GS01 0163
Analysis of Microarray Data

Keith Baggerly and Kevin Coombes
Department of Bioinformatics and Computational Biology
UT M. D. Anderson Cancer Center
kabagg@mdanderson.org
kcoombes@mdanderson.org

9 October 2007

Lecture 12: Rank-based tests of differential expression

- Wilcoxon rank-sum test
- Empirical Bayes
- The Tail-Rank Test
- Looking at the results

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Nonparametric tests

The t-test for differences in mean expression between two groups of samples assumes that the measurements in each group are normally distributed. If this assumption is far from the truth, then the t-statistics and p-values you get may be meaningless. (Actually, departures from normality tend to increase the Type II error, especially when the sample size is small.)

Statistically, the dispute over log-transforming microarray data reduces to whether a normal distribution better describes the data on the raw scale or on the log scale.

We can avoid this problem entirely by using a statistical test that does not assume anything about the distributions. These tests are usually called distribution-free or nonparametric.

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Wilcoxon rank-sum test

The most common nonparametric test for a difference in mean expression is the Wilcoxon rank-sum test, which is also known as the Mann-Whitney test.

We assume that we have sample measurements from two groups:

$$X_1, X_2, \ldots, X_{n_X}$$

$$Y_1, Y_2, \ldots, Y_{n_Y}$$

We then rank these values from smallest to largest, getting something like

$$\begin{array}{ccccccccccc} X_3 & \leq & Y_5 & \leq & X_1 & \leq & X_{10} & \leq & Y_1 & \leq \cdots \leq & X_2 \\ 1 & & 2 & & 3 & & 4 & & 5 & & n_X + n_Y \end{array}$$

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Computing rank-sums

Next, we compute a statistic W by summing the ranks of the measurements from the first group. In our example,

$$W = 1 + 3 + 4 + \cdots + (n_X + n_Y).$$

W is always an integer, and it is easy to compute its minimum and maximum values. The minimum occurs when all the X values are smaller than all the Y values. Thus, all the X values are at the beginning of the list, and we have

$$W \geq 1 + 2 + \cdots + n_X = \frac{n_X(n_X + 1)}{2}.$$

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

The maximum occurs when all the X values come after all the Y values, giving

$$\begin{aligned} W &\leq (n_Y + 1) + (n_Y + 2) + \cdots + (n_Y + n_X) \\ &= n_X n_Y + (1 + 2 + \cdots n_X) \\ &= n_X n_Y + \frac{n_X(n_X + 1)}{2} \\ &= \frac{n_X(2n_Y + n_X + 1)}{2}. \end{aligned}$$

Those formulas are very nice, but let's see what happens when we have 10 samples in each group. The range of values for W in this case is

$$(1 + 2 + \cdots + 10) = 55 \leq W \leq (11 + 12 + \cdots + 20) = 155$$

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

When is a rank-sum significant?

Intuitively, if we get a value of W near its extreme values, then we strongly suspect that the two groups are different. If, however, we get a value near the middle, then we suspect that there is no difference. How can we make this idea more precise?

First, let's do some exploration. We start by generating an unstructured random data matrix:

```
> n.genes <- 40000
> n.samples <- 20
> type <- rep(c("A", "B"), each = 10)
> data <- matrix(rnorm(n.genes * n.samples),
+     ncol = n.samples)
```

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Next, we rank the values in each row:

```
> ranked.data <- apply(data, 1, rank)
> dim(data)

[1] 40000    20

> dim(ranked.data)

[1]    20 40000
```

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Notice that the matrix of ranks is transposed when compared to the original data matrix. So, we can compute the Wilcoxon rank-sum statistics by summing the correct ranks by column:

```
> wilstat <- apply(ranked.data[type == "A",
+     ], 2, sum)
> summary(wilstat)
```

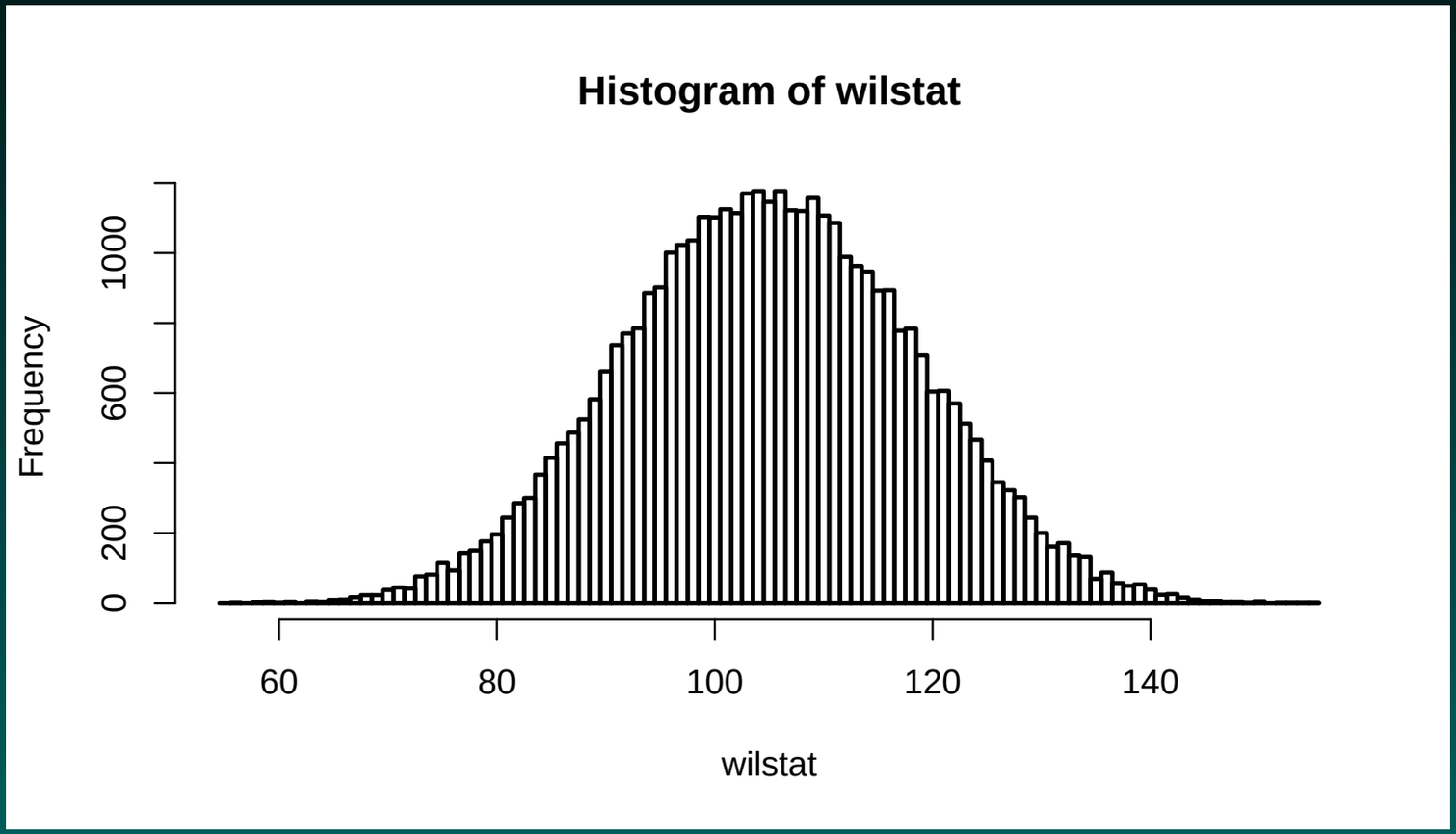
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
     56      96     105     105     114     155
```

We don't quite get to the extremes...

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Null distribution of Wilcoxon rank-sum

```
> hist(wilstat, breaks = seq(54.5, 155.5, by = 1))
```



© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

What happens with real data?

We will return to the prostate cancer data set used in the last lecture. Recall that this data set contains the log ratios of $42,129$ genes measured using two-color fluorescent microarrays and a common reference channel. Recall also that we selected a subset of 10 samples from normal prostate and 10 samples of prostate cancer.

We compute the Wilcoxon rank-sum statistics for this data set:

```
> ranked.data <- apply(expression.data, 1, rank)
> dim(ranked.data)

[1]    20 42129

> status <- clinical.info[, "Status"]
> ranksum <- apply(ranked.data[status == "N",
```

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

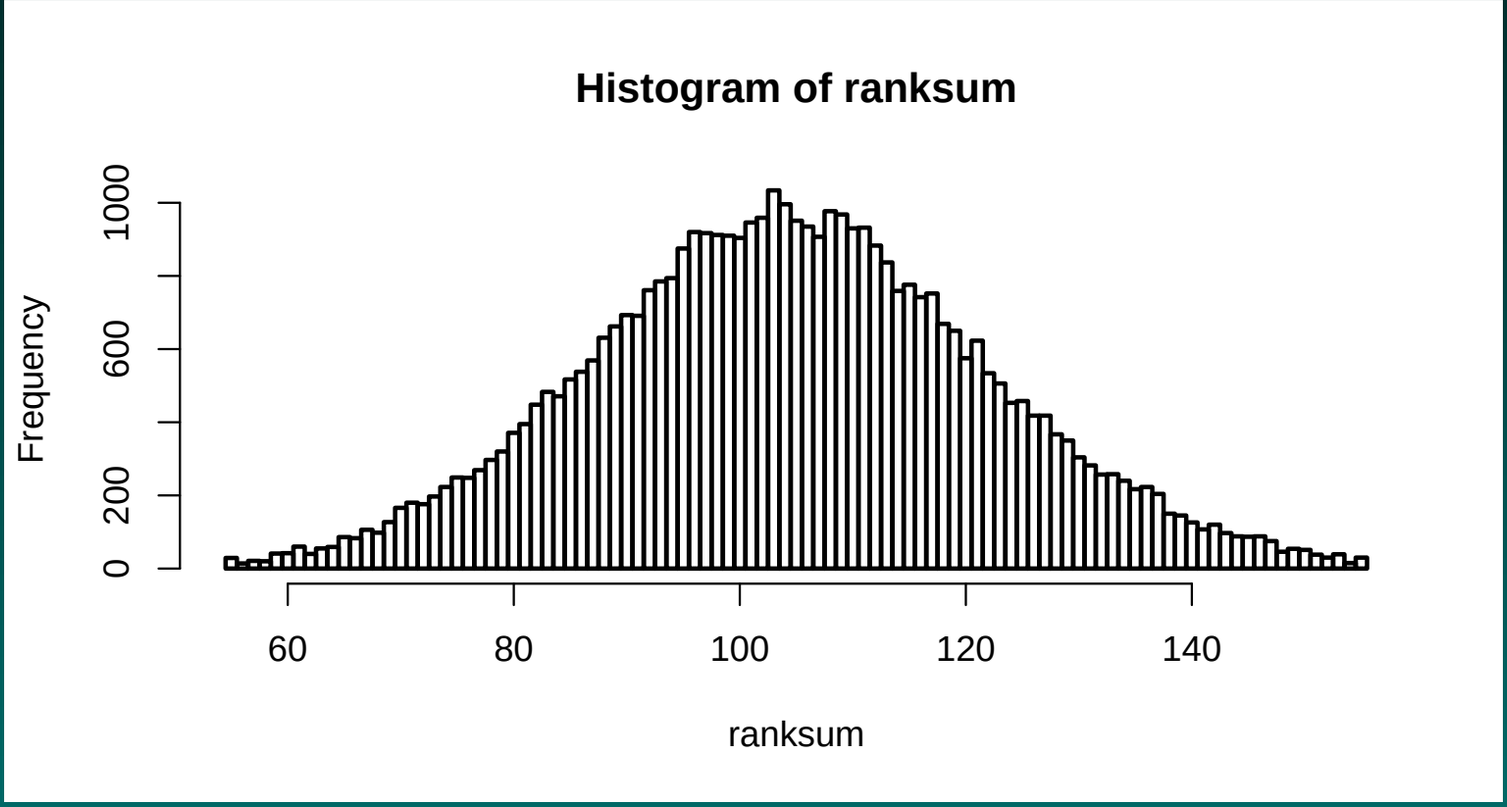
```
+     ], 2, sum)
```

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Real data yields extreme statistics

