LMGene User's Guide

Geun-Cheol Lee and David M. Rocke

May 18, 2006

Contents

1 Introduction **1**

2 Data preparation **1**

3 Diagnostic methods **3**

4 G-log transformation **6**

5 Finding differentially expressed genes **7**

1 Introduction

This article introduces a short usage of `LMGene` package. LMGene package has been developed mainly for analysis of microarray data using a linear model and glog data transformation in the R statistical package. This package also provides good visual supports which give overall behavior of microarray data to users.

2 Data preparation

`LMGene` package uses objects of `exprSet` class as its input data , which is the standard data sturcture of the `Biobase` package. Hence, if data which is `exprSet` class is ready, the user can jump to further steps, like diagnostic plotting or g-log transformation. Otherwise, the user needs to generate new `exprSet` class data. For more detail, please see the vignette, 'Textual Description of Biobase' in the `Biobase` package.
Note: exprSet. In this package, an object of `exprSet` class must contain `exprs` and `phenoData` slots with proper data.
`Example. LMGene` package includes a sample array data which is a class of `exprSet`. Let's take a look this sample data.

1. First, load the necessary packages in your R session.

```
> library(LMGene)
```

```
Loading required package: Biobase
Loading required package: multtest
Loading required package: survival
Loading required package: splines
Loading required package: survival
```

```
> library(Biobase)
> library(tools)
```

2. Load the sample `exprSet` class data in the package `LMGene`.

```
> data(Smpd0)
```

3. View the data structure of the sample data and the details of `exprs` and `phenoData` slots in the data.

```
> slotNames(Smpd0)
```

```
[1] "exprs"        "se.exprs"     "description"  "annotation"   "notes"
[6] "reporterInfo" "phenoData"
```

```
> dim(Smpd0@exprs)
```

```
[1] 100   8
```

```
> Smpd0@exprs[1:3, ]
```

```
    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
780  355  311  175  224  209  254  316  517
297   95  102   89  119   91  112  192  255
939  123  104  104   85  112  118  192  184
```

```
> Smpd0@phenoData
```

```
         phenoData object with 3 variables and 8 cases
         varLabels
                dye: dye
                slide: slide
                treat: treat
```

```
> slotNames(Smpd0@phenoData)
```

```
[1] "pData"       "varLabels"   "varMetadata"
```

`Data generation.` If you don't have `exprSet` class data, you need to make one. `LMGene` package provides a function that can generate an object of `exprSet` class, assuming that there are array data of `matrix` class and experimental data of `list` class.

1. The package has sample array and experimental data, `Smpd` and `vlist`.

```
> data(Smpd)
> dim(Smpd)

[1] 100   8

> data(vlist)
> vlist

$dye
[1] R G R G R G R G
Levels: G R

$slide
[1] 1 1 2 2 3 3 4 4
Levels: 1 2 3 4

$treat
[1] P S S P P S S P
Levels: P S
```

2. Generate `exprSet` class data using `neweS` function.

```
> Smpd1 <- neweS(Smpd, vlist)
> class(Smpd1)

[1] "exprSet"
attr(,"package")
[1] "Biobase"

> identical(Smpd0, Smpd1)

[1] FALSE
```

