pheno2
[1] "y2"

$n.ind
[1] 100

$loglik
[1] -123.5318 -140.4604 -141.5803 -123.4834

$model.dim
[1] 6 6 6 7

$R2
[1] 0.4407170 0.2153583

CMST tests - output

Covariance matrix of the log-likelihood scores.

> out1[7]
$S.hat
$S.hat
[1,] 0.26221327 -0.01323094 0.010924311 -0.275444212 -0.251288963 0.02415525
[2,] -0.01323094 0.36275299 0.012080993 0.375983930 0.025311930 -0.35067200
[3,] 0.01092431 0.01208099 0.001115354 0.001155681 -0.009808958 -0.01096563
[4,] -0.27544421 0.37598393 0.001155681 0.651428142 0.276600893 0.33970636
[5,] -0.25128896 0.2531193 -0.009808958 0.276600893 0.241480006 0.33970636
CMST tests - output

> out1[8:12]

$BICs
[1] 274.6946 308.5518 310.7917 279.2030

$Z.bic
[1,] NA 3.305926 2.9966507 6.749745
[2,] NA       NA 0.1387598 -2.986200
[3,] NA       NA       NA -2.709873
[4,] NA       NA       NA       NA

$pvals.p.BIC
[1] 0.001364817 0.999526684 0.998635183 1.000000000

$pvals.np.BIC

$pvals.j.BIC
[1] 0.003779558 0.999946885 0.999669186 1.000000000

CMST tests - output

> out1[13:17]

$AICs
[1] 259.0636 292.9208 295.1606 260.9668

$Z.aic
[1,] NA 3.305926 2.9966507 2.849429
[2,] NA       NA 0.1387598 -3.251273
[3,] NA       NA       NA -2.933361
[4,] NA       NA       NA       NA

$pvals.p.AIC
[1] 0.002189889 0.999526684 0.998635183 0.997810111

$pvals.np.AIC

$pvals.j.AIC
[1] 0.005993868 0.999466885 0.999669186 1.000000000
CMST tests

Fit one phenotype against a list of phenotypes.

```r
> out2 <- CMSTtestsList(Cross,
+   phenol = nms[1],
+   phenos = nms[-1],
+   Q.chr = 1,
+   Q.pos = 55,
+   addcov1 = NULL,
+   addcov2 = NULL,
+   intcov1 = NULL,
+   intcov2 = NULL,
+   cross.type = "bc",
+   method = "par",
+   penalty = "bic")
```

CMST tests

```r
> out2

$R^2$s

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>y1_y2</td>
<td>0.440717</td>
<td>0.2153583</td>
</tr>
<tr>
<td>y1_y3</td>
<td>0.440717</td>
<td>0.2914979</td>
</tr>
</tbody>
</table>

$BIC.stats$

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>y1_y2</td>
<td>274.6946</td>
<td>308.5518</td>
<td>310.7917</td>
<td>279.2030</td>
<td>3.305926</td>
</tr>
<tr>
<td>y1_y3</td>
<td>270.4445</td>
<td>294.0943</td>
<td>325.3707</td>
<td>274.6665</td>
<td>2.339472</td>
</tr>
</tbody>
</table>

$z.23$ $z.24$ $z.34$

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>y1_y2</td>
<td>0.1387598</td>
<td>-2.986200</td>
</tr>
<tr>
<td>y1_y3</td>
<td>1.9587743</td>
<td>-2.126754</td>
</tr>
</tbody>
</table>

$pvals.p.BIC$

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>y1_y2</td>
<td>0.001364817</td>
<td>0.9995267</td>
<td>0.9986352</td>
</tr>
<tr>
<td>y1_y3</td>
<td>0.009655499</td>
<td>0.9903445</td>
<td>0.9999871</td>
</tr>
</tbody>
</table>
```
Simulate data

We simulate data from a $F_2$ cross with 500 ind. and 5 chr of len 100 cM, containing 11 equally spaced markers per chr. We simulated one QTL per pheno. The QTLs, $Q_t$, $t = 1, 2, 3, 4, 5$, were placed at the middle marker on chr $t$. We set additive and dominance QTL effects to 1 and 0, respectively.

```r
> library(qtlnet)
> set.seed(12345)
> Map <- sim.map(len = rep(100, 5), n.mar = 11, eq.spacing = TRUE,
+ include.x = FALSE)
> Cross <- sim.cross(map = Map, n.ind = 500, type = "f2")
> crosses <- vector(mode = "list", length = 5)
> add.effects <- c(1, 1, 1, 1, 1)
> for (i in 1:5) {
+ map <- sim.map(len = rep(100, i), n.mar = 11, eq.spacing = TRUE,
+ include.x = FALSE)
+ crosses[[i]] <- sim.cross(map = map, n.ind = 500, type = "f2",
+ model = c(i, 50, add.effects[i], 0))
+ Cross$geno[[i]] <- crosses[[i]]$geno[[i]]
+ }
```
Simulate data
The pheno data was simulated according to the network below, using regr equations with regr coeffs set to 1.

