Hotspots and Causal Inference For Yeast Data

Elias Chaibub Neto[*]and Brian S Yandell[†]

October 2, 2012

Here we reproduce the analysis of the budding yeast genetical genomics data-set presented in Chaibub Neto et al. (2012). The data represents a cross of a standard yeast laboratory strain, and a wild isolate from a California vineyard (Brem and Kruglyak 2005). It consists of expression measurements on 5,740 transcripts measured on 112 segregant strains with dense genotype data on 2,956 markers. Processing of the expression measurements raw data was done as described in Brem and Kruglyak (2005), with an additional step of converting the processed measurements to normal quantiles by the transformation $\Phi^{-1}[(r_i - 0.5)/112]$, where Φ is the standard normal cumulative density function, and the r_i are the ranks.

The data were provided by Rachel Brem and further edited by Jun Zhu and Bin Zhang (formerly of Sage Bionetworks). Elias Chaibub Neto and Brian Yandell have organized the data and analysis into this R statistical package.

We first load the yeast cross object (`yeast.orf`), and compute the conditional genotype 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 2cM.

```
> library(qtlhot)
> library(qtlyeast)
> data(yeast.orf)
> yeast.orf <- calc.genoprob(yeast.orf, step = 2)
```

The following command does an genome scan for QTL using R/qtl for all the traits using Haley-Knott regression (Haley and Knott 1992).

```
> scan.orf <- scanone(yeast.orf, pheno.col = seq(nphe(yeast.orf)), method = "hk")
```

To save space, we work with only the genome regions that are above the single trait LOD threshold and within 1.5 LOD of the maximum per chromosome. We do this after we determine the permutation LOD threshold below.

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

1 Hotspot Inference

Plan of action: 1. Find Churchill-Doerge 5% LOD threshold 2. Determine hotspot counts relative to LOD threshold (Jansen method) 3. conduct permutation test (using CHTC) 4. report Jansen and Chaibub-Neto results 5. identify hotspots.

1.1 Churchill-Doerge LOD threshold

Since we are using normal scores on the traits, we need only conduct permutation threshold calculation with a normal response. Here we create one trait and then do 1000 permutations. We have saved this as `perm.orf`.

```
> cross <- yeast.orf
> cross$pheno <- data.frame(norm = rnorm(nind(cross)))

> set.seed(12345)
> perm.orf <- scanone(cross, method = "hk", n.perm = 1000)

> data(perm.orf)
> summary(perm.orf)

LOD thresholds (1000 permutations)
     lod
5%  3.48
10% 3.08

> lod.thr <- c(summary(perm.orf, alpha = 0.05))
```

Now we save only the high lods of the `scan.orf` object to save space.

```
> highlod.orf <- highlod(scan.orf, lod.thr = lod.thr, drop.lod = 1.5)
```

This takes considerable time, so we have actually saved the completed scans as object `scan.orf`. However, the `scan.orf` object is 203Mb, so we don't keep it in the package. Instead we have saved `highlod.orf`.

```
> data(highlod.orf)
```

1.2 Hotspots for Yeast Data above LOD threshold

Now we show the hotspots. We can get summary and plot from `highlod.orf`, but this is actually first turned into a `hotsize` object. We will use this directly.

```
> hotsize.orf <- hotsize(highlod.orf, lod.thr = lod.thr)
> summary(hotsize.orf)
```