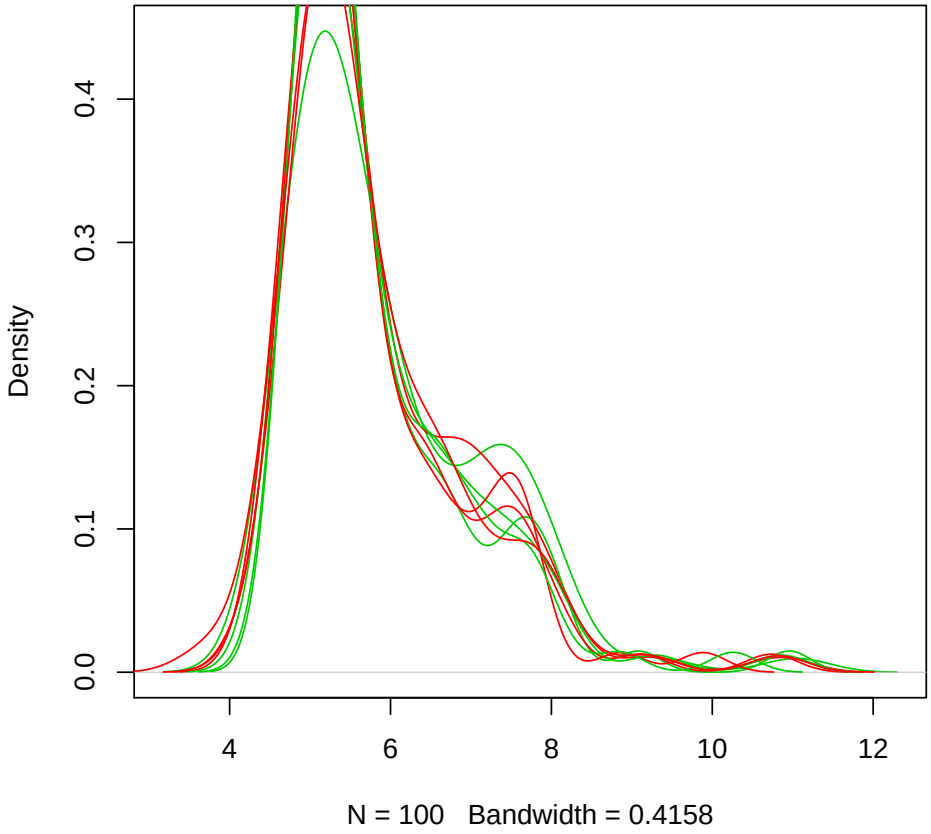
`c.f.` If you have different types of array data, such as `RGList`, `marrayRaw`, and so on, you can convert them into `exprSet` class by using `as` method after installing `convert` package.

3 Diagnostic methods

`LMGene` package provides several plotting functions that show the overall behavior of the array data.

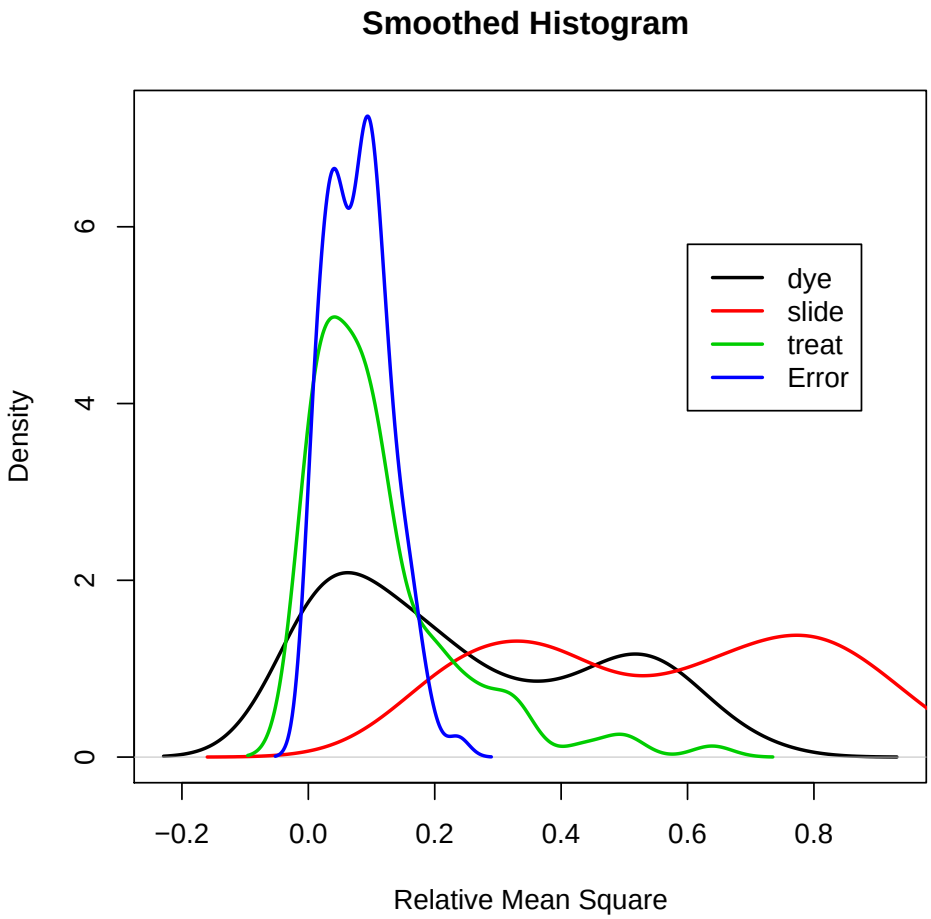
- `Plotting density function:` First, the package can plot the kernel density of each channel in the array data. Before the plotting, the data is log-trasformed and normalized.

```
> rgplot(Smpd0)
```



