Multiple testing

Sliding Haplin run, single SNPs

Load data:

```
pres.data <- genDataLoad(filename = "data_preprocessed",
 dir.in = "data")
```

Run:

```
 result <- haplinSlide(data = pres.data, markers = 1:100,
   use.missing = T, table.output = T, response = "mult",
   reference = "ref.cat", winlength = 1, cpus = 4)
```

COLLECT RESULTS FROM haplinSlide IN A FULL haptable

First option:

```
result1 <- haptable(result)
head(result1)
```

```
  window row.win marker alleles    counts    HWE.pv Original After.
1    rs1       1    rs1     c/G  146/3040 0.7075356      559
2    rs1       2   <NA>    <NA>      <NA>        NA      559
3    rs3       1    rs3     A/t  2460/706 0.3620846      559
4    rs3       2   <NA>    <NA>      <NA>        NA      559
5    rs5       1    rs5     a/T 1196/1978 0.3014432      559
6    rs5       2   <NA>    <NA>      <NA>        NA      559
  After.rem.Mend.inc. After.rem.unused.haplos pv.overall haplos  ha
1                 559                     559  0.4733520      c 0.0
2                 559                     559  0.4733520      G 0.9
... etc.
```

Complete, BUT every second row strictly speaking redundant

COLLECT RESULTS FROM haplinSlide IN A FULL haptable

Second option:

```
result1 <- toDataFrame(result, reduce = T)
head(result1)
```

```
   element marker alleles    counts    HWE.pv Original After.rem.NA
1      rs1    rs1     c/G  146/3040 0.7075356      559           559
3      rs3    rs3     A/t  2460/706 0.3620846      559           559
5      rs5    rs5     a/T 1196/1978 0.3014432      559           559
7      rs6    rs6     c/G  276/2906 0.3472571      559           559
9      rs7    rs7     c/G 1274/1898 0.3554434      559           559
11     rs8    rs8     A/g  2330/836 0.2787112      559           559
   After.rem.Mend.inc. After.rem.unused.haplos pv.overall haplos
1                  559                     559 0.47335197      c
3                  559                     559 0.67199029      t
... etc.
```

Leads to ONE row per SNP

MULTIPLE TESTING ISSUES

Top hits:

```
result2 <- result1[order(result1$pv.overall),]
head(result2, 3)
```

```
   element marker alleles    counts     HWE.pv Original
43    rs28   rs28     A/t  2590/586 0.24716713      559
23    rs16   rs16     a/T  505/2659 0.95500149      559
71    rs45   rs45     a/T 1364/1808 0.05498588      559
   After.rem.NA After.rem.Mend.inc. After.rem.unused.haplos
43          559                 559                    559
23          559                 559                    559
71          559                 559                    559
   pv.overall haplos haplofreq haplofreq.lower haplofreq.upper
43 0.01458538      t 0.2135287       0.1895629       0.2398940
23 0.01816661      a 0.1854451       0.1625915       0.2104145
71 0.02319043      a 0.4654631       0.4346627       0.4965631
   reference   RR.est.  RR.lower  RR.upper RR.p.value RRdd.est.
43         -  0.7640836 0.6136162 0.9452186      0.0154 0.5838237
```

MULTIPLE TESTING ISSUES: QQ-PLOT

Results look OK in terms of Mendelian inconsistencies, HWE testing and MAFs.

But what to expect?

