shape: Shape analysis of high-throughput experiments data

Kwame Okrah*, Héctor Corrada-Bravo

Applied Mathematics, Statistics, and Scientific Computation
Center for Bioinformatics and Computational Biology
University of Maryland, College Park

*kwame.okrah (at) gmail.com

January 6, 2015

Abstract

Given a dataset we first try to characterize its central value (location) with the sample mean (or sample median) and its spread around the location (variance) with the sample standard deviation (or sample range). When the sample size, n, is sufficiently large we can begin to assess the shape of the data in some meaningful way. Distributional shape is often characterized by two features (1) Skewness: a measure of how far the shape of the distribution deviates from symmetry around its location and (2) Kurtosis: a measure of how much weight is at the tails of the distribution relative to the weight around the location. This vignette describes the statistical analysis of the shape (skewness and kurtosis) of high-throuput experiments data (eg. RNA-seq, microarry) using the **shape** package. It also demonstrates the detection of transcripts (eg. genes) within a dataset whose sample shape is markedly different from the majority of transcripts in the same dataset. The ability to describe the shape of high-throughput genomics data is useful for two reasons: 1. It enriches the exploratory data analysis step, and 2. It provides a means of checking the distributional assumptions of statisical methods.

If you use **shape** (version 1.0.0) in your work, please cite:

- K. Okrah and H. Bravo. *Shape analysis of high-throughput experiments data.* Under review.
- J. Hosking. *L-moments: analysis and estimation of distributions using linear combinations of order statistics.* Journal of the Royal Statistical Society. Series B (Methodological), **52**, 105–124, 1990.
- K. Okrah. `shape`: Shape analysis of high-throughput experiments data. R package version 1.0.0, 2014 (`https://github.com/kokrah/shape`).

Contents

1 Introduction

Before we begin we first give a brief description of the theory on which our methods are based.

1.1 A numerical summary of shape

Similar to traditional moments, the theory of L-moments forms the basis of many statistical methods such as parameter estimation, hypothesis testing, and model selection. However, L-moments enjoy many theoretical and practical advantages over traditional moments (see [1–4]). In this vignette we focus on its ability to provide robust statistics that summarize a given dataset. The first four L-moments $\lambda_1, \lambda_2, \lambda_3$ and λ_4 measure location, variance, skewness, and kurtosis of data respectively. Unitless measures of relative variance, skewness, and kurtosis are defined as: L-CV $\tau = \lambda_2/\lambda_1$, L-skew $\tau_3 = \lambda_3/\lambda_2$, and L-kurt $\tau_4 = \lambda_4/\lambda_2$. The L-skew ($\tau_3$) coefficient can take any value between (-1, 1); where $\tau_3 = 0$ implies symmetry and $\tau_3 > 0$ ($\tau_3 < 0$) implies skewness to the right (left). The symmetry-outlier plot (SO-plot) conveniently summarizes the shape of each gene as a point on a 2-dimensional plot (see L-moments ratio diagram in [2]). The interpretation of the L-kurt (τ_4) depends on L-skew (in general one would expect a highly skewed data to have a high kurtosis). At $\tau_3 = 0$ and $\tau_4 = 0$ the data has the shape of a uniformly distributed random variable. As τ_4 (positive) increases the data becomes bell shaped, for example at $\tau_4 = 0.1226$ the data is normally distributed. As τ_4 (negative) decreases the data becomes U-shaped (indicating two possible groups). See Figure 1 below for examples.