\[
\begin{align*}
> & \text{beta} <- 1 \\
> & \text{Cross$\text{pheno}[1]} <- \text{crosses[[1]]$\text{pheno}} \\
> & \text{Cross$\text{pheno}[2]} <- \text{crosses[[2]]$\text{pheno} + beta * Cross$\text{pheno}[1]} \\
> & \text{Cross$\text{pheno}[3]} <- \text{crosses[[3]]$\text{pheno} + beta * Cross$\text{pheno}[2]} \\
> & \text{Cross$\text{pheno}[4]} <- \text{crosses[[4]]$\text{pheno} + beta * Cross$\text{pheno}[2]} \\
> & \text{Cross$\text{pheno}[5]} <- \text{crosses[[5]]$\text{pheno} + beta * Cross$\text{pheno}[3] + beta * Cross$\text{pheno}[4]} \\
> & \text{names(Cross$\text{pheno})} <- \text{paste("y", 1:5, sep = "")}
\end{align*}
\]

Permutation test threshold

We determine the QTL mapping LOD threshold via permutation test.

\[
\begin{align*}
> & \text{Cross <- calc.genoprob(Cross, step = 1)} \\
> & \text{set.seed(12345)} \\
> & \text{perm.test <- scanone(Cross, n.perm = 1000, method = "hk")} \\
& \text{Doing permutation in batch mode ...} \\
> & \text{summary(perm.test)} \\
& \text{LOD thresholds (1000 permutations)} \\
& \text{lod} \\
& 5\% 3.04 \\
& 10\% 2.70
\end{align*}
\]

We adopt a LOD threshold of 3.04, that aims to control GWER < 5\%.
QDG routines

We perform QTL mapping with Haley-Knott regression for all 5 phenotypes.

> Scan <- scanone(Cross, pheno.col = 1:5, method = "hk")

Next we determine the QTLs for each phenotype, and create a list with objects of class qtl that is needed as impute for the qdg function.

> Cross <- sim.geno(Cross, n.draws = 1)
> marker.nms <- allqtls <- vector(mode = "list", length = 5)
> names(marker.nms) <- names(allqtls) <- paste("y", 1:5, sep = "")
> for (i in 1:5) {
+   aux <- summary(Scan[, c(1, 2, i + 2)], thr = 3.04)
+   marker.nms[[i]] <- find.marker(Cross, chr = aux[, 1], pos = aux[, 2])
+   allqtls[[i]] <- makeqtl(Cross, chr = aux[, 1], pos = aux[, 2])
+ }

QDG routines

Fit the QDG algorithm.

> out1 <- qdg(cross = Cross,
+   +phenotype.names = paste("y", 1:5, sep = ""),
+   +marker.names = marker.nms,
+   +QTL = allqtls,
+   +alpha = 0.005,
+   +n.qdg.random.starts = 10,
+   +addcov = NULL,
+   +intcov = NULL,
+   +skel.method = "pcskel")
>
> out1$UDG

   node1 node2 edge
1     y1   y2   1
3     y2   y3   1
4     y2   y4   1
6     y3   y5   1
8     y4   y5   1
QDG routines

```r
> out1$DG
   node1 direction node2 lod score
1    y1    ---->    y2  24.135325
2    y2    ---->    y3  23.990280
3    y2    ---->    y4  32.013798
4    y3    ---->    y5   3.119176
5    y4    ---->    y5   9.726617
```

```r
> out1$Solutions
$solutions
$solutions[[1]]
   node1 direction node2 lod
1    y1    ---->    y2 24.13533
2    y2    ---->    y3 61.02425
3    y2    ---->    y4 69.14849
4    y3    ---->    y5 54.17467
5    y4    ---->    y5 69.25563
```

```r
$loglikelihood
[1] -3595.164
```