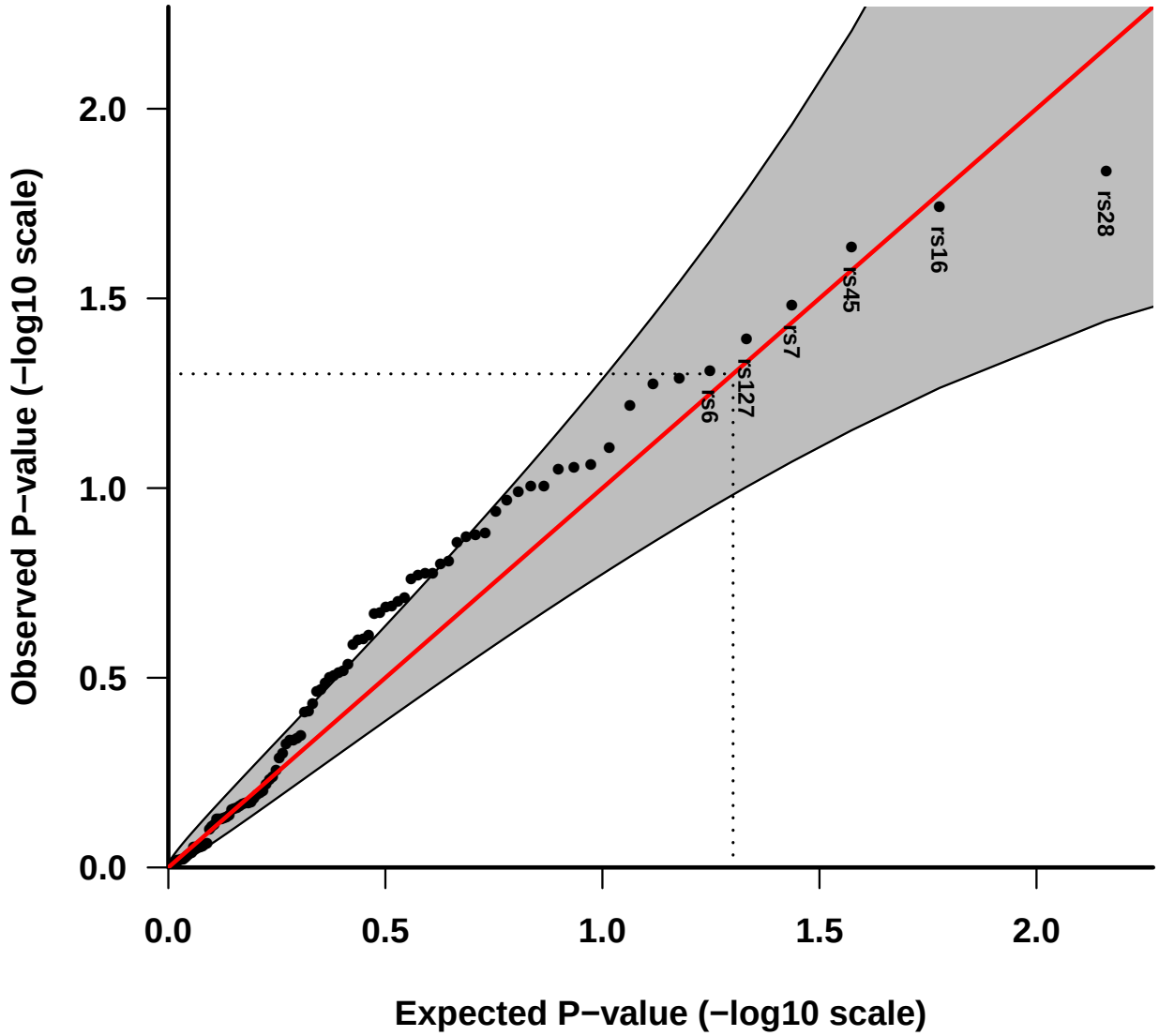
QQ-plot with confidence bands

A QQ-plot will show deviations from overall H_0 of no effects

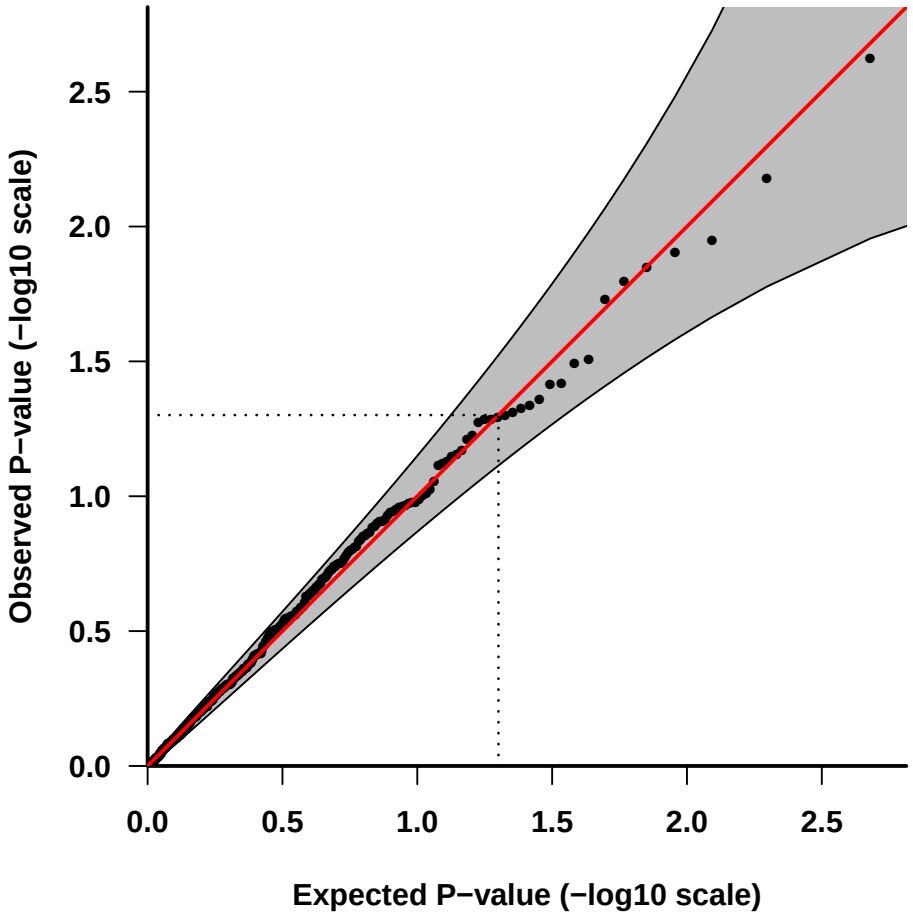
```
pvals <- result1$pv.overall
names(pvals) <- result1$window
pQQ(pvals)
```

- Names added to `pvals` will show up in plot.
- Red diagonal will show "neutral" position,
  i.e. what's expected under H_0 of no effects.
- Dotted lines show traditional 0.05.
- Points above grey confidence area may suggest real effects.