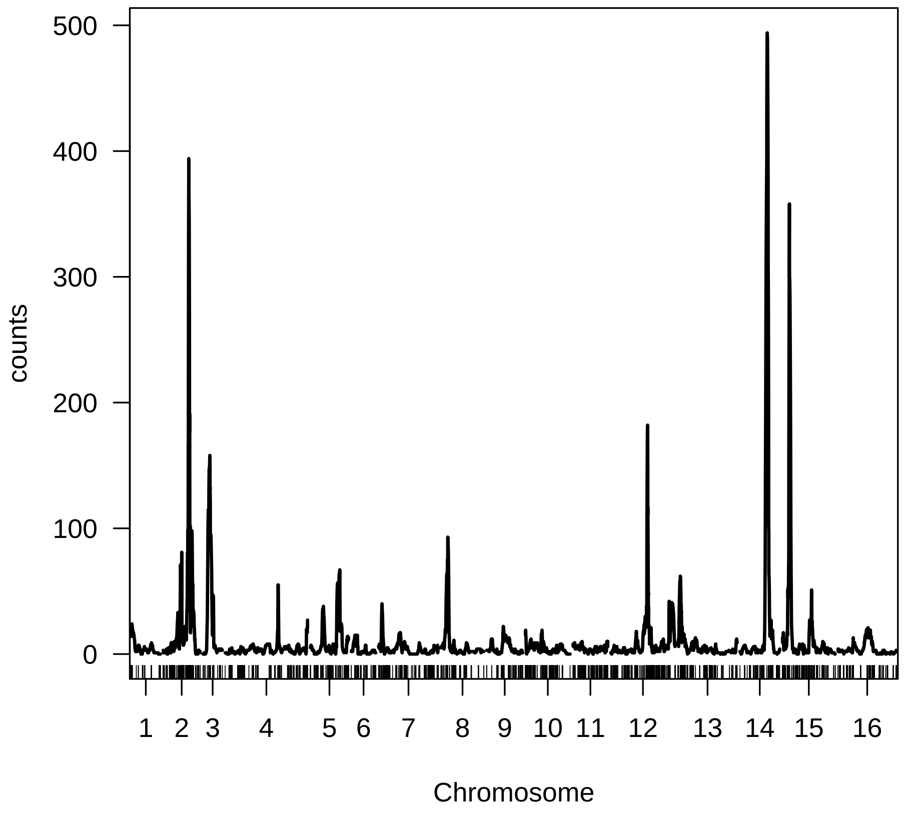
```
hotsize elements:  chr pos max.N
LOD threshold: 3.475609

                        chr    pos max.N
c1.loc6                   1   6.00    24
YBR154C_chr2@548401       2 224.11   394
c3.loc60                  3  60.01   158
YDR233C_chr4@929769       4 464.02    55
c4.loc464                 4 464.02    55
YDR233C.1_chr4@929895     4 464.92    55
c4.loc466                 4 466.02    55
YER129W_chr5@420595       5 257.84    67
c6.loc32                  6  32.01    15
NFL013C.2_chr6@5854       6  32.71    15
NFL013C.3_chr6@5870       6  32.71    15
NFL013C.4_chr6@5872       6  32.71    15
NFL013C.5_chr6@5873       6  32.71    15
c6.loc34                  6  34.01    15
c6.loc48                  6  48.01    15
c6.loc50                  6  50.01    15
YFL056C_chr6@15106        6  51.87    15
c6.loc52                  6  52.01    15
YFL056C.1_chr6@15195      6  53.70    15
YFL055W_chr6@18384        6  53.70    15
YFL055W.1_chr6@18546      6  53.70    15
YFL055W.2_chr6@18552      6  53.70    15
NFL012W_chr6@28029        6  53.70    15
NFL012W.1_chr6@28041      6  53.70    15
NFL012W.2_chr6@28050      6  53.70    15
YFL051C_chr6@30378        6  53.70    15
YGL235W_chr7@55458        7  29.33    40
YGL235W.1_chr7@55464      7  29.33    40
NHR001C.2_chr8@111682     8  58.51    93
NHR001C.3_chr8@111683     8  58.51    93
NHR001C.4_chr8@111686     8  58.51    93
NHR001C.5_chr8@111687     8  58.51    93
NHR001C.6_chr8@111690     8  58.51    93
c9.loc146                 9 146.02    22
YJL212C_chr10@34086      10   0.94    19
c10.loc144               10 144.02    19
YKL141W_chr11@180221     11  80.46    10
c11.loc298               11 298.00    10
gKR07_chr11@643655       11 299.85    10
gKR07.1_chr11@645247     11 299.85    10
```

