The new `summary.scantwo` and `plot.scantwo`
Karl W Broman, 27 Oct 2006

In R/qtl version 1.04, the functions `summary.scantwo` and `plot.scantwo` have been changed quite substantially. Also, the permutations with `scantwo` have been changed to match the new format for `summary.scantwo`.

In this document, I describe the revisions and how to use the new functions. We'll look at the `hyper` data as an example.

First we need to load the package and the data.

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

I'm going to use `scantwo` with `method="em"`. First I run `calc.genoprob`, and then `scantwo` as before.

```
> hyper <- calc.genoprob(hyper, step=2.5)
> out2 <- scantwo(hyper)
```

The output, in this new version of R/qtl, is slightly different. The LOD scores for the full model (two QTL plus interaction) are there as before, but in place of the epistasis LOD scores, I save the LOD scores for the additive QTL model. Also, we now always run the single-QTL analysis with `scanone`, since the results are necessary to make sense of the output of `scantwo`.

The big change is in `summary.scantwo`. Consider a pair of positions in the genome, s and t. We consider four models.

$$\begin{aligned}
\text{Full:} &\quad y = \mu + \beta_1 q_1 + \beta_2 q_2 + \beta_3 (q_1 \times q_2) + \epsilon \\
\text{Add:} &\quad y = \mu + \beta_1 q_1 + \beta_2 q_2 + \epsilon \\
\text{One:} &\quad y = \mu + \beta_1 q_1 + \epsilon \\
\text{Null:} &\quad y = \mu + \epsilon
\end{aligned}$$

Let $l_f(s,t)$ be the $\log_{10}$ likelihood for the full model with QTL at s and t, $l_a(s,t)$ be the $\log_{10}$ likelihood for the additive model with QTL at s and t, $l_1(s)$ be the $\log_{10}$ likelihood for the single-QTL model with the QTL at s, and l_0 be the $\log_{10}$ likelihood under the null (with no QTL).

Define the LOD scores as follows.

$$\begin{aligned}
\text{LOD}_f(s,t) &= l_f(s,t) - l_0 \\
\text{LOD}_a(s,t) &= l_a(s,t) - l_0 \\
\text{LOD}_1(s) &= l_1(s) - l_0
\end{aligned}$$

Now for the new part. Following a suggestion from Gary Churchill, we consider a pair of chromosomes j and k. (We include the case $j = k$.) Let $c(s)$ denote the chromosome for position s. We now consider the maximum LOD scores over that pair of chromosomes.

$$\begin{aligned} M_f(j,k) &= \max_{c(s)=j,c(t)=k} \mathrm{LOD}_f(s,t) \\ M_a(j,k) &= \max_{c(s)=j,c(t)=k} \mathrm{LOD}_a(s,t) \\ M_1(j,k) &= \max_{c(s)=j \text{ or } k} \mathrm{LOD}_1(s) \end{aligned}$$

So $M_f(j,k)$ is the $\log_{10}$ likelihood ratio comparing the full model with QTL on chromosomes j and k to the null model, and $M_a(j,k)$ is the analogous thing for the additive model. Note that the pair of positions at which the full model is maximized may be different from the pair of positions at which the additive model is maximized. $M_1(j,k)$ is the $\log_{10}$ likelihood ratio comparing the model with a single QTL on either chromosomes j or k to the null model.

We derive three further LOD scores from the above.

$$\begin{aligned} M_i(j,k) &= M_f(j,k) - M_a(j,k) \\ M_{fv1}(j,k) &= M_f(j,k) - M_1(j,k) \\ M_{av1}(j,k) &= M_a(j,k) - M_1(j,k) \end{aligned}$$

$M_i(j,k)$ is the $\log_{10}$ likelihood ratio comparing the full model with QTL on chromosomes j and k to the additive model with QTL on chromosomes j and k, and so indicates evidence for an interaction between QTL on chromosomes j and k, assuming that there is precisely one QTL on each chromosome (or, for $j = k$, that there are two QTL on the chromosome).

$M_{fv1}(j,k)$ is the $\log_{10}$ likelihood ratio comparing the full model with QTL on chromosomes j and k to the single-QTL model, with a single QTL on either chromosome j or k. Thus, it indicates evidence for a second QTL, allowing for the possibility of epistasis.