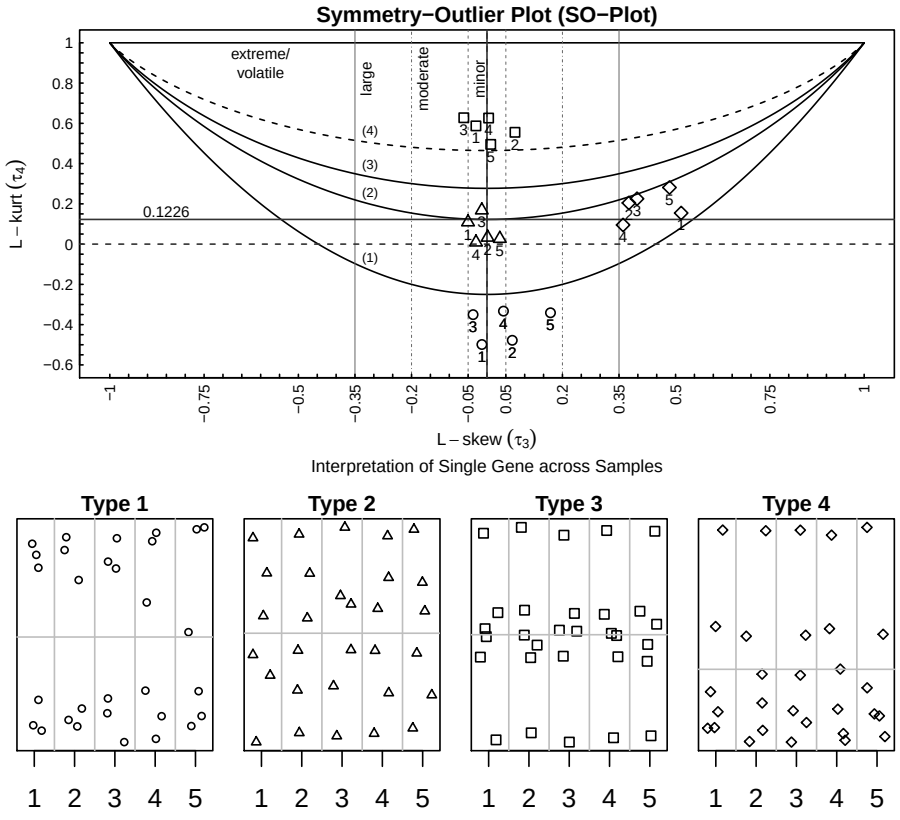


Figure 1: **Interpretation of the SO-plot.** We have shown examples, based on a sample of size 6, of the four main types of sample shape (bottom panels) and where they occur on the SO-plot (top panel). Every single point on the SO-plot corresponds to a summary of the shape of a single gene. The measure of skewness and kurtosis are not independent, as one would expect a highly skewed sample would tend to have a high kurtosis. This relationship is indicated by the parabolas on the SO-plot. Starting from below: (1) we have the theoretical lower bound for L-kurt, in terms of L-skew, $\tau_4 = 0.25(5\tau_3^2 - 1)$. Between the second (2) and third (3) parabolas is a region where the L-kurt measure is considered moderate. Between the third (3) and fourth (4) parabolas indicate a region where the L-kurt measure is considered to be high. Above the fourth curve is considered extreme. One should keep in mind that these descriptive measures are independent of the expression level and variance of the gene. Interpretation of the SO-plot ultimately rests on the context of the data.

1.2 Package overview

The purpose of this package is to compute the shape statistics of each gene in a high-throughput dataset. Using these statistics we can find genes within a dataset whose sample shape is markedly different from the majority of genes in the same dataset. When put together these shape statistics give an overall description of the entire high-throughput dataset. The ability to describe the shape of high-throughput genomics data is useful for two reasons: 1. It enriches the exploratory data analysis step, and 2. It provides a means of checking the distributional assumptions of statisical methods.

There are three main functions in the **shape** package:

1. `fitShape()`
2. `computeDvals()`
3. `plotSO()`

Given a dataset such as a high-throughput expression matrix (or just a vector of measurements) the function `fitShape()` will compute and return the L-CV, L-skew, and L-kurt estimates for each gene.

Given the shape (ie. L-skew and L-kurt) estimate of each gene, the function `computeDvals()` computes a dissimilarity distance (d-values) between each gene's shape estimate and the typical gene's shape estimate. The d-values range from 0 to 1; where 1 is very close and 0 is very far.

The function `plotSO()` shows each gene's shape estimate on a single plot (SO-plot).

1.3 Vignette overview

We will demonstrate the utility of the **shape** package as part of the data exploratory steps in the analysis of high-throughput experiments data. A total of three publicly available datasets will be used in this vignette :

1. The **Pickrell dataset** [5] is part of the International HapMap Project. RNA samples were extracted from the lymphoblastoid cell lines of 69 unrelated Nigerian individuals, 29 males and 40 females. See [6] for details on alignment and counting.
2. The **Hammoud dataset** [7] contains 10 samples of mRNA samples of 8-week old wild type mice (strain: C57BL/6). Of the 10 RNA samples 5 were obtained from spermatids (cells) and the other 5 from spermatocytes (cells). Summarized counts in the form of FPKM can be downloaded at GEO:GSE49622. For the details on alignment and counting please see [7].
3. The **Bottomly dataset** [8] was obtained from the ReCount webpage. See [6] for details on alignment and counting. It contains counts summarizing an RNA-seq experiment that includes 21 samples from inbred mouse strains. Eleven of the samples came from the strain DBA/2J and 10 from the strain C57BL/6J.

2 The Pickrell dataset

The datasets in this package are bundled together in the form of a list. Each component of the list constains the expression measures and its experimental design. We will use the Pickrell dataset to illustrate the main functions in this package.

We begin by loading the **shape** package into an R session and looking at the datasets available in the package.

```
> library(shape)
> data(examplesData) # Load datasets.
> names(examplesData)

[1] "bottomly" "hammoud"  "pickrell"
```

2.1 Filter counts and normalize

First we will filter out genes with low expression by only keeping genes whose counts per million (cpm) is more than 1 in at least 29 samples (where 29 is the minimum of the 29 male samples and 40 female samples).