QDG routines

Plot the QDGs

```r
> gr1 <- graph.qdg(out1, include.qtl = FALSE)
> plot(gr1)
> gr2 <- graph.qdg(out1, include.qtl = TRUE)
> plot(gr2)
```

*(cannot export eps from R. pdf has no margins)*

Although the structure of the phenotype network is correct, the genetic architecture is not.
Unconditional versus conditional QTL mapping

Here we plot the LOD profiles for all phenotypes using both unconditional mapping analysis, and conditional mapping (where the parents of each phenotype are used as additive covariates in the QTL mapping).

```r
> par(mfrow = c(2, 5), cex.lab = 1.5, cex.axis = 1.5, cex.main = 2)
> uncond.nms <- paste("Y", 1:5, sep = "")
> for (i in 1:5) {
+ plot(Scan, lodcolumn = i, main = uncond.nms[i], ylab = "lod")
+ }
> plot(Scan, lodcolumn = 1, main = uncond.nms[1], ylab = "lod")
> cond.nms <- c("Y1", "Y2 | Y1", "Y3 | Y2", "Y4 | Y2", "Y5 | Y3, Y4")
> pheno.parents <- list(NULL, 1, 2, 2, c(3, 4))
> for (i in 2:5) {
+ CondScan <- scanone(Cross, pheno.col = i, method = "hk",
+ addcov = Cross$pheno[, pheno.parents[[i]]])
+ plot(CondScan, main = cond.nms[i], ylab = "lod")
+ }
```

Unconditional versus conditional QTL mapping

![LOD profile plots for Y1 to Y5](image1.png)
QTLnet routines - basic functionality

Fit the QTLnet algorithm.

```r
> out2 <- mcmc.qtlnet(cross = Cross,
+                   pheno.col = 1:5,
+                   threshold = 3.04,
+                   addcov = NULL,
+                   intcov = NULL,
+                   nSamples = 1000,
+                   thinning = 3,
+                   max.parents = 4,
+                   M0 = NULL,
+                   burnin = 0.2,
+                   method = "hk",
+                   random.seed = 987654321,
+                   init.edges = 0,
+                   saved.scores = NULL,
+                   rev.method = "nbhd",
+                   verbose = TRUE)
```

QTLnet routines - basic functionality

```r
> summary(out2)
```

Model-averaged network: (min.prob = 0.5)

<table>
<thead>
<tr>
<th>cause</th>
<th>effect</th>
<th>prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>y1</td>
<td>y2</td>
</tr>
<tr>
<td>2</td>
<td>y2</td>
<td>y3</td>
</tr>
<tr>
<td>3</td>
<td>y2</td>
<td>y4</td>
</tr>
<tr>
<td>4</td>
<td>y3</td>
<td>y5</td>
</tr>
<tr>
<td>5</td>
<td>y4</td>
<td>y5</td>
</tr>
</tbody>
</table>

Posterior probabilities by direction:

| node1 node2  -->  <--  no |
|-------------------------|-------------------------|
| 1 y1 y2 1.000 0.000 0.000 |
| 2 y1 y3 0.019 0.000 0.981 |
| 3 y1 y4 0.073 0.000 0.927 |
| 4 y1 y5 0.080 0.000 0.920 |
| 5 y2 y3 1.000 0.000 0.000 |

Acceptance frequency for MCMC: 0.9996667
QTLnet routines - basic functionality

> print(out2)

Model averaged probabilities for edge direction (row -> col):

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>0</td>
<td>1</td>
<td>0.019</td>
<td>0.073</td>
<td>0.080</td>
</tr>
<tr>
<td>[2,]</td>
<td>0</td>
<td>0</td>
<td>1.000</td>
<td>1.000</td>
<td>0.094</td>
</tr>
<tr>
<td>[3,]</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0.054</td>
<td>1.000</td>
</tr>
<tr>
<td>[4,]</td>
<td>0</td>
<td>0</td>
<td>0.029</td>
<td>0.000</td>
<td>1.000</td>
</tr>
<tr>
<td>[5,]</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Posterior probabilities by causal model:

<table>
<thead>
<tr>
<th></th>
<th>post.prob</th>
<th>BIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)(2</td>
<td>1)(3</td>
<td>2)(4</td>
</tr>
<tr>
<td>(1)(2</td>
<td>1)(3</td>
<td>2)(4</td>
</tr>
<tr>
<td>(1)(2</td>
<td>1)(3</td>
<td>2)(4</td>
</tr>
<tr>
<td>(1)(2</td>
<td>1)(3</td>
<td>2)(4</td>
</tr>
<tr>
<td>(1)(2</td>
<td>1)(3</td>
<td>2)(4</td>
</tr>
<tr>
<td>(1)(2</td>
<td>1)(3</td>
<td>2,4)(4</td>
</tr>
<tr>
<td>(1)(2</td>
<td>1)(3</td>
<td>2)(4</td>
</tr>
</tbody>
</table>

