Limma: Linear Models for Microarray Data

Gordon K. Smyth

August 13, 2005

1 Introduction

Limma is a package for differential expression analysis of data arising from microarray experiments. The package is designed to analyze complex experiments involving comparisons between many RNA targets simultaneously while remaining reasonably easy to use for simple experiments. The central idea is to fit a linear model to the expression data for each gene. The expression data can be log-ratios, or sometimes log-intensities, from two color microarrays or log-intensity values from one channel technologies such as Affymetrix™. Empirical Bayes and other shrinkage methods are used to borrow information across genes making the analyses stable even for experiments with small number of arrays [1, 2].

Limma is designed to be used in conjunction with the affy or affyPLM packages for Affymetrix™ data. With two color microarray data, the marray package may be used for pre-processing. Limma itself also provides input and normalization functions which support features especially useful for the linear modeling approach.

2 Data Representations

The starting point for this chapter and many other chapters in this book is that an experiment has been performed using a set of microarrays hybridized with two or more different RNA sources. The arrays have been scanned and image-analyzed to produce output files containing raw intensities, usually one file for each array. The arrays may be *one-channel* with one RNA sample hybridized to each array or they may be *two-channel* or *two-color* with two RNA samples hybridized competitively to each array.

Expression data from experiments using one-channel arrays can be represented as a data matrix with rows corresponding to probes and columns to arrays. The `rma()` function in the affy package produces such a matrix for Affymetrix™ arrays. The output from `rma()` is an *exprSet* object with the matrix of log-intensities in the `exprs` slot.

Experiments using two-color arrays produce two data matrices, one each for the green and red channels. The green and red channel intensities are usually kept separate until normalization, after which they are summarized by a matrix of log-ratios (M-values) and a matrix of log-averages (A-values).

Two-color experiments can be divided into those for which one channel of every array is a common reference sample and those which make direct comparisons between the RNA samples of interest without the intermediary of a common reference. Common reference experiments can be treated similarly to one-channel experiments with the matrix of log-ratios taking the place of the matrix of log-intensities. Direct two-color designs require some special techniques. Many features of limma are motivated by the desire to obtain full information from direct designs and to treat all types of experiment in a unified way.

When discussing linear models, we will assume that a normalized data object called `MA` or `eset` is available. The object `eset` is assumed to be of class *exprSet* containing normalized probe-set log-intensities from an Affymetrix™ experiment while `MA` is assumed to contain normalized M and A-values from an experiment using two-color arrays. The data object `MA` might be an *marrayNorm* object produced by `maNorm()` in the marray package or an *MAList* object produced by `normalizeWithinArrays()` or `normalizeBetweenArrays()` in the limma package, although *marrayNorm* objects usually need some further processing after normalization before being used for linear modeling.

Apart from the expression data itself, microarray data sets need to include information about the *probes* printed on the arrays and information about the *targets* hybridized to the arrays. The targets are of particular interest when setting up a linear model. In this chapter the target labels and any associated covariates are assumed to be available in a *targets frame* called `targets`, which is just a `data.frame` with rows corresponding to arrays in the experiment. In an *exprSet* object this data frame is often stored as part of the `phenoData` slot, in which case it can be extracted by `targets <- pData(eset)`. Despite the name, there is no implication that the covariates are phenotypic in nature, in fact they often indicate genotypes such as wild-type or knockout. In an *marrayNorm* object the targets frame is often stored as part of the `maTargets` slot, in which case it can be extracted by `targets <- maInfo(maTargets(MA))`. Limma provides the function `readTargets()` for reading the targets frame directly from a text file, and doing so is often the first step in a microarray data analysis.

3 Linear Models

Limma uses *linear models* to analyze designed microarray experiments [3, 1]. This approach allows very general experiments to be analyzed nearly as easily as a simple replicated experiment. The approach requires two matrices to be specified. The first is the *design matrix* which provides a representation of the different RNA targets which have been hybridized to the arrays. The second is the *contrast matrix* which allows the coefficients defined by the design matrix to be combined into contrasts of interest. Each contrast corresponds to a comparison of interest between the RNA targets. For very simple experiments the contrast matrix may not need to be specified explicitly.

