Robust Analysis of MicroArray
The rama package

Raphael Gottardo*

April 30, 2018

*Department Statistics, University of Washington
http://www.rglab.org
raph@stat.washington.edu

Contents

1 Overview

The `rama` package consists of several functions for robust estimation of two color microarray intensities with replicates.

The robust estimation is achieved by hierarchical Bayesian modeling. The model we use here is a Bayesian linear model (Lindley and Smith 1972), with t-distributed sampling errors to accommodate outliers (Besag and Higdon 1999). We also explicitly model the nonconstant variance by using an exchangeable priors for the gene precisions (Lewin et al. 2003). Our model includes design effects to deal with normalization issues, similarly to the ANOVA model of Kerr, Martin, and Churchill (2000). Gottardo et al. (2003) give a full description of the model.

Realizations are generated from the posterior distribution via Markov chain Monte Carlo (MCMC) algorithms (Gelfand and Smith 1990; Brooks 1998). Where the full conditionals are of simple form, Gibbs updates are used; where the full conditionals were not of standard form, slice sampling is used. We use the "stepping out" procedure for the slice sampling as introduced in Neal (2003).

Convergence of the MCMC can be assessed using the `coda` library available from CRAN. Our experience with the software is that 50,000 iterations are usually enough to estimate quantities of interest such as posterior means, credible intervals, *etc.* Because the number of genes I can be quite large, the computing time can be long. We recommend running the functions provided in the package in batch mode. This can be done via `R CMD BATCH`. An example is presented in the next section.

Help files. As with any R package, detailed information on functions, their arguments and value, can be obtained in the help files. For instance, to view the help file for the function `fit.model` in a browser, use `?fit.model`.

2 Installation

2.1 Windows

You can install the package using the menu *Packages* → *Install Packages from local zip ....* Then browse you local directory to find the package. Make sure you have root privilegies if you want to install the package in the default directory.

2.2 Unix/Linux

Under Unix, go into your favorite shell window and type `R CMD INSTALL rama.tar.gz` (the package has to be in the directory you are executing the command from). By default R will install the package in `/usr/local/R/lib/R ...` and you will need to have root privilegies in order to do so. You can also install the package locally by using the option -l. If you want to install the package in /home/user/Rlib/ use the command `R CMD INSTALL rama.tar.gz -l /home/user/Rlib`. If you install the package in a directory different from the R default directory you will need to specify the full path when you load the package, i.e.

```
> library(rama,lib.loc='/home/user/Rlib')
```

3 Data Import

You can read your data into R, using the `read.table` command. It is hard to give a general guideline as different data will have different format. In order to use the package you will need to create two data sets, one for each sample (control and treatment). The data should be on the raw scale, i.e. not transformed. The two datasets should be arranged such that rows correspond to genes and columns to replicates. If your data are issued from a dye-swap experiment, replicates with same dye (color) should be arranged together. Table 1 give and example of a data set in the right format.

Table 1: Format of the control and treatment datasets used in the rama package. The two samples must be seperated into two datasets. If the dyes have been swaped, replicates with the same color must be grouped together.

sample	1	1	1	1	2	2	2	2
color	R	R	G	G	G	G	R	R
replicate	1	2	3	4	1	2	3	4
	Dataset 1				Dataset 2			
gene 1	...	...	...	...	...	...	...	...
gene 2	...	...	...	...	...	...	...	...
⋮	...	...	...	...	...	...	...	...
gene n	...	...	...	...	...	...	...	...

4 Fitting the model

We demonstrate the functionality of this package using gene expression data from an HIV study of van't Wout et al. (2003). To load the HIV dataset, use `data(hiv)`, and to view a description of the experiments and data, type `?hiv`. We first load the hiv dataset.

```
> library(rama)
> data(hiv)
```

This data set consists of 4 experiments using the same RNA preparation on 4 different slides. The expression levels of 7000 cellular RNA transcripts were assessed in CD4-T-cell lines at time $t = 24$ hour after infection with HIV virus type 1. The first 4 columns correspond to the first treatment state (hiv infected). The second four represent the control state. The experiment is a balanced dye swap experiment. Finally, the last two columns contain the row and column positions of each gene on the array (slide).

Our goal is to obtain an estimate of the log intensities of each gene in each sample. To demonstrate the functions of the `rama` package we will only use a subset of 640 genes.

We model $y^* = \log_2(y + \kappa)$ where y are the raw intensities and κ is a positive additive constant. This transformation was proposed by Tukey (1957) among the power transformations and studied in detail by Box and Cox (1964). The purpose of κ is to avoid taking the logarithm of negative numbers and to reduce the variance at low intensities. The number κ is estimated beforehand and will be treated as fixed during the MCMC procedure. The estimation procedure for κ is demonstrated in the next section.