Let us define a function to filter out the low count genes

```
> filterCounts <- function (pcounts, thresh, minSamples) {
   cpm <- t(t(pcounts) /  colSums(pcounts)) * 1e+06
   keep <- rowSums(cpm > thresh) >= minSamples
   filteredPcounts <- pcounts[keep, ]
   filteredPcounts
 }
```

and apply it to the Pickrell dataset.

```
> counts <- examplesData$pickrell$exprs
> gender <- examplesData$pickrell$cond
> (tab <- table(gender))

gender
female   male
    40     29

> pcounts <- counts + 1 # pseudo-counts
> minSamples <- min(tab)
> dim(pcounts) # Before filteration.

[1] 38415    69

> pcounts <- filterCounts(pcounts, 1, minSamples)
> dim(pcounts) # After filteration.

[1] 17604    69
```

After filteration we normalize for library size and transform the data to $\log_2$ scale. In this vignette we use the DEseq method [9], however any of the library size normalization methods can be used.

```
> ref <- exp(rowMeans(log(pcounts)))
> deseqScal <- apply(pcounts / ref, 2, median)
> pcounts <- t(t(pcounts) / deseqScal)
> y <- log2(pcounts)
```

2.2 `fitShape()`: computation of sample shape

We are now ready to compute the L-skew (τ_3) and L-kurt (τ_4) estimates (ie. shape) of each gene. This is done by calling the function `fitShape()`.

```
> # Compute the L-skew (t3) and L-kurt (t4) of each gene.
> res <- fitShape(y)
> class(res)

[1] "list"

> lapply(res, head, n=3)

$lcv
ENSG00000127720 ENSG00000242018 ENSG00000051596
     0.08185207      0.08437542      0.02878125

$lrats
                           t3         t4
ENSG00000127720  0.0009948179 0.08783691
ENSG00000242018 -0.0398215800 0.14262165
ENSG00000051596  0.0670900379 0.08541339

$lmoms
                     l1        l2            l3         l4
ENSG00000127720 4.115234 0.3368404  0.0003350948 0.02958702
ENSG00000242018 4.275824 0.3607744 -0.0143666072 0.05145424
ENSG00000051596 8.540288 0.2458002  0.0164907419 0.02099463
```

2.3 `computeDvals()`: finding outlier genes

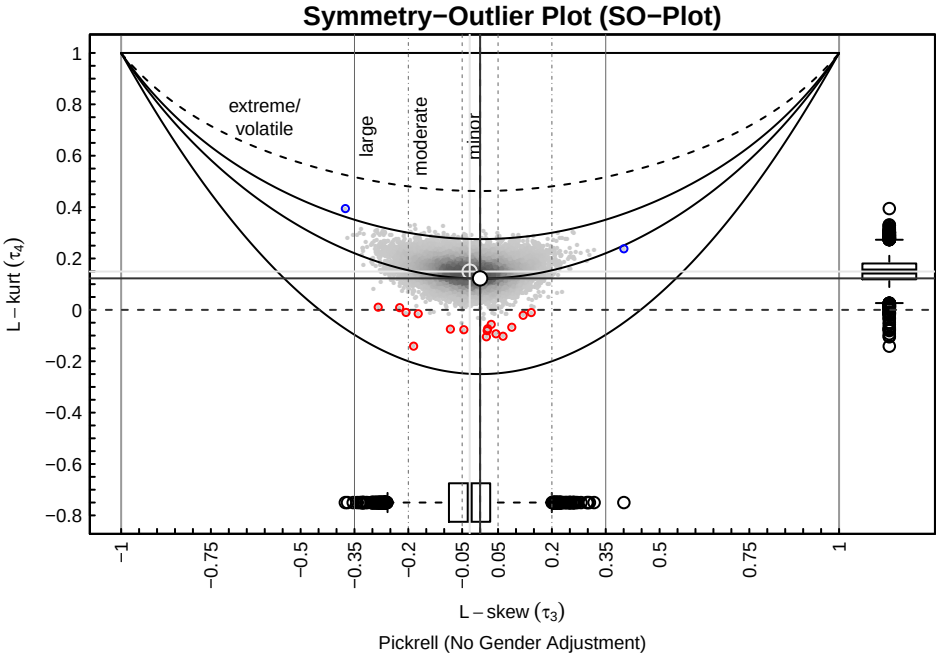
Given the shape of each gene in the dataset the function `computeDvals()` computes the dissimilarity score (d-values) between each gene's shape and the typical gene's shape. The d-values range from 0 to 1; where 1 is very close and 0 is very far. See section 2.5 for details.

```
> # Compute d-values
> t3 <- res$lrats[, "t3"] # Grab L-skew estimates.
> t4 <- res$lrats[, "t4"] # Grab L-kurt estimates.
> dvals <- computeDvals(t3, t4)
```

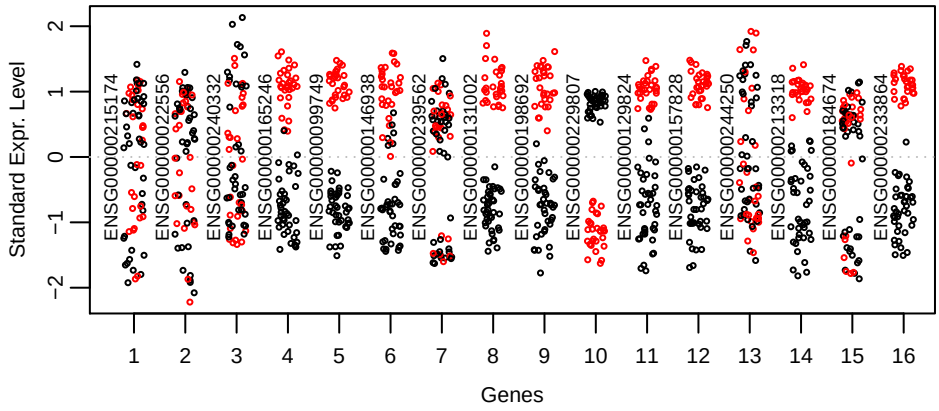
2.4 `plotSO()`: the symmetry outlier plot (SO-plot)

We now construct the SO-plot. On the SO-plot we highlight genes that have very low ($< 10^{-4}$) d-values (aka. outlier genes). This criterion is abitrary and is at the users discretion. For illustrative reasons we separate the outlier genes into two gorups; those with the extreme skew (blueGroup) from the rest (redGroup).