The first step is to fit a linear model using `lmFit()` which fully models the systematic part of the data. Each row of the design matrix corresponds to an array in the experiment and each column corresponds to a coefficient. With one-channel data or common reference data, the number of coefficients will be equal to the number of distinct RNA sources. With direct-design two-color data there will be one fewer coefficient than distinct RNA targets, or the same number if a dye-effect is included. One purpose of this step is to estimate the variability in the data.

In practice one might be interested in more or fewer comparisons between the RNA targets than there are coefficients. The contrast step, which uses the function `contrasts.fit()`, allows the fitted coefficients to be compared in as many ways as there are questions to be answered, regardless of how many or how few these might be.

Mathematically we assume a linear model $\mathrm{E}[\mathbf{y}_j] = \mathbf{X}\boldsymbol{\alpha}_j$ where $\mathbf{y}_j$ contains the expression data for the gene j, $\mathbf{X}$ is the design matrix and $\boldsymbol{\alpha}_j$ is a vector of coefficients. Here $\mathbf{y}_j^T$ is the jth row of the expression matrix and contains either log-ratios or log-intensities. The contrasts of interest are given by $\boldsymbol{\beta}_j = \mathbf{C}^T\boldsymbol{\alpha}_j$ where $\mathbf{C}$ is the contrasts matrix. The `coefficients` component of the fitted model produced by `lmFit()` contains estimated values for the $\boldsymbol{\alpha}_j$. After applying `contrasts.fit()`, the `coefficients` component now contains estimated values for the $\boldsymbol{\beta}_j$.

With one-channel or common reference microarray data, linear modeling is much the same as ordinary ANOVA or multiple regression except that a model is fitted for every gene. With data of this type, design matrices can be created in the same way that one would do when modeling univariate data. With data from two-color direct designs, linear modeling is very flexible and powerful but the formation of the design matrix may be less familiar. The function `modelMatrix()` is provided to simplify the construction of appropriate design matrices for two-color data.

4 Statistics for Differential Expression

Limma provides functions `topTable()` and `decideTests()` which summarize the results of the linear model, perform hypothesis tests and adjust the p-values for multiple testing. Results include (log) fold changes, standard errors, t-statistics and p-values. The basic statistic used for significance analysis is the *moderated t-statistic*, which is computed for each probe and for each contrast. This has the same interpretation as an ordinary t-statistic except that the standard errors have been moderated across genes, i.e., shrunk towards a common value, using a simple Bayesian model. This has the effect of borrowing information from the ensemble of genes to aid with inference about each individual gene [1]. Moderated t-statistics lead to p-values in the same way that ordinary t-statistics do except that the degrees of freedom are increased, reflecting the greater reliability associated with the smoothed standard errors.

A number of summary statistics are presented by `topTable()` for the top genes and the selected contrast. The M-value (M) is the value of the contrast. Usually this represents a $\log_2$-fold change between two or more experimental conditions although sometimes it represents a $\log_2$-expression level. The A-value (A) is the average $\log_2$-expression level for that gene across all the arrays and channels in the experiment. Column t is the moderated t-statistic. Column p-value is the associated p-value after adjustment for multiple testing. The most popular form of adjustment is "fdr" which is Benjamini and Hochberg's method to control the false discovery rate [4]. The meaning of "fdr" adjusted p-values is as follows. If all genes with p-value below a threshold, say 0.05, are selected as differentially expressed, then the expected proportion of false discoveries in the selected group is controled to be less than the theshold value, in this case 5%.

The B-statistic (lods or B) is the log-odds that the gene is differentially expressed [1, Section 5]. Suppose for example that $B = 1.5$. The odds of differential expression is exp(1.5)=4.48, i.e, about four and a half to one. The probability that the gene is differentially expressed is 4.48/(1+4.48)=0.82, i.e., the probability is about 82% that this gene is differentially expressed. A B-statistic of zero corresponds to a 50-50 chance that the gene is differentially expressed. The B-statistic is automatically adjusted for multiple testing by assuming that 1% of the genes, or some other percentage specified by the user in the call to eBayes(), are expected to be differentially expressed. The p-values and B-statistics will normally rank genes in the same order. In fact, if the data contains no missing values or quality weights, then the order will be precisely the same.

As with all model-based methods, the p-values depend on normality and other mathematical assumptions which are never exactly true for microarray data. It has been argued that the p-values are useful for ranking genes even in the presence of large deviations from the assumptions [5, 2]. Benjamini and Hochberg's control of the false discovery rate assumes independence between genes, although Reiner et al [6] have argued that it works for many forms of dependence as well. The B-statistic probabilities depend on the same assumptions but require in addition a prior guess for the proportion of differentially expressed genes. The p-values may be preferred to the B-statistics because they do not require this prior knowledge.

