Causal Model Selection Hypothesis Tests in Systems Genetics: a tutorial

Elias Chaibub Neto* and Brian S Yandell†

November 6, 2012

1 Motivation

Current efforts in systems genetics have focused on the development of statistical approaches that aim to disentangle causal relationships among molecular phenotypes in segregating populations. Model selection criterions, such as the AIC and BIC, have been widely used for this purpose, in spite of being unable to quantify the uncertainty associated with the model selection call. In this tutorial we illustrate the use of software implementing the causal model selection hypothesis tests proposed by Chaibub Neto et al. (2012).

2 Overview

This tutorial illustrates the basic functionality of the CMST routines in the `qtlhot` R package using few simulated toy examples. The analysis of a yeast genetical genomics data-set presented in Chaibub Neto et al. (2012) is reproduced in a separate package, `R/qtlyeast`. The `R/qtlhot` package depends on `R/qtl` (Broman et al. 2003), and we assume the reader is familiar with it.

3 Basic functionality

Here, we illustrate the basic functionality of the CMST routines in the `R/qtlhot` package in a toy simulated example.

```
> library(qtlhot)
```

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. The simulated cross data set, `Cross`, is composed of: 100 individuals (`n.ind = 100`); 3 chromosomes of length 100cM (`len = rep(100, 3)`); 101 unequally spaced markers per chromosome (`n.mar = 101` and `eq.spacing = FALSE`); additive genetic effect set to 1 (`add.eff = 1`); dominance genetic effect set to 0 (`dom.eff =`

*Department of Computational Biology, Sage Bionetworks, Seattle WA
†Department of Statistics, University of Wisconsin-Madison, Madison WI

0); residual variances for y_1 (sig2.1) and the other phenotypes (sig2.2) set to 0.4 and 0.1, respectively; backcross cross type (cross.type = "bc"); and phenotype data transformed to normal scores (normalize = TRUE). The argument beta = rep(0.5, 2), represents the causal effect of y_1 on the other phenotypes (i.e., coefficients of the regressions of $y_2 = 0.5\, y_1 + \epsilon$ and $y_3 = 0.5\, y_1 + \epsilon$). The length of beta controls the number of phenotypes to be simulated.

```
> set.seed(987654321)
> CMSTCross <- 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,
+                          eq.spacing = FALSE,
+                          cross.type = "bc",
+                          normalize = TRUE)
```

We compute the genotype conditional probabilities using Haldane's map function, genotype error rate of 0.0001, and setting the maximum distance between positions at which genotype probabilities were calculated to 1cM.

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

We perform QTL mapping using Haley-Knott regression (Haley and Knott 1992), and summarize the results for the 3 phenotypes. Figure 1 presents the LOD score profiles for all 3 phenotypes. The black, blue and red curves represent the LOD profiles of phenotypes y_1, y_2 and y_3, respectively.

```
> Scan <- scanone(CMSTCross, pheno.col = 1 : 3, method = "hk")
> 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
```

```
> plot(Scan, lodcolumn = 1 : 3, ylab = "LOD")
```