MULTIPLE TESTING ISSUES: QQ-PLOT

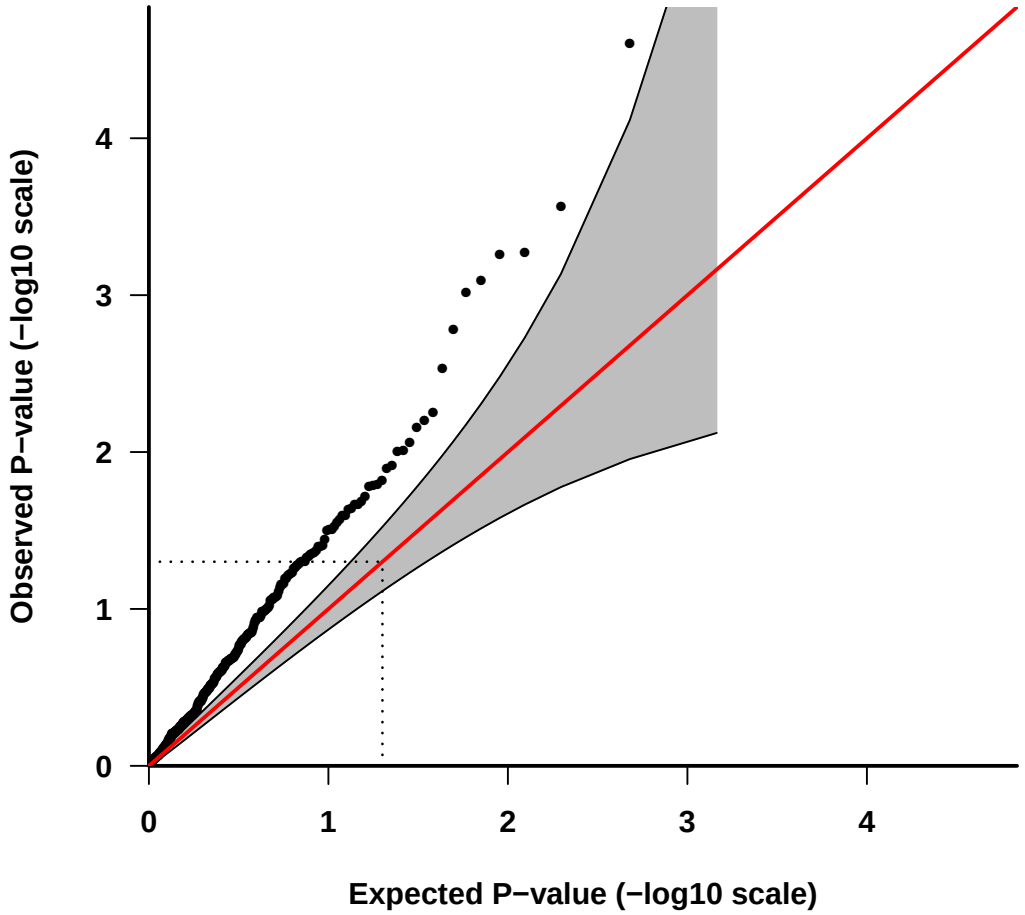


QQ-PLOT TO CHECK MULTIPLE TESTING



331 randomly generated p-values

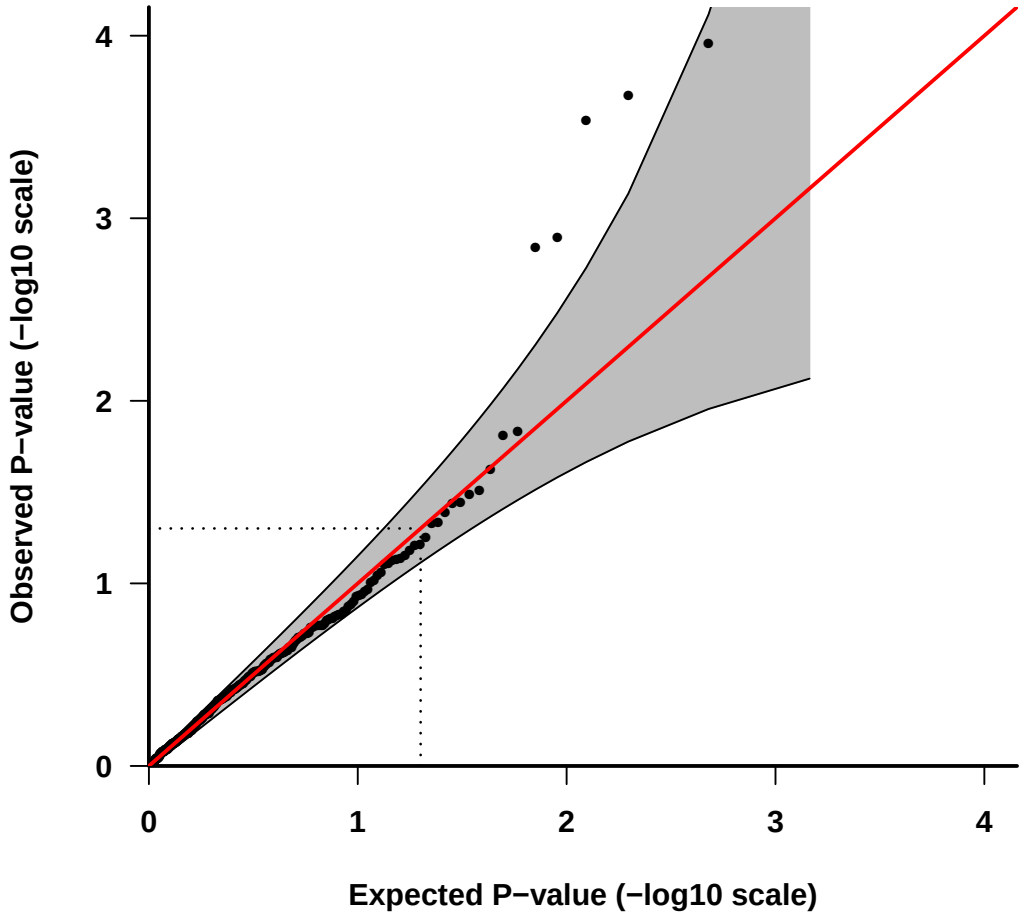