> loci.qtlnet(out2)

$y1
[1] "chr1@50"

$y2
[1] "chr2@50"

$y3
[1] "chr3@49"

$y4
[1] "chr4@49"

$y5
[1] "chr5@49"
QTLnet routines - basic functionality

> plot(out2)

QTLnet routines - basic functionality

> par(mfrow = c(1, 1))
> plotbic.qtlnet(out2, smooth = FALSE)
QTLnet routines - parallel implementation

The most expensive part of calculations is running `scanone` on each phenotype with parent phenotypes as covariates. Our strategy is to pre-compute the BIC contributions using a cluster and save them for later use.

We divide the job into four steps:

1. Determine parents and divide into reasonable sized groups.
2. Compute BIC scores using `scanone` on a grid of computers.
3. Compute multiple MCMC runs on a grid of computers.
4. Catenate the outputs of the multiple MCMC runs into a single output object.

We illustrate this approach with a simple example of “parallel” analysis.

QTLnet routines - parallel implementation - step 1

**STEP 1:** defines how the computations are going to break up (that are carried out on steps 2 and 3).

```r
> pheno.col <- 1:5
> max.parents <- 4
> size.qtlnet(pheno.col, max.parents)
[1] 80
> parents <- parents.qtlnet(pheno.col, max.parents)
> groups <- group.qtlnet(parents = parents, group.size = 10)
> save(Cross, pheno.col, max.parents, parents, groups,
+       file = "Step1.RData", compress = TRUE)
```

The function `size.qtlnet` determines the number of `scanone` calculations possible for a network with nodes `pheno.col` and maximum parent size `max.parents`.

```r
> size.qtlnet(pheno.col, max.parents)
[1] 80
```
QTLnet routines - parallel implementation - step 1

The `parents.qtlnet` function creates a list of all possible parent sets (up to `max.parents` in size) to be used as covariates of the child phenotypes in the `scanone` computations.

The `parents` column shows the possible parent sets. The `n.child` column represents the number of possible child nodes to the parent set.

```r
> parents <- parents.qtlnet(pheno.col, max.parents)
> parents
  parents n.child
    1      1       4
    2      2       4
    ...  1,2     1,2       3
    ...  
```

No parents (5 scanones): $y_1 \sim 1$, $y_2 \sim 1$, $y_3 \sim 1$, $y_4 \sim 1$, and $y_5 \sim 1$.

With $y_1$ as a parent (4 scanones): $y_2 \sim y_1$, $y_3 \sim y_1$, $y_4 \sim y_1$, and $y_5 \sim y_1$.

With $y_1$ and $y_2$ as parents (3 scanones): $y_3 \sim y_1 + y_2$, $y_4 \sim y_1 + y_2$, and $y_5 \sim y_1 + y_2$.

---

QTLnet routines - parallel implementation - step 1

The function `group.qtlnet` groups the parent sets into roughly equal size groups for parallel computations.

```r
> groups <- group.qtlnet(parents = parents, group.size = 10)
> groups
  begin end
    1  1  2
    2  3  4
    3  5  7
    4  8 10
    5 11 14
    6 15 18
    7 19 23
    8 24 30
    9 31 31
> pa <- summary(parents)
> N <- rep(NA, nrow(groups))
> for (i in 1:nrow(groups))
+   N[i] <- sum(pa[seq(groups[i, 1], groups[i, 2]), 2])
> N
 [1]  9  8 11  9 12 10 10 10  1
```
STEP 2: Pre-compute BIC scores for selected parents.

```r
> load("Step1.RData")
> for (i in seq(nrow(groups))) {
+ bic <- bic.qtlnet(Cross,
+ pheno.col,
+ threshold = 3.04,
+ max.parents = max.parents,
+ parents = parents[seq(groups[i,1], groups[i,2])])
+ save(bic, file = paste("bic", i, ".RData", sep = ""), compress = TRUE)
+ }
```