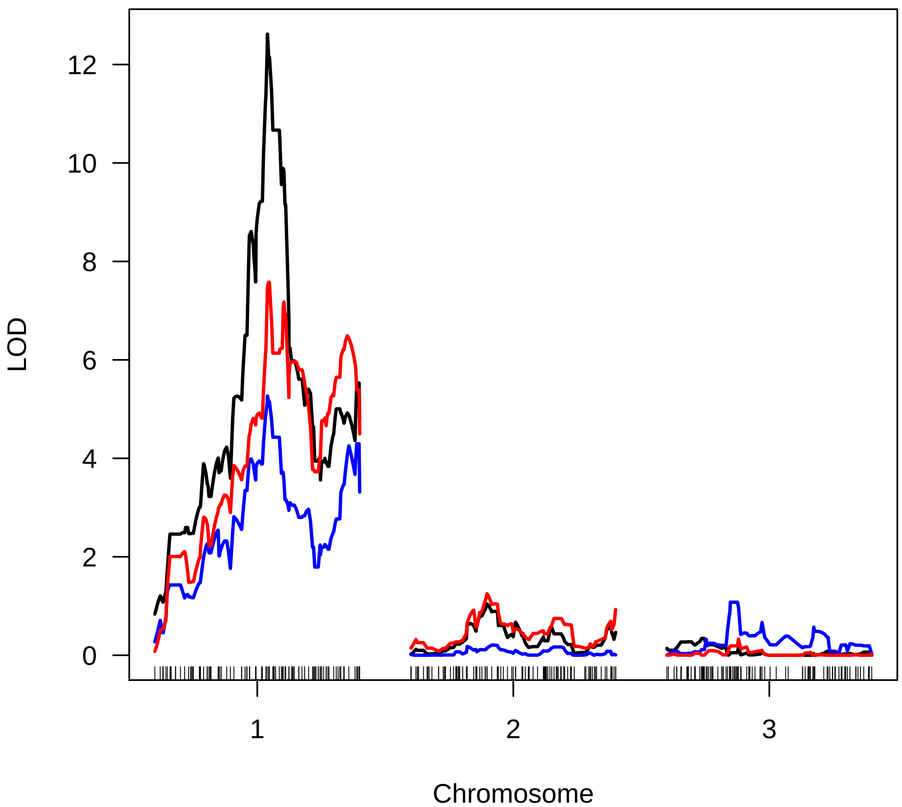

Figure 1: LOD score profiles for phenotypes y_1 (black curve), y_2 (blue curve) and y_3 (red curve).

Phenotypes y_1 and y_2 map to exactly same QTL at position 55 cM on chromosome 1. Phenotype y_3 maps to a QTL at position 55.5 cM. Whenever two phenotypes map to close, but not exactly identical, positions we are faced with the question of which QTL to use as causal anchor. Instead of making a (sometimes) arbitrary choice, our approach is to compute the joint LOD profile of both phenotypes and use the QTL detected by this joint mapping approach as the causal anchor. The function `GetCommonQtls` performs the joint QTL mapping for phenotypes whose marginal LOD peak positions are higher than a certain LOD threshold (`thr`), and are less than a fixed distance apart (`peak.dist`). The function can also handle separate additive and interacting covariates for each phenotype (`addcov1`, `intcov1`, `addcov2`, `intcov2`). In this simulated example the QTL detected by the joint analysis agreed with phenotype's y_1 QTL.

```
> commqtls <- GetCommonQtls(CMSTCross,
+                           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
```

Now, we fit our causal model selection tests for phenotypes y_1 and y_2 using the `CMSTtests` function. The `Q.chr` and `Q.pos` arguments specify the chromosome and position (in cM) of the QTL to be used as a causal anchor. The argument `method` specify which version of the CMST test should be used. The options `"par"`, `"non.par"` and `"joint"` represent, respectively, the parametric, non-parametric, joint parametric versions of the CMST test. The option `"all"` fits all three versions. The `penalty` argument specifies whether we should test statistics based on the AIC (`"aic"`), BIC (`"bic"`), or both (`"both"`) penalties. In this particular call we computed all 3 versions using both penalties fitting 6 separate CMST tests.

```
> nms <- names(CMSTCross$pheno)
> out1 <- CMSTtests(CMSTCross,
+                   pheno1 = nms[1],
+                   pheno2 = nms[2],
+                   Q.chr = 1,
+                   Q.pos = 55,
+                   addcov1 = NULL,
+                   addcov2 = NULL,
+                   intcov1 = NULL,
+                   intcov2 = NULL,
+                   method = "all",
+                   penalty = "both")
```

The output of the `CMSTtests` function is composed of a list with 17 elements. It returns the names of the phenotypes and number of individuals (`n.ind`):

```
> out1[1:3]

$pheno1
[1] "y1"

$pheno2
```

```
[1] "y2"

$n.ind
[1] 100
```

The log-likelihood scores (`loglik`) of models M_1, M_2, M_3, and M_4 (see Chaibub Neto et al. 2012 for details):

```
> out1[4]

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

The dimensions of the models (`model.dim`):