The eBayes() function computes one more useful statistic. The moderated F-statistic (F) combines the t-statistics for all the contrasts into an overall test of significance for that gene. The F-statistic tests whether any of the contrasts are non-zero for that gene, i.e., whether that gene is differentially expressed on any contrast. The denominator degrees of freedom is the same as that of the moderated-t. Its p-value is stored as fit$F.p.value. It is similar to the ordinary F-statistic from analysis of variance except that the denominator mean squares are moderated across genes.

5 Fitted Model Objects

The output from `lmFit()` is an object of class *MArrayLM*. This section gives some mathematical details describing what is contained in such objects, following on from the Section 3. This section can be skipped by readers not interested in such details.

The linear model for gene j has residual variance σ_j^2 with sample value s_j^2 and degrees of freedom f_j. The output from `lmFit()`, `fit` say, holds the s_j in component `fit$sigma` and the f_j in `fit$df.residual`. The covariance matrix of the estimated $\hat{\boldsymbol{\beta}}_j$ is $\sigma_j^2 \mathbf{C}^T (\mathbf{X}^T \mathbf{V}_j \mathbf{X})^{-1} \mathbf{C}$ where $\mathbf{V}_j$ is a weight matrix determined by prior weights, any covariance terms introduced by correlation structure and any iterative weights introduced by robust estimation. The square-roots of the diagonal elements of $\mathbf{C}^T (\mathbf{X}^T \mathbf{V}_j \mathbf{X})^{-1} \mathbf{C}$ are called unscaled standard deviations and are stored in `fit$stdev.unscaled`. The ordinary t-statistic for the kth contrast for gene j is $t_{jk} = \hat{\beta}_{jk}/(u_{jk}s_j)$ where u_{jk} is the unscaled standard deviation. The ordinary t-statistics can be recovered by

```
> tstat.ord <- fit$coef/fit$stdev.unscaled/fit$sigma
```

after fitting a linear model if desired.

The empirical Bayes method assumes an inverse Chisquare prior for the σ_j^2 with mean s_0^2 and degrees of freedom f_0. The posterior values for the residual variances are given by

$$\tilde{s}_j^2 = \frac{f_0 s_0^2 + f_j s_j^2}{f_0 + f_j}$$

where f_j is the residual degrees of freedom for the jth gene. The output from `eBayes()` contains s_0^2 and f_0 as `fit$s2.prior` and `fit$df.prior` and the $\tilde{s}_j^2$ as `fit$s2.post`. The moderated t-statistic is

$$\tilde{t}_{jk} = \frac{\hat{\beta}_{jk}}{u_{jk}\tilde{s}_j}$$

This can be shown to follow a t-distribution on $f_0 + f_j$ degrees of freedom if $\beta_{jk} = 0$ [1]. The extra degrees of freedom f_0 represent the extra information which is borrowed from the ensemble of genes for inference about each individual gene. The output from `eBayes()` contains the $\tilde{t}_{jk}$ as `fit$t` with corresponding p-values in `fit$p-value`.

References

[1] Smyth, G.: Linear models and empirical bayes methods for assessing differential expression in microarray experiments. Statistical Applications in Genetics and Molecular Biology. 3: Article 3, 2004.

[2] Smyth, G. K., Michaud, J., and Scott, H.: The use of within-array replicate spots for assessing differential expression in microarray experiments. Bioinformatics. 21: to appear, 2005.

[3] Yang, Y. H. and Speed, T. P. Design and analysis of comparative microarray experiments. In Speed, T. P. (Ed.): Statistical Analysis of Gene Expression Microarray Data, pp 35–91. Chapman & Hall/CRC Press, 2003.

[4] Benjamini, Y. and Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. R. Statist. Soc. B. 57: 289–300, 1995.

[5] Smyth, G. K., Yang, Y. H., and Speed, T.: Statistical issues in cDNA microarray data analysis. Methods Mol Biol. 224: 111–36, 2003.

[6] Reiner, A., Yekutieli, D., and Benjamini, Y.: Identifying differentially expressed genes using false discovery rate controlling procedures. Bioinformatics. 19: 368–375, 2003.