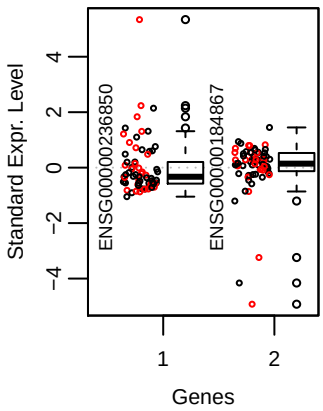
```
> # Symmetry-Outlier plot.
> plotSO(t3, t4, dataName="Pickrell (No Gender Adjustment)", verbose = TRUE)

[1] "Pickrell (No Gender Adjustment) L-skew: (25%, 50%, 75%) = (-0.09, -0.03, 0.03)"

> # Pick volatile / outlier genes.
> sel <- which(dvals < 0.0001) # select 0.01% cutoff
>
> # Seperate outlier genes into 2 groups for illustration purposes
> blueGroup <- sel[abs(t3[sel]) > 0.3]
> redGroup <- sel[abs(t3[sel]) <= 0.3]
> points(t3[blueGroup], t4[blueGroup], cex=0.5, col="blue")
> points(t3[redGroup], t4[redGroup], cex=0.5, col="red")
```



Let us take a closer look at the genes called outliers. Keep in mind that outlier here means that the shape of the gene is different from the majority of gene shapes in the data; independent of the gene's variance and expression level. First we begin with the redGroup (contains 16 genes).



As we can see some of these genes exhibit two groups. The genes are colored by sex. Black is female and red is male. Genes 4, 5, 6, 8, 9, 10, 11, 12, 14, and 16 probably form two groups due to gender differences. Genes 7, 13, and 15 show two groups but probably not due to gender. Perhaps they are due to some other unkown factors (biological or technical) or they are just due to chance. For the blueGroup



gene 1 appears to be skewed sytematically whereas gene 2 appears to be influenced by three extreme levels.

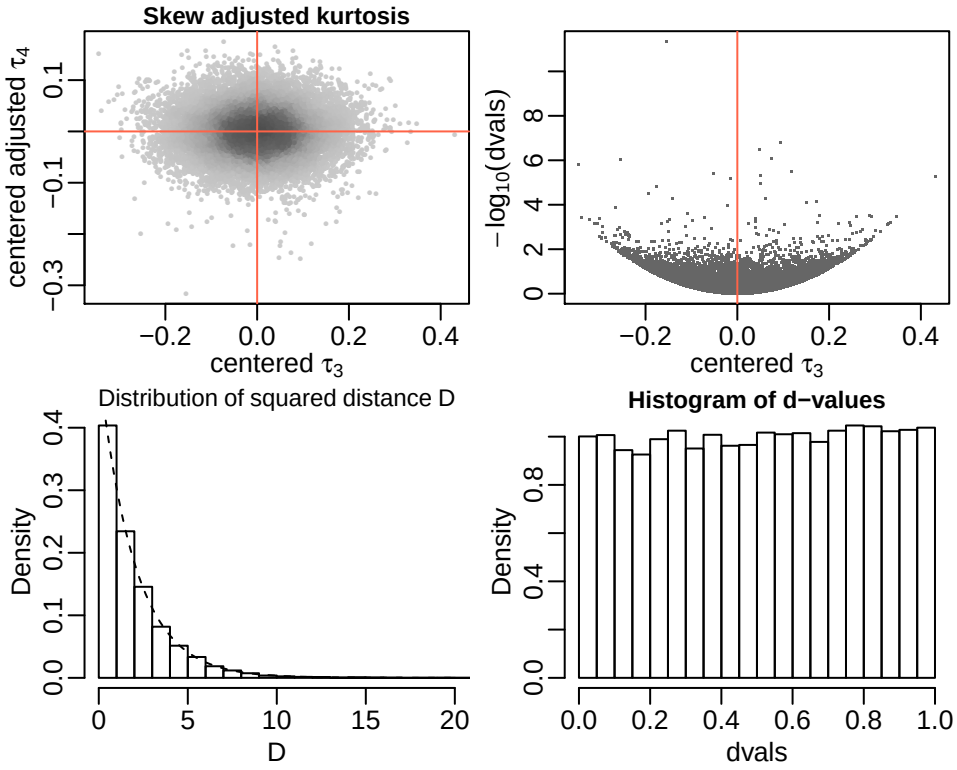
2.5 Steps in outlier computation

We know describe how we assign d-values to the genes. There are three main steps:

1. Estimate the dependence of L-kurt (τ_4) on L-skew (τ_3) with a lowess fuction. And ajdust the L-kurt estimates by subtracting the predicted lowess values.
2. Model the adjusted (τ_4) estimates and (τ_3) estimates with a bivariate Gaussion. And compute the statistical distance of each point from the mean.
3. From the statistical distance obtain the exceedance probalitiy using a chi-square distribution with 2 degrees of freedom.