```
> out1[5]

$model.dim
[1] 6 6 6 7
```

The R^2 values (`R2`) relative to the regression of phenotypes 1 and 2 on the causal anchor:

```
> out1[6]

$R2
[1] 0.4407170 0.2153583
```

The covariance matrix (`S.hat`) with the variances and covariances of the penalized log-likelihood ratios of models $M_1 \times M_2$, $M_1 \times M_3$, $M_1 \times M_4$, $M_2 \times M_3$, $M_2 \times M_4$, and $M_3 \times M_4$:

```
> out1[7]

$S.hat
            [,1]        [,2]         [,3]         [,4]         [,5]        [,6]
[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  0.001156681 -0.009808958 -0.01096564
[4,] -0.27544421  0.37598393  0.001156681  0.651428142  0.276600893 -0.37482725
[5,] -0.25128896  0.02531193 -0.009808958  0.276600893  0.241480006 -0.03512089
[6,]  0.02415525 -0.35067200 -0.010965639 -0.374827248 -0.035120888  0.33970636
```

The BIC scores (`BICs`):

```
> out1[8]

$BICs
[1] 274.6946 308.5518 310.7917 279.2030
```

The BIC-based penalized log-likelihood test statistics (`Z.bic`):

```
> out1[9]

$Z.bic
     [,1]     [,2]      [,3]      [,4]
[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
```

The BIC-based model selection p-values for the parametric CMST (`pvals.p.BIC`), non-parametric CMST (`pvals.np.BIC`) and joint parametric CMST (`pvals.j.BIC`):

```
> out1[10:12]

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

$pvals.np.BIC
[1] 6.289575e-06 9.999977e-01 9.999999e-01 1.000000e+00

$pvals.j.BIC
[1] 0.003779474 0.999946885 0.999669186 1.000000000
```

The analogous AIC-based quantities:

```
> out1[13:17]

$AICs
[1] 259.0636 292.9208 295.1606 260.9668

$Z.aic
     [,1]     [,2]      [,3]      [,4]
[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
[1] 6.289575e-06 9.999977e-01 1.000000e+00 9.999977e-01

$pvals.j.AIC
[1] 0.005993868 0.999946885 0.999669186 1.000000000
```

The function `CMSTtests` can also computes CMST tests of a single phenotype against a list of phenotypes. Its output is less detailed though. In this particular call we test y_1 against y_2 and y_3.

```
> out2 <- CMSTtests(CMSTCross,
+                   pheno1 = nms[1],
+                   pheno2 = nms[-1],
+                   Q.chr = 1,
+                   Q.pos = 55.5,
+                   addcov1 = NULL,
+                   addcov2 = NULL,
+                   intcov1 = NULL,
+                   intcov2 = NULL,
+                   method = "all",
+                   penalty = "both")
> out2
```

```
$R2s
      R2.Y1 ~ Q R2.Y2 ~ Q
y1_y2 0.4286585 0.2112760
y1_y3 0.4286585 0.2945801

$AIC.stats
         AIC.1    AIC.2    AIC.3    AIC.4     z.12     z.13      z.14      z.23
y1_y2 261.1967 293.4397 297.8127 263.0819 3.136952 3.034372 2.6436961 0.2659898
y1_y3 256.9466 278.0272 311.4368 258.2783 2.177343 3.876750 0.8229369 2.0030490
           z.24      z.34
y1_y2 -3.084095 -2.975873
y1_y3 -2.329987 -4.023391

$BIC.stats
         BIC.1    BIC.2    BIC.3    BIC.4     z.12     z.13     z.14      z.23
y1_y2 276.8278 309.0707 313.4437 281.3181 3.136952 3.034372 6.297065 0.2659898
y1_y3 272.5777 293.6583 327.0678 276.5145 2.177343 3.876750 2.432884 2.0030490
           z.24      z.34
y1_y2 -2.819431 -2.752652
y1_y3 -2.022629 -3.826214

$pvals.j.BIC
           pval.1    pval.2    pval.3 pval.4
y1_y2 0.003366319 0.9998806 0.9997017      1
y1_y3 0.035842249 0.9974573 0.9999900      1