gKR07.2_chr11@645253	11	299.85	10
gKR07.3_chr11@645409	11	299.85	10
gKR07.4_chr11@645547	11	299.85	10
gKR07.5_chr11@645577	11	299.85	10
gKR07.6_chr11@645589	11	299.85	10
gKR07.7_chr11@645727	11	299.85	10
gKR07.8_chr11@645998	11	299.85	10
gKR07.9_chr11@646028	11	299.85	10
gKR07.10_chr11@646042	11	299.85	10
gKR07.11_chr11@646049	11	299.85	10
gKR07.12_chr11@646054	11	299.85	10
gKR07.13_chr11@646060	11	299.85	10
gKR07.14_chr11@646065	11	299.85	10
gKR07.15_chr11@646146	11	299.85	10
gKR07.16_chr11@646245	11	299.85	10
gKR07.17_chr11@646285	11	299.85	10
gKR07.18_chr11@646408	11	299.85	10
gKR07.19_chr11@646556	11	299.85	10
gKR07.20_chr11@646820	11	299.85	10
gKR07.21_chr11@647318	11	299.85	10
gKR07.22_chr11@647807	11	299.85	10
gKR07.23_chr11@647868	11	299.85	10
gKR07.24_chr11@647911	11	299.85	10
gKR07.25_chr11@648010	11	299.85	10
gKR07.26_chr11@648055	11	299.85	10
gKR07.27_chr11@648146	11	299.85	10
gKR07.28_chr11@648221	11	299.85	10
gKR07.29_chr11@648280	11	299.85	10
gKR07.30_chr11@648354	11	299.85	10
gKR07.31_chr11@648430	11	299.85	10
gKR07.32_chr11@648545	11	299.85	10
gKR07.33_chr11@648673	11	299.85	10
gKR07.34_chr11@648742	11	299.85	10
YKR102W_chr11@649174	11	299.85	10
YKR102W.1_chr11@649228	11	299.85	10
YKR102W.2_chr11@649234	11	299.85	10
YKR102W.3_chr11@649240	11	299.85	10
YKR102W.4_chr11@649258	11	299.85	10
YKR102W.5_chr11@649270	11	299.85	10
YKR102W.6_chr11@649288	11	299.85	10
YKR102W.7_chr11@649300	11	299.85	10
YKR102W.8_chr11@649324	11	299.85	10
YKR102W.9_chr11@649348	11	299.85	10
YKR102W.10_chr11@649408	11	299.85	10

```
YKR102W.11_chr11@649414  11 299.85    10
YKR102W.12_chr11@649438  11 299.85    10
YKR102W.13_chr11@649468  11 299.85    10
YKR102W.14_chr11@649480  11 299.85    10
gKR08_chr11@649545       11 299.85    10
gKR08.1_chr11@649704     11 299.85    10
gKR08.2_chr11@650208     11 299.85    10
gKR08.3_chr11@650256     11 299.85    10
gKR08.4_chr11@650325     11 299.85    10
gKR08.5_chr11@650334     11 299.85    10
gKR08.6_chr11@650626     11 299.85    10
gKR08.7_chr11@650638     11 299.85    10
gKR08.8_chr11@650644     11 299.85    10
gKR08.9_chr11@650800     11 299.85    10
gKR08.10_chr11@650845    11 299.85    10
gKR08.11_chr11@651100    11 299.85    10
gKR08.12_chr11@651178    11 299.85    10
gKR08.13_chr11@651668    11 299.85    10
gKR08.14_chr11@652016    11 299.85    10
gKR08.15_chr11@652166    11 299.85    10
gKR08.16_chr11@652304    11 299.85    10
gKR08.17_chr11@652394    11 299.85    10
gKR08.18_chr11@652406    11 299.85    10
gKR08.19_chr11@652530    11 299.85    10
gKR08.20_chr11@652596    11 299.85    10
gKR08.21_chr11@652636    11 299.85    10
gKR08.22_chr11@652646    11 299.85    10
gKR08.23_chr11@652684    11 299.85    10
gKR08.24_chr11@652688    11 299.85    10
gKR08.25_chr11@652752    11 299.85    10
gKR08.26_chr11@652835    11 299.85    10
gKR08.27_chr11@652923    11 299.85    10
gKR08.28_chr11@653047    11 299.85    10
gKR08.29_chr11@653219    11 299.85    10
gKR08.30_chr11@653244    11 299.85    10
gKR08.31_chr11@653507    11 299.85    10
gKR08.32_chr11@653697    11 299.85    10
c11.loc300               11 300.00    10
gKR08.33_chr11@653763    11 301.67    10
c11.loc302               11 302.00    10
gKR08.35_chr11@654154    11 303.49    10
gKR08.36_chr11@654229    11 303.49    10
gKR08.37_chr11@654284    11 303.49    10
gKR08.38_chr11@654788    11 303.49    10
```

```
gKR08.39_chr11@654923     11 303.49    10
gKR08.40_chr11@655159     11 303.49    10
gKR08.41_chr11@655316     11 303.49    10
gKR08.42_chr11@655559     11 303.49    10
gKR08.43_chr11@655634     11 303.49    10
gKR08.44_chr11@655648     11 303.49    10
gKR08.45_chr11@655662     11 303.49    10
gKR08.46_chr11@655678     11 303.49    10
YKR103W_chr11@655967      11 303.49    10
YKR103W.1_chr11@655991    11 303.49    10
YKR103W.2_chr11@656099    11 303.49    10
YKR103W.3_chr11@656105    11 303.49    10
YKR103W.4_chr11@656141    11 303.49    10
YKR103W.5_chr11@656147    11 303.49    10
YKR103W.6_chr11@656171    11 303.49    10
YKR103W.7_chr11@656177    11 303.49    10
YKR103W.8_chr11@656225    11 303.49    10
YKR103W.9_chr11@656231    11 303.49    10
YKR103W.10_chr11@656279   11 303.49    10
YKR104W_chr11@657035      11 303.49    10
YKR104W.1_chr11@657065    11 303.49    10
YKR104W.2_chr11@657071    11 303.49    10
YKR104W.3_chr11@657077    11 303.49    10
YKR104W.4_chr11@657107    11 303.49    10
YKR104W.5_chr11@657119    11 303.49    10
c11.loc304                11 304.00    10
YKR104W.6_chr11@657197    11 305.31    10
c11.loc306                11 306.00    10
YKR104W.11_chr11@657329   11 307.13    10
YKR104W.12_chr11@657359   11 307.13    10
YKR104W.13_chr11@657365   11 307.13    10
gKR09_chr11@663632        11 307.13    10
gKR09.1_chr11@663758      11 307.13    10
gKR09.2_chr11@666052      11 307.13    10
YLR257W_chr12@659357      12 323.64   182
YLR258W_chr12@662627      12 323.64   182
c12.loc324                12 324.02   182
c13.loc20                 13  20.03    62
c14.loc240                14 240.01   494
gOL02_chr15@170945        15  61.14   358
gOL02.1_chr15@174364      15  61.14   358
c16.loc264                16 264.07    21
```

```
> plot(hotsize.orf)
```



This shows hotspots, but there is no way yet to assess their significance. To do that, we must run some further permutations across all the traits together, preserving their correlation structure. This takes even more time, so we will do it offline and show the results.