The backround steps can be shown when calling `computeDvals()` by setting the argument `plot=TRUE`.

```
> head(computeDvals(t3, t4, plot=TRUE))
```




```
ENSG00000127720 ENSG00000242018 ENSG00000051596 ENSG00000236211
      0.47352534      0.99400885      0.21526143      0.08799414
ENSG00000213697 ENSG00000135541
      0.79532942      0.74748301
```

In the top left panel we have shown the adjusted L-kurt and L-skew estimates (both are centered). These points are assumed to be generated from a bivariate Gaussian distribution. See [2] for the basis of this assumption. Statistical distances are computed for each point. The square of these distances follow a chi-square distribution with 2 degrees of freedom. In the bottom left panel we have shown the histogram of the squared distances obtained from the Pickrell dataset. On the top of this histogram we have shown the density of the chi-square 2-df distribution (broken curve). The d-value for a gene is defined as the Pr(chi-squre 2df $>$ gene's squared distance). In the top right panel we show the $-\log_{10}$(d-values) versus the centered L-skew estimates. In the bottom right we show a histogram of the d-values. Also shown are the d-values for the first 6 genes in the Pickrell dataset. We have called the statsitics d-values instead of p-values in order to avoid the confusion that it is a formal statistical test. The d-value is used here as a descriptive measure.

3 The Hammoud dataset

Let us now analyze the Hammoud dataset. It contains 10 samples of mRNA profiles of 8-week old wild type mice (strain: C57BL/6). Of the 10 RNA samples 5 were obtained from spermatids (cells) and the other 5 from spermatocytes (cells). The data has been normalized and are in RPKM units.

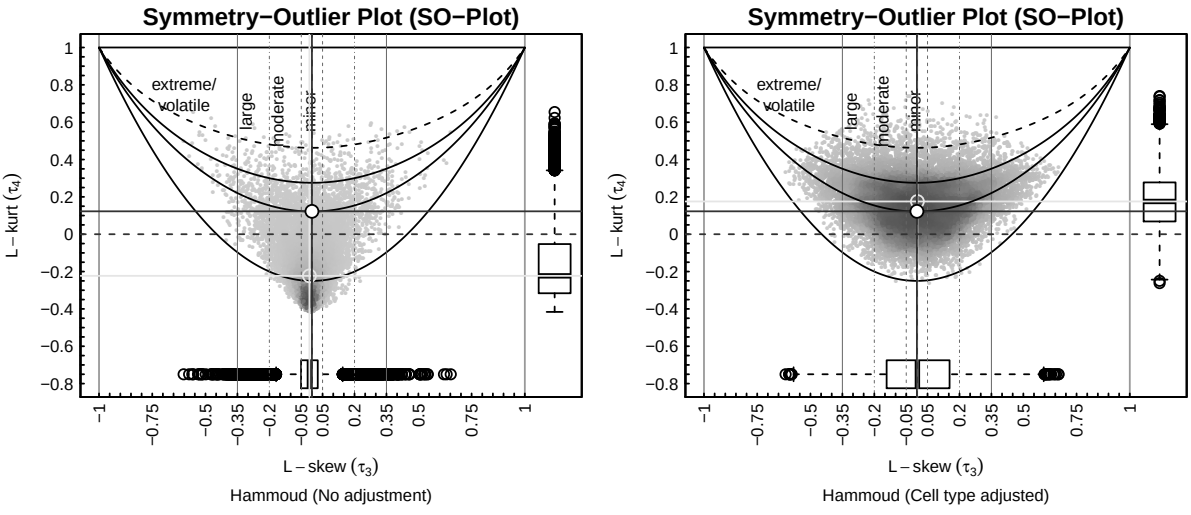
```
> hammoud <- examplesData$hammoud
> rpkm <- hammoud$exprs
> cond <- hammoud$cond
> (tab <- table(cond))

cond
   Spermatids Spermatocytes
            5             5
```