4.1 Robust estimation

The estimation of the intensities is done using the function `fit.model`. This function output an object of type `mcmc` containing the sampled values from the posterior distribution. The parameters of interests are $\boldsymbol{\gamma_1}$ and $\boldsymbol{\gamma_2}$, that is the gene effects in each sample.

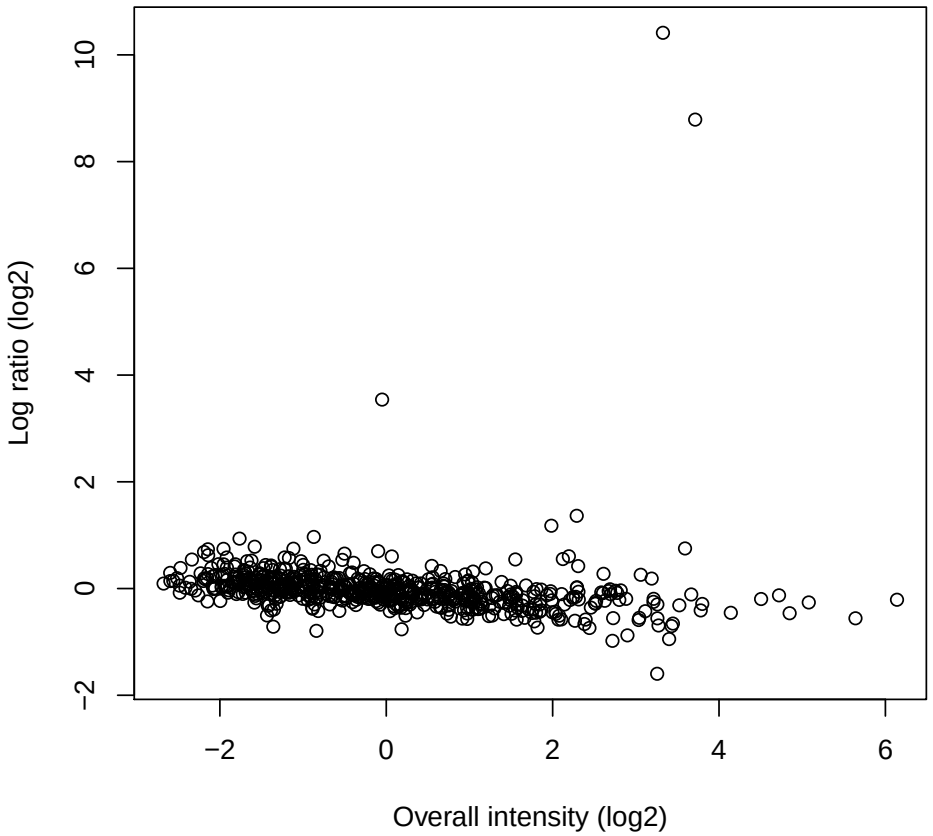
```
> mcmc.hiv<-fit.model(hiv[1:640,c(1:4)],hiv[1:640,c(5:8)],B=5000,min.iter=4000,batch=1,shift=3
```

Note that the model is fitted on the log shifted intensities: see below for details on the estimation of the shift. If `shift` is set to NULL, the shift is automatically estimated with the function `est.shift`. Once you have fitted the model you may view or plot the sampled parameters from the posterior distribution. We can also obtain point estimates of the paramaters of interest such as the gene effects

```
> gamma1<-mat.mean(mcmc.hiv$gamma1)[,1]
> gamma2<-mat.mean(mcmc.hiv$gamma2)[,1]
```

