The Tail Rank Test

Kevin R. Coombes

January 10, 2007

Contents

1 Introduction 1
2 Getting Started 1
3 Performing the Tail Rank Test 2
   3.1 Which genes are significant? . . . . . . . . . . . . . . . . 3
4 Power Computations 4
5 References 6

1 Introduction

OOMPA is a suite of object-oriented tools for processing and analyzing large biological data sets, such as those arising from mRNA expression microarrays or mass spectrometry proteomics.

This vignette documents the tail rank test, which provides an alternative method for discovering potential biomarkers in large data sets. The idea is that one starts with a target specificity for a gene as a univariate biomarker, and uses the “normal” or “baseline” samples to estimate a threshold that yields that specificity. Then, for each gene, one counts the number of “cancer” or “experimental” samples that exceed the gene-specific threshold. Significance is determined based on control of the family-wise error rate (FWER).

2 Getting Started

As usual, we start by loading the library.

> library(TailRank)

The TailRank package comes with some sample data that we can use to illustrate the methods. The sample data consists of a subset containing 2000
genes from a prostate cancer study on glass arrays reported by Lapointe and colleagues [1]. The next set of commands loads the data.

```r
data(expression.data)
data(gene.info)
data(clinical.info)
```

There are 112 samples in the study. The Subgroups column of the clinical.info data frame refers to the subgroups discovered in the original publication by clustering based on the gene expression data. The ChipType column identifies the two different generations of glass arrays that were combined in the study. The Status column classifies the samples as normal prostate (N), primary prostate tumor (T), or lymph node metastasis (L). Since there is a natural order to this status in terms of the severity of the disease, we are going to make certain that it is used:

```r
clinical.info$Status <- ordered(clinical.info$Status, levels = c("N", "T", "L"))
```

```r
dim(clinical.info)

[1] 112 6
```

```r
summary(clinical.info)

    Arrays Reference Sample Status Subgroups ChipType
p16090 : 1 CRG1 : 1 PL114 : 1 N:41 I :11 new:86
p16093 : 1 CRG10 : 1 PL115 : 1 T:62 II :39 old:26
p16095 : 1 CRG100 : 1 PL116 : 1 L: 9 III:19
p16097 : 1 CRG101 : 1 PL118.3: 1 N :41
p16098 : 1 CRG102 : 1 PL122 : 1 O : 2
p16101 : 1 CRG103 : 1 PL129 : 1 (Other):106 (Other):106 (Other):106
```

### 3 Performing the Tail Rank Test

The main function in the package is the TailRankTest. We start by invoking this function with the default values of the arguments. The summary includes details on the parameters that were used, along with the fact that 49 of the 2000 genes were more highly expressed in non-normal samples than would be expected by chance, based on a 5% FWER.

```r
trt <- TailRankTest(expression.data, clinical.info$Status)
summary(trt)
```

A tail-rank test object in the up direction.
The test was performed using the bb model.
Specificity: 0.95 computed with tolerance 0.5
Significance cutoff: 24 based on a family-wise error rate less than 0.05
There are 49 tail-rank statistics that exceed the cutoff

2
In the next example, we increase both the target specificity (from the default of 95% to a desired value of 99%) and the desired confidence limits (to 99% from the default of 95%). With this more stringent criteria, only 25 of the genes remain significant.

```r
trt2 <- TailRankTest(expression.data, clinical.info$Status, specificity = 0.99, +   confidence = 0.99)
> summary(trt2)
A tail-rank test object in the up direction.
The test was performed using the bb model.
Specificity: 0.99 computed with tolerance 0.5
Significance cutoff: 19 based on a family-wise error rate less than 0.01
There are 25 tail-rank statistics that exceed the cutoff
```

### 3.1 Which genes are significant?

After performing an analysis that identifies a gene list like this, it is, of course, natural to want to know which genes were selected. The `as.logical` method converts the results of the tial rank test into a logical vector that selects these significant genes. Using this method, we can verify that the 25 genes selected by the more stringent criteria are a subset of the 49 genes selected using the weaker criteria.

```r
> sel <- as.logical(trt)
> sel2 <- as.logical(trt2)
> sum(sel2 & sel)
[1] 25
```

Since this vector serves as an index into the `gene.info` database, we can figure out which genes were actually selected.

```r
> gene.info[sel2, 3:6]
(Clone.ID Gene.Symbol Cluster.ID Accession)
X2180 IMAGE:506669 LOC170394 Hs.157728 AA708916
X23774 IMAGE:244350 Hs.484965 N54811
X5918 IMAGE:26883
X11386 IMAGE:302331 MYL4 Hs.356717 AI688645
X27346 IMAGE:376764 UTRN Hs.250607 AA046146
X26538 IMAGE:364934 DAPK1 Hs.244318 AA024655
X17798 IMAGE:838829 PLU-1 Hs.143323 AA464869
X7405 IMAGE:814528 TP53INP1 Hs.75497 AA459364
X27228 IMAGE:47475 CYFIP2 Hs.211201 H12043
X40508 IMAGE:811582 Hs.459841 AA454597
X12648 IMAGE:809421 PCBD Hs.3192 AA442959
X17040 IMAGE:258175 Hs.23754 N30900
```
4 Power Computations