$pvals.p.BIC
```

```
           pval.1    pval.2    pval.3    pval.4
y1_y2 0.001205187 0.9991464 0.9987948 1.0000000
y1_y3 0.014727493 0.9852725 0.9999471 0.9925105

$pvals.np.BIC
            pval.1    pval.2 pval.3    pval.4
y1_y2 2.346206e-06 0.9999992      1 1.0000000
y1_y3 1.758821e-03 0.9991050      1 0.9999607

$pvals.j.AIC
          pval.1    pval.2    pval.3 pval.4
y1_y2 0.01109575 0.9998806 0.9997017      1
y1_y3 0.38662933 0.9985143 0.9999950      1

$pvals.p.AIC
           pval.1    pval.2    pval.3    pval.4
y1_y2 0.004100312 0.9991464 0.9987948 0.9958997
y1_y3 0.205271925 0.9900966 0.9999713 0.7947281

$pvals.np.AIC
            pval.1    pval.2 pval.3    pval.4
y1_y2 1.608001e-05 0.9999992      1 0.9999937
y1_y3 4.431304e-02 0.9991050      1 0.9715560
```

4 Other Functions

There are several other functions involved in simulation and in data analysis that are not well documented yet. See R/qtlyeast available at GITHUB for further analysis. Here we do scans for the three traits, and create a reduced object with only high LOD values.

```
> CMSTscan <- scanone(CMSTCross, pheno.col = 1:3, method = "hk")
> CMSThigh <- highlod(CMSTscan)
```

For our purposes, we place the three traits on chromosome 1 at some arbitrary positions, with trait y1 having causal "targets" of the other two traits.

```
> traits <- names(CMSTCross$pheno)
> annot <- data.frame(name = traits, traits = traits, chr = rep(1, 3),
+  Mb.pos = c(55,10,100))
> annot$cM.pos <- annot$Mb.pos
> annot

  name traits chr Mb.pos cM.pos
1   y1     y1   1     55     55
```

```
2   y2     y2   1     10     10
3   y3     y3   1    100    100
```

```
> targets <- list(y1 = c("y2","y3"))
```

Now we used the scans (via `CMSThigh`) and the annotation to identify candidate regulators, the subset of cis-acting candidate regulators, and co-mapping targets.

```
> cand.reg <- GetCandReg(CMSThigh, annot, traits)
> cand.reg

  gene phys.chr phys.pos peak.chr peak.pos  peak.lod
1   y1        1       55        1 55.00000 12.618418
2   y2        1       10        1 55.00000  5.266431
3   y3        1      100        1 55.54525  7.577615

> cis.cand.reg <- GetCisCandReg(CMSThigh, cand.reg)
> cis.cand.reg

  gene phys.chr phys.pos peak.pos peak.lod peak.pos.lower peak.pos.upper
1   y1        1       55       55 12.61842             53       57.61146

> comap.targets <- GetCoMappingTraits(CMSThigh, cand.reg)
> comap.targets

$y1
[1] "y2" "y3"

$y2
[1] "y1" "y3"

$y3
[1] "y1" "y2"
```

Next, we perform tests to infer causal relationships.

```
> tests <- list()
> for(k in seq(names(comap.targets))) {
+   tests[[k]] <- FitAllTests(CMSTCross, pheno1 = names(comap.targets)[k],
+                         pheno2 = comap.targets[[k]],
+                         Q.chr = cand.reg[k, 4],
+                         Q.pos = cand.reg[k, 5])
+ }
```

```
pheno2 =  1
pheno2 =  2
CIT pheno2 =  y2
CIT pheno2 =  y3
pheno2 =  1
pheno2 =  2
CIT pheno2 =  y1
CIT pheno2 =  y3
pheno2 =  1
pheno2 =  2
CIT pheno2 =  y1
CIT pheno2 =  y2
```

```
> names(tests) <- names(comap.targets)
> tests <- JoinTestOutputs(comap.targets, tests)
> tests
```