3.1 SO-plot: with and without cell type adjustment

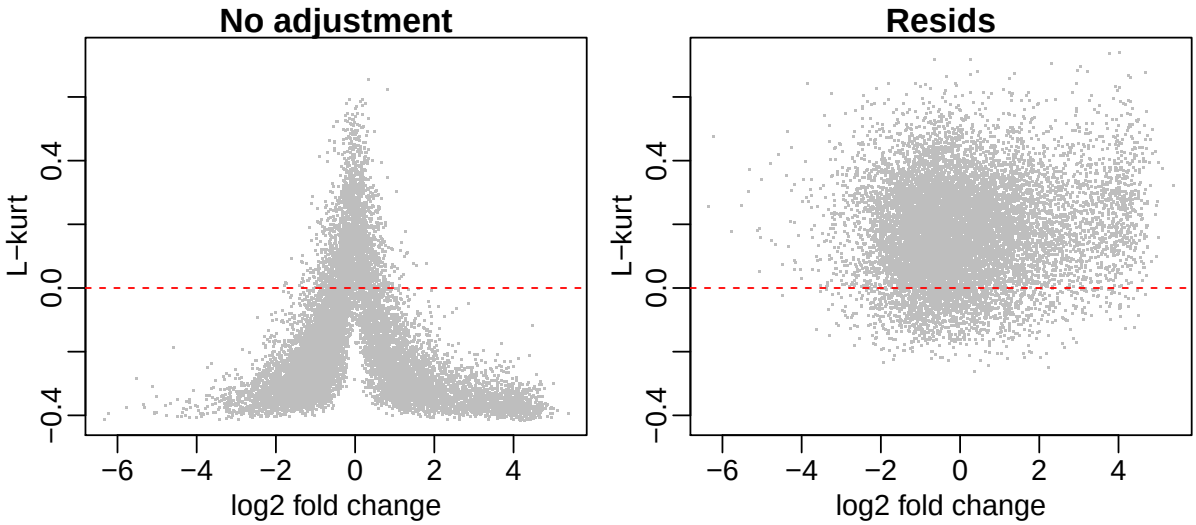
First we filter out genes with low expression by only keeping genes whose RPKM is more than 1 in at least 5 samples (the minimum number of samples per group). Next we transform the RPKM to $\log_2(\text{RPKM}+1)$.

In this analysis we will look at the shape of the genes with and without adjustment for cell type. Let us denote the filtered log transformed RPKM with no cell type adjustment as `log2RPKM`. The adjusted data is then obtained by subtraction the group (spermatid group and spermatocyte group) means from the corresponding samples. We will denote this data as `resids`.

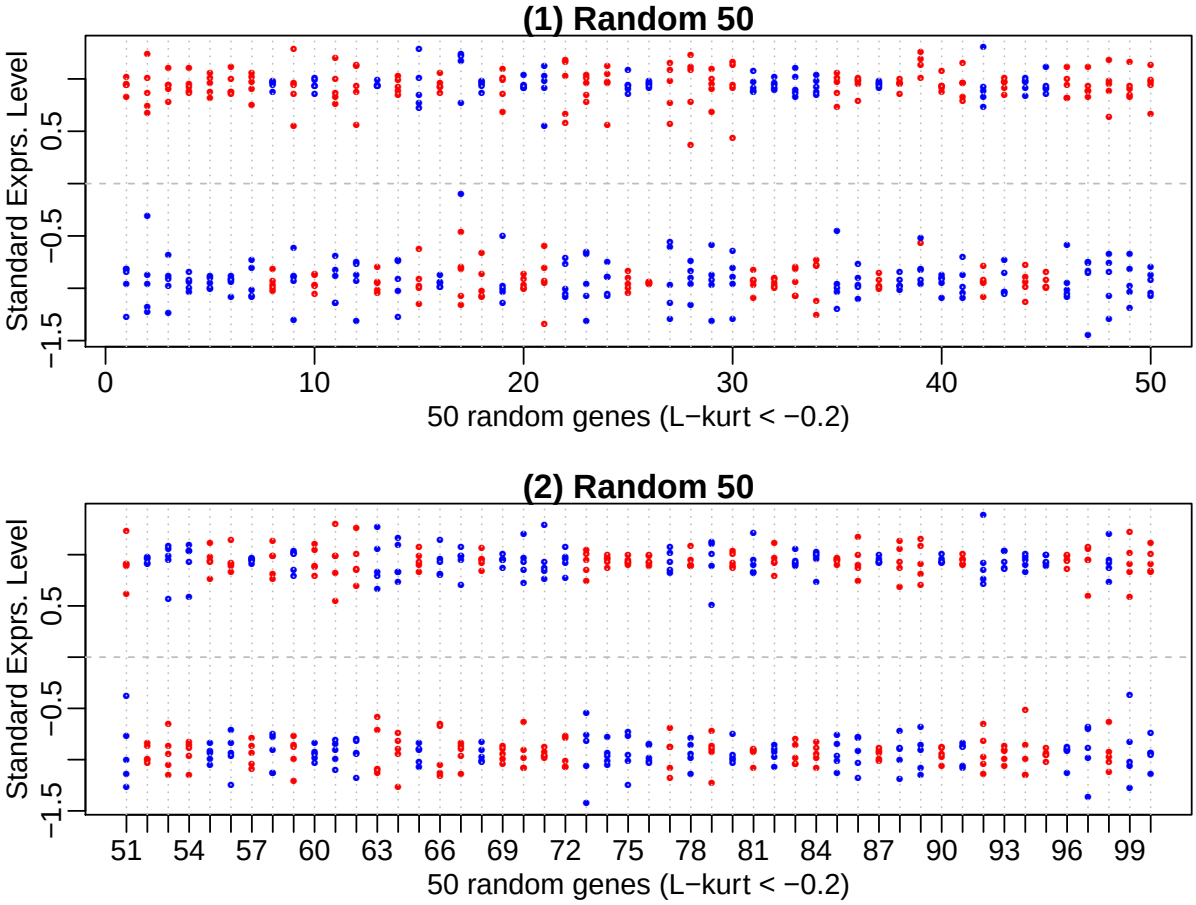
```
> par(mgp=c(1.5, 0.5, 0), mar=c(2.5, 2.5, 1, 0.5), mfrow=c(1, 2))
>
> res1 <- fitShape(log2RPKM) # Without adjustment for cell type.
> res2 <- fitShape(resids) # With adjustment for cell type.
>
> plotSO(res1$lrats[, "t3"], res1$lrats[, "t4"],
        dataName = "Hammoud (No adjustment)")
> plotSO(res2$lrats[, "t3"], res2$lrats[, "t4"],
        dataName="Hammoud (Cell type adjusted)")
```