The power depends on the number of genes (G), the number of healthy samples (N1), the number of cancer samples (N2), the target specificity (psi), the confidence (conf = 1 - FWER), and the sensitivity that you want to be able to detect (phi). Here is an example using the sizes from the prostate cancer data set, showing that we have more than 70% power to detect a marker with 40% sensitivity.

```r
> tailRankPower(2000, N1 = 41, N2 = 71, psi = 0.95, phi = 0.4, + conf = 0.95)
[1] 0.7135006
```

The next example shows that the power decreases to 43% when using the same number of samples with a whole genome array containing 40000 gene probes. (This was the size of the full study from which these 2000 genes were randomly selected.)

```r
> tailRankPower(40000, N1 = 41, N2 = 71, psi = 0.95, phi = 0.4, + conf = 0.95)
[1] 0.4271892
```

We can determine the power for a variety of cancer sample sizes, keeping everything else the same

```r
> tailRankPower(40000, N1 = 41, N2 = seq(40, 100, by = 10), psi = 0.95, + phi = 0.4, conf = 0.95)
[1] 0.2063922 0.3067931 0.3920020 0.4033145 0.4648195 0.4673351 0.5137604
```

More generally, we can create power tables:
```r
> biomarkerPowerTable(G = c(10000, 20000, 40000), N1 = 41, N2 = seq(40, + 100, by = 10), psi = 0.95, conf = 0.95, phi = seq(0.3, 0.5, + by = 0.05))

[[1]]
An object of class `AIJBMPT`
Slot "G":
[1] 10000

Slot "psi":
[1] 0.95

Slot "conf":
[1] 0.95

Slot "power":
30 35 40 45 50
40 0.09033665 0.1991811 0.3571932 0.5409076 0.7144119
50 0.12613959 0.2663173 0.4525563 0.6462431 0.8062082
60 0.12068814 0.2640932 0.4577585 0.6584144 0.8203311
70 0.14804875 0.3129456 0.5212980 0.7202351 0.8659786
80 0.1247308 0.3543683 0.5719672 0.7654303 0.8962882
90 0.19413312 0.3895314 0.6121234 0.7992573 0.9171966
100 0.21332331 0.4195350 0.6449771 0.8251594 0.9321230

[[2]]
An object of class `AIJBMPT`
Slot "G":
[1] 20000

Slot "psi":
[1] 0.95

Slot "conf":
[1] 0.95

Slot "power":
30 35 40 45 50
40 0.05886264 0.1420590 0.2765896 0.4504793 0.6328099
50 0.09058714 0.2069445 0.3772638 0.5720731 0.7487870
60 0.12068814 0.2640932 0.4577585 0.6584144 0.8203311
70 0.11639120 0.2621789 0.4617095 0.6678071 0.8309751
80 0.14046885 0.3057095 0.5182606 0.7220098 0.8698565
90 0.16239855 0.3434290 0.5642925 0.7629450 0.8967888
100 0.15428558 0.3344203 0.5581562 0.7609404 0.8972354
```
An object of class aĂĐJEMPȚĂĂ
Slot "G":
[1] 40000

Slot "psi":
[1] 0.95

Slot "conf":
[1] 0.95

Slot "power":

<table>
<thead>
<tr>
<th></th>
<th>30</th>
<th>35</th>
<th>40</th>
<th>45</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>0.03682533</td>
<td>0.09740169</td>
<td>0.2063922</td>
<td>0.3628918</td>
<td>0.5449930</td>
</tr>
<tr>
<td>50</td>
<td>0.06324274</td>
<td>0.15651685</td>
<td>0.3067931</td>
<td>0.4956936</td>
<td>0.6837994</td>
</tr>
<tr>
<td>60</td>
<td>0.09038819</td>
<td>0.21232817</td>
<td>0.3920020</td>
<td>0.5948291</td>
<td>0.7729542</td>
</tr>
<tr>
<td>70</td>
<td>0.09001682</td>
<td>0.21627298</td>
<td>0.4033145</td>
<td>0.6121954</td>
<td>0.7908250</td>
</tr>
<tr>
<td>80</td>
<td>0.11292231</td>
<td>0.26053337</td>
<td>0.4648195</td>
<td>0.6752875</td>
<td>0.8392830</td>
</tr>
<tr>
<td>90</td>
<td>0.11006564</td>
<td>0.25911220</td>
<td>0.4673351</td>
<td>0.6813917</td>
<td>0.8459462</td>
</tr>
<tr>
<td>100</td>
<td>0.12947138</td>
<td>0.29490766</td>
<td>0.5137604</td>
<td>0.7248561</td>
<td>0.8757409</td>
</tr>
</tbody>
</table>

5 References