Read in saved BIC scores and combine into one object.

```r
> load("Step1.RData")
> bic.group <- list()
> for (i in seq(nrow(groups))) {
+ load(paste("bic", i, ".RData", sep = "]"))
+ bic.group[[i]] <- bic
+ cat("group =", i, 
"\n")
+ }
> saved.scores <- bic.join(Cross, pheno.col, bic.group, max.parents = 4)
```
QTLnet routines - parallel implementation - step 2

```r
# saved.scores
y1  y2  y3  y4  y5
1 1132.647 1414.8944 1776.265 1780.912 2437.802
2 1304.698 1222.2440 1394.943 1434.985 2005.474
3 1291.897 1242.0799 1682.636 1687.116 1953.474
4 1299.858 1156.7878 1273.851 1246.465 1917.227
1,2 1137.728 1059.9072 1400.437 1439.585 2011.442
1,3 1138.785 1089.4257 1627.265 1631.393 1917.990
1,4 1138.526 1023.7947 1276.038 1241.347 1885.897
2,3 1263.316 1002.2426 1401.154 1441.172 1800.790
2,4 1287.025 1110.6142 1221.812 1218.454 1759.518
3,4 1279.066 1128.8087 1210.769 1186.155 1424.405
1,2,3 1143.837 896.6137 1406.650 1445.789 1805.105
1,2,4 1143.942 984.3935 1225.789 1222.706 1765.651
1,3,4 1144.734 1000.8086 1202.754 1171.667 1430.608
2,3,4 1269.522 1008.2210 1091.324 1087.177 1429.712
1,2,3,4 1149.933 902.5707 1096.584 1093.126 1435.644
```

QTLnet routines - parallel implementation - step 3

**STEP 3**: Sample Markov chain (MCMC).

```r
# set.seed(54321)
# n.runs <- 3
# for (i in seq(n.runs)) {
#   cat("run =", i, "\n")
#   ## Run MCMC with randomized initial network.
#   mcmc <- mcmc.qtlnet(Cross,
#                        pheno.col,
#                        threshold = 3.04,
#                        thinning = 1,
#                        max.parents = max.parents,
#                        saved.scores = saved.scores,
#                        init.edges = NULL)
#   save(mcmc, file = paste("mcmc", i, ".RData", sep = ""),
#        compress = TRUE)
# }
```
QTLnet routines - parallel implementation - step 4

**STEP 4:** Combine results for post-processing.

```r
> n.runs <- 3
> outs.qtlnet <- list()
> for (i in seq(n.runs)) {
+   load(paste("mcmc", i, ".RData", sep = ""))
+   outs.qtlnet[[i]] <- mcmc
+ }
> out3 <- c.qtlnet(outs.qtlnet)
```

The function `c.qtlnet` catenates the outputs of the 3 separate runs together.

---

QTLnet routines - parallel implementation - outputs

```r
> summary(out3)
```

Model-averaged network: (min.prob = 0.5)

<table>
<thead>
<tr>
<th>cause</th>
<th>effect</th>
<th>prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>y1</td>
<td>y2</td>
<td>0.915556</td>
</tr>
<tr>
<td>y2</td>
<td>y3</td>
<td>0.925556</td>
</tr>
<tr>
<td>y2</td>
<td>y4</td>
<td>0.912963</td>
</tr>
<tr>
<td>y3</td>
<td>y5</td>
<td>0.9085185</td>
</tr>
<tr>
<td>y4</td>
<td>y5</td>
<td>0.9103704</td>
</tr>
</tbody>
</table>

Posterior probabilities by direction:

<table>
<thead>
<tr>
<th>node1</th>
<th>node2</th>
<th>---&gt;</th>
<th>&lt;--</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>y1</td>
<td>y2</td>
<td>0.916</td>
<td>0.084</td>
<td>0.000</td>
</tr>
<tr>
<td>y1</td>
<td>y3</td>
<td>0.019</td>
<td>0.015</td>
<td>0.966</td>
</tr>
<tr>
<td>y1</td>
<td>y4</td>
<td>0.033</td>
<td>0.023</td>
<td>0.947</td>
</tr>
<tr>
<td>y1</td>
<td>y5</td>
<td>0.028</td>
<td>0.006</td>
<td>0.966</td>
</tr>
<tr>
<td>y2</td>
<td>y3</td>
<td>0.926</td>
<td>0.074</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Acceptance frequency for MCMC: 0.999
QTLnet routines - parallel implementation - outputs

> plot(out3)

QTLnet routines - parallel implementation - outputs

> plotbic.qtlnet(out3, smooth = FALSE)