These two SO-plots are very different. The unadjusted SO-plot has the bulk of its genes below $\tau_4 = 0$ and concentrated at $\tau_3 = 0$ (symmetry). This suggets that a lot of genes are differentially expressed across the cell type. Let us investigate further by exploring the relationship between L-kurt and log fold change.



We can see clearly that there is a strong relationship between L-kurt estimates and fold-change in the No adjustment plot whereas there is none in the Resids plot. Let us randomly select and plot a few (100) of the genes with L-kurt estimates less than -0.2. The spermatids samples are colored blue and the spermatocyte samples are colored red.



4 The Bottomly dataset

The Bottomly dataset contains counts summarizing an RNA-seq experiment that includes 21 samples from inbred mouse strains. Eleven of the samples came from the strain DBA/2J and 10 from the strain C57BL/6J.

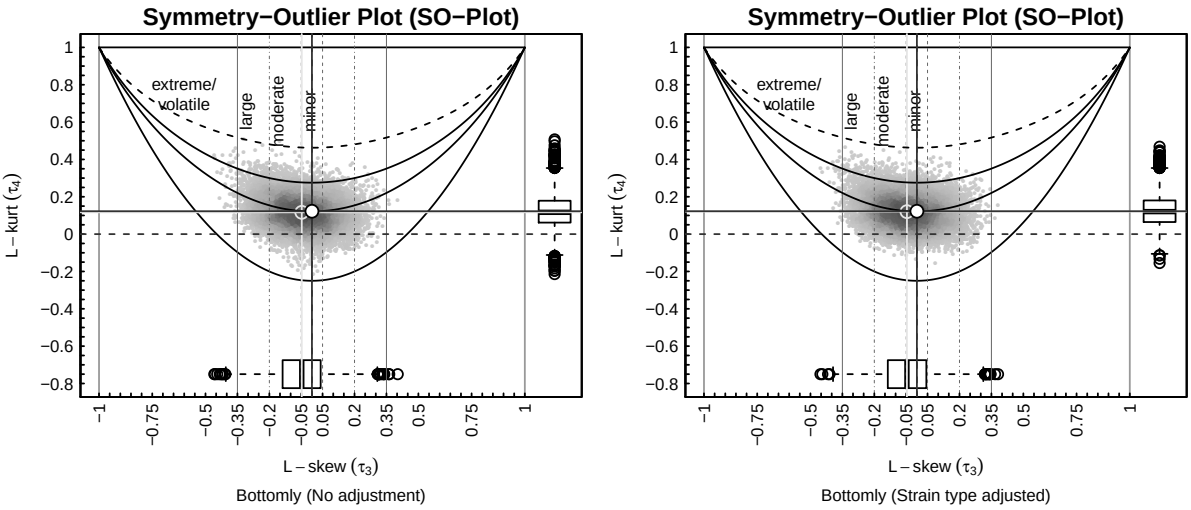
```
> bottomly <- examplesData$bottomly
> counts <- bottomly$exprs
> cond <- bottomly$cond
> (tab <- table(cond))

cond
C57BL/6J   DBA/2J
      10       11
```

4.1 SO-plot: with and without strain type adjustment

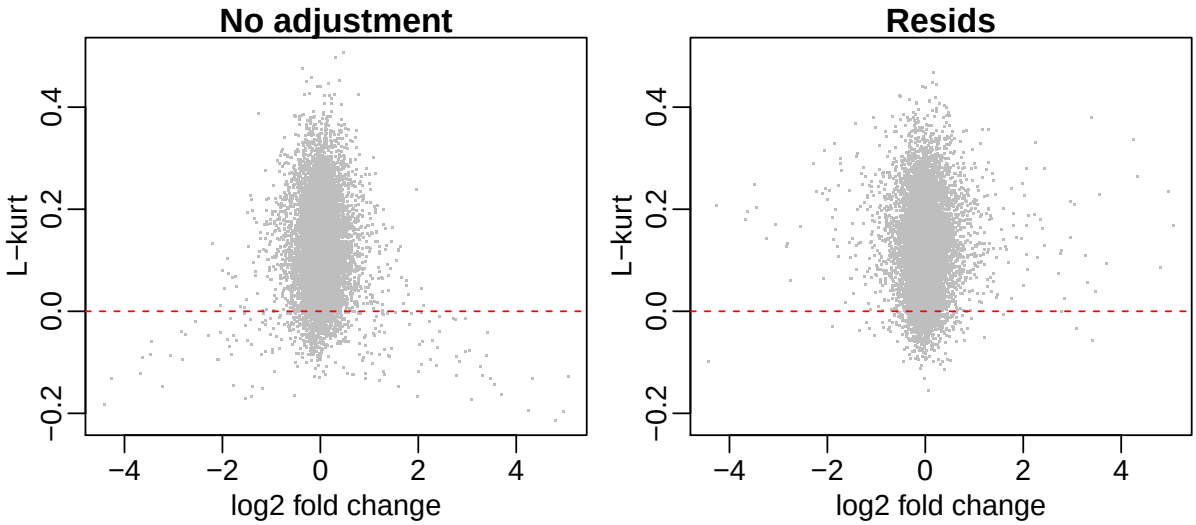
First we filter out genes with low expression by only keeping genes whose cpm is more than 1 in at least 10 samples (the minimum number of samples per condition). Next we normalize the counts + 1 (pseudo-counts) using DESeq's method and transform the normalized pseudo-counts to $\log_2$(normalized pseudo-counts).