```
$R2s
      R2.Y1 ~ Q R2.Y2 ~ Q
y1_y2 0.4407170 0.2153583
y1_y3 0.4407170 0.2914979
y2_y1 0.2153583 0.4407170
y2_y3 0.2153583 0.2914979
y3_y1 0.2945801 0.4286585
y3_y2 0.2945801 0.2112760

$AIC.stats
         AIC.1    AIC.2    AIC.3    AIC.4      z.12      z.13       z.14
y1_y2 259.0636 292.9208 295.1606 260.9668  3.305926 2.9966507  2.8494289
y1_y3 254.8135 278.4632 309.7396 256.4303  2.339472 4.2078724  1.3197398
y2_y1 292.9208 259.0636 295.1606 260.9668 -3.305926 0.1387598 -3.2512727
y2_y3 226.3940 216.1866 231.6614 213.0150 -1.014912 0.4428305 -2.0302949
y3_y1 278.0272 256.9466 311.4368 258.2783 -2.177343 2.0030490 -2.3299872
y3_y2 215.7506 226.9129 231.7444 212.9632  1.098857 1.2871586 -0.5813799
           z.23       z.24      z.34
y1_y2 0.1387598 -3.2512727 -2.933361
y1_y3 1.9587743 -2.4119422 -4.279798
y2_y1 2.9966507  2.8494289 -2.933361
y2_y3 1.2380872 -0.6314165 -2.084435
y3_y1 3.8767496  0.8229369 -4.023391
y3_y2 0.3909483 -2.0014294 -2.073127

$BIC.stats
         BIC.1    BIC.2    BIC.3    BIC.4      z.12      z.13        z.14
```

```
y1_y2 274.6946 308.5518 310.7917 279.2030  3.305926 2.9966507  6.74974480
y1_y3 270.4445 294.0943 325.3707 274.6665  2.339472 4.2078724  3.44622282
y2_y1 308.5518 274.6946 310.7917 279.2030 -3.305926 0.1387598 -2.98619967
y2_y3 242.0250 231.8176 247.2924 231.2511 -1.014912 0.4428305 -1.63495434
y3_y1 293.6583 272.5777 327.0678 276.5145 -2.177343 2.0030490 -2.02262858
y3_y2 231.3816 242.5439 247.3754 231.1994  1.098857 1.2871586 -0.03799597
           z.23       z.24      z.34
y1_y2 0.1387598 -2.9861997 -2.709873
y1_y3 1.9587743 -2.1267544 -4.070649
y2_y1 2.9966507  6.7497448 -2.709873
y2_y3 1.2380872 -0.1127673 -1.793211
y3_y1 3.8767496  2.4328842 -3.826214
y3_y2 0.3909483 -1.6276524 -1.785559

$pvals.j.BIC
           pval.1      pval.2    pval.3    pval.4
y1_y2 0.003779474 0.999946885 0.9996692 1.0000000
y1_y3 0.023770207 0.998342606 0.9999985 1.0000000
y2_y1 0.999946885 0.003779474 0.9996692 1.0000000
y2_y3 0.991331495 0.727786029 0.9881747 0.9533151
y3_y1 0.997457264 0.035842249 0.9999900 1.0000000
y3_y2 0.708326515 0.990042453 0.9878292 0.9698693

$pvals.p.BIC
           pval.1      pval.2    pval.3    pval.4
y1_y2 0.001364817 0.999526684 0.9986352 1.0000000
y1_y3 0.009655499 0.990344501 0.9999871 0.9997158
y2_y1 0.999526684 0.001364817 0.9986352 1.0000000
y2_y3 0.948970690 0.544892461 0.9635304 0.4551075
y3_y1 0.985272507 0.014727493 0.9999471 0.9925105
y3_y2 0.515154554 0.948200693 0.9629147 0.4848454

$pvals.np.BIC
            pval.1       pval.2    pval.3    pval.4
y1_y2 6.289575e-06 9.999977e-01 0.9999999 1.0000000
y1_y3 2.043886e-04 9.999084e-01 1.0000000 0.9999997
y2_y1 9.999977e-01 6.289575e-06 0.9999999 1.0000000
y2_y3 9.556870e-01 8.643735e-01 0.9997956 0.1841008
y3_y1 9.991050e-01 1.758821e-03 1.0000000 0.9999607
y3_y2 8.158992e-01 9.895106e-01 0.9997956 0.2420592