QQ-PLOT TO CHECK MULTIPLE TESTING



331 randomly generated p-values, but incorrect calibration

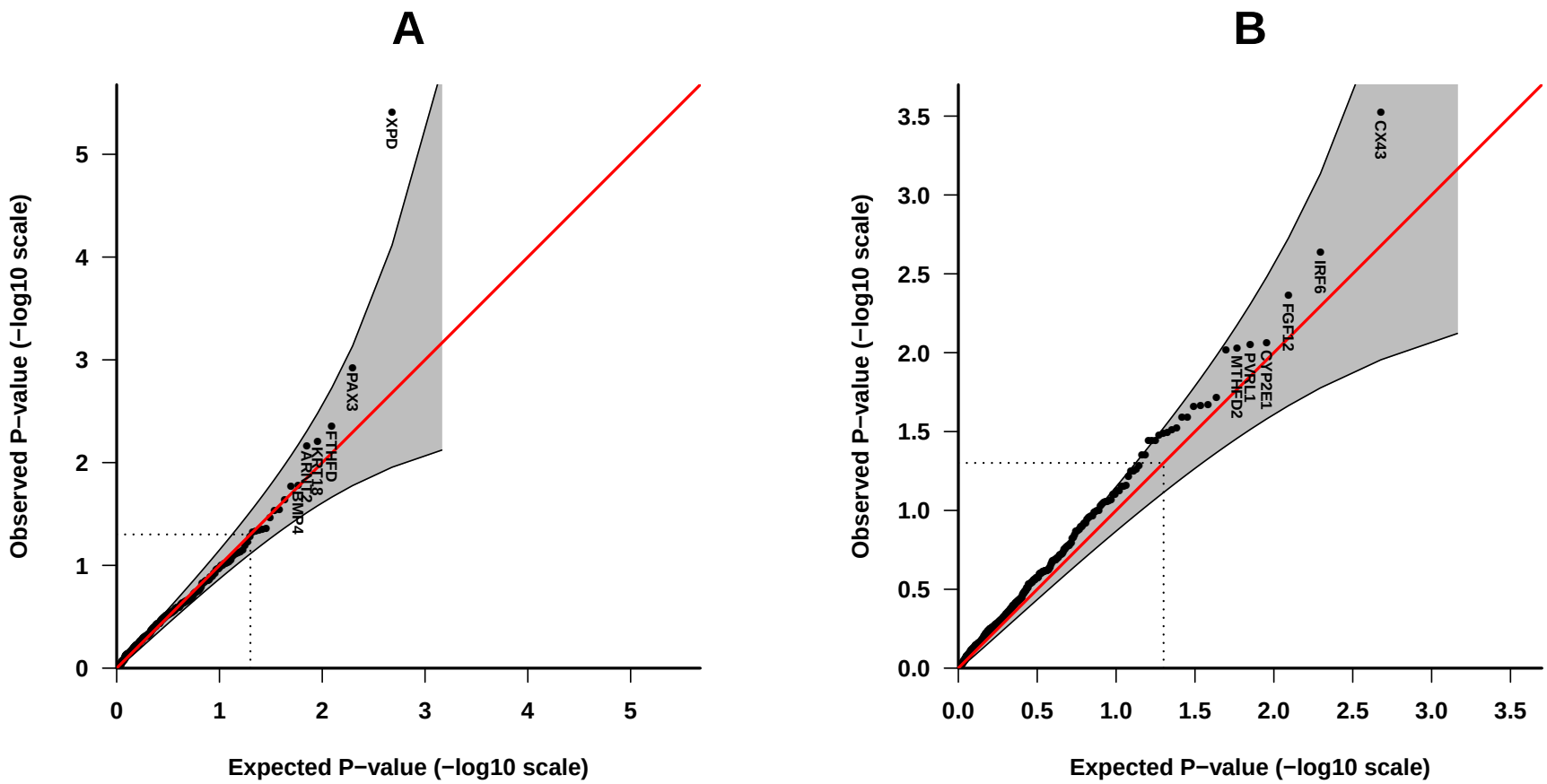
(For instance: population admixture in case-control studies)