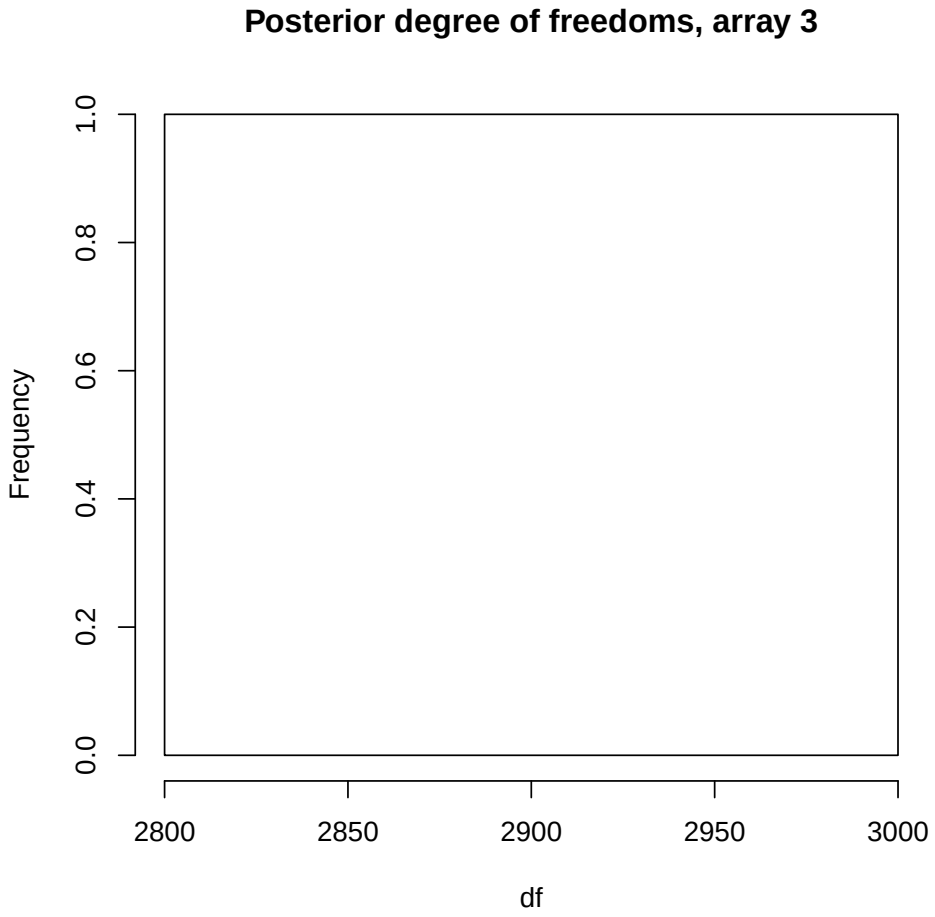
and/or plot the resulting log ratio estimates.

```
> ratio.plot(mcmc.hiv,col=1,pch=1)
```



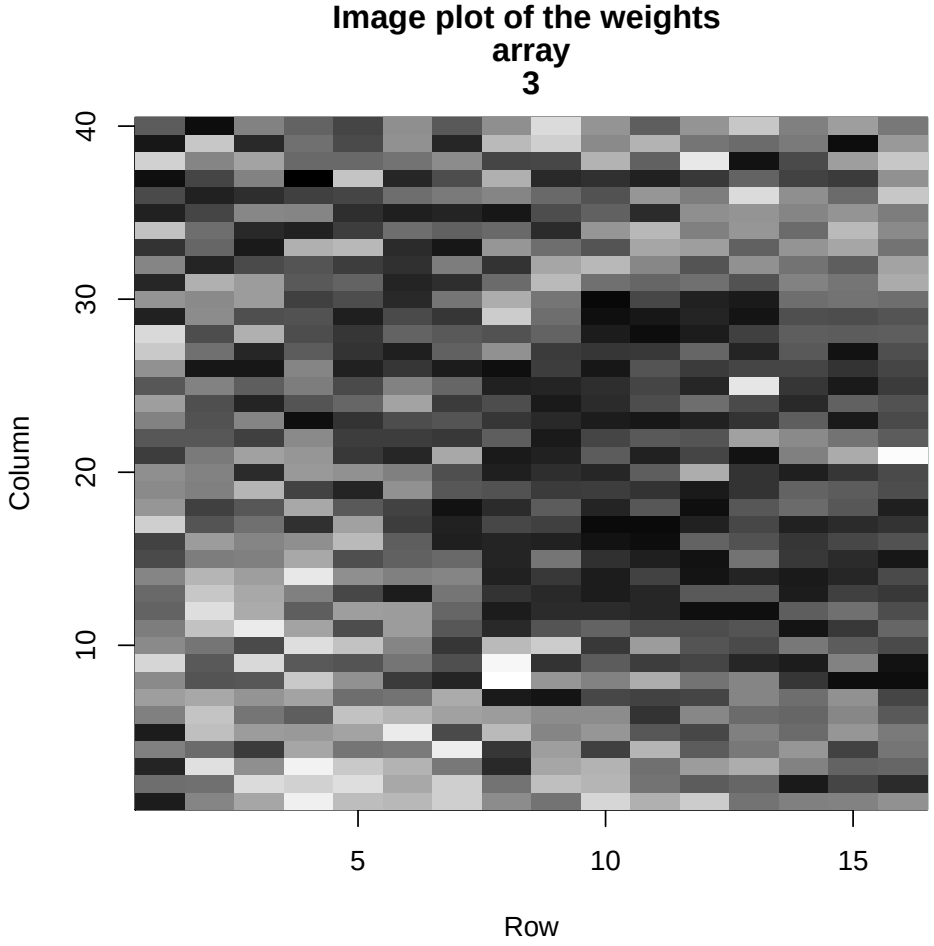
Robustness is achieved by using a t-distribution for the errors with a different degree of freedoms for each replicate array. This is formally equivalent to giving each replicate a different weight in the estimation process for each gene. A low number of degrees of freedom for one array will indicate that the corresponding array contains numerous outliers. One could use the function `hist` to look at the posterior mode of the degrees of freedom for each array.

```
> hist(mcmc.hiv$df[3,],main="Posterior degree of freedoms, array 3",xlab="df",50)
```