$pvals.j.AIC
           pval.1      pval.2    pval.3    pval.4
y1_y2 0.005993868 0.999946885 0.9996692 1.0000000
```

```
y1_y3 0.195697275 0.998720666 0.9999990 1.0000000
y2_y1 0.999946885 0.005993868 0.9996692 1.0000000
y2_y3 0.997728473 0.880223915 0.9949729 0.6993183
y3_y1 0.998514322 0.386629331 0.9999950 1.0000000
y3_y2 0.875137894 0.997069515 0.9947476 0.7328496

$pvals.p.AIC
           pval.1      pval.2    pval.3    pval.4
y1_y2 0.002189889 0.999526684 0.9986352 0.9978101
y1_y3 0.093460955 0.992066102 0.9999906 0.9065390
y2_y1 0.999526684 0.002189889 0.9986352 0.9978101
y2_y3 0.978836716 0.736115878 0.9814397 0.2638841
y3_y1 0.990096585 0.205271925 0.9999713 0.7947281
y3_y2 0.719507792 0.977326933 0.9809198 0.2804922

$pvals.np.AIC
            pval.1       pval.2    pval.3     pval.4
y1_y2 6.289575e-06 9.999977e-01 1.0000000 0.99999765
y1_y3 3.318560e-03 9.999084e-01 1.0000000 0.99824118
y2_y1 9.999977e-01 6.289575e-06 1.0000000 0.99999765
y2_y3 9.823999e-01 9.895106e-01 0.9997956 0.02844397
y3_y1 9.991050e-01 4.431304e-02 1.0000000 0.97155603
y3_y2 9.823999e-01 9.895106e-01 0.9997956 0.02844397

$pvals.cit
         pval.1       pval.2
y2 1.641765e-06 7.741579e-01
y3 1.770222e-06 6.029575e-01
y1 7.741579e-01 1.641765e-06
y3 2.692473e-02 8.015192e-04
y1 4.219075e-01 4.617830e-06
y2 1.351854e-04 3.160420e-02

$phenos
     [,1] [,2]
[1,] "y1" "y2"
[2,] "y1" "y3"
[3,] "y2" "y1"
[4,] "y2" "y3"
[5,] "y3" "y1"
[6,] "y3" "y2"
```

Finaly, we compare the inferred causal relationships to the known `targets` to assess precision, true positive rate and false positive rate.

```
> PrecTpFpMatrix(alpha = seq(0.01, 0.10, by = 0.01),
+   val.targets = targets, all.orfs = CMSThigh$names, tests = tests,
+   cand.reg = cand.reg, cis.cand.reg = cis.cand.reg)

$Prec1
       0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1
aic    1.00 1.00 1.00    1    1    1    1    1    1   1
bic    1.00 1.00 1.00    1    1    1    1    1    1   1
j.bic  1.00 1.00 1.00    1    1    1    1    1    1   1
p.bic  1.00 1.00 1.00    1    1    1    1    1    1   1
np.bic 1.00 1.00 1.00    1    1    1    1    1    1   1
j.aic  1.00 1.00 1.00    1    1    1    1    1    1   1
p.aic  1.00 1.00 1.00    1    1    1    1    1    1   1
np.aic 1.00 1.00 1.00    1    1    1    1    1    1   1
cit    0.67 0.67 0.67    1    1    1    1    1    1   1

$Prec2
       0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1
aic       1    1    1    1    1    1    1    1    1   1
bic       1    1    1    1    1    1    1    1    1   1
j.bic     1    1    1    1    1    1    1    1    1   1
p.bic     1    1    1    1    1    1    1    1    1   1
np.bic    1    1    1    1    1    1    1    1    1   1
j.aic     1    1    1    1    1    1    1    1    1   1
p.aic     1    1    1    1    1    1    1    1    1   1
np.aic    1    1    1    1    1    1    1    1    1   1
cit       1    1    1    1    1    1    1    1    1   1