QQ-PLOT TO CHECK MULTIPLE TESTING



326 randomly generated p-values + 5 interesting ones

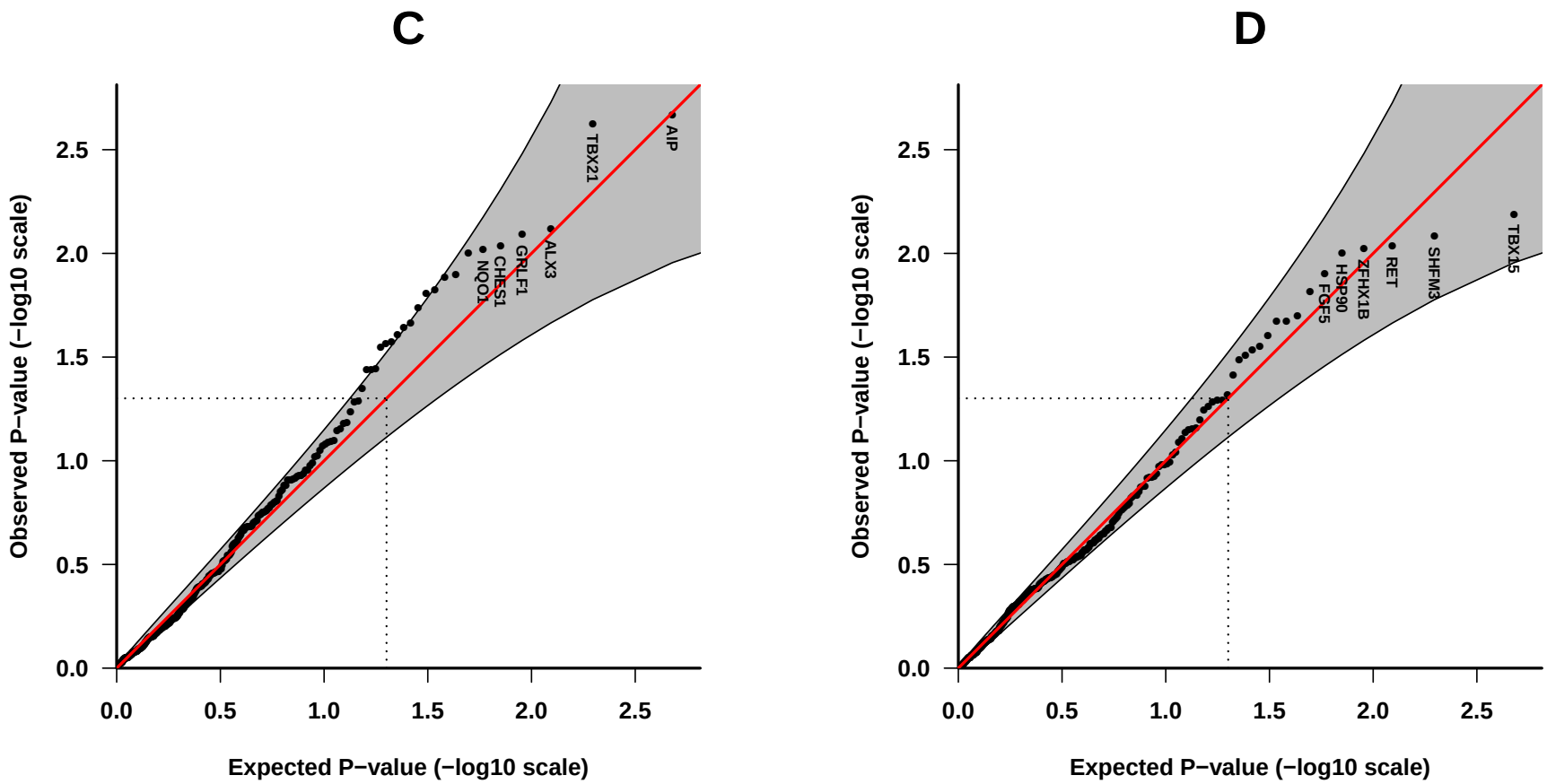
QQ-PLOT TO CHECK MULTIPLE TESTING (FIND THE TRUE RESULTS!)



331 candidate genes for cleft lip (with or without cleft palate)

(Norway and Denmark combined)

