

Bimodality Index

Kevin R. Coombes

May 6, 2019

Contents

1 Simulated Data	1
2 Computing the Bimodal Index	1
3 Appendix	2

1 Simulated Data

We simulate a dataset.

```
> set.seed(564684)
> nSamples <- 60
> nGenes <- 3000
> dataset <- matrix(rnorm(nSamples*nGenes), ncol=nSamples, nrow=nGenes)
> dimnames(dataset) <- list(paste("G", 1:nGenes, sep=''),
+                           paste("S", 1:nSamples, sep=''))
```

At present, this dataset has no interesting structure; all genes have their expression patterns drawn from a common normal distribution. So, we shift the means by three standard deviations for half the samples for the first 100 genes.

```
> dataset[1:100, 1:30] <- dataset[1:100, 1:30] + 3
```

2 Computing the Bimodal Index

In order to compute the bimodal index from Wang et al. (2009) <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2730180>, we must load the package.

```
> library(BimodalIndex)
```

Now we call the basic function:

```
> bim <- bimodalIndex(dataset)
```

```
1 .....
2 .....
3 .....
4
```

```
> summary(bim)
```

mu1	mu2	sigma	delta
Min. : -4.3546	Min. : -0.1689	Min. : 0.3941	Min. : 0.3182
1st Qu.: -0.8958	1st Qu.: 0.3900	1st Qu.: 0.6742	1st Qu.: 1.5785
Median : -0.5944	Median : 0.6270	Median : 0.7590	Median : 2.0552
Mean : -0.6996	Mean : 0.7922	Mean : 0.7690	Mean : 1.9962
3rd Qu.: -0.3454	3rd Qu.: 0.9623	3rd Qu.: 0.8579	3rd Qu.: 2.4705
Max. : 0.5800	Max. : 4.0833	Max. : 1.3067	Max. : 4.6638

pi	BI
Min. : 0.01682	Min. : 0.1589
1st Qu.: 0.37812	1st Qu.: 0.6341
Median : 0.50043	Median : 0.8560
Mean : 0.49958	Mean : 0.8546
3rd Qu.: 0.62829	3rd Qu.: 1.0646
Max. : 0.98309	Max. : 2.2457

Here we see a suggestion that at least some of the values are likely to be above a reasonable cutoff to be called significant.

Next, we plot the results, with the known bimodal genes colored red (Figure ??). As expected, most (but not all) of the large BI values arise from the known bimodal genes. We can then use the simulations from the null model to estimate reasonable significance cutoffs when using 60 samples.

```
> summary(bim$BI[101:3000])
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.1589	0.6251	0.8418	0.8285	1.0431	1.7491

```
> cutoffs <- quantile(bim$BI[101:3000], probs=c(0.90, 0.95, 0.99))
```

```
> cutoffs
```

90%	95%	99%
1.214219	1.310714	1.476804

Now we can assess the sensitivity of the test when using the derived cutoffs.

```
> sapply(cutoffs, function(x) sum(bim$BI[1:100] > x))
```

90%	95%	99%
94	91	78

With real data, of course, we would need to determine the significance by simulating a large number of genes from the null model, using the simulations to compute empirical p-values. Because these p-values would still be computed one gene at a time, it would be advisable to incorporate a multiple testing criterion by, for example, estimating the false discovery rate.

3 Appendix

This analysis was performed in the following directory:

```
> getwd()
```

```
> plot(bim$BI, col=rep(c("red", "black"), times=c(100, 2900)),  
+       xlab="Gene", ylab="Bimodal Index")
```

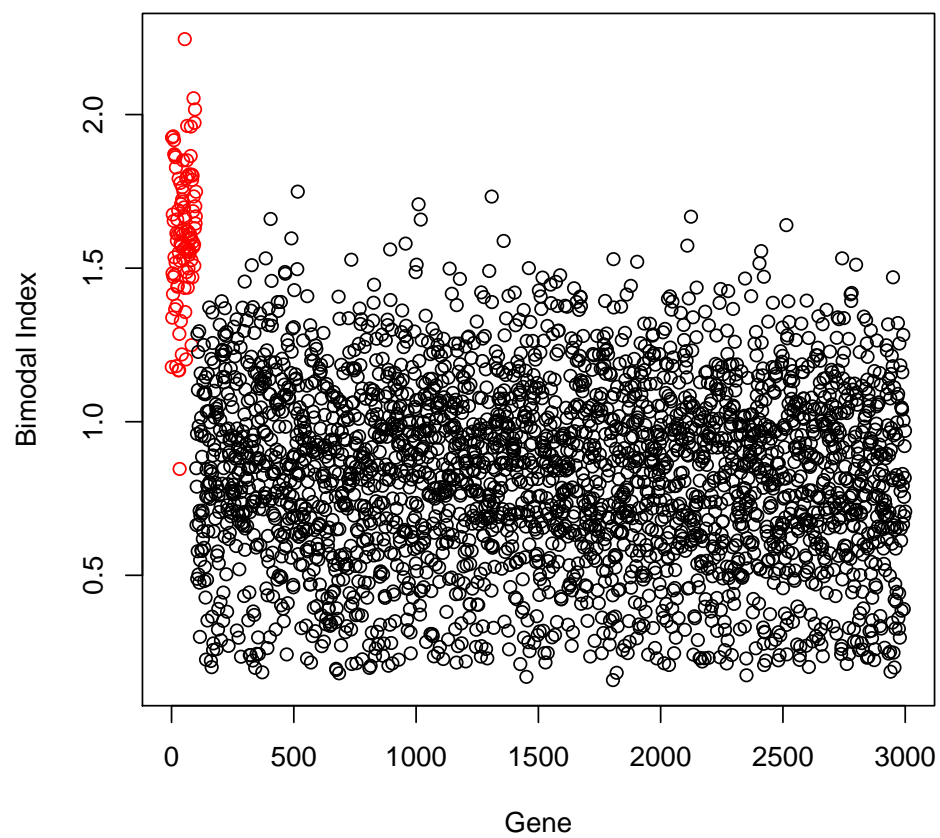


Figure 1: Scatter plot of the bimodal indices of all genes.

```
[1] "/tmp/RtmpjppF Xu/Rbuild5da2f12dbd0/BimodalIndex/vignettes"
```

This analysis was performed in the following software environment:

```
> sessionInfo()
```

```
R version 3.5.3 (2019-03-11)
```

```
Platform: x86_64-pc-linux-gnu (64-bit)
```

```
Running under: Debian GNU/Linux buster/sid
```

```
Matrix products: default
```

```
BLAS: /srv/R/R-patched/build.19-04-05/lib/libRblas.so
```

```
LAPACK: /srv/R/R-patched/build.19-04-05/lib/libRlapack.so
```

```
locale:
```

```
[1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C              LC_TIME=en_US.UTF-8
[4] LC_COLLATE=C             LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=en_US.UTF-8     LC_NAME=C                 LC_ADDRESS=C
[10] LC_TELEPHONE=C          LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
```

```
attached base packages:
```

```
[1] stats      graphics  grDevices  utils      datasets  methods    base
```

```
other attached packages:
```

```
[1] BimodalIndex_1.1.9
```

```
loaded via a namespace (and not attached):
```

```
[1] compiler_3.5.3  mclust_5.4.3    tools_3.5.3     ompaBase_3.2.8  cluster_2.0.9
```