```
> summary(ranksum)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   55.0    93.0   104.0   104.4   116.0   155.0
```



© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Distributions matter

When we simulated data, we got values of the rank-sum statistic between 56 and 155.

When we looked at real data, we got rank-sum statistics that spanned the full range of possible values from 55 to 155.

One (pessimistic) interpretation of this result is that rank-sum statistics are only useful in microarray experiments if they find genes where all the values in one group are less than all the values in the other group.

We can do better by looking more carefully at the distributions.

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

R and Wilcoxon

R contains functions to explore the distribution of the rank-sum statistics:

rwilcox generate random values from the Wilcoxon distribution

dwilcox probability density function

pwilcox cumulative probability function

qwilcox quantile function

This set of functions parallels those for other distributions (like `rnorm`, `dnorm`, `pnorm`, and `qnorm` for the normal distribution).

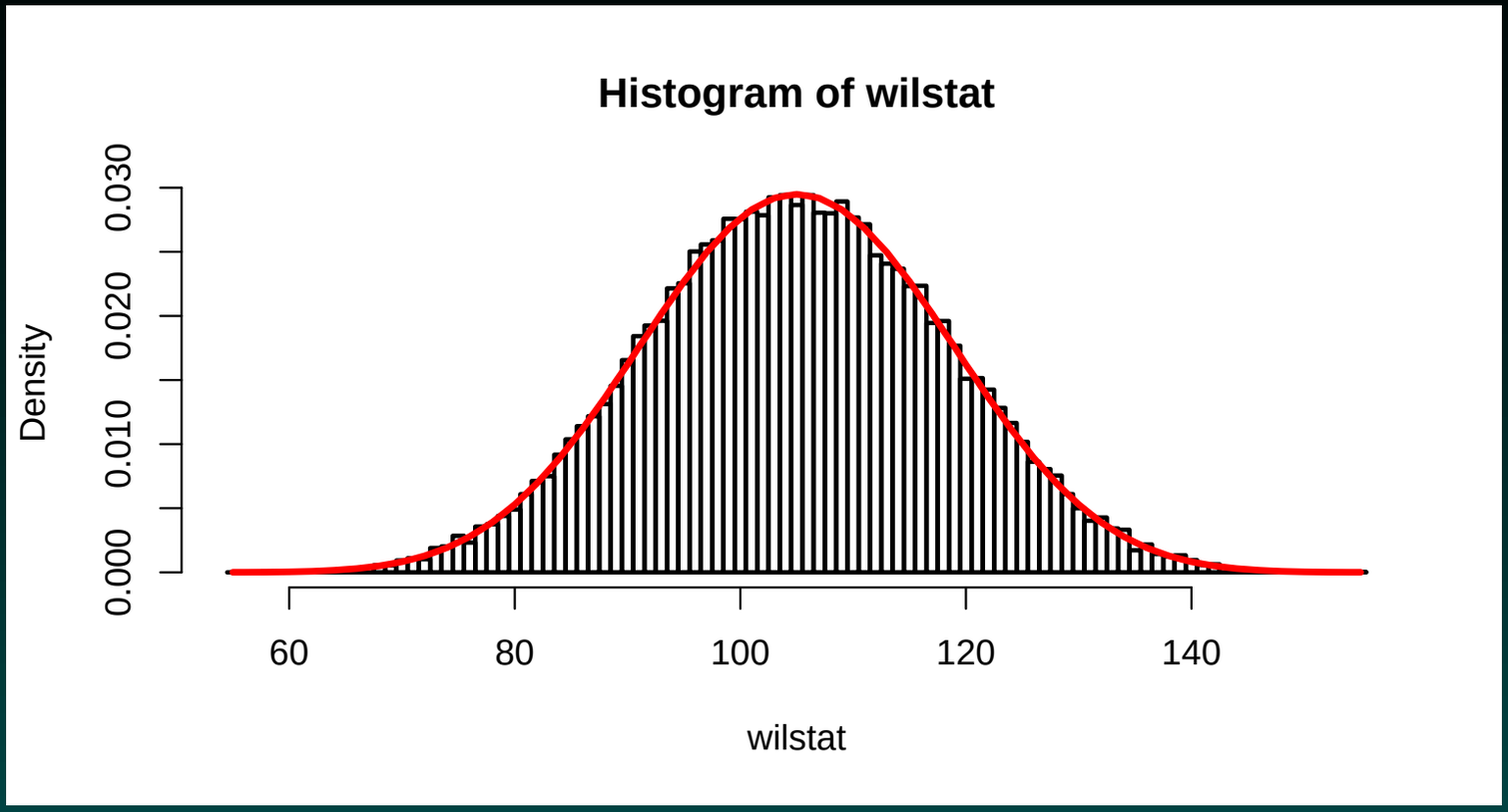
One should note, however, that the rank-sum statistics in R are shifted so that the smallest value is 0 instead of n_X.

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

The null distribution

```
> minW <- sum(1:10); maxW <- sum(11:20)
> breaker <- seq(minW-0.5, maxW+0.5, by=1)
> hist(wilstat, breaks=breaker, prob=TRUE)
> lines(minW:maxW, dwilcox(0:(maxW-minW), 10, 10),
+       col='red', lwd=3)
```

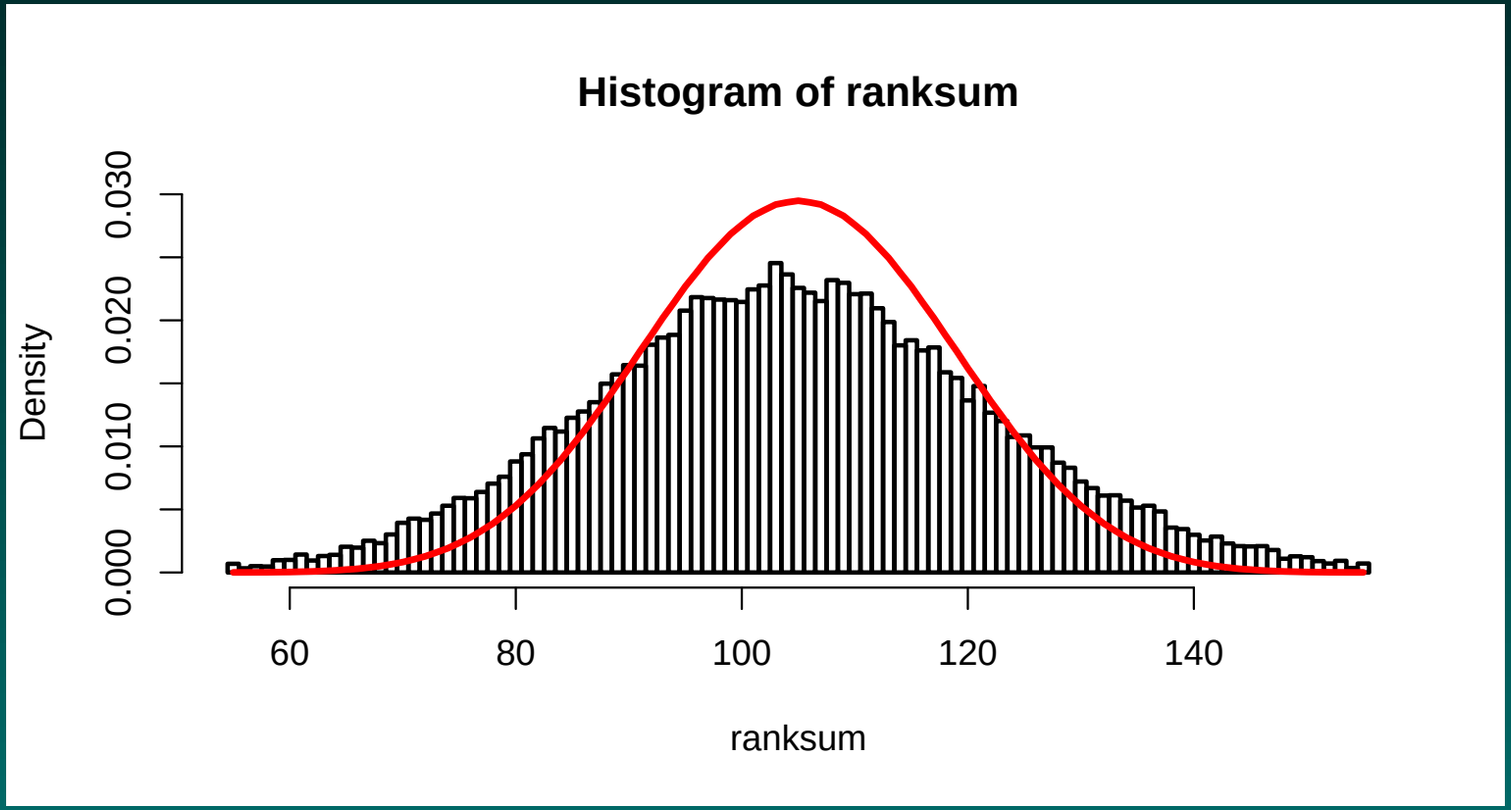
© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly



© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

The real distribution

```
> hist(ranksum, breaks = breaker, prob = TRUE,
+      ylim = c(0, 0.03))
> lines(minW:maxW, dwilcox(0:(maxW - minW),
+      10, 10), col = "red", lwd = 3)
```



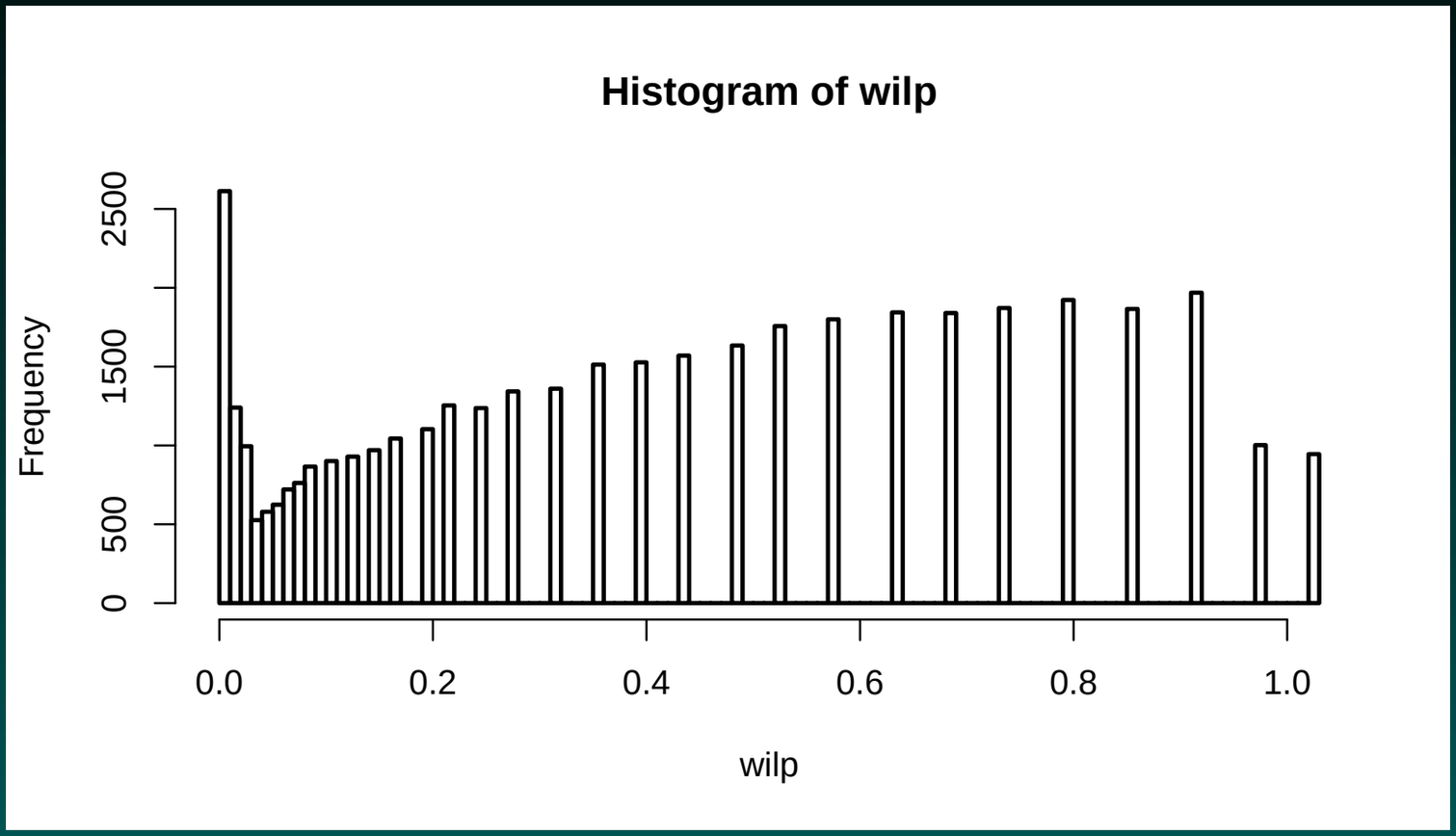
© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Wilcoxon p-values