$M_{av1}(j,k)$ is the $\log_{10}$ likelihood ratio comparing the additive model with QTL on chromosomes j and k to the single-QTL model, with a single QTL on either chromosome j or k. Thus, it indicates evidence for a second QTL, assuming no epistasis.

In `summary.scantwo`, we must provide thresholds for each of the five LOD scores, $M_f(j,k)$, $M_{fv1}(j,k)$, $M_i(j,k)$, $M_a(j,k)$ and $M_{av1}(j,k)$. A pair of chromosomes (j,k) is reported as interesting if either of the following holds:

- $M_f(j,k) \geq T_f$ and $[M_{fv1}(j,k) \geq T_{fv1}$ or $M_i(j,k) \geq T_i]$
- $M_a(j,k) \geq T_a$ and $M_{av1}(j,k) \geq T_{av1}$

I'm inclined towards ignoring $M_i(j,k)$ in this rule (i.e. setting $T_i = \infty$), and using a common significance level ($\alpha = 5$ or 10%) for the four remaining thresholds.

By default, `summary.scantwo` now calculates the five LOD scores above, keeping track of the positions at which $M_f(j,k)$ and $M_a(j,k)$ were maximized. It either prints the best results on all pairs of chromosomes, or we must provide five thresholds (T_f, T_{fv1}, T_i, T_a and T_{av1}, in that order).

The thresholds can be obtained by a permutation test (see below), but this is extremely time-consuming. For a mouse backcross, we suggest the thresholds (6.0, 4.7, 4.4, 4.7, 2.6) for the full, conditional-interactive, interaction, additive, and conditional-additive LOD scores, respectively. For a mouse intercross, we suggest the thresholds (9.1, 7.1, 6.3, 6.3, 3.3) for the full, conditional-interactive, interaction, additive, and conditional-additive LOD scores, respectively. These were obtained by 10,000 simulations of crosses with 250 individuals, markers at a 10 cM spacing, and analysis by Haley-Knott regression.

```
> summary(out2, thresholds=c(6.0, 4.7, 4.4, 4.7, 2.6))

       pos1f pos2f lod.full lod.fv1 lod.int     pos1a pos2a lod.add lod.av1
c1:c4   68.3    30    14.31    6.67   0.306      68.3  30.0   14.00    6.36
c6:c15  60.0    18     7.02    5.16   3.230      22.5  20.5    3.78    1.93
```

M_f, M_{fv1} and M_i are `lod.full`, `lod.fv1` and `lod.int`, respectively, and correspond to positions `pos1f` and `pos2f`.

M_a and M_{av1} are `lod.add` and `lod.av1`, respectively, and correspond to positions `pos1a` and `pos2a`.

The above is the default output, with `what="best"`. The argument `what` may also be given as `"full"`, `"add"`, or `"int"`, in which case, for each pair of chromosomes, we pull out the pair of positions with maximum full, additive, or interactive LOD score, respectively, and calculate, for example, the interaction LOD score as the difference between the full and additive LOD scores *for that fixed pair of positions*, rather than allow the full and additive models to be maximized at different positions. (This is more like what we did before, and is included just for completeness.) The same set of five thresholds is required.

```
> summary(out2, thresholds=c(6.0, 4.7, 4.4, 4.7, 2.6), what="full")

       pos1 pos2 lod.full lod.fv1 lod.int lod.add lod.av1
c1:c4  68.3   30    14.31    6.67   0.306   14.00    6.36
c6:c15 60.0   18     7.02    5.16   4.062    2.95    1.10

> summary(out2, thresholds=c(6.0, 4.7, 4.4, 4.7, 2.6), what="add")

      pos1 pos2 lod.full lod.fv1 lod.int lod.add lod.av1
c1:c4 68.3   30     14.3    6.67   0.306      14    6.36
```

```
> summary(out2, thresholds=c(6.0, 4.7, 4.4, 4.7, 2.6), what="int")

       pos1 pos2 lod.full lod.fv1 lod.int lod.add lod.av1
c6:c15   60   18     7.02    5.16    4.06    2.95    1.10
```