Fitting a t-distribution can be seen as a weighted least square problem with a special hierarchical structure for the variances. Similarly to a weighted least squares function, our function computes weights associated with each replicate. The weights can be useful as a tool for diagnosing the quality of the replicate measurements. If one knows the exact positions of the genes on the array, it is possible to look at the spatial variation of the weights. This can be done via the function `weight.plot`.
Here is an example of such a plot:

```
> weight.plot(mcmc.hiv,hiv[1:640,9:10],array=3)
```



Small weights, corresponding to light colors in the image plot, indicate outliers. The image plot of the weights could be useful to detect possible artifacts or array outliers, *etc.*

4.2 Estimating the shift

The log transformation with shifted origin is often used in the context of gene expression data (Kerr, Martin, and Churchill 2000; Cui, Kerr, and Churchill 2002). One question is how to choose the shift, κ. We estimate the shift by fitting the same model as `fit.model` with Gaussian sampling errors, constant variance and a vague uniform prior for κ. Then we use the posterior mean of κ as a point estimate for the general model.

```
> mcmc.shift<-est.shift(hiv[1:640,c(1:4)],hiv[1:640,c(5:8)],B=2000,min.iter=1000,batch=10,mcmc.
```

Note that the shift can also be estimated automatically with the `fit.model` function (See above).

4.3 Computing time and batch mode

The computing time can be quite long depending on the number of genes and replicates. It takes about 5 hours to run the MCMC on the full HIV data set using a Linux machine with AMD Athlon

at 2GHz. However we feel that this additional computational cost is worth the effort and can lead to some very good results. The code can be run in batch mode without any user intervention; this optimizes computing resources and allows the user to perform other tasks at the same time. R code can be run in batch mode by typing `R CMD BATCH myfile.R` at the shell prompt. The file `myfile.R` contains the command to execute. An example file could be the following,

```
> library(rama)
> data(hiv)
> mcmc.hiv<-fit.model(hiv[,c(1:4)],hiv[,c(5:8)],B=50000,min.iter=5000,batch=90,shift=30,mcmc.ob
> save(mcmc.hiv,file='mcmc.hiv.R')
```

Then, when the execution is done one can load the resulting output into R using the `load` command. For further details about the `BATCH` command type `?BATCH` in R.

5 Acknowledgment

This research was supported by NIH Grant 8 R01 EB002137-02.

References

Besag, J. and D. M. Higdon (1999). Bayesian analysis of agricultural field experiments (with discussion). *Journal of the Royal Statistical Society, Series B 61*, 691–746.

Box, G. and D. Cox (1964). An analysis of transformations. *Journal of the Royal Statistical Society 26*, 211–252.

Brooks, S. (1998). Markov chain monte carlo and its application. *The Statistician 47*, 69–100.

Cui, X., M. K. Kerr, and G. A. Churchill (2002). Data transformations for cDNA microarray data. Technical report, The Jackson Laboratory.

Gelfand, A. E. and A. F. M. Smith (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association 85*, 398–409.

Gottardo, R., A. Raftery, K. Yeung, and R. Bumgarner (2003). Robust estimation of microarray intensities with replicates. Technical Report TR-438, Statistics Department, University of Washington.

Kerr, M., M. Martin, and G. Churchill (2000). Analysis of variance for gene expression microarray data. *Journal of Computational Biology 7*, 819–837.

Lewin, A., C. Marshall, and S. Richardson (2003). Bayesian hierarchical modelling of gene expression data. Technical report, Imperial College, London.

Lindley, D. V. and A. F. M. Smith (1972). Bayes estimates for the linear model. *Journal of the Royal Statistical Society 34*, 1–41.

Neal, R. (2003). Slice sampling. *The annals of statistics 31*(3).

Tukey, J. W. (1957). On the comparative anatomy of transformations. *Annals of Mathematical Statistics 28*, 602–632.

van't Wout, A. B., G. K. Lehrma, S. A. Mikheeva, G. C. O'Keeffe, M. G. Katze, R. E. Bumgarner, G. K. Geiss, and J. I. Mullins (2003). Cellular gene expression upon human immunodeficiency virus type 1 infection of $CD4^+ - T -$ Cell lines. *Journal of Virology* *77*(2), 1392–1402.