```
> wilp <- sapply(ranksum - minW, function(w,
+     m, n) {
+     if (w > m * n/2)
+         2 * (1 - pwilcox(w, m, n))
+     else 2 * pwilcox(w, m, n)
+ }, 10, 10)
```

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

```
> hist(wilp, breaks = 100)
```



© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Empirical Bayes

The discreteness of the values of the Wilcoxon statistics makes the distribution of p-values problematic for the application of something like BUM to sort out the significance in the face of multiple testing. Instead, we are going to use a differnt approach.

Reference: Efron and Tibshirani. Empirical Bayes methods and false discovery rates for microarrays. *Genetic Epidemiology*, 2002; **23**: 70–86.

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Basic idea

Assume that there are two classes of genes, Different and Not Different. We assume prior probabilities

- $p_0 =$ Prob(Not Different)
- $p_1 = 1 - p_0 =$ Prob(Different)

and density functions

- $f_0(y)$, known Wilcoxon, if Not Different
- $f_1(y)$, unknown, if Different

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Mixtures

The overall probability density function is a mixture

$$f(y) = p_0 f_0(y) + p_1 f_1(y).$$

Bayes Theorem: $P(H|D) = P(D|H)P(H)/P(D)$

Applying Bayes Theorem gives posterior estimates:

$$p_1(y) \equiv Prob(Diff|Y = y) = 1 - p_0 f_0(y)/f(y)$$

and

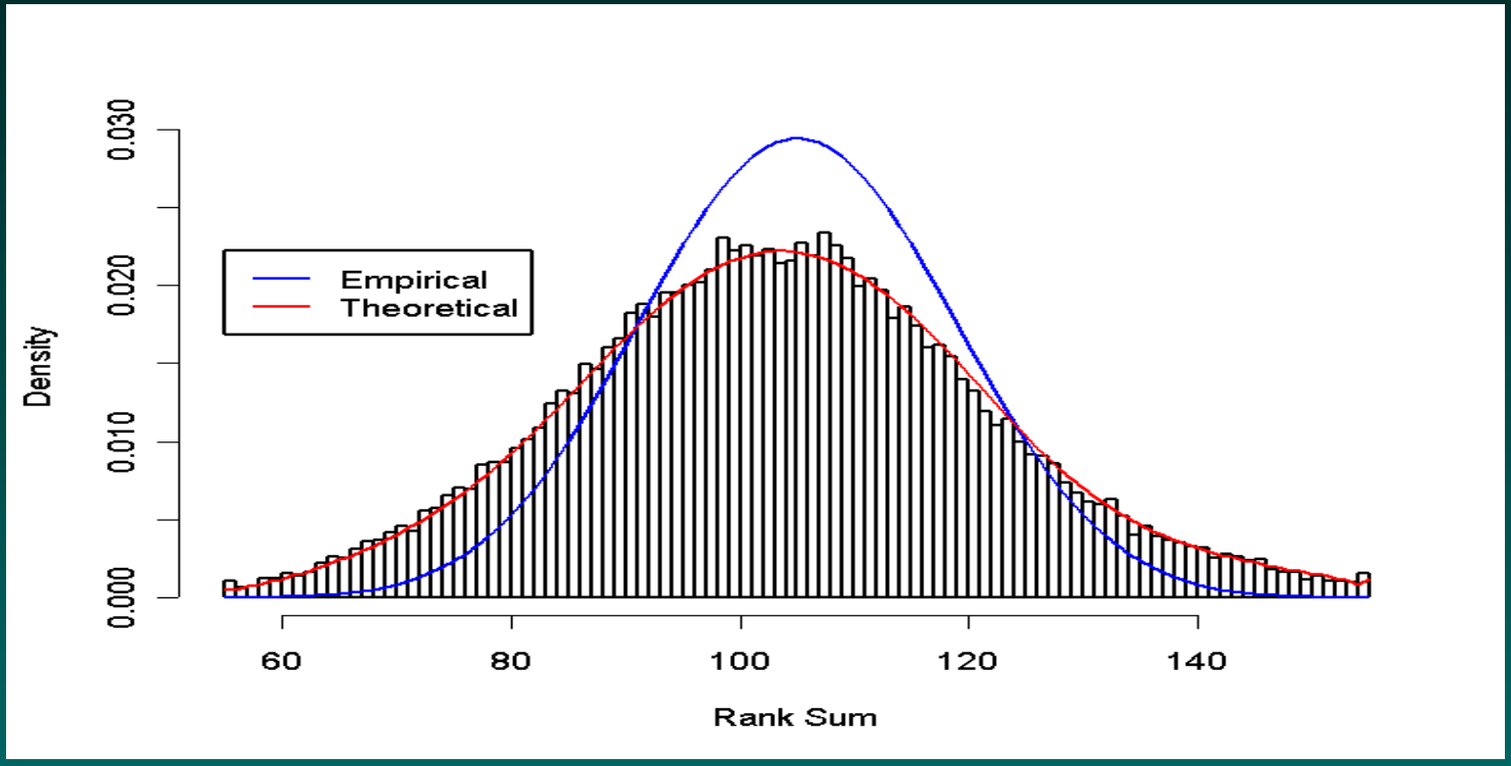
$$p_0(y) \equiv Prob(NotDiff|Y = y) = p_0 f_0(y)/f(y)$$

We can use the observed data to estimate the overall density function by $\hat{f}(y)$ (typically by log-transforming the observed function and fitting a curve.)

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

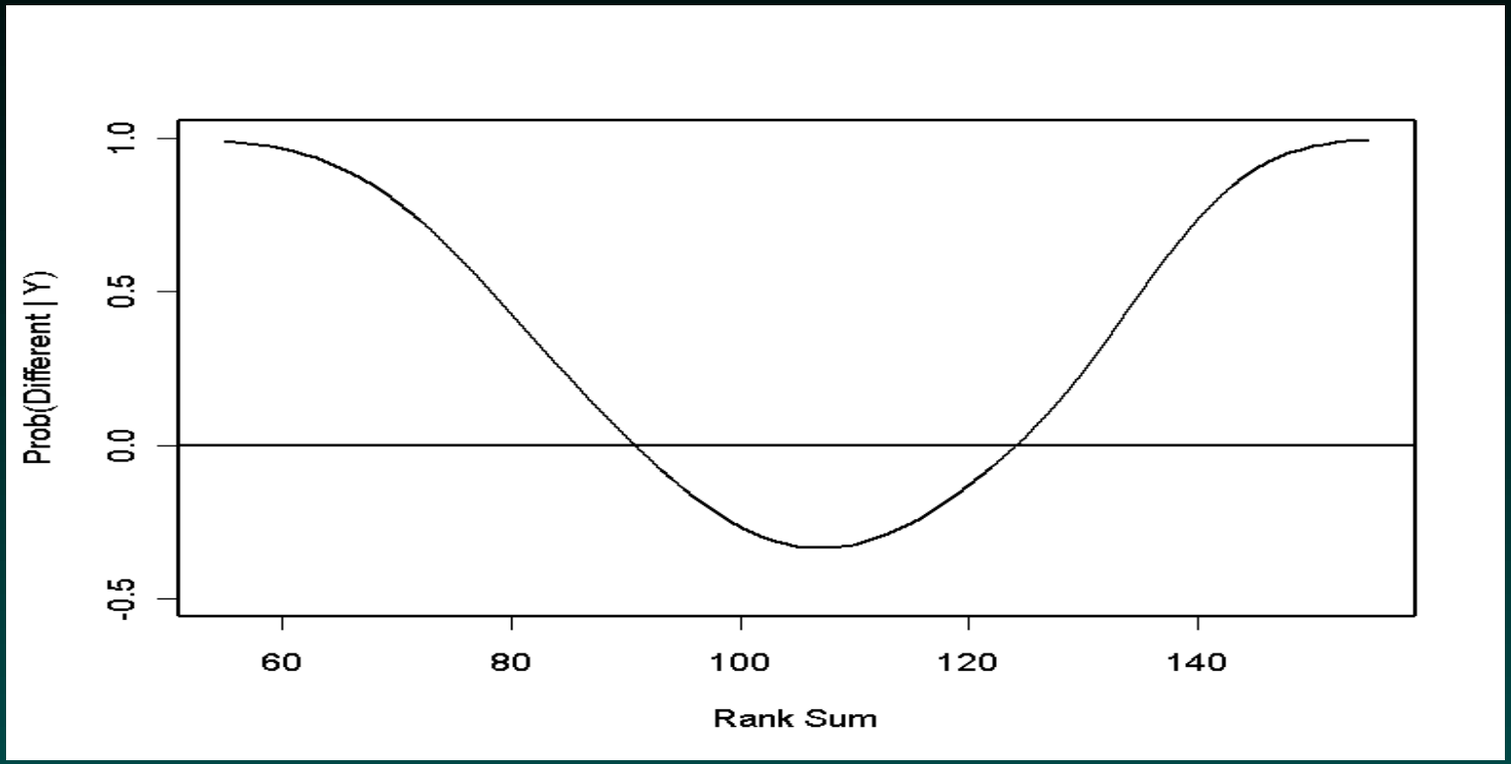
Empirical Bayes

The "empirical" nature of this Bayesian idea is that we can adjust the "prior" p_0 after looking at the data, and thus obtain some reasonable values for it. First, here is how well we fit the distribution (mentally swap the labels, since they are wrong):



© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

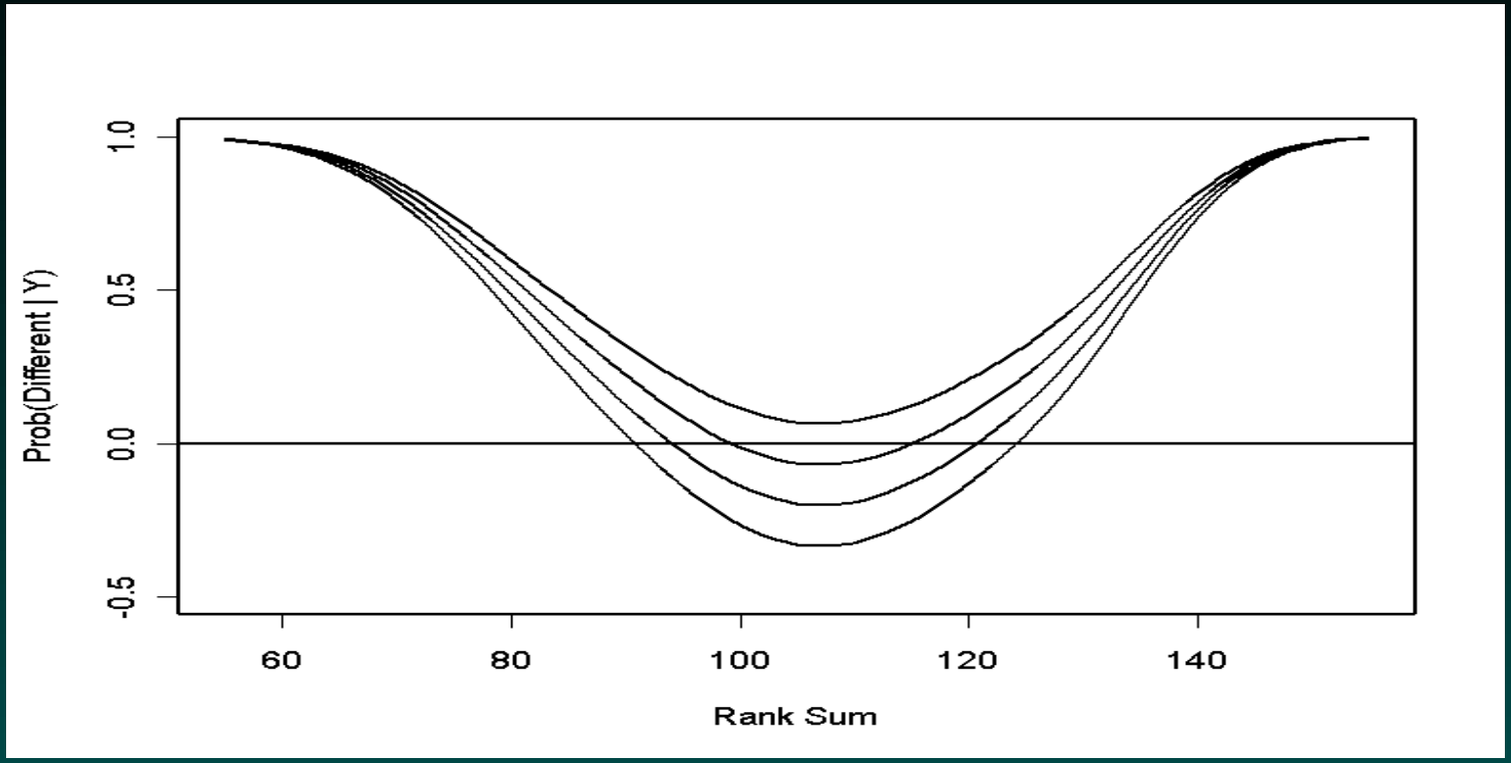
Plot of Posterior Probability of Difference



This graph assumes $p_0 = 1$, so no genes are different. The posterior probability of being different becomes negative in the middle of the graph. This results from the "empirical" nature of the estimate without imposing a full model. We can, however, adjust p_0 to prevent seeing any negative probabilities.

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

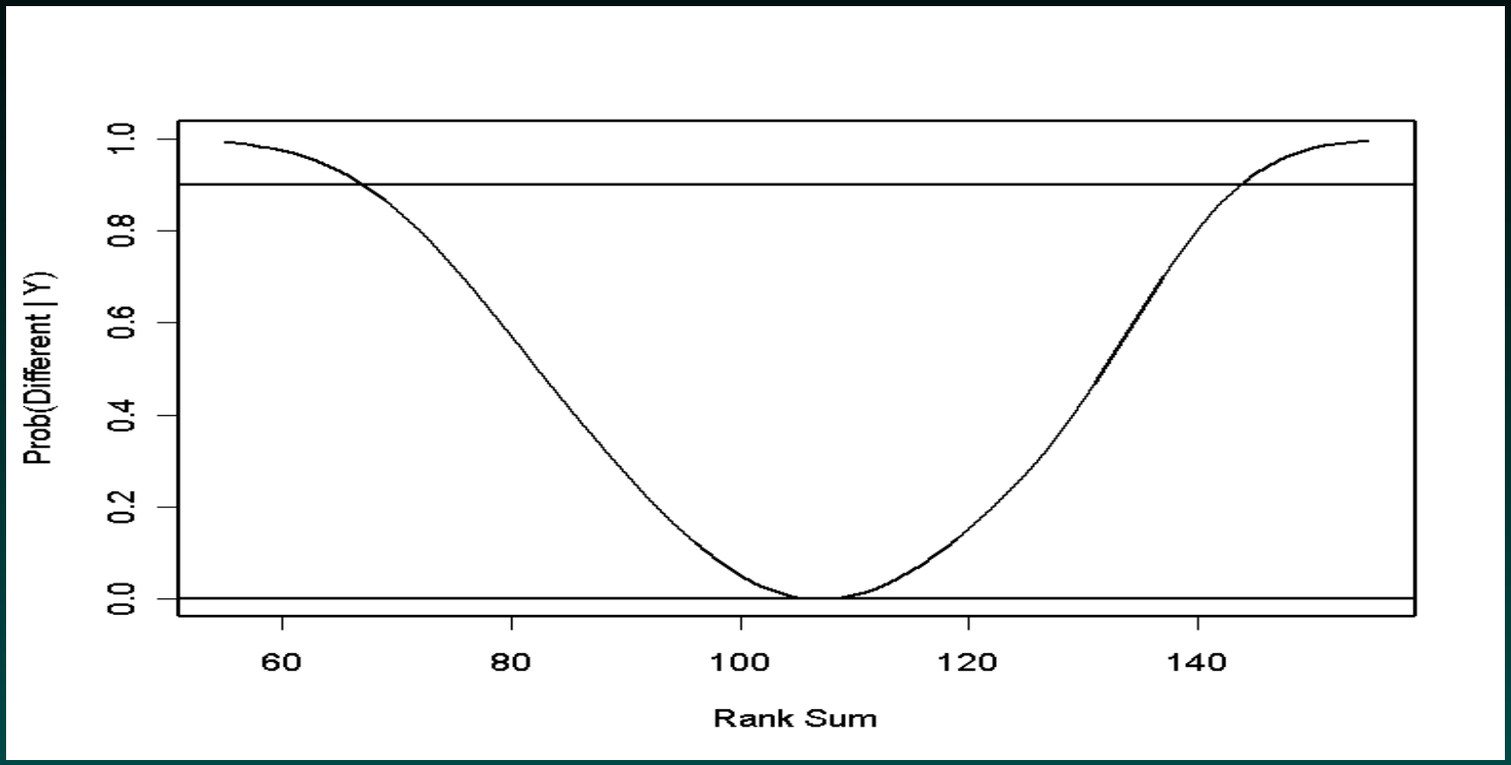
Plot of Posterior Probability of Difference



This shows posterior probabilities with $p_0 = 0.7, 0.8, 0.9, 1.0$. Somewhere between $p0 = 0.7$ and $p_0 = 0.8$, all the posterior probabilities become positive.

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Plot of Posterior Probability of Difference

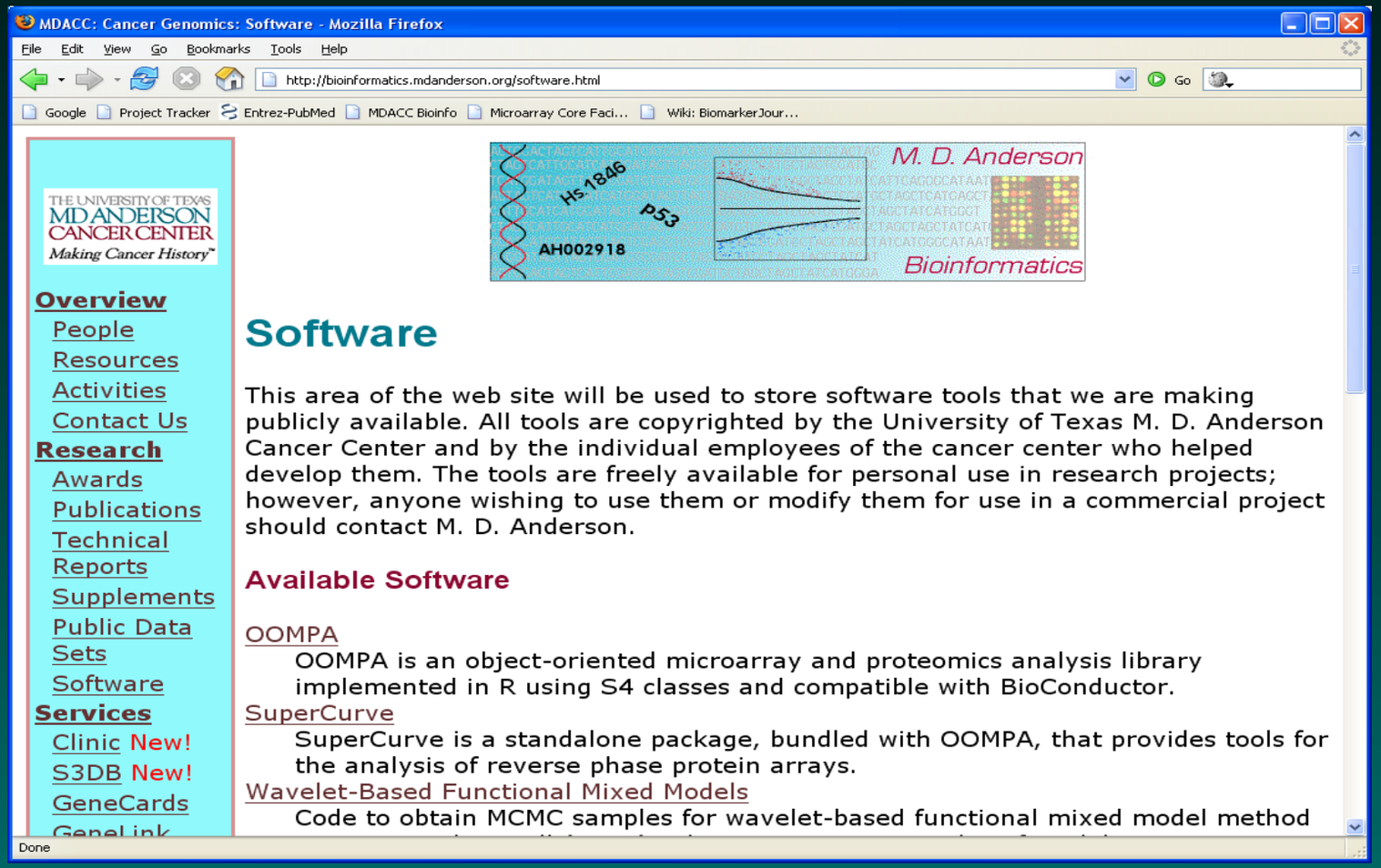


This plot uses $p_0 = 0.75$, which is essentially the largest value we can use for p_0 and ensure that all the posterior probabilities are positive. The horizontal line indicates a posterior probability of 90% that a gene is differentially expressed.

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

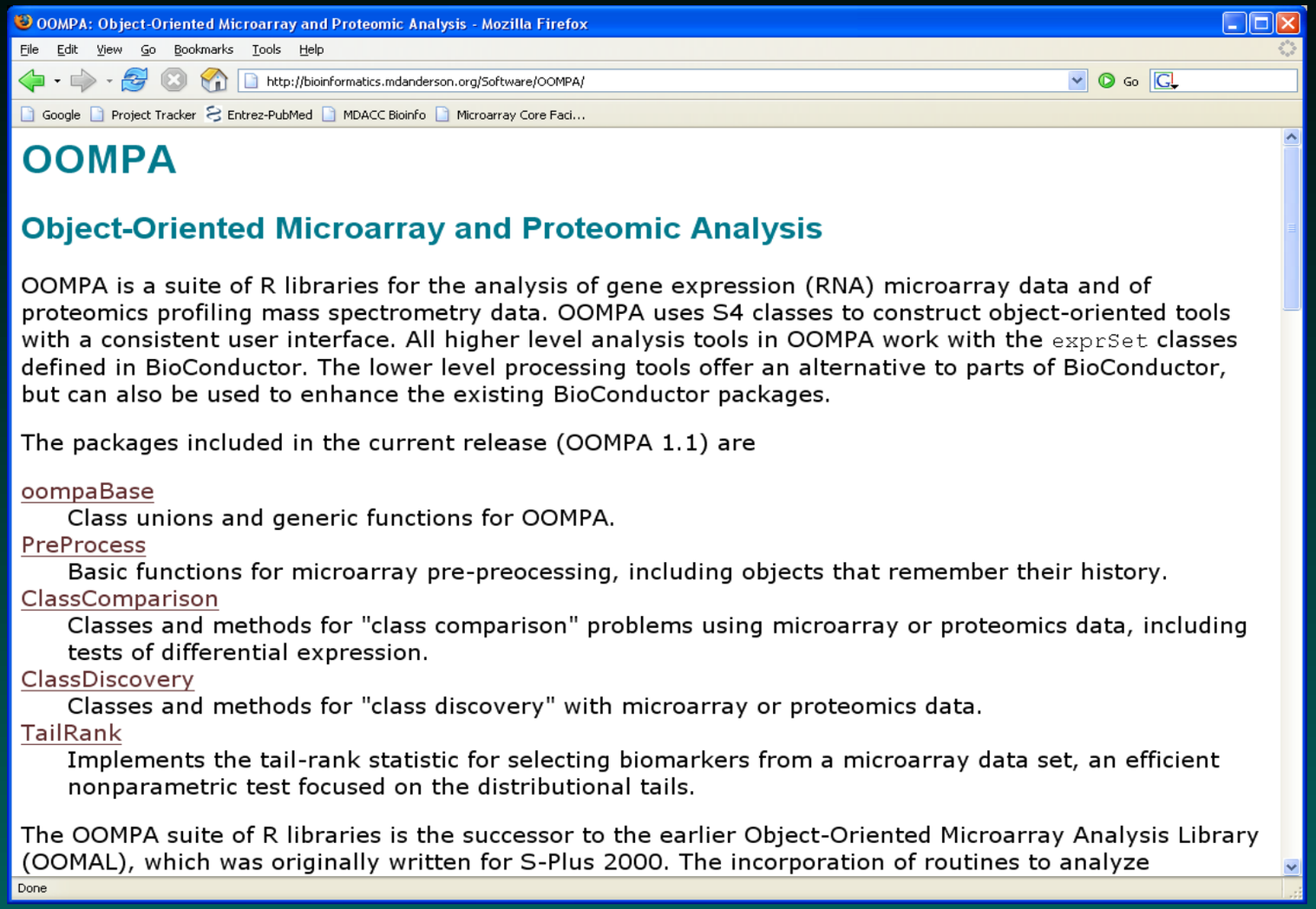
How does this work in R?

We have implemented this idea in an R package:
http://bioinformatics.mdanderson.org/software.html



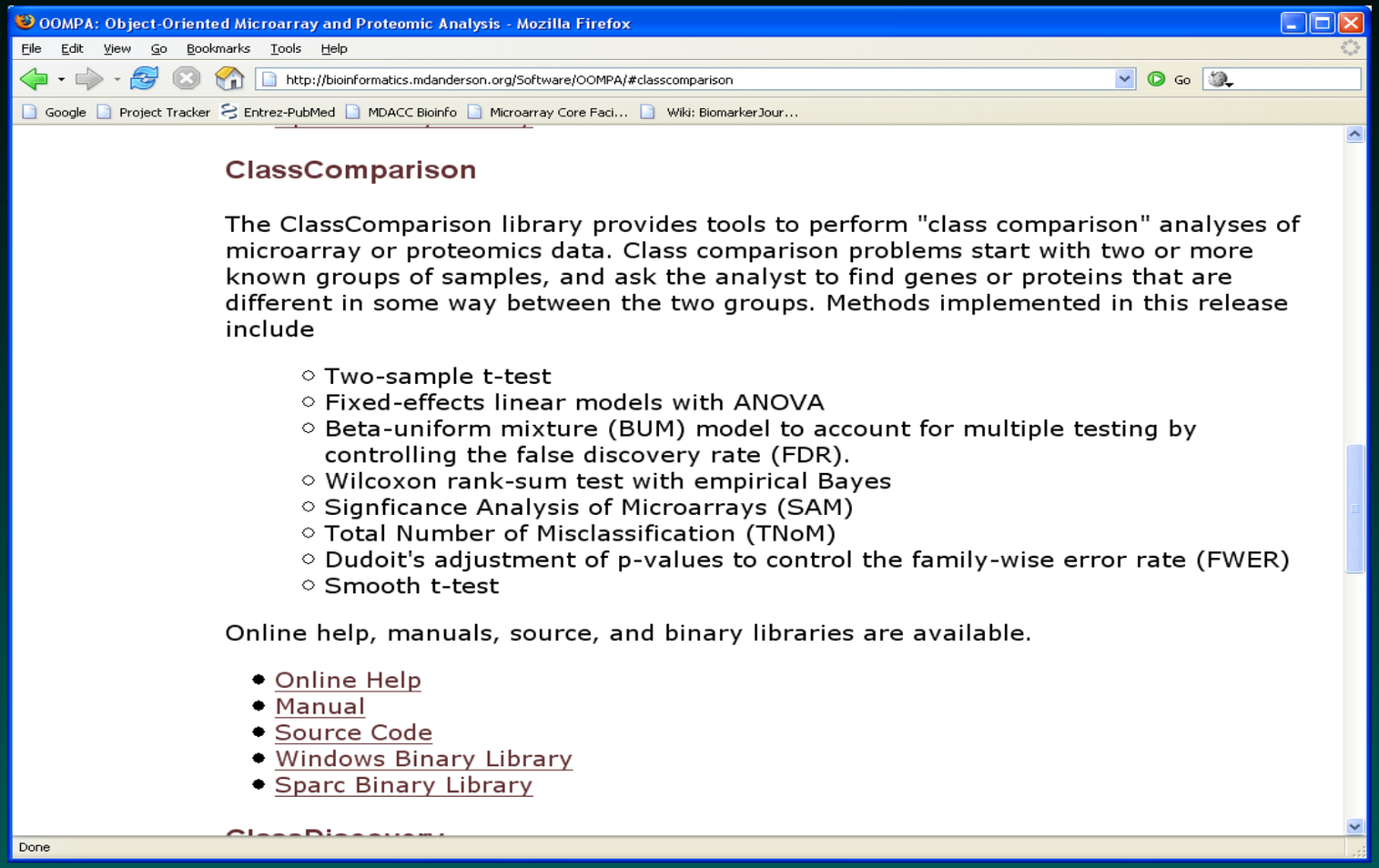
© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

The OOMPA home page



© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

The ClassComparison package



© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Empirical Bayes in R

The ClassComparison package implements the empirical Bayes method with Wilcoxon statistics. For the computations shown above, do the following:

```
> require(ClassComparison)
> efron <- MultiWilcoxonTest(expression.data,
+   status)
> hist(efron)
> plot(efron, prior=c(0.7, 0.8, 0.9, 1.0),
+   signif=NULL)
> abline(h=0)
> plot(efron, prior=0.75, ylim=c(0,1))
> abline(h=0)
```

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

How many genes are differentially expressed?

As a crude estimate, the fact that we had to take $p_0 = 0.75$ suggests that 25% of the genes may be different; this is consistent with the BUM estimate from last time. With a cutoff of 90% on the posterior probability, we get:

```
> cutoffSignificant(efron, prior=0.75, signif=0.9)
$low
[1] 68
$high
[1] 143
> sum(efron@rank.sum.statistics < 68)
[1] 839
> sum(efron@rank.sum.statistics > 143)
[1] 825
```

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

The Tail-Rank Test

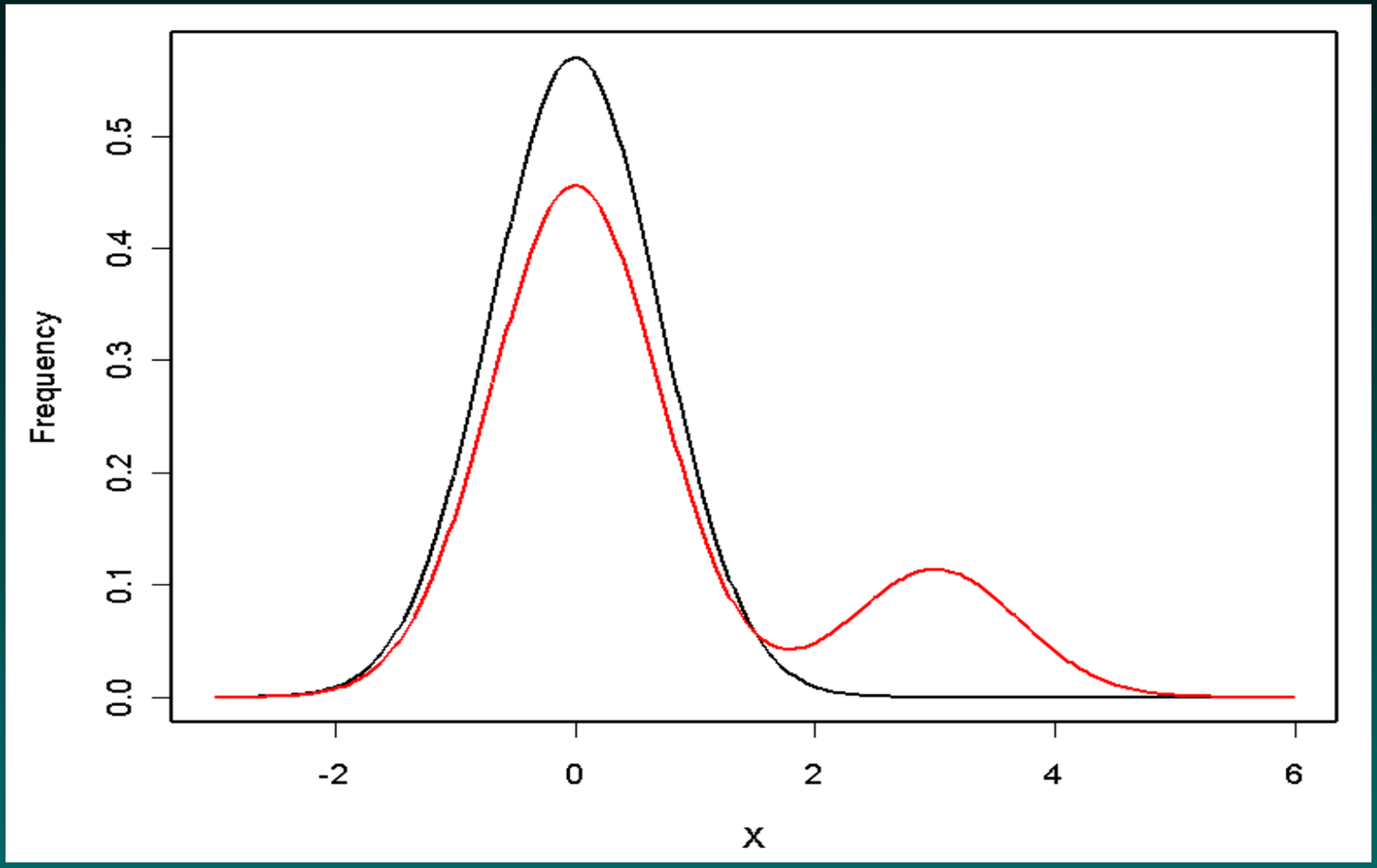
The Wilcoxon rank-sum test and the t-test both look at the same property: is the mean expression the same? When looking for cancer biomarkers, this may well be the wrong question.

Cancers that are histologically the same are not identical. Deletion of part of chromosome 3 (3p14-p23) is found in 50% of non-small-cell lung cancers; MYC amplification is found in 14% of stomach cancers; BRCA1 mutations are found in a subset of breast cancers; a translocation between chromosomes 11 and 14 occurs in 35% of mantle cell lymphomas. These genetic abnormalities directly causes specific differences in gene expression that only occur in a subset of cancers. Statistically, these results suggest that the distributions of gene expression in cancer patients are likely to differ from the healthy distributions in much more than the location of the center.

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Motivation: Subset Biomarkers

If a biomarker is only present in 20% of the cancer samples, then the distributions might look something like this.



© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Outline of The Tail-Rank Test

- Collect data on G genes from n_H healthy individuals. Write $X_{g,i}$ for measurement of gene g on individual i. Assume for fixed g that $X_{g,i} \sim X_g$ are IID.
- Specify a target value ψ for specificity.
- Estimate, for each g, a threshold τ_g such that $Prob(X_g < \tau_g) = \psi$.
- Collect data from n_C cancer patients. Count the number Y_g of cancer patients for which the measured expression level of gene g exceeds τ_g; we call Y_g the tail-rank statistic.
- Call g a biomarker if Y_g exceeds a certain threshold.

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

The null distribution

Null hypothesis: gene g is not a useful biomarker.

More precisely: the measurements on cancer patients have the same distribution as the measurements from healthy individuals.

Then: all Y_g have identical binomial distributions,

$$Y_g \sim Y = Binom(n_C, 1 - \psi).$$

The point here is that the probability of being in the tail is the same for healthy and cancer, and is given by $1 - \psi$, where ψ was the desired specificity.

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Even when we perform the same test for G genes, the expected maximum value of G independent instances of Y_g remains small.

Let $M_G = max_{g=1...G}(Y_g)$ be the maximum over G IID binomial random variables. Also, let

$$\begin{aligned} \alpha &= \alpha(m) = Prob(Y > m) \\ \gamma &= Prob(M_G > m) \end{aligned}$$

Then

$$\begin{aligned} 1 - \gamma &= Prob(M_G \leq m) \\ &= Prob(Y_1 \leq m, \ldots, Y_G \leq m) \\ &= Prob(Y \leq m)^G = (1 - \alpha)^G. \end{aligned}$$

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

The maximum value expected by chance

Solving,

$$\alpha = 1 - (1 - \gamma)^{1/G}.$$

and m is the $(1 - \alpha)^{\text{th}}$ quantile of a single binomial distribution:

	$\gamma = 0.01$, $\psi = 0.99$			
n_C	$G = 100$	1000	10000	100000
10	3	3	4	4
20	3	4	5	5
50	5	6	6	7
100	6	7	8	9
250	10	12	13	14
500	15	17	19	20

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Interpretation

One needs to specify two parameters in order to apply the tail-rank test.

1. ψ, the desired specificity of the biomarker

2. γ, the desired bound on the FWER

Then, given the number of genes and the number of cancer samples, the values m in a table like the previous one represent the largest value of Y_g that we would expect to see by chance over the entire microarray. Any gene where we observe $Y_g > m$ is a potential biomarker.

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Tail-rank and real data

We return yet again to our prostate cancer data set. We will now start using the entire data set, which contains 14 samples from normal prostate, 61 prostate cancer samples, and 9 samples from lymph node metastases of prostate cancer. With this number of samples, taking $\gamma = 0.95$ and $\psi = 0.95$, a gene was called a biomarker if at least 16 of the 71 cancer samples were above the threshold.

We assumed that the log ratios of the normal prostate samples were normally distributed. We computed 90% tolerance bounds for the 5^{th} and 95^{th} percentiles, and counted the number of combined prostate cancer samples whose log ratios fell outside these boundaries.

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Tail-rank results

We identified $1,766$ spots that were "positive" biomarkers, since they were present at higher than normal levels in at least 16 cancer samples. We also identified $1,930$ spots that were "negative" biomarkers, since they were expressed at lower than normal levels in at least 16 samples. In total, we identified 3,692 spots as candidate biomarkers.

Although the theory told us the number of false positives should be close to zero, we decided to test this using both simulations and a permutaion test. We simulated completely random (IID normal) data 100 times, and we permuted the samples labels on the real data 100 times.

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

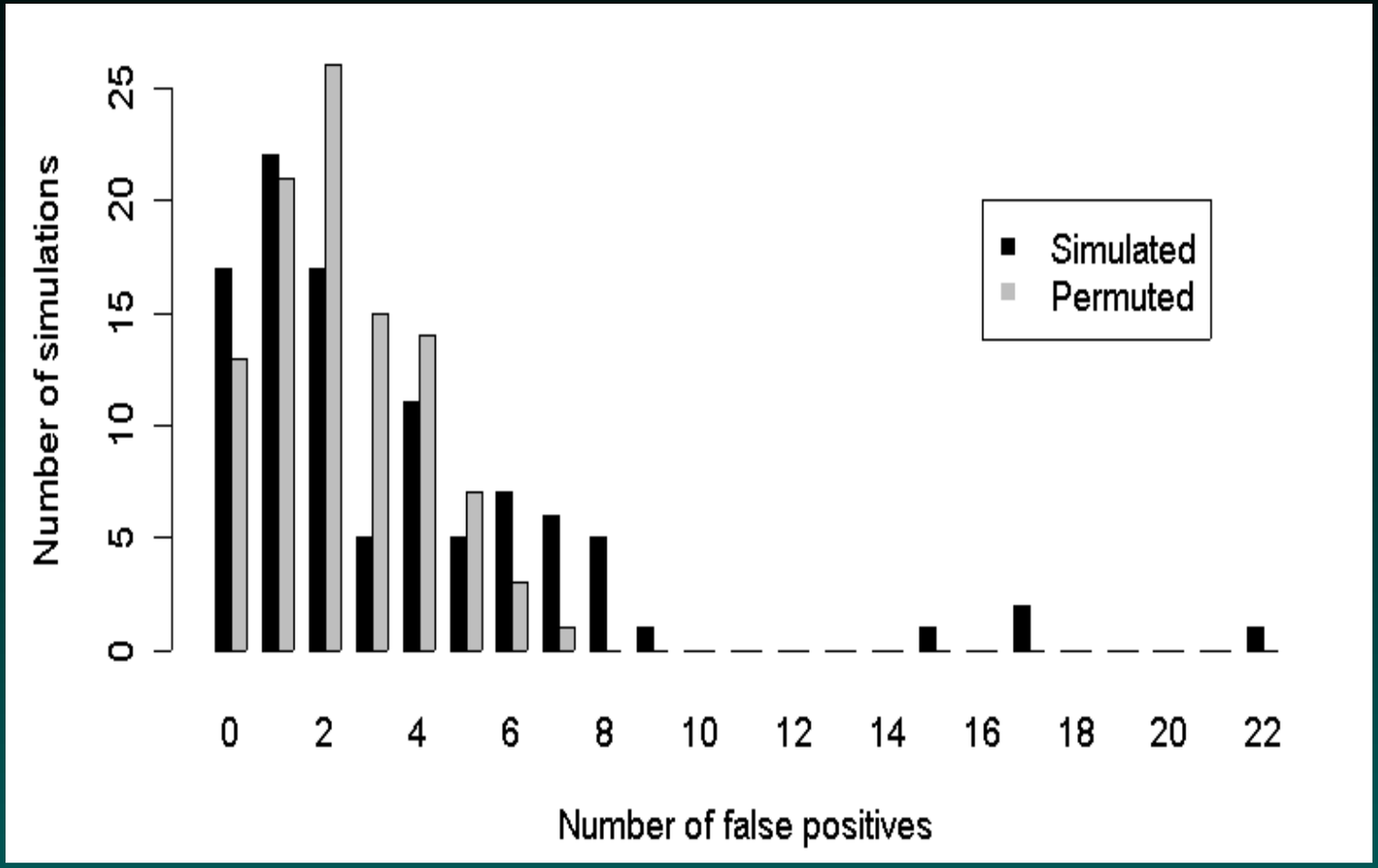
Using the tail-rank test in R

We developed the tail-rank test. A preprint, along with an R package is available on the web at
http://bioinformatics.mdanderson.org/TailRank