One may also restrict the summary to just the case of $j = k$, to look at evidence for linked QTL on each chromosome, by using `allpairs=FALSE`.

```
> summary(out2, allpairs=FALSE)

        pos1f pos2f lod.full lod.fv1 lod.int     pos1a pos2a lod.add lod.av1
c1 :c1   38.3  78.3    5.239   1.710   0.094      45.8  80.8   5.145  1.6156
c2 :c2   62.7  85.2    3.205   1.592   0.089      62.7  85.2   3.116  1.5035
c3 :c3   37.2  44.7    4.541   3.756   0.218      37.2  44.7   4.323  3.5387
c4 :c4   30.0  72.5    8.479   0.839   0.214      30.0  72.5   8.266  0.6250
c5 :c5   62.5  65.0    3.066   1.513   0.475      62.5  65.0   2.591  1.0379
c6 :c6   62.5  65.0    3.534   1.683   0.829      62.5  65.0   2.705  0.8535
c7 :c7   26.1  41.1    2.105   1.706   1.166      26.1  41.1   0.938  0.5399
c8 :c8   39.1  59.1    1.839   1.049   0.470      36.6  59.1   1.369  0.5783
c9 :c9   62.0  67.0    2.220   1.499   0.935      57.0  62.0   1.285  0.5645
c10:c10   4.7  74.7    0.773   0.512   0.415      12.2  49.7   0.357  0.0967
c11:c11  32.2  57.2    1.936   1.270   0.120       4.7  12.2   1.816  1.1502
c12:c12   1.1   6.1    2.128   1.700   0.868      11.1  16.1   1.260  0.8317
c13:c13  10.7  20.7    1.890   1.583   1.257      15.7  25.7   0.633  0.3256
c14:c14   0.0  20.0    0.822   0.716   0.277       0.0  20.0   0.545  0.4394
c15:c15  20.5  35.5    3.098   1.375   1.003      25.5  28.0   2.094  0.3713
c16:c16   0.0  12.5    2.654   2.290   1.683       0.0   7.5   0.971  0.6072
c17:c17  28.6  38.6    1.856   1.681   0.705       3.6   8.6   1.151  0.9753
c18:c18  19.7  29.7    1.408   0.920   0.225      19.7  29.7   1.183  0.6953
c19:c19   0.0  42.5    1.720   0.928   0.737       0.0  55.0   0.983  0.1914
cX :cX   28.6  31.1    1.902   0.904   0.670      13.6  23.6   1.231  0.2335
```

Note also that the degrees of freedom associated with each LOD score may be displayed, via `df=TRUE`:

```
> summary(out2, thresholds=c(6.0, 4.7, 4.4, 4.7, 2.6), df=TRUE)

Degrees of freedom:
        full fv1 int add av1
   AA:     3   2   1   2   1
   AX:     3   2   1   2   1
```

```
   XX:     3   2   1   2   1

       pos1f pos2f lod.full lod.fv1 lod.int     pos1a pos2a lod.add lod.av1
c1:c4   68.3    30    14.31    6.67   0.306      68.3  30.0   14.00    6.36
c6:c15  60.0    18     7.02    5.16   3.230      22.5  20.5    3.78    1.93
```

The old version of `summary.scantwo` is still available, though it is now called `summary.scantwo.old`.

```
> summary.scantwo.old(out2, thresholds=c(6, 4, 4))

       pos1 pos2   LODjnt -logP   LODint -logP   LODq1 -logP   LODq2 -logP
c1: c4   68   30    14.31  13.5     0.31   0.6    6.36   7.2   10.66  11.6
c6:c15   60   18     7.02   6.3     4.06   4.8    1.24   1.8    1.90   2.5
```

The permutation test with `scantwo` has also changed. At each permutation replicate, we record the maxima for each of the $M_f(j,k)$, $M_{fv1}(j,k)$, $M_i(j,k)$, $M_a(j,k)$ and $M_{av1}(j,k)$. The output is given class `"scantwoperm"`, and there is a `summary.scantwoperm` function for getting LOD thresholds. These permutations can take a very long time, and so one would generally use a multi-processor computer or cluster and do multiple shorter runs in parallel. And so we have added a function `c.scantwoperm` for combining such runs together.