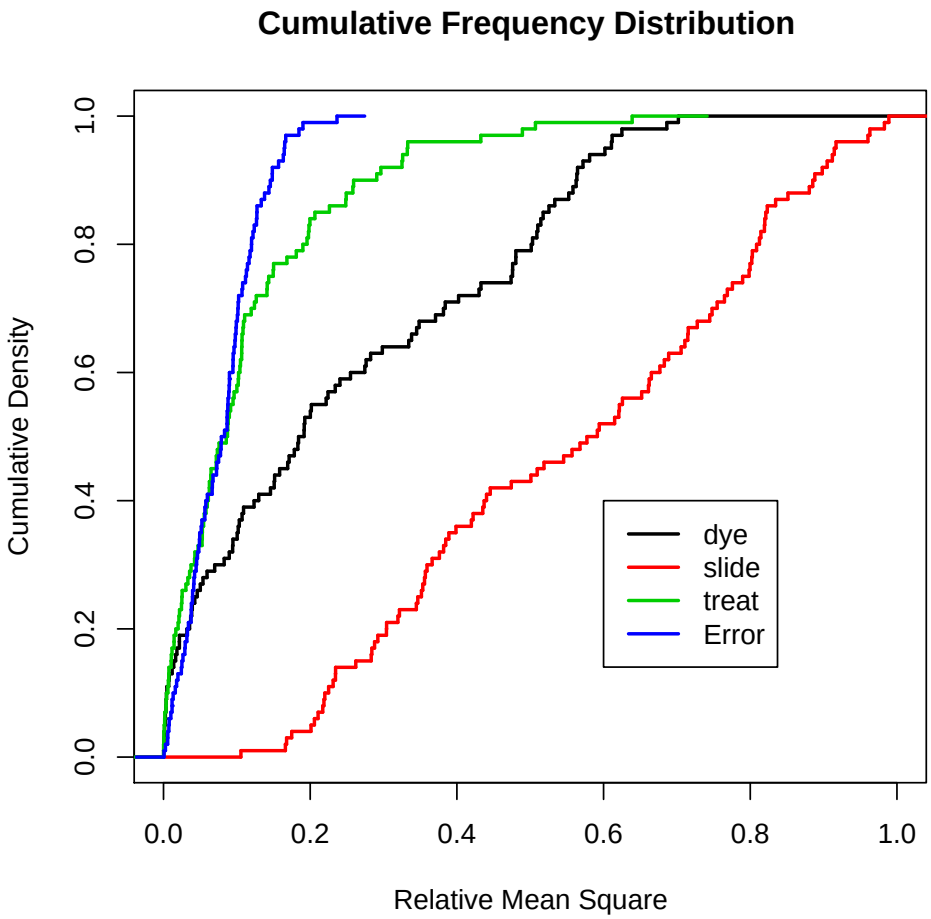
- `Smoothed histogram:` To evaluate the overall sources of variation in the array data produced in this study, we use a visual tool. Before making data for the plotting, we perform the analysis of variance (ANOVA) for each gene considering all factors. Using this analysis, relative mean square values of the factors (including error factor) for all genes can be obtained and the overall results of the data can be presented with a smoothed histogram.

```
> arrplotd(Smpd0)
```



- `Cumulative frequency distribution:` The same results of the relative mean square values from the previous analysis, a different graph, cumulative frequency distribution, can be plotted.

  ```
  > arrplote(Smpd0)
  ```