```
> require(TailRank)
> tr.test <- TailRankTest(expression.data, status,
+    direction="two")
> summary(tr.test)
A tail-rank test object in the two-sided direction.
Specificity: 0.95 computed with tolerance 0.9.
Significance cutoff: 15 based on a family-wise error
rate less than 0.05.
There are 3692 tail-rank statistics that exceed the
cutoff.
```

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Reviewing significance



Pretty good, when you consider that the test with the real data detected a few thousand potential markers!

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Differential expression results

We repeated the t-test analysis on the full data set (adjusing for multiple testing using BUM). With $FDR < 0.05$, we used a cutoff at $p < 0.000045$ or $|t| > 4.25$. We detected 3,522 differentially expressed spots. Of these, $1,415$ spots were overexpressed in prostate cancer and $2,107$ spots were underexpressed.

We also repeated the Wilcoxon test with the empirical Bayes approach. In order to get comparable results, we selected a cutoff corresponding to a posterior probability of 99.9%. We detected 3,627 differentially expressed spots. Of these, $1,498$ spots were overexpressed and $2,129$ spots were underexpressed in prostate cancer.

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

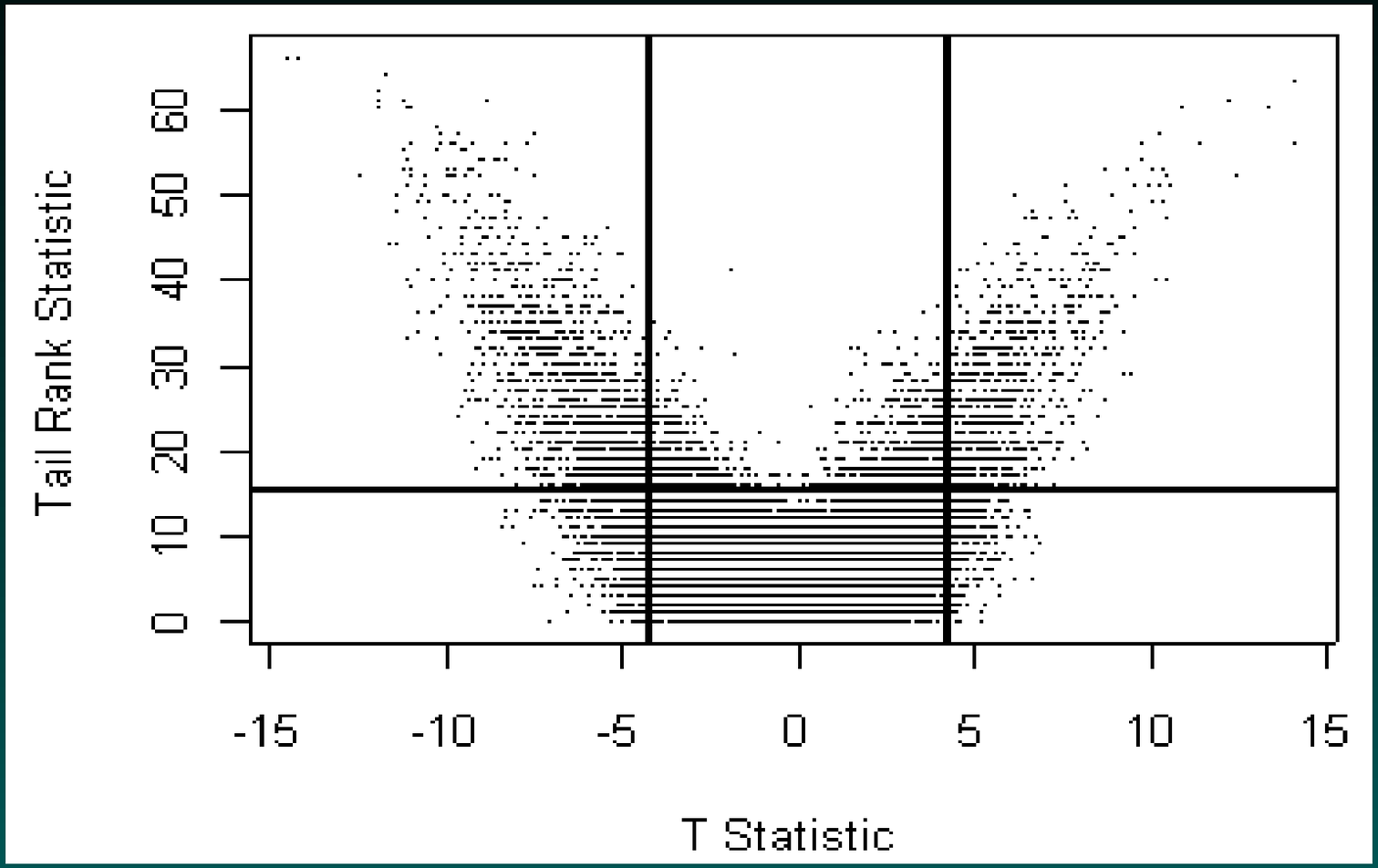
Comparing tests

The number of genes found by the three tests was very similar. Are they finding the same things?

There was good agreement between the t-test and the Wilcoxon test. More than 90% $(1,905)$ of underexpressed and 88% $(1,244)$ of overexpressed spots that were found by the t-test were also detected by the Wilcoxon test. So, we only need to compare one of these to the tail-rank test.

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Comparing tests



Lower left and right = different by T, not by tail-rank

Upper center = different by tail-rank, not by T.

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Looking at the results

Since the tests give different answers, which one should we believe?

Both, since they are giving the answers to different questions!

Whether you perform one test or many, however, it is useful to look at the expression values for some of the genes that you find, if only to make sure you believe the results.

A useful R function for this purpose is `stripchart`. Here is an example for our data set. First, we get the clinical status as an `ordered` factor.

```
> x <- ordered(clinical.info$Status,
+     levels=c('N', 'T', 'L'))
```

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Selecting interesting genes

Now we look at genes with small tail-rank statistics (< 2) and significant t-statistics.

```
> tr.stats <- getStatistic(tr.test)
> k.weird <- tr.stats < 2 & (abs(t.stats) > 4.25)
> sum(k.weird)
[1] 38
```

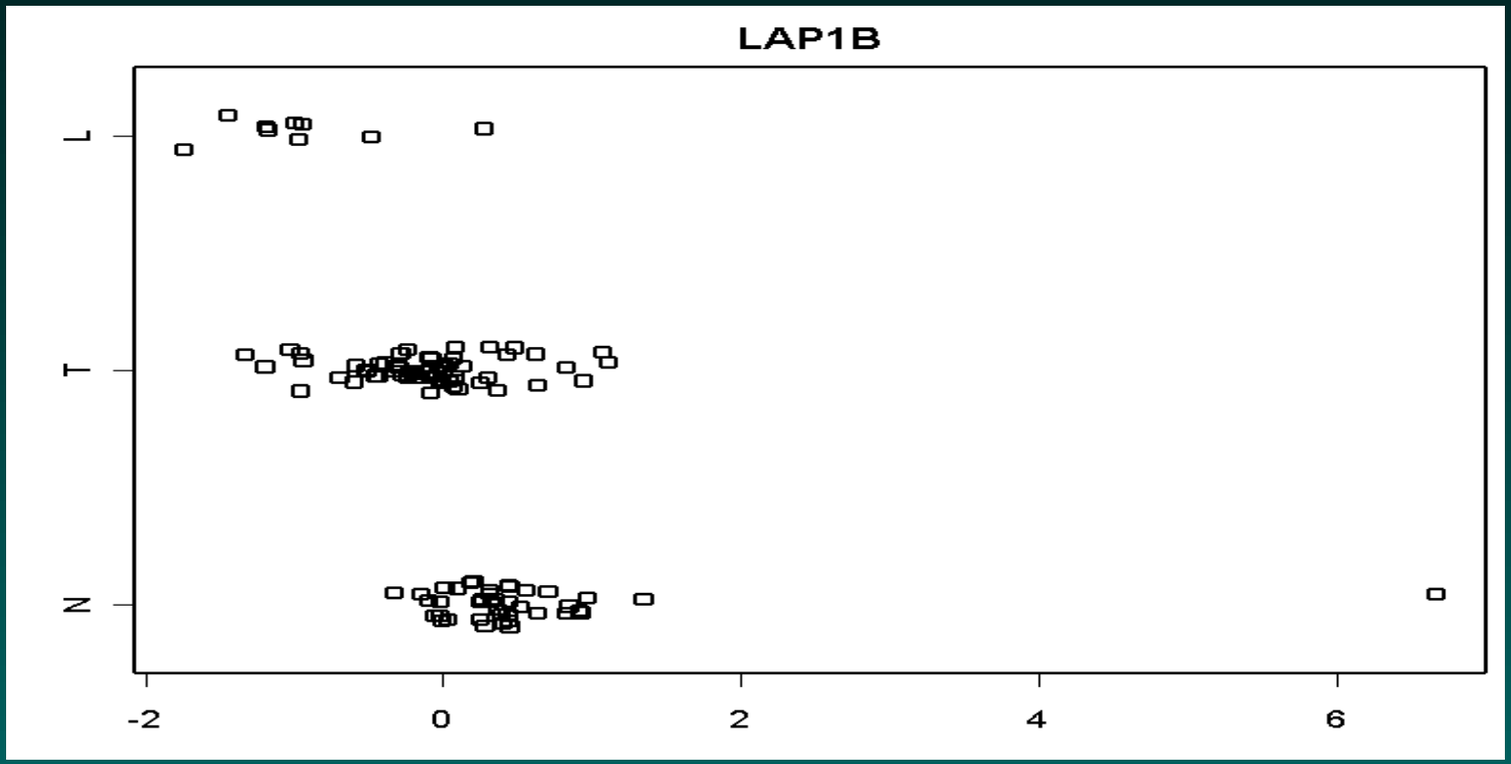
We can select one of the "weird" genes and get its expression data.

