Lapmix: A package for fitting a Laplace mixture model to microarray data

Yann Ruffieux

October 28, 2009

This package fits a Laplace mixture model (Bhowmick et al. 2006) to differential gene expression data. The model is based on the one proposed by Lönnstedt and Speed (2002), which assumes normality of the mean expression in the genes. Here the Gaussian distribution is replaced by a heavier-tailed Laplace distribution. Also implemented is a maximum likelihood estimation method for the hyperparameters of the model. We have five such parameters: γ and α are respectively the shape and scale parameters of the inverse gamma prior distribution for the between-gene error variance, V represents a signal to noise ratio, ω is the prior probability of differential expression, and β is an asymmetry parameter in the Laplace distribution. If we fix $\beta = 0$ we assume a symmetric Laplace distribution in the model. See Bhowmick et al. (2006) for more details on the parameter estimation.

The data are assumed to take the form of normalized base 2 logarithms of the expression ratios, and are stored in a $G \times n$ matrix, G being the number of genes and n the number of replicates per gene. If there are different numbers of replicates between genes, one can insert NaN's where appropriate to indicate 'missing' replicates. The data may alternatively be stored in a list of arrays, or in an object of class `eSet` or `ExpressionSet`.

Below we go through a simple example, using simulated data. First we load the library, of course:

```
> library(lapmix)
```

Next we simulate some symmetric Laplace microarray data:

```
> set.seed(1011)
> G <- 3000
> Y <- NULL
> sigma_sq <- 1/rgamma(G, shape = 2.8, scale = 0.04)
> mu <- rexp(G, rate = 1/(sigma_sq * 1.2)) - rexp(G, rate = 1/(sigma_sq *
+     1.2))
> is.diff <- sample(c(0, 1), replace = TRUE, prob = c(0.9, 0.1),
+     size = G)
> mu <- mu * is.diff
> for (g in 1:G) Y <- rbind(Y, rnorm(4, mu[g], sd = sqrt(sigma_sq[g])))
```

We have generated 3000 genes with 4 replicates each, giving us the matrix `Y`. Now we fit this data to the Laplace model:

```
> res <- lapmix.Fit(Y)
```

The resulting list `res` contains several things of interest. For instance the empirical Bayes estimates of the hyperparameters can be accessed using:

```
> res$estimates

$w
[1] 0.09452177

$V
[1] 1.166841

$beta
[1] 0

$gamma
[1] 2.988225

$alpha
[1] 0.03809164
```

By default the parameters are estimated using a two-step likelihood-based approach. They can also be obtained from maximization of the full marginal likelihood, by adding the argument `two.step=FALSE` in the call to `lapmix.Fit`.

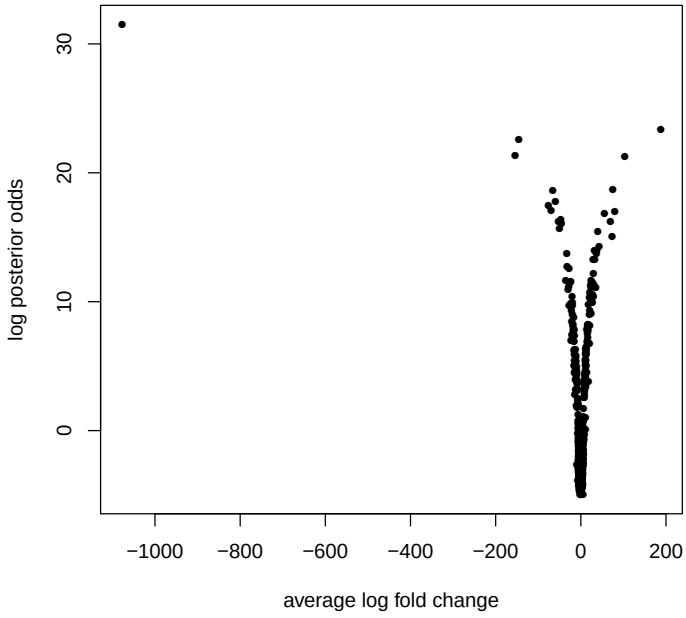
We will be mostly interested in the posterior odds of differential expression (the L- or AL-stat) for each gene. The `laptopTable` ranks the top m genes based on this criterion. This is very similar to the `topTable` from the `limma` package.

```
> m <- 12
> laptopTable(res, m)

   gene           M log.odds
1   647 -1077.18985 31.53057
2   584   187.71534 23.33870
3  1466  -145.86372 22.60668
4   715  -154.29803 21.30894
5  2510   103.12168 21.27225
6  2106    75.53487 18.68461
7  2182   -65.55155 18.66217
8  2089   -59.52450 17.75526
9  1845   -75.57873 17.48861
10 1988   -69.35740 17.07079
11  213    80.11059 16.98297
12 2313    55.56550 16.83399
```

We can also produce a 'volcano plot' based on the L-stat.

```
> lap.volcanoplot(res)
```



Above we have assumed a symmetric Laplace distribution for mean of differential expression. We can also fit an asymmetric Laplace model:

```
> res2 <- lapmix.Fit(Y, asym = TRUE)
> res2$estimates

$w
[1] 0.09452301

$V
[1] 1.166813

$beta
[1] 0.001647244

$gamma
[1] 2.988225
```

```
$alpha
[1] 0.03809164
```

and the result can be treated as in the symmetric case.

By default the `lapmix.Fit` routine uses a 'fast' hyperparameter estimation method: a very small proportion of the integrals in the marginal likelihoods cannot be computed via the t-distribution and are thus neglected (see Bhomwick et al. 2006, p. 632). A 'slow' method can be invoked by inserting the argument `fast=FALSE` in the call to `lapmix.Fit`. This will compute the problem integrals numerically using the `integrate` routine. This is not advised, however: experiments suggest there is very little difference between the fast and slow methods, and the latter may cause convergence problems when used in conjunction with the one-step estimation approach.

References

Bhowmick, D., Davison, A. C., Goldstein, D. R., and Ruffieux, Y. (2006) A Laplace mixture model for identification of differential expression in microarray experiments. *Biostatistics* **7**, 4: 630-641.

Lönnstedt, I. and Speed, T. P. (2002). Replicated microarray data. *Statistica Sinica* **12**: 31-46.