$Tp1
       0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1
aic       2    2    2    2    2    2    2    2    2   2
bic       2    2    2    2    2    2    2    2    2   2
j.bic     1    1    2    2    2    2    2    2    2   2
p.bic     2    2    2    2    2    2    2    2    2   2
np.bic    2    2    2    2    2    2    2    2    2   2
j.aic     1    1    1    1    1    1    1    1    1   1
p.aic     1    1    1    1    1    1    1    1    1   2
np.aic    2    2    2    2    2    2    2    2    2   2
cit       2    2    2    2    2    2    2    2    2   2

$Tp2
       0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1
aic       2    2    2    2    2    2    2    2    2   2
bic       2    2    2    2    2    2    2    2    2   2
```

```
j.bic      1    1    2    2    2    2    2    2    2   2
p.bic      2    2    2    2    2    2    2    2    2   2
np.bic     2    2    2    2    2    2    2    2    2   2
j.aic      1    1    1    1    1    1    1    1    1   1
p.aic      1    1    1    1    1    1    1    1    1   2
np.aic     2    2    2    2    2    2    2    2    2   2
cit        2    2    2    2    2    2    2    2    2   2

$Fp1
       0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1
aic       0    0    0    0    0    0    0    0    0   0
bic       0    0    0    0    0    0    0    0    0   0
j.bic     0    0    0    0    0    0    0    0    0   0
p.bic     0    0    0    0    0    0    0    0    0   0
np.bic    0    0    0    0    0    0    0    0    0   0
j.aic     0    0    0    0    0    0    0    0    0   0
p.aic     0    0    0    0    0    0    0    0    0   0
np.aic    0    0    0    0    0    0    0    0    0   0
cit       1    1    1    0    0    0    0    0    0   0

$Fp2
       0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1
aic       0    0    0    0    0    0    0    0    0   0
bic       0    0    0    0    0    0    0    0    0   0
j.bic     0    0    0    0    0    0    0    0    0   0
p.bic     0    0    0    0    0    0    0    0    0   0
np.bic    0    0    0    0    0    0    0    0    0   0
j.aic     0    0    0    0    0    0    0    0    0   0
p.aic     0    0    0    0    0    0    0    0    0   0
np.aic    0    0    0    0    0    0    0    0    0   0
cit       0    0    0    0    0    0    0    0    0   0
```

5 References

1. Brem R., L. Kruglyak, 2005 The landscape of genetic complexity across 5,700 gene expression trait in yeast. PNAS **102:** 1572-1577.

2. Broman K., H. Wu, S. Sen, G. A. Churchill, 2003 R/qtl: QTL mapping in experimental crosses. Bioinformatics **19**: 889-890.

3. Chaibub Neto et al. (2012) Causal model selection hypothesis tests in systems genetics. Genetics (under review)

4. Churchill G. A., R. W. Doerge, 1994 Empirical threshold values for quantitative trait mapping. Genetics **138**: 963-971.

5. Haley C., S. Knott, 1992 A simple regression method for mapping quantitative trait loci in line crosses using flanking markers. Heredity **69**: 315-324.

6. Hughes T. R., M. J. Marton, A. R. Jones, C. J. Roberts, R. Stoughton, et al, 2000 Functional discovery via a compendium of expression profiles. Cell **102:** 109-116.

7. Manichaikul A., J. Dupuis, S. Sen, and K. W. Broman, 2006 Poor performance of bootstrap confidence intervals for the location of a quantitative trait locus. Genetics **174:** 481-489.

8. Schadt E. E., J. Lamb, X. Yang, J. Zhu, S. Edwards, et al., 2005 An integrative genomics approach to infer causal associations between gene expression and disease. Nature Genetics **37**: 710-717.

9. Zhu J., B. Zhang, E. N. Smith, B. Drees, R. B. Brem, L. Kruglyak, R. E. Bumgarner, E. E. Schadt, 2008 Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. Nature Genetics **40**: 854-861.