2 Causal Inference

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. We illustrate analysis of the Brem and Kruglyak (2005 PNAS) data using software implemented in R/qtlhot.

In order to evaluate the precision of the causal predictions made by the methods we used validated causal relationships extracted from a data-base of 247 knock-out experiments in yeast (Hughes et al. 2000, Zhu et al. 2008). In each of these experiments, one gene was knocked-out, and the expression levels of the remainder genes in control and knocked-out strains were interrogated for differential expression. The set of differentially expressed genes form the knock-out signature (ko-signature) of the knocked-out gene (ko-gene), and show direct evidence of a causal effect of the ko-gene on the ko-signature genes.

Next, we load a yeast annotation data.frame, `yeast.annot`, that provides the orf, gene symbol, and chromosome location (in both Mb and cM) of each one of the 5,740 transcripts. (This information will be needed to determine which ko-genes show significant QTLs.) [Need to describe where these data come from.]

```
> data(yeast.annot)
> head(yeast.annot)

          orf  gene chr   Mb.pos   cM.pos
3952 YAL001C  TFC3   1 0.151168 102.4066
3951 YAL002W  VPS8   1 0.143709 101.3745
3950 YAL003W  EFB1   1 0.142176 101.1623
1330 YAL005C  SSA1   1 0.141433 101.0595
3934 YAL007C  ERP2   1 0.138347 100.1245
3933 YAL008W FUN14   1 0.136916 100.1245
```

Next, we load the list of ko-signatures derived from the knock-out experiments in Hughes et al. (2000) and Zhu et al. (2008). We show below the first knock-out signature.

```
> data(ko.list)
> length(ko.list)

[1] 247

> ko.list[[1]]

 [1] "YAR073W"   "YBL013W"   "YBL032W"   "YBL042C"   "YBL054W"   "YBL064C"
 [7] "YBR013C"   "YBR054W"   "YBR072W"   "YBR126C"   "YBR155W"   "YBR186W"
[13] "YCL030C"   "YDL038C"   "YDL234C"   "YDL244W"   "YDR001C"   "YDR018C"
[19] "YDR055W"   "YDR077W"   "YDR085C"   "YDR399W"   "YDR518W"   "YDR533C"
[25] "YDR534C"   "YER055C"   "YER062C"   "YFL014W"   "YFL030W"   "YFL058W"
[31] "YGL156W"   "YGL162W"   "YGL187C"   "YGL234W"   "YGR032W"   "YGR043C"
[37] "YGR138C"   "YGR161C"   "YGR171C"   "YGR213C"   "YGR250C"   "YHL040C"
[43] "YHR087W"   "YHR096C"   "YHR104W"   "YHR216W"   "YIL125W"   "YJL034W"
[49] "YJL054W"   "YJL116C"   "YJR151C"   "YKL029C"   "YKL090W"   "YKL097W.A"
[55] "YKL163W"   "YKL165C"   "YKR061W"   "YLL019C"   "YLL060C"   "YLR120C"
[61] "YLR121C"   "YLR142W"   "YLR178C"   "YLR194C"   "YLR350W"   "YLR359W"
[67] "YML130C"   "YML131W"   "YMR040W"   "YMR090W"   "YMR173W"   "YMR181C"
[73] "YMR300C"   "YNL112W"   "YNL134C"   "YNL160W"   "YNL220W"   "YOL151W"
[79] "YOL031C"   "YOR173W"   "YOR289W"   "YOR338W"   "YOR382W"   "YPL088W"
[85] "YPL277C"   "YPR156C"   "YAR075W"   "YDR243C"   "YFR024C.A" "YOL053C.A"
```

Next, we determine which of the 247 ko-genes also showed a significant QTL in our data set, according to a permutation test (Churchill and Doerge 1994) aiming to control GWER < 0.05. For each one of the ko-genes with a significant QTL, that is, with LOD score above `lod.thr` $= 3.47$, the function `GetCandReg` returns the ko-gene's chromosome (`phys.chr`) and physical position in cM (`phys.pos`), as well as, the LOD score (`peak.lod`) at the peak position (`peak.pos`), and the chromosome where the peak is located (`peak.chr`). In total, we observed 135 ko-genes with significant QTLs. These ko-genes are our candidate regulators. We show below the information on the first 10 candidate regulators. Note that some ko-genes map to the same chromosome where they are physically located, while other map to different chromosomes.

```
> cand.reg <- GetCandReg(highlod.orf, yeast.annot, names(ko.list))
> dim(cand.reg)

[1] 135   6

> head(cand.reg)

      gene phys.chr phys.pos peak.chr    peak.pos peak.lod
2  YMR282C       13 473.2316       14 236.0138450 3.692560
3  YER017C        5 152.3216       14 238.0138450 6.597231
7  YER069W        5 211.7280        3  54.0140660 3.975861
9  YOR058C       15 188.5460        8   0.9067482 3.569372
10 YGL017W        7 227.0394        7 221.6439074 5.894020
14 YMR055C       13 235.2625       13 246.0276440 5.578000
```