We could perform the `scantwo` permutations in five batches, as follows.

```
> operm2A <- scantwo(hyper, n.perm=200)
> operm2B <- scantwo(hyper, n.perm=200)
> operm2C <- scantwo(hyper, n.perm=200)
> operm2D <- scantwo(hyper, n.perm=200)
> operm2E <- scantwo(hyper, n.perm=200)
> operm2 <- c(operm2A, operm2B, operm2C, operm2D, operm2E)
```

The 5 and 20% thresholds could be calculated as follows.

```
> summary(operm2, alpha=c(0.05,0.20))

bp (1000 permutations)
    full 2v1.int  int  add 2v1.add
5%  6.28    4.93 4.66 4.52    2.37
20% 5.29    4.22 3.89 3.71    1.93
```

The permutation results may also be used within the `summary.scantwo` function, to automatically calculate the thresholds for desired significance levels. In this case, rather than provide `thresholds`, one provides `alphas`, which again should be a vector of length 5, giving the significance levels for $M_f(j,k)$, $M_{fv1}(j,k)$, $M_i(j,k)$, $M_a(j,k)$ and $M_{av1}(j,k)$, in that order.

```
> summary(out2, perms=operm2, alphas=rep(0.05, 5))

       pos1f pos2f lod.full lod.fv1 lod.int     pos1a pos2a lod.add lod.av1
c1:c4   68.3    30    14.31    6.67   0.306      68.3  30.0   14.00    6.36
c6:c15  60.0    18     7.02    5.16   3.230      22.5  20.5    3.78    1.93
```

My version of the decision rule, in which $M_i(j,k)$ is ignored, could be obtained as follows:

```
> summary(out2, perms=operm2, alphas=c(0.05, 0.05, 0, 0.05, 0.05))

       pos1f pos2f lod.full lod.fv1 lod.int     pos1a pos2a lod.add lod.av1
c1:c4   68.3    30    14.31    6.67   0.306      68.3  30.0   14.00    6.36
c6:c15  60.0    18     7.02    5.16   3.230      22.5  20.5    3.78    1.93
```

In the case that permutation results are provided, genome-scan-adjusted p-values may also be displayed, via `pvalues=TRUE`:

```
> summary(out2, perms=operm2, alphas=c(0.05, 0.05, 0, 0.05, 0.05),
+         pvalues=TRUE)

       pos1f pos2f lod.full  pval lod.fv1  pval lod.int  pval     pos1a
c1:c4   68.3    30    14.31 0.000    6.67 0.001   0.306 1.000      68.3
c6:c15  60.0    18     7.02 0.013    5.16 0.036   3.230 0.539      22.5
       pos2a lod.add  pval lod.av1 pval
c1:c4   30.0   14.00 0.000    6.36  0.0
c6:c15  20.5    3.78 0.181    1.93  0.2
```

I have also made an important change in `plot.scantwo`. The arguments `upper` and `lower` control what is plotted in the upper-left and lower-right triangles, respectively. The options are `"full"`, `"add"`, `"cond-int"`, `"cond-add"`, and `"int"`. The case `"full"` is what was previously called `"joint"`, but this and the case `"add"` are not changed; the LOD scores for the full model (two QTLs plus interaction) and the additive model are displayed.

The other two cases, `"cond-int"`, and `"cond-add"`, are quite different. We now plot the following LOD scores:

$$\texttt{"cond-int"}: \quad \mathrm{LOD}_{fv1}(s,t) = \mathrm{LOD}_f(s,t) - M_1[c(s),c(t)]$$
$$\texttt{"cond-add"}: \quad \mathrm{LOD}_{av1}(s,t) = \mathrm{LOD}_a(s,t) - M_1[c(s),c(t)]$$

When these values are negative, they are replaced with 0. Before, we had subtracted off the maximum of the single-QTL LOD scores at the points s and t. Now we subtract off the maximum of the single-QTL LOD scores for the chromosomes containing s and t.

Note that these LOD scores will be maximized at the same positions as LOD_f and LOD_a. Indeed, except for the negative values being changed to 0's, they will have the same shape as LOD_f and LOD_a.

