Using farms - Factor Analysis for Robust Microarray Summarization

Djork-Arné Clevert

April 25, 2007

Contents

1 Introduction

The *farms* package provides a new summarization algorithm called FARMS - Factor Analysis for Robust Microarray Summarization. The summarization method is based on a factor analysis model for which a Bayesian Maximum a Posteriori method optimizes the model parameters under the assumption of Gaussian measurement noise Hochreiter et al. (2006). Thereafter, the RNA concentration is estimated from the model. *farms* does not use background correction and uses either quantile normalization Bolstad et al. (2003) or cyclic loess Yang et al. (2002); Dudoit et al. (2002). *farms* uses quantile normalization as default normalization procedure because it is computational efficient. It does not apply PM corrections and uses PMs only. We set **weigth = 8**, **mu = 0**, and **scale = 2.0** for quantile normalization and **scale = 1.5** for cyclic loess as default. We further set the default vaues for the maximal EM-Steps to **cyc = 100** and the termination criteriea factor analysis to **tol = 0.00001** if the λ-update vector has length smaller than . For the sake of convenience *farms* package provides three wrapper function for *affy*- `expresso`:

- `q.farms` is a wrapper function to `expresso` and uses no background correction and quantile normalization as default normalization procedure.
- `l.farms` performs like `q.farms`, but uses loess normalization as default normalization procedure.
- The function `exp.farms` is a transparent wrapper to `expresso` and permits further preprocessing options.

Note: If you use this package please cite Hochreiter et al. (2006). This package is only free for non-commercial users. Non-academic users **MUST** have a valid license.

2 Getting Started

As usual, it is necessary to load the package.

```
> library(farms)
```

In the following, we use the `affybatch.example` data set as it is provided by the *affy* package to illustrate how to compute expression measures with *farms*.

```
> data(affybatch.example)
> eset <- q.farms(affybatch.example)

background correction: none
normalization: quantiles
PM/MM correction : pmonly
expression values: farms
background correcting...done.
normalizing...done.
150 ids to be processed
|                    |
|####################|
```

This will store expression values, in the object `eset`, as an object of class `exprSet` (see the *Biobase* package).

```
> data(affybatch.example)
> eset <- exp.farms(affybatch.example, bgcorrect.method = "rma",
+     pmcorrect.method = "pmonly", normalize.method = "constant")

background correction: rma
normalization: constant
PM/MM correction : pmonly
expression values: farms
background correcting...done.
normalizing...done.
150 ids to be processed
|                    |
|####################|
```

The available preprocessing options can be queried by using `normalize.AffyBatch.methods`, `pmcorrect.methods` or `bgcorrect.methods`.

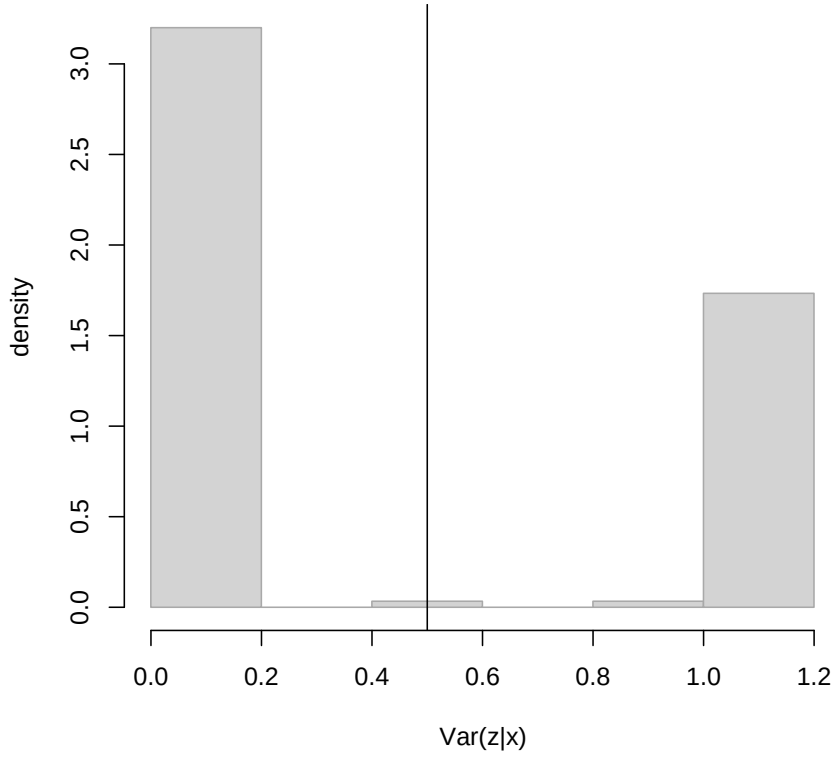
The following example shows how this summarization method can be used as a filtering tool, based on informative / non-informative calls. Please note, that due to the small sample size of the `affybatch.example` data set, additional informative probes were inserted. This was a necessary adjustment to make the example executalbe. (`se.exprs(eset)[100:150,]<-1`)

```
> data(affybatch.example)
> eset <- exp.farms(affybatch.example, bgcorrect.method = "rma",
+     pmcorrect.method = "pmonly", normalize.method = "constant")

background correction: rma
normalization: constant
PM/MM correction : pmonly
expression values: farms
background correcting...done.
normalizing...done.
150 ids to be processed
|                    |
|####################|
```

```
> se.exprs(eset)[100:150, ] <- 1
> IN_genes <- makeINICalls(eset)
> cat(paste(IN_genes, " percentage of the genes are informative",
+     sep = " "))

0.353333333333333  percentage of the genes are informative
```



Enjoy!

References

B. M. Bolstad, R. A. Irizarry, M. Astrand, and T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003.

S. Dudoit, Y. H. Yang, M. J. Callow, and T. P. Speed. Statistical methods for identifying genes with differential expression in replicate cDNA microarray experiments. *Stat. Sin*, 12(1):111–139, 2002.

Sepp Hochreiter, Djork-Arne Clevert, and Klaus Obermayer. A new summarization method for affymetrix probe level data. *Bioinformatics*, page btl033, 2006. doi: 10.1093/bioinformatics/btl033. URL `http://bioinformatics.oxfordjournals.org/cgi/content/abstract/btl033v1`.

Y. H. Yang, S. Dudoit, P. Luu, D. Lin, V. Peng, J. Ngai, and T. Speed. Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res*, 30(4):e15, 2002.