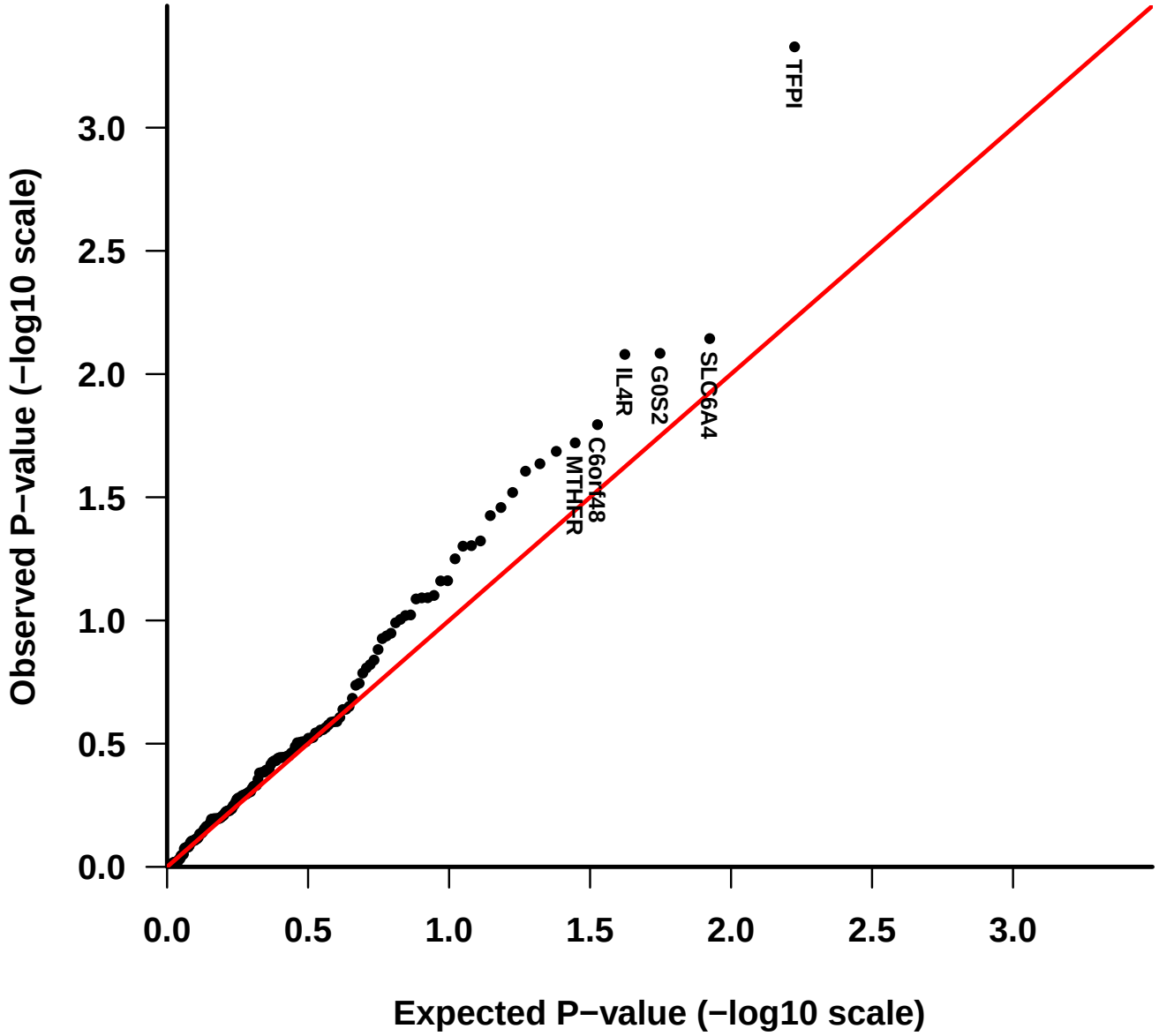
QQ-PLOT TO CHECK MULTIPLE TESTING (FIND THE TRUE RESULTS!)



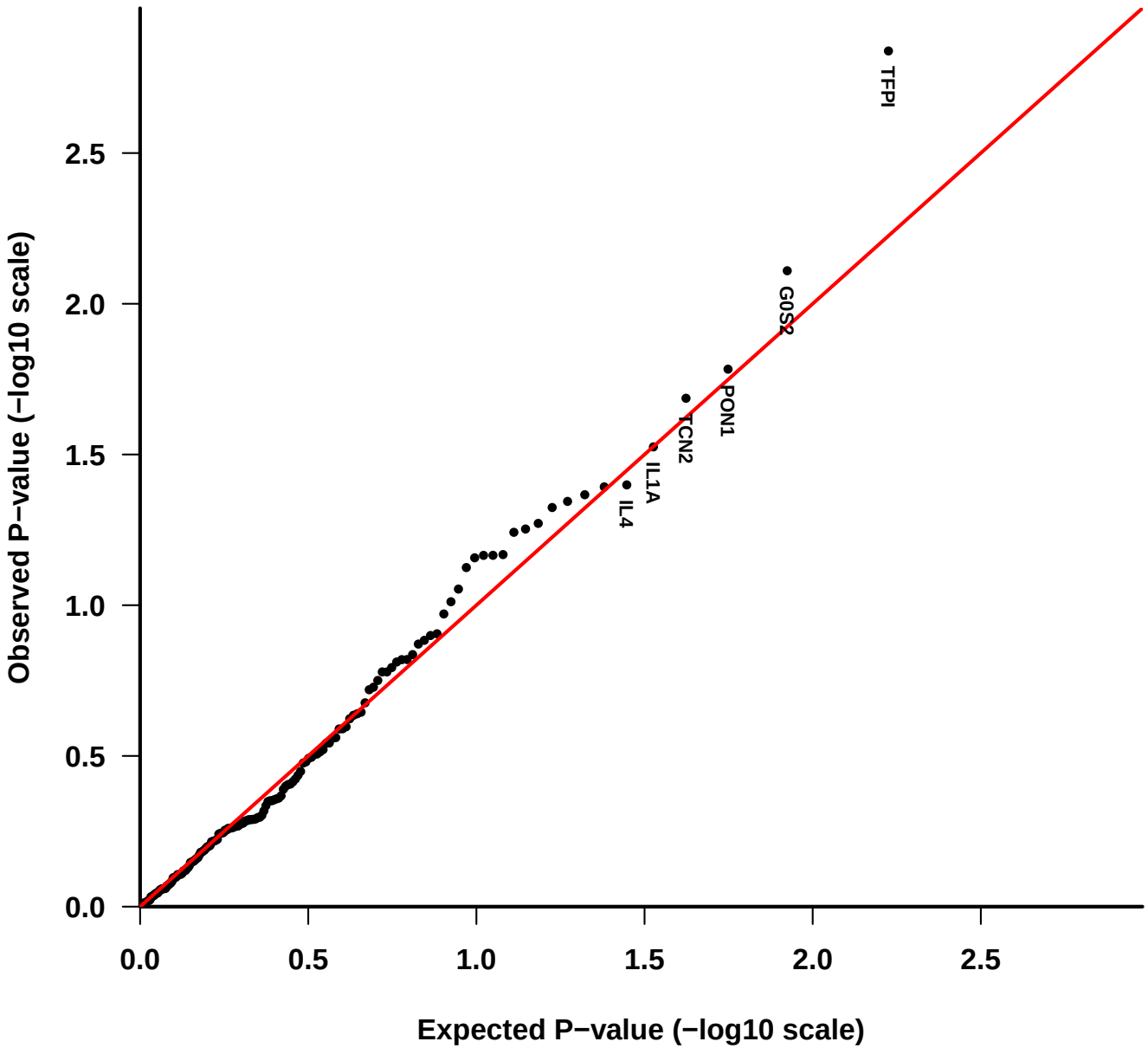
331 candidate genes for cleft lip (with or without cleft palate)