Genes that map to positions close to their physical locations are said to map in *cis* (local-linkages). Genes that map to positions away from their physical locations are said to map in *trans* (distal-linkages). There is no unambiguous way the determine how close a gene needs to map to its physical location in order to be classified as cis. Our choice is to classify a gene as cis if the 1.5-LOD support interval (Manichaikul et al. 2006) around the LOD peak contains the gene's physical location, and if the LOD score at its physical location is higher the the LOD threshold. The function `GetCisCandReg` determines which of the candidate regulators map in cis. We see that only 30 out of the 135 candidate regulators, show cis-linkages. (The additional columns `peak.pos.lower` and `peak.pos.upper` show, respectively, the lower and upper bounds of the 1.5-LOD support interval around `peak.pos`.)

```
> cis.cand.reg <- GetCisCandReg(highlod.orf, cand.reg)
> cis.cand.reg

      gene phys.chr  phys.pos peak.chr  peak.pos  peak.lod peak.pos.lower
10  YGL017W        7 227.03943        7 221.64391  5.894020      210.00371
14  YMR055C       13 235.26248       13 246.02764  5.578000      144.02764
16  YMR275C       13 467.31829       13 460.02764  5.508846      420.02764
48  YLR342W       12 402.50870       12 402.50870  8.666742      400.01961
61  YNL021W       14 278.51987       14 242.01385  4.359721      222.01385
```

```
63  YOR038C       15 179.17085       15 174.00115  5.060326       156.00115
69  YMR035W       13 167.42650       13 238.76631  4.464391       140.02764
77  YGR040W        7 251.47006        7 246.21456  8.558638       234.00371
79  YJL030W       10 157.02011       10 157.08904  8.976738       154.85559
97  YKL043W       11 151.09330       11 152.00059  9.323551       144.00059
101 YDR004W        4 237.54203        4 238.01772  4.574174       228.01772
103 YOR101W       15 236.94091       15 238.07680  5.644913       234.00115
121 YER047C        5 186.44500        5 185.20813 18.350191       183.38430
134 YJR104C       10 279.90532       10 281.49259  5.166115       267.83540
154 YLR234W       12 292.15862       12 294.71293  4.349759       292.01961
184 YHR022C        8  93.10339        8 116.23247  4.515100        86.00578
186 YHR034C        8 103.59995        8 104.00578  5.306633        86.00578
190 YJL107C       10  84.29753       10  84.05471  9.096395        82.02221
196 YMR010W       13 131.68979       13 132.02764  5.240367       115.22523
203 YMR034C       13 166.94996       13 218.02764 11.791949       144.02764
215 YOR051C       15 188.03288       15 188.54599  9.870057       184.92557
222 YLR450W       12 506.31866       12 509.03922  6.139989       494.01961
230 YBR158W        2 225.66303        2 228.00942 23.243023       225.01577
231 YHR005C        8  61.46146        8  58.51417 10.844566        57.61323
233 YCL009C        3  66.45451        3  54.01407  5.900022        44.01407
234 YCL018W        3  61.74737        3  62.01407 25.674740        60.01407
235 YNL085W       14 241.18612       14 240.01385  6.381722       220.01385
236 YOL084W       15  58.21656       15  61.14207 19.713682        58.00115
    peak.pos.upper
10       230.74609
14       260.02764
16       472.02764
48       410.01961
61       280.01385
63       184.00115
69       240.12475
77       254.00371
79       162.02221
97       158.00059
101      246.01772
103      244.00115
121      188.00539
134      292.02221
154      294.71293
184      136.00578
186      114.00578
190       94.02221
196      144.02764
203      238.76631
```

```
215      194.00115
222      554.01961
230      230.68527
231       66.00578
233       72.01407
234       64.01407
235      250.01385
236       62.00115
```