```
> i.k.weird <- (1:length(k.weird))[k.weird]
> i <- sample(i.k.weird, 1)
> y <- as.vector(t(expression.data[i,]))
```

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

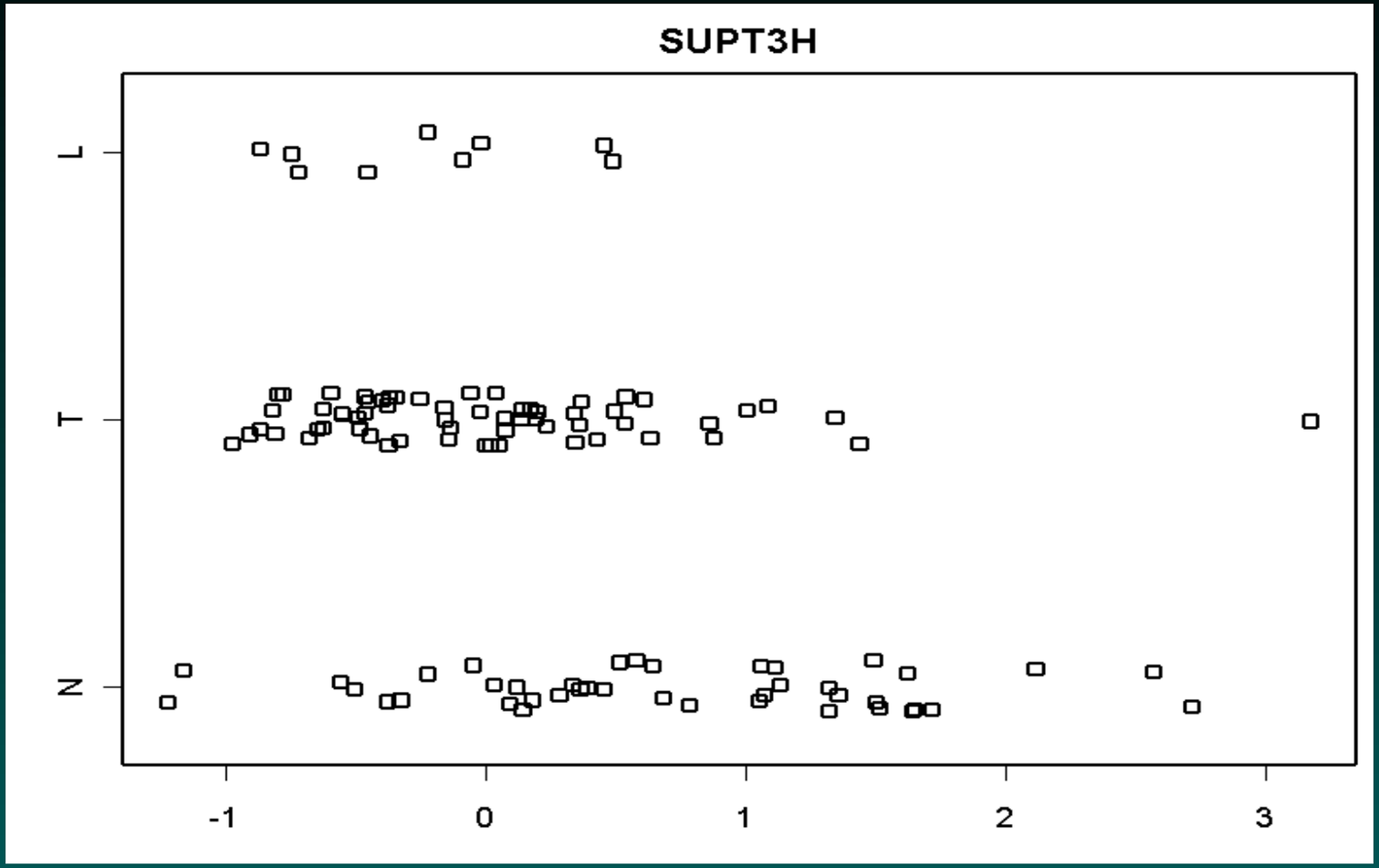
Outliers can throw off the estimates

```
> label <- as.character(gene.info$Gene.Symbol[i])
> stripchart(y ~ x, xlab='', main=label,
+   method='jitter')
```



© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Some genes are normally variable



© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

Selecting interesting biomarkers

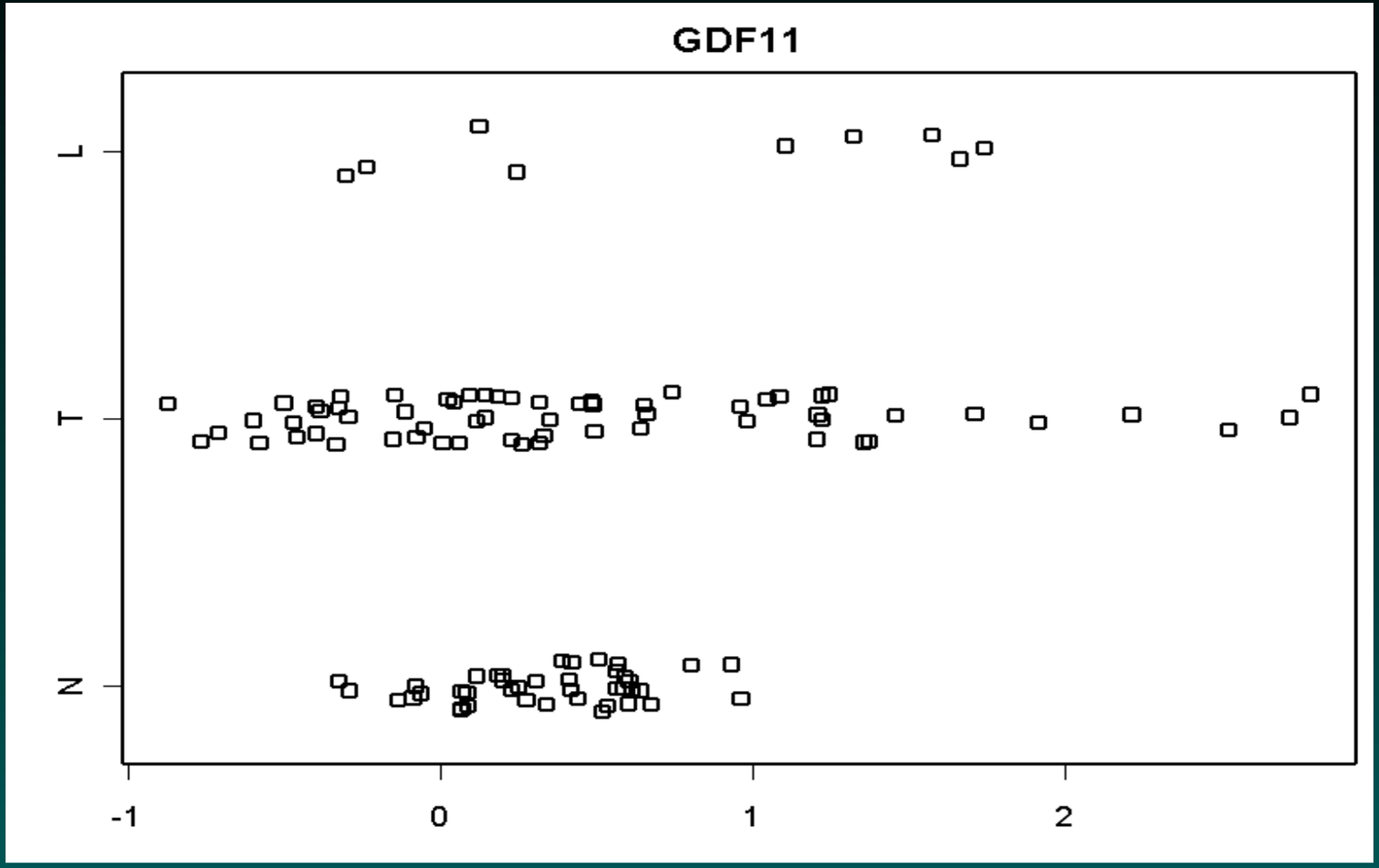
Now we look at genes with significant tail-rank statistics and small t-statistics ($|t| < 1.25$).