In the following code, we plot LOD_f in the lower triangle and LOD_i in the upper triangle, for chromosomes 1, 4, 6, and 15. The result appears in Figure 1.

```
> plot(out2, chr=c(1,4,6,15))
```

The same plot, but with the $fv1$-type LOD scores in the upper triangle, would be obtained as follows. The result appears in Figure 2.

```
> plot(out2, chr=c(1,4,6,15), upper="cond-int")
```

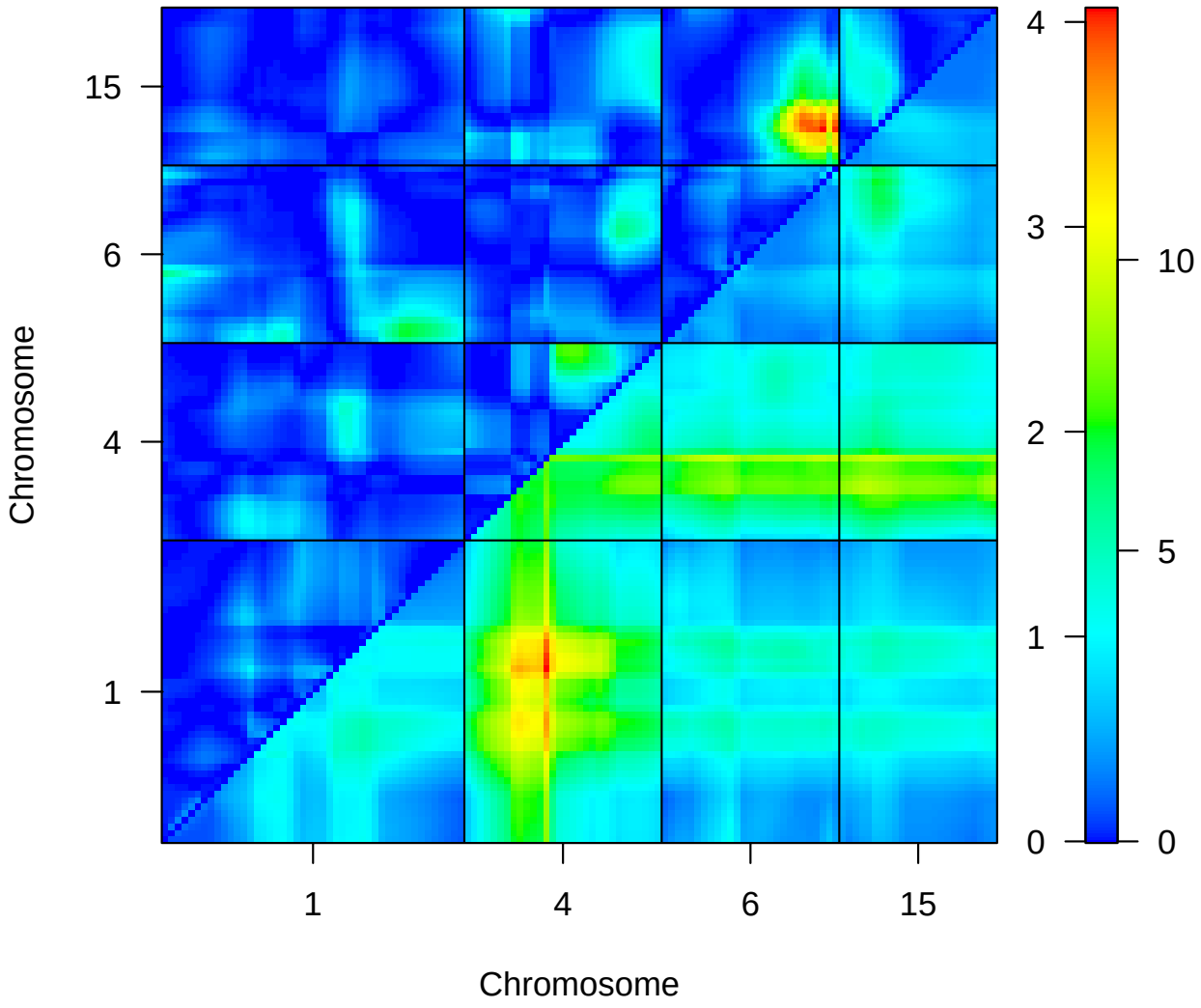

Figure 1: LOD scores for selected chromosomes for a two-dimensional scan with the `hyper` data. Epistasis LOD scores are in the upper triangle and LOD_f is in the lower triangle.