Of these, 2 (YNL135C, YMR223W) have 1.5-LOD support intervals that drop below `lod.thr`. A third gene (YMR035W) is still detected, although it has a large support interval with regions that dip below `lod.thr`. Adding the argument `lod.thr` to the call to `GetCisCandReg` restricts to LOD support intervals above `lod.thr`.

```
> cis.high <- GetCisCandReg(highlod.orf, cand.reg, lod.thr)
> cis.high
```

```
       gene phys.chr  phys.pos peak.chr  peak.pos  peak.lod peak.pos.lower
10  YGL017W        7 227.03943        7 221.64391  5.894020      210.00371
14  YMR055C       13 235.26248       13 246.02764  5.578000      144.02764
16  YMR275C       13 467.31829       13 460.02764  5.508846      420.02764
48  YLR342W       12 402.50870       12 402.50870  8.666742      400.01961
61  YNL021W       14 278.51987       14 242.01385  4.359721      222.01385
63  YOR038C       15 179.17085       15 174.00115  5.060326      156.00115
69  YMR035W       13 167.42650       13 238.76631  4.464391      140.02764
77  YGR040W        7 251.47006        7 246.21456  8.558638      234.00371
79  YJL030W       10 157.02011       10 157.08904  8.976738      154.85559
97  YKL043W       11 151.09330       11 152.00059  9.323551      144.00059
101 YDR004W        4 237.54203        4 238.01772  4.574174      228.01772
103 YOR101W       15 236.94091       15 238.07680  5.644913      234.00115
121 YER047C        5 186.44500        5 185.20813 18.350191      183.38430
134 YJR104C       10 279.90532       10 281.49259  5.166115      267.83540
154 YLR234W       12 292.15862       12 294.71293  4.349759      292.01961
184 YHR022C        8  93.10339        8 116.23247  4.515100       86.00578
186 YHR034C        8 103.59995        8 104.00578  5.306633       86.00578
190 YJL107C       10  84.29753       10  84.05471  9.096395       82.02221
196 YMR010W       13 131.68979       13 132.02764  5.240367      115.22523
203 YMR034C       13 166.94996       13 218.02764 11.791949      144.02764
215 YOR051C       15 188.03288       15 188.54599  9.870057      184.92557
222 YLR450W       12 506.31866       12 509.03922  6.139989      494.01961
230 YBR158W        2 225.66303        2 228.00942 23.243023      225.01577
231 YHR005C        8  61.46146        8  58.51417 10.844566       57.61323
233 YCL009C        3  66.45451        3  54.01407  5.900022       44.01407
234 YCL018W        3  61.74737        3  62.01407 25.674740       60.01407
235 YNL085W       14 241.18612       14 240.01385  6.381722      220.01385
```

```
236 YOL084W       15  58.21656       15  61.14207 19.713682       58.00115
    peak.pos.upper
10       230.74609
14       260.02764
16       472.02764
48       410.01961
61       280.01385
63       184.00115
69       240.12475
77       254.00371
79       162.02221
97       158.00059
101      246.01772
103      244.00115
121      188.00539
134      292.02221
154      294.71293
184      136.00578
186      114.00578
190       94.02221
196      144.02764
203      238.76631
215      194.00115
222      554.01961
230      230.68527
231       66.00578
233       72.01407
234       64.01407
235      250.01385
236       62.00115
```

For each one of the 135 candidate ko-genes, we determined which other genes also co-mapped to the same QTL of the ko-gene. The co-mapping genes represent the putative targets of a ko-gene. The function `GetCoMappingTraits` returns a list with the putative targets of each ko-gene. A gene is included in the putative target list of a ko-gene when its LOD peak is greater than `lod.thr` and the 1.5 LOD support interval around the peak contains the location of the ko-gene's QTL. The number of targets vary from ko-gene to ko-gene (from 1 to 570), and we show below the putative targets of one ko-gene (YMR275C) with 4 putative targets. In total, the 135 candidate regulators have 31,936 targets.

```
> comap.targets <- GetCoMappingTraits(highlod.orf, cand.reg)
> summary(sapply(comap.targets, length))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    1.0    63.5   188.0   236.9   479.0   569.0
```

```
> comap.targets[[7]]

[1] "YDL013W" "YGL254W" "YML069W" "YMR247C"

> length(unlist(comap.targets))

[1] 31975
```

Next, we use the function `FitAllTests` to fit the causality tests of each candidate regulator ko-gene (`pheno1`) to its putative targets (`phenos`). We use the candidate regulator's QTL (`Q.chr` and `Q.pos`) as a causal anchor. This function fits: the AIC and BIC model selection criterions (Schadt et al. 2005); the AIC- and BIC-based versions of the joint, parametric and non-parametric CMST tests (Chaibub Neto et al. 2012); and the CIT test (Millstein et al. 2009). We do not run it here because this step can take a few hours, as we perform a total of 31,936 tests for each of the 9 approaches. The function `JoinKoOutputs` joins together the outputs of the 135 separate fits of the `FitAllTests` function.

```
###### don't run

set.seed(123456789) # we fix a seed because cit uses bootstrap
for (k in 1 : 135) {
  cat("trait=", k, "\n")
  out <- FitAllTests(cross = yeast.orf,
                     pheno1 = cand.reg[k, 1],
                     pheno2 = comap.targets[[k]],
                     Q.chr = cand.reg[k, 4],
                     Q.pos = cand.reg[k, 5])
  save(out, file=paste("output_ko_validation", cand.reg[k, 1], "RData",
       sep = "."), compress = TRUE)
}
######
ko.tests <- JoinTestOutputs(x = comap.targets)
save(ko.tests, file = "ko.tests.RData", compress=TRUE)
```