(Norway and Denmark combined)

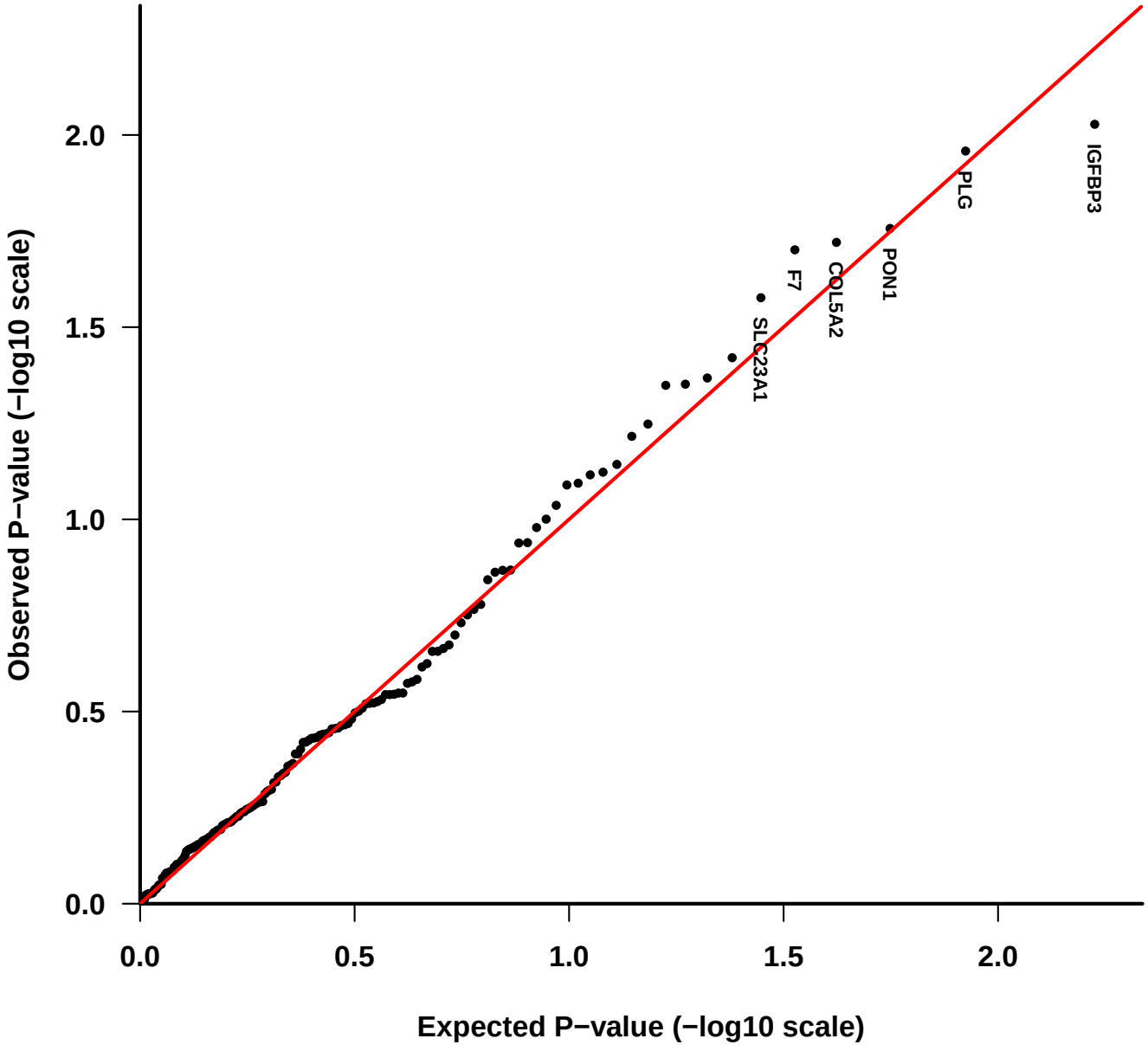
PRETERM BIRTH (ONLY CASE-CONTROL)