4 G-log transformation

1. `Estimating parameters for g-log transformation.` Estimate the parameters λ and α of the generalized log transform $\log(y - \alpha + \sqrt{(y-\alpha)^2 + \lambda})$. Using the function tranest0 as follows

```
> tranpar <- tranest(Smpd0)
> tranpar

$lambda
[1] 1

$alpha
[1] 24.54102
```

The optional parameter **ngenes** controls how many genes are used in the estimation. the default is all of them, but it may be impractical in some cases. A typical call using this

parameter would be

```
> tranpar <- tranest(Smpd0, 100)
> tranpar

$lambda
[1] 1

$alpha
[1] 24.54102
```

In this case, 100 genes are chosen at random and used to estimate the transformation parameter. The routine returns a list containing values for lambda and alpha.

2. `G-log transformation.` Using the obtained two parameters, g-log transformed array data (`matrix` class) can be calculated as follows.

```
> Smpd[1:3, ]

     X1  X2  X3  X4  X5  X6  X7  X8
780 355 311 175 224 209 254 316 517
297  95 102  89 119  91 112 192 255
939 123 104 104  85 112 118 192 184

> Glogged <- glog(Smpd - tranpar$alpha, tranpar$lambda)
> Glogged[1:3, ]

          X1       X2       X3       X4       X5       X6       X7       X8
780 6.493632 6.350746 5.706849 5.988762 5.910582 6.128876 6.368049 6.892559
297 4.948228 5.042937 4.859236 5.241341 4.889789 5.164350 5.813895 6.133225
939 5.282813 5.068428 5.068428 4.795181 5.164350 5.230698 5.813895 5.764944
```

Execution time for tranest and lmgene can be substantial, up to several hours, depending on the size of the data set and the choice (for tranest) of the parameter ngenes. It may be worthwhile to try tranest first with a relatively small number of genes or probes, say 100-1000, to check execution time before starting with the full set of genes/probes.

5 Finding differentially expressed genes

1. `Transformation and Normalization.` Before finding differentially expressed genes, the array data needs to be log-like transformed and normalized. Thus, we first generate `exprSet` class data which contains g-log transformed and lowess normalized array matrix data.

```
> Glogged <- glog(Smpd - tranpar$alpha, tranpar$lambda)
> LLdGlogged <- lnorm(Glogged)
> Smpd2 <- neweS(LLdGlogged, vlist)
```

2. `Finding differentially expressed genes` The lmgene routine computes significant genes using the method of Rocke (2003). A typical call would be

   ```
   > siggenes <- LMGene(Smpd2)
   ```

   There is an optional argument, level, which is the test level, .05 by default. A call using this optional parameter would by typically

   ```
   > siggenes <- LMGene(Smpd2, 0.01)
   ```

   The result is a list whose components have the effect names in the model that have any significant genes, and have as values the significant genes for the test of that effect or else the message "No significant genes".

The routine LMGene requires the multtest package.

References

[1] Durbin, B.P., Hardin, J.S., Hawkins, D.M., and Rocke, D.M. (2002) "A variance-stabilizing transformation for gene-expression microarray data," *Bioinformatics*, **18**, S105–S110.

[2] Durbin, B. and Rocke, D. M. (2003a) "Estimation of transformation parameters for microarray data," *Bioinformatics*, **19**, 1360–1367.

[3] Durbin, B. and Rocke, D. M. (2003b) "Exact and approximate variance-stabilizing transformations for two-color microarrays," submitted for publication.

[4] Geller, S.C., Gregg, J.P., Hagerman, P.J., and Rocke, D.M. (2003) "Transformation and normalization of oligonucleotide microarray data," *Bioinformatics*, **19**, 1817–1823.

[5] Rocke, David M. (2004) "Design and Analysis of Experiments with High Throughput Biological Assay Data," *Seminars in Cell and Developmental Biology* , **15**, 708–713.

[6] Rocke, D., and Durbin, B. (2001) "A model for measurement error for gene expression arrays," *Journal of Computational Biology*, **8**, 557–569.

[7] Rocke, D. and Durbin, B. (2003) "Approximate variance-stabilizing transformations for gene-expression microarray data," *Bioinformatics*, **19**, 966–972.