```
> k.weird <- tr.stats > 15 & (abs(t.stats) < 1.25)
> sum(k.weird)
[1] 52
```

We use the same idea to select some of these genes and plot stripcharts to see if the values agree with what we tihnk the test should be doing.

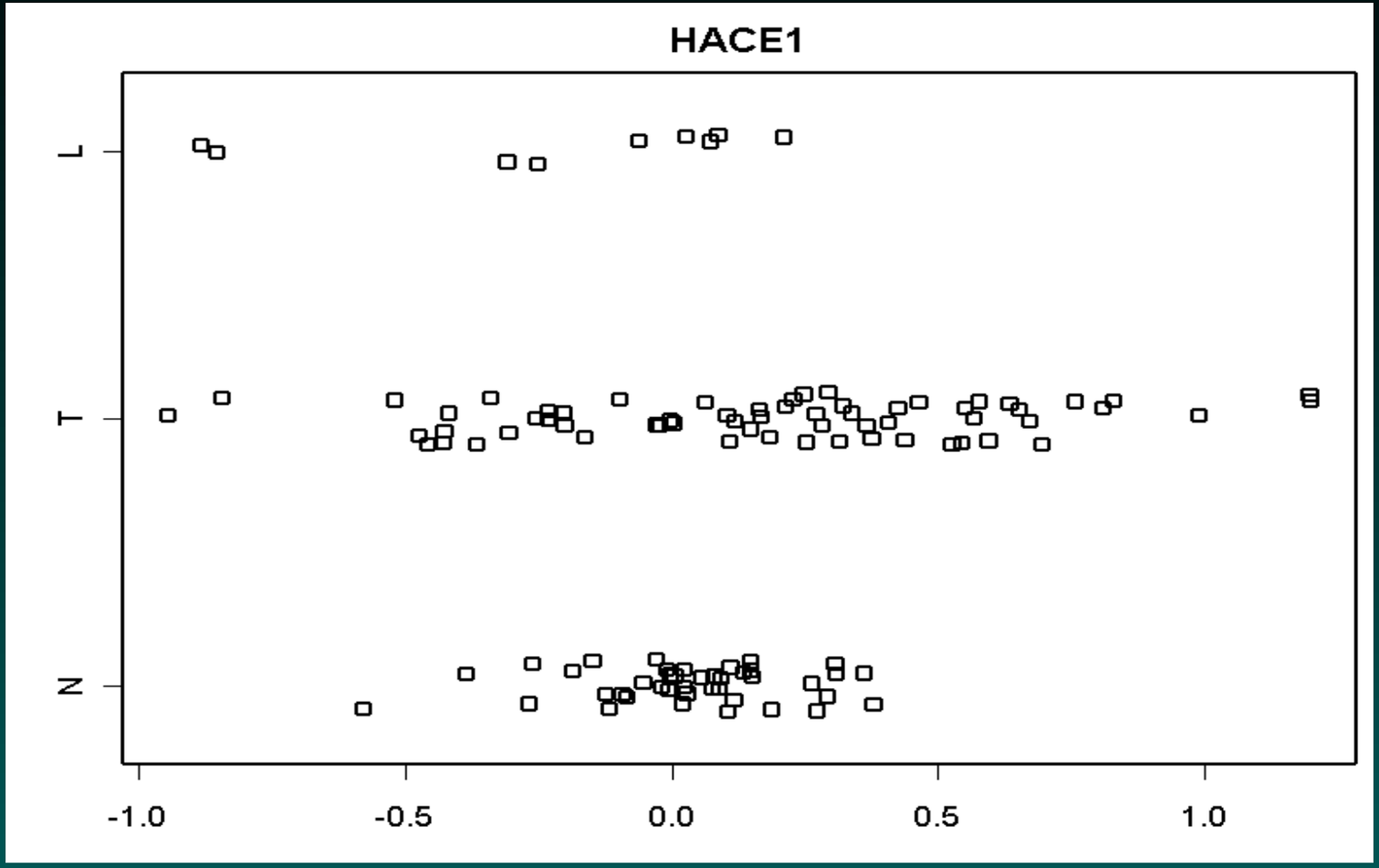
© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

GDF11



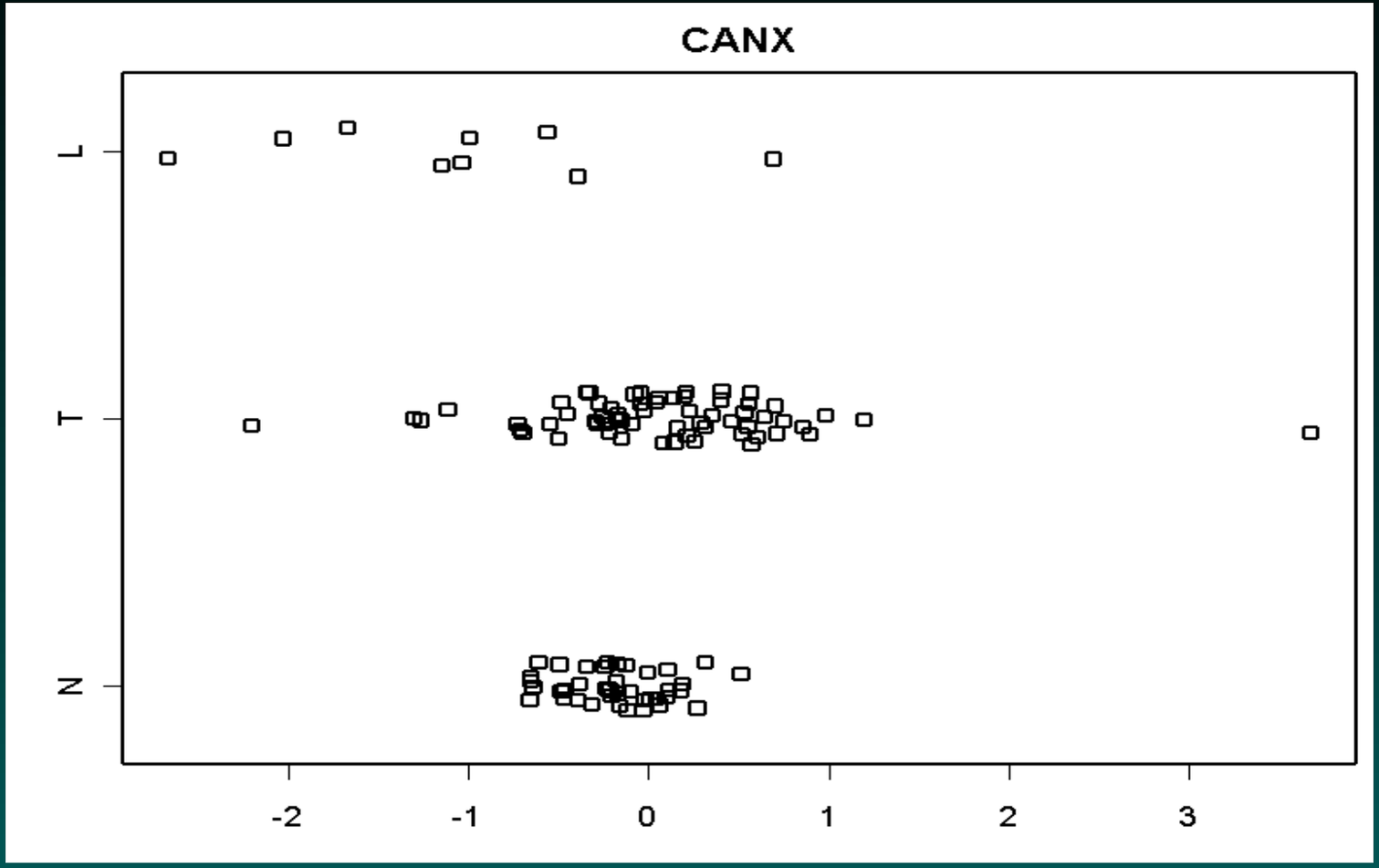
© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

HACE1



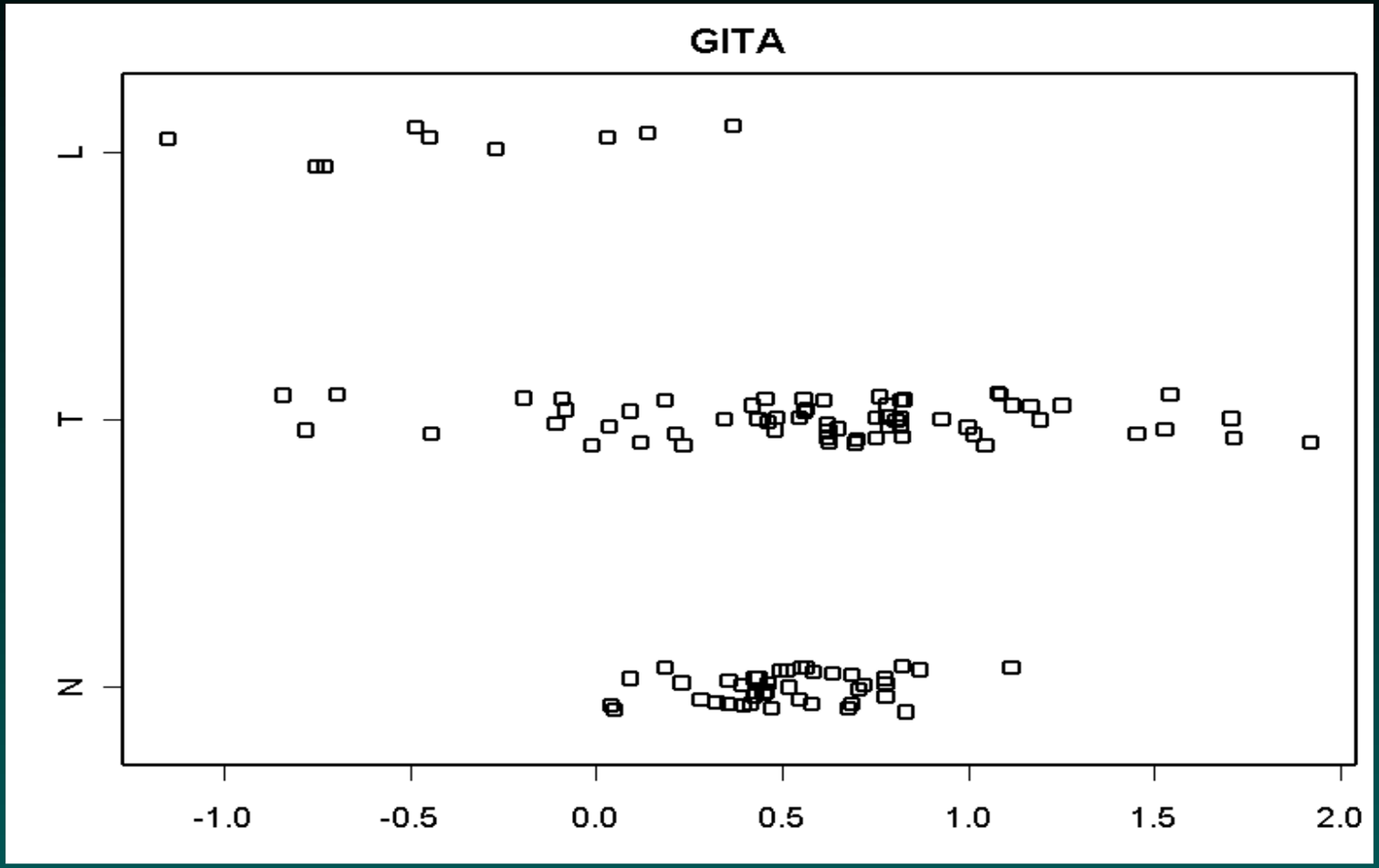
© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

CANX



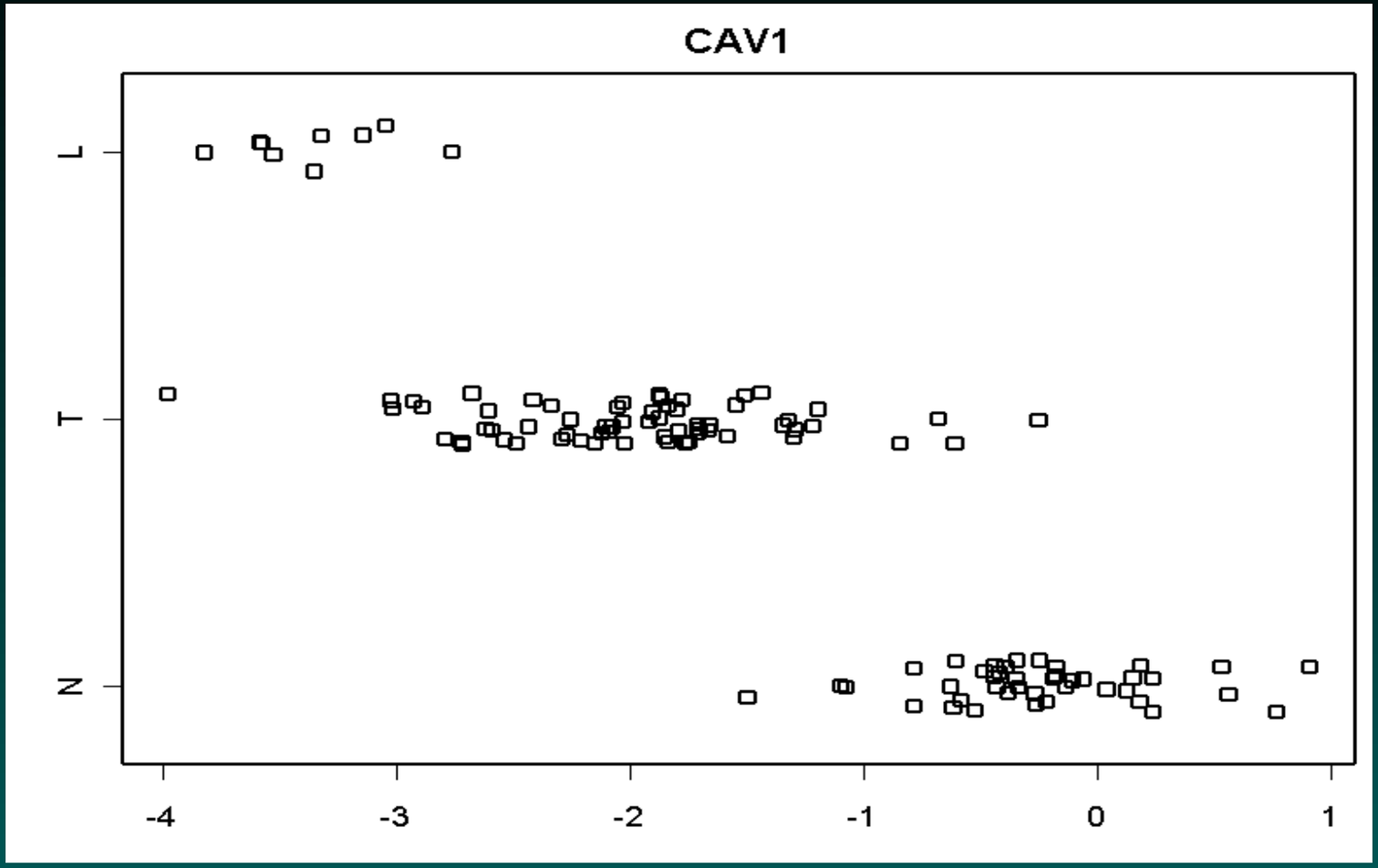
© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

GITA



© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

When T and Tail-Rank agree



© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly

More Examples with ClassComparison

```
> require(ClassComparison}
> # perform t-tests
> tea <- MultiTtest(exprs(prostate), status)
> # beta-uniform mixture model
> bum <- Bum(tea@p.values)
> hist(bum)
> # Dudoit's method
> dudoit <- Dudoit(exprs(prostate), status)
> plot(dudoit)
> # significane analysis of microarrays
> sam <- Sam(exprs(prostate), status)
> plot(sam)
```

© Copyright 2004–2007, Kevin R. Coombes and Keith A. Baggerly