PRETERM BIRTH (HYBRID DESIGN)



PRETERM BIRTH (ONLY CASE TRIAD)



Multiple testing, sliding windows

SLIDING WINDOW APPROACH

gene	snp
ABCA1	rs2230806
ABCA1	rs2230808
ABCA1	rs2472384
ABCA1	rs2487054
ABCA1	rs2740479
ABCA1	rs3858075
ABCA1	rs4149272
ABCA1	rs4149313

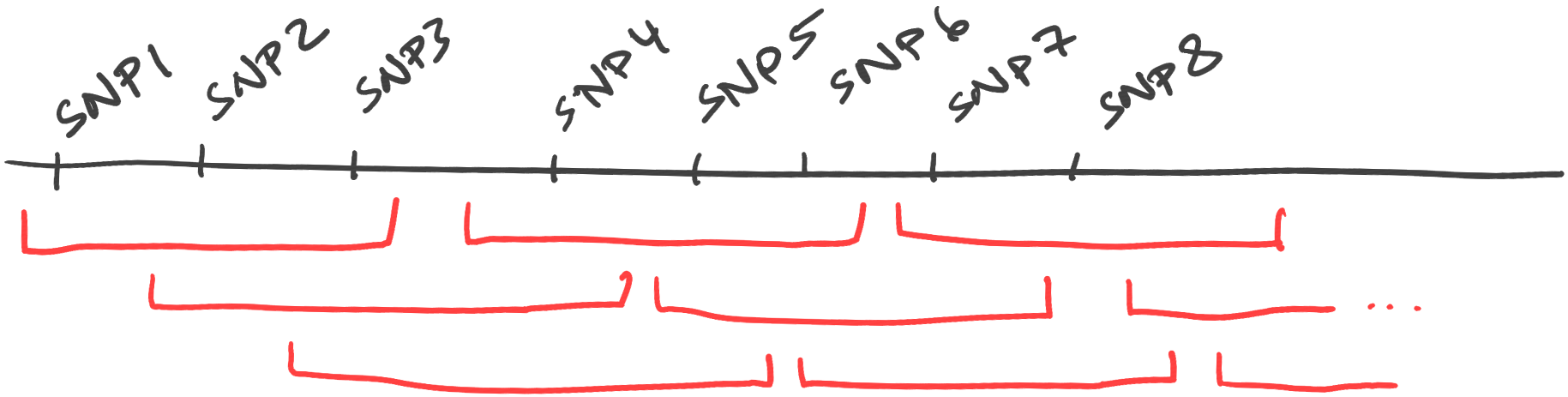
Problem:

- More than 4 SNPs demand heavy computations
- Long haplotypes “break up” (don’t really exist in population)

Solution:

- Sliding window approach
- “Bracketing” of mutations

SLIDING WINDOW APPROACH



- Sliding window, haplotypes of length, say, 2, 3, or 4.
- Compute p-value for each window
- Combine p-values
- For instance: $\min(p_1, p_2, \ldots)$ (Highly popular, for obvious reasons!)
  BUT requires control for multiple testing

WITHIN-GENE/LOCUS MULTIPLE TESTING PROBLEM

Problems:

- Overlapping windows
- Linkage disequilibrium (LD) = dependence among SNPs

That is, p-values are *dependent*

What if we want full Family-Wise Error Rate (FWER) control?

Possible solutions:

- Bonferroni (brutal)
- Bootstrap or jackknife (heavy)
- "Extended" score test
  (Seemingly related to "Seemingly unrelated estimation"!)

WITHIN-GENE/LOCUS MULTIPLE TESTING PROBLEM: `suest`

Use the `suest` function in Haplin:

```
result <- haplinSlide(data = pres.data, markers = 15:25,
  use.missing = T, table.output = F, response = "mult",
  reference = "ref.cat", winlength = 3, cpus = 3)

suest(result)
```

Important:

- Set `table.output = F` to be able to run `suest`

WITHIN-GENE/LOCUS MULTIPLE TESTING PROBLEM: suest

```
Individual score p-values:
[1] 0.80109934 0.82524993 0.81806066 0.26556298 0.06119231
[6] 0.02122326 0.08373334 0.08533893 0.24453079
```

The individual score p-values should be similar to the pv.overall (from the LRT test), just for checking

```
summary(result[[1]])
```

```
LIKELIHOOD RATIO TEST:
 Loglike null model:          -2256.6635
 Loglike full model:          -2256.1638
 df:                              3.0000
 Likelihood ratio p-value:        0.8014
```

WITHIN-GENE/LOCUS MULTIPLE TESTING PROBLEM: `suest`

The adjusted minimum score p-value:

```
Multiple testing corrected p-value, based on minimum
 of individual values:
[1] 0.10254
```

- Here, the most significant (0.0212) is not significant after correction (0.10254).
- Bonferroni correction would be $10 \times 0.0212 = 0.212$.

THUS:

- `suest` computes a joint overall p-value for a region
- In this case for 10 overlapping windows
- Adjusted for multiple testing, taking LD into account
- Not intended for long scans, only limited regions