After loading the joined results we use the function `PrecTpFpMatrix` to summarize the performance of the different methods in terms of "biologically validated" true positives, false positives and precision, of the inferred causal relations. Since we already have the results of the knock-out experiments (recall that `ko.list.all` holds the ko-signatures of the ko-genes), we define a true positive as a statistically significant causal relation between a ko-gene and a putative target gene, when the putative target gene belongs to the ko-signature of the ko-gene. Similarly, we define a false positive as a statistically significant causal relation between a ko-gene and a putative target gene when the target gene doesn't belong to the ko-signature. (For the AIC and BIC methods, that do not provide a p-value measuring the significance of the causal call, we simply use the detected causal relations in the computation of true and false positives). The "validated precision", is computed as the ratio of true positives by the sum of true and false positives. The PrecTpFpMatrix computes these measures to both all ko-genes, and to cis ko-genes only. The argument alpha sets the significant levels at each the summaries are computed.

[This has the file name passed down, which is messy. Need to clean up.]

```
> data(ko.tests)
> roc.aux <- PrecTpFpMatrix(alpha = seq(0.01, 0.10, by = 0.01),
+                         nms = cand.reg[, 1],
+                         val.targets = ko.list.all,
+                         all.orfs = names(yeast.orf$pheno),
+                         tests = ko.tests,
+                         cis.index = cis.cand.reg[[2]])
```

Before we show plots, here are some preliminary plot settings and a simple plot routine that will be used repeatedly for figures.

```
> lwd <- 2
> xaxis <- seq(0.01, 0.10, by=0.01)
> my.lty <- c(rep(1, 4), rep(2, 4), 1)
> my.lty <- rep(1, 9)
> my.pch <- c(1, 21, 24, 23, 25, 2, 5, 6, 8)
> par(mfrow=c(1, 3))
> par(mar=c(5, 4.1, 4, 2) + 0.1)
> myplot <- function(sum.type, sum.label) {
+   ymax <- max(sum.type)
+   yaxis <- seq(0, ymax,length.out = length(xaxis))
+   plot(xaxis, yaxis, type = "n", ylab = sum.label, cex = 1.5,
+        xlab = "Target significance level", cex.axis = 1.5,
+        cex.lab = 1.7, main = "(a)", cex.main = 2)
+   for (k in 1 : 9) {
+     lines(xaxis, sum.type[k,], type="b", lwd=lwd, pch=my.pch[k], cex=1.5,
+           col = "black", bg = "black")
+   }
+ }
```

Below we reproduce Figure 5 of Chaibub Neto et al. (2012). This figure presents the number of inferred true positives, number of inferred false positives and the prediction precision across varying significance levels for each one of the methods. The results were computed using all 135 ko-gene/putative target lists.

Next, we reproduce Figure 6 of Chaibub Neto et al. (2012). This figure was generated using the results of the 27 cis ko-gene/putative targets lists.

3 References

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

```
> myplot(roc.aux$Tp1, "Number of true positives")
> myplot(roc.aux$Fp1, "Number of false positives")
> myplot(roc.aux$Prec1, "Precision")
```