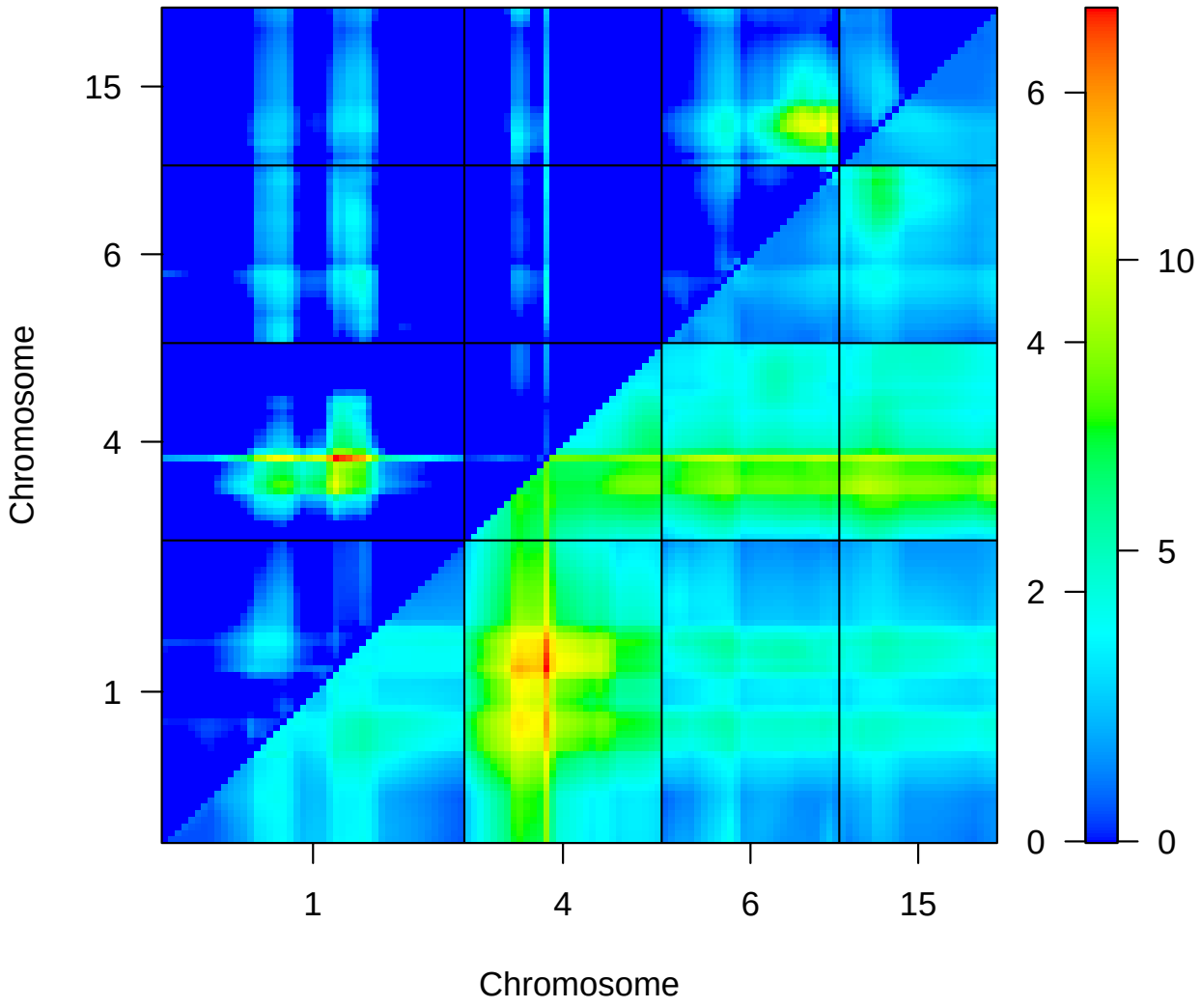

Figure 2: LOD scores for selected chromosomes for a two-dimensional scan with the `hyper` data. LOD_{fv1} is in the upper triangle and LOD_f is in the lower triangle.