In this analysis we will look at the shape of the genes with and without adjustment for strain. Let us denote the filtered log transformed counts with no strain adjustment as `log2pcounts`. The adjusted data is then obtained by subtracting the group (DBA/2J group and C57BL/6J group) means from the corresponding samples. We will denote this data as `resids`.

```
> par(mgp=c(1.5, 0.5, 0), mar=c(2.5, 2.5, 1, 0.5), mfrow=c(1, 2))
>
> res1 <- fitShape(log2pcounts) # Without adjustment for strain.
> res2 <- fitShape(resids) # With adjustment for strain.
>
> plotSO(res1$lrats[, "t3"], res1$lrats[, "t4"],
        dataName = "Bottomly (No adjustment)")
> plotSO(res2$lrats[, "t3"], res2$lrats[, "t4"],
        dataName="Bottomly (Strain type adjusted)")
```



These two SO-plots are very similar. This suggests that most genes are not differentially expressed. Let us investige further by exploring the relationship between L-kurt and log fold change.



Only a few genes change L-kurt estimates after we a adjust for strain.

5 Summary and discussion

We have built on the sound statistical properties of the L-moments ratio estimators to provide a framework for exploring the distributional shapes of genes and the detection of genes (volatile/outlier genes) with shapes that are markedly different from the majority in a given high-throughput transcriptome dataset (SO-plot). The SO-plot (symmetry-outlier) is informative for samples sizes as little as $n \geq 6$. This makes the SO-plot a very powerful tool for exploratory purposes.

Although we analyzed RNA-seq data other types of high-throughput data can benefit from this kind of analysis. In the future examples of analyzing microarray data and methylation data will be included in this vignette.

6 Session Information

```
> sessionInfo()

R version 3.1.2 (2014-10-31)
Platform: x86_64-apple-darwin10.8.0 (64-bit)

locale:
[1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods
[7] base

other attached packages:
[1] shape_1.0.0 knitr_1.8

loaded via a namespace (and not attached):
[1] KernSmooth_2.23-13 evaluate_0.5.5     formatR_1.0
[4] highr_0.4          stringr_0.6.2      tools_3.1.2
```

[1] W. Kirby. Algebraic boundedness of sample statistics. *Water Resources Research*, 10:220–222, 1974.

[2] J. Hosking. L-moments: analysis and estimation of distributions using linear combinations of order statistics. *Journal of the Royal Statistical Society. Series B (Methodological)*, 52:105–124, 1990.

[3] R. Vogel and N. Fennessey. L-moment diagrams should replace product moment diagrams. *Water Resources Research*, 29:1745–1752, 1993.

[4] P. Delicado and M. Goria. A small sample comparison of maximum likelihood, moments and L-moments methods for the asymmetric exponential power distribution. *Computational Statistics & Data Analysis*, 52:1661–1673, 2008.

[5] K. Pickrell, J. Marioni, A. Pai, J. Degner, B. Engelhardt, E. Nkadori, J. Veyrieras, M. Stephens, Y. Gilad, and J. Pritchard. Understanding mechanisms underlying human gene expression variation with RNA-sequencing. *Nature*, 464:768–772, 2010.

[6] A. Frazee, B. Langmead, and J. Leek. Recount: a multi-experiment resource of analysis-ready RNA-seq gene count datasets. *BMC bioinformatics*, 12:449, 2011.

[7] S. Hammoud, D. Low, C. Yi, D. Carrell, E. Guccione, and B. Cairns. Chromatin and transcription transitions of mammalian adult germline stem cells and spermatogenesis. *Cell stem cell*, 2014.

[8] D. Bottomly, N. Walter, J. Hunter, P. Darakjian, S. Kawane, K. Buck, R. Searles, M. Mooney, S. McWeeney, and R. Hitzemann. Evaluating gene expression in C57BL/6J and DBA/2J mouse striatum using RNA-seq and microarrays. *PloS one*, 6:e17820, 2011.

[9] S. Anders and W. Huber. Differential expression analysis for sequence count data. *Genome biology*, 11:R106, 2010.