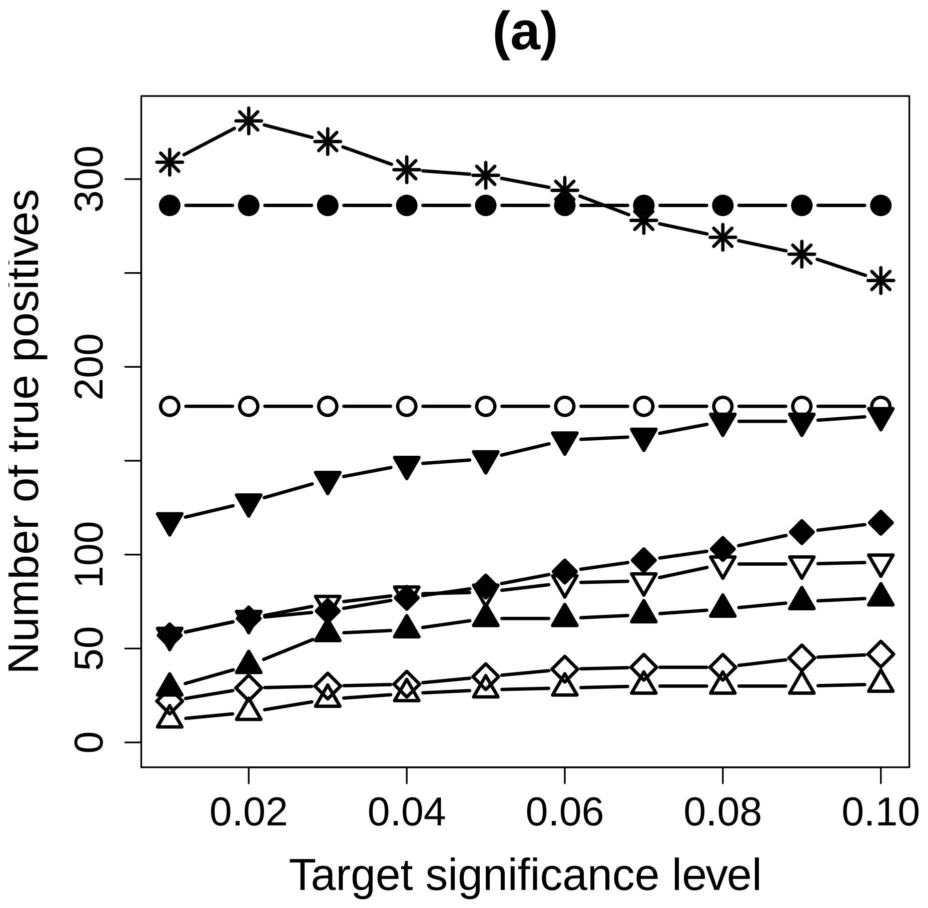


Figure 1: Reproduction of Figure 5 on Chaibub Neto et al. 2012. Overall number of true positives, number of false positives and precision across all 135 ko-gene/putative target lists. Asterisk represents the CIT. Empty and filled symbols represent, respectively, AIC- and BIC-based methods. Diamonds: parametric CMST. Point-down triangles: non-parametric CMST. Point-up triangles: joint-parametric CMST. Circles: AIC and BIC.

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)

```
> myplot(roc.aux$Tp2, "Number of true positives")
> myplot(roc.aux$Fp2, "Number of false positives")
> myplot(roc.aux$Prec2, "Precision")
```

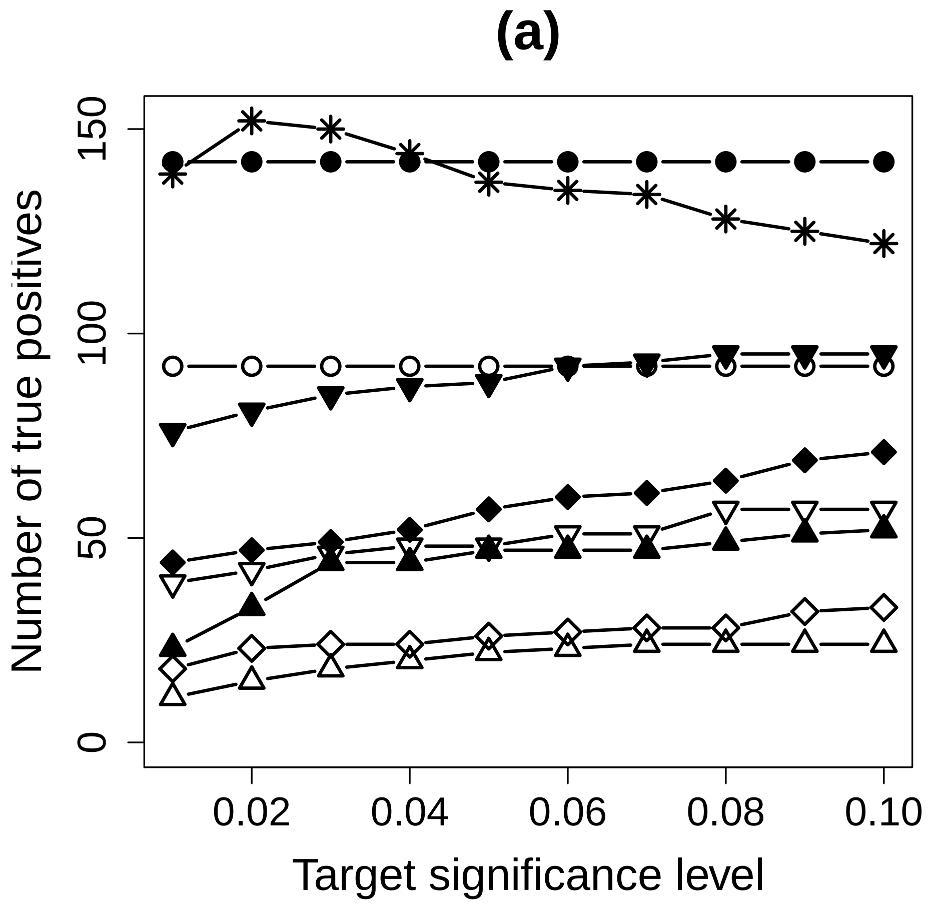


Figure 2: Reproduction of Figure 6 on Chaibub Neto et al. 2012. Overall number of true positives, number of false positives and precision restricted to 27 cis ko-gene/putative target lists. Asterisk represents the CIT. Empty and filled symbols represent, respectively, AIC- and BIC-based methods. Diamonds: parametric CMST. Point-down triangles: non-parametric CMST. Point-up triangles: joint-parametric CMST. Circles: AIC and BIC.

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.