Using FARMS for summarization
Using I/NI-calls for gene filtering

Djork-Arné Clevert

Institute of Bioinformatics, Johannes Kepler University Linz
Altenberger Str. 69, 4040 Linz, Austria
okko@clevert.de

Version 1.12.0, April 4, 2013

Contents

1 Introduction **3**

2 FARMS **3**
2.1 Getting Started 4

3 I/NI calls **5**
3.1 Original I/NI call 5
3.2 Laplacian I/NI call 6

1 Introduction

The `farms` package provides a new summarization algorithm called FARMS - Factor Analysis for Robust Microarray Summarization and a novel unsupervised feature selection criterion called I/NI-calls.

2 FARMS

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). Neverthess any other `affy` preprocessing method can be applied as well. `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 the hyperparameters of the prior distribution by default to $\mathbf{weigth} = \mathbf{0.5}$, $\mathbf{mu} = \mathbf{0}$. We further set the default values for the maximal EM-iterations to $\mathbf{cyc} = \mathbf{100}$ and the termination criteriea to $\mathbf{tol} = \mathbf{0.00001}$, which express that the iteration will stop if the change of $\mathbf{var}(\mathbf{z}|\mathbf{x})$ after the update step is smaller than that tolerance value. If probes of a probe set are governed by a common latent variable, then we associate this variable with the mRNA concentration and its variation with the mRNA variation, i.e. with the signal. Intuitively speaking, if probes of a probe set change synchronously across the arrays then this effect is very unlikely produced by noise and one should assume they are driven by a signal. But if no common variable exists, the covariance structure can solely explain by the noise variance and implies that factor loadings are zero. In this case the expression values will be constant. Some post-processing methods e.g. t-tests face problems with constant results, therefore we introduced a boolean parameter called $\mathbf{robust}$, which prevent results with zero variance. This parameter is by default set to TRUE. **Nevertheless, we highly recommend to filter out nonrelevant probe sets by applying I/NI-calls, as described in section 3.** For the sake of convenience `farms` package provides three wrapper function for `affy- expresso`:

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

Note: If you use this package please cite Hochreiter *et al.* (2006) and Talloen *et al.* (2007). This package is only free for non-commercial users. Non-academic users **MUST** have a valid license.

2.1 Getting Started

As usual, it is necessary to load the package.

```
> library(farms)
> library(affydata)
```

```
     Package    LibPath                                    Item
[1,] "affydata" "E:/biocbld/bbs-2.12-bioc/R/library" "Dilution"
     Title
[1,] "AffyBatch instance Dilution"
```

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(Dilution)
> eset <- qFarms(Dilution)
```

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

```
> data(Dilution)
> eset <- expFarms(Dilution , bgcorrect.method = "rma", pmcorrect.method = "pmonly",
     normalize.method = "constant")
```

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

Standard FARMS assumes Gaussian factor distribution that is a Gaussian distribution of the mRNA concentration across the samples. This assumption is suited well suited for most experiments. However, under some condition other, e.g. compounds studies, or very unbalanced data sets, where differentially expressed gene are only expected in few individuals other assumptions seem to be more adequate. Such rare events are hard to detect with the original FARMS as they would be interpreted as noise. An appropriate model would be a sparse distribution of the factor, that is the factor takes for most cases its default value and deviates only in few cases considerably from this value. Therefore we additionally propose a factor analysis model with a Laplacian prior which leads to a sparse factor distribution. But now the likelihood is analytically intractable due to the non-Gaussian form of the prior. To tackle this problem, we implemented an algorithm that applies a variational expectation maximization algorithm which optimizes a lower bound on the likelihood by representing the prior as the maximum of a Gaussian function family. The following example shows how to switch from the Gaussian to the Laplacian prior:

```
> data(Dilution)
> eset <- qFarms(Dilution, laplacian=TRUE)
```

3 I/NI calls

In this section, we show how to apply the I/NI-calls to a data set. Informative/ non-informative (I/NI) calls is an objective feature filtering technique for Affymetrix GeneChips. It uses the multiple probes measuring the same target mRNA as repeated measures to quantify the signal-to-noise ratio of that specific probe set. By incorporating probe level information to assess the noisy nature of probe sets, I/NI calls provide a highly powerful and objective tool for gene filtering. I/NI calls consequently offers a key solution to the main problem in the analysis of high-dimensional microarray data, being multiple testing and overfitting. I/NI calls can be used in combination with summarization techniques like FARMS, but also with any other summarization technique like MAS5 or (GC)RMA.

3.1 Original I/NI call

The following example shows how this summarization method can be used as a filtering tool, based on informative / non-informative calls.

```
> data(Dilution)
> eset<-qFarms(Dilution)

background correction: none
normalization: quantiles
PM/MM correction : pmonly
expression values: farms
background correcting...done.
normalizing...done.
12625 ids to be processed
|                    |
|####################|

> INIs <- INIcalls(eset)  # apply I/NI calls
> summary(INIs)

Summary
Informative probe sets      : 5.35%
Non-Informative probe sets  : 94.65%

> I_data <- getI_Eset(INIs) # affybatch containing only  informative probe sets
> I_data

ExpressionSet (storageMode: lockedEnvironment)
assayData: 675 features, 4 samples
  element names: exprs, se.exprs
protocolData: none
phenoData
```

```
  sampleNames: 20A 20B 10A 10B
  varLabels: liver sn19 scanner
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation: hgu95av2
```

```
>   plot(INIs)  # draws a density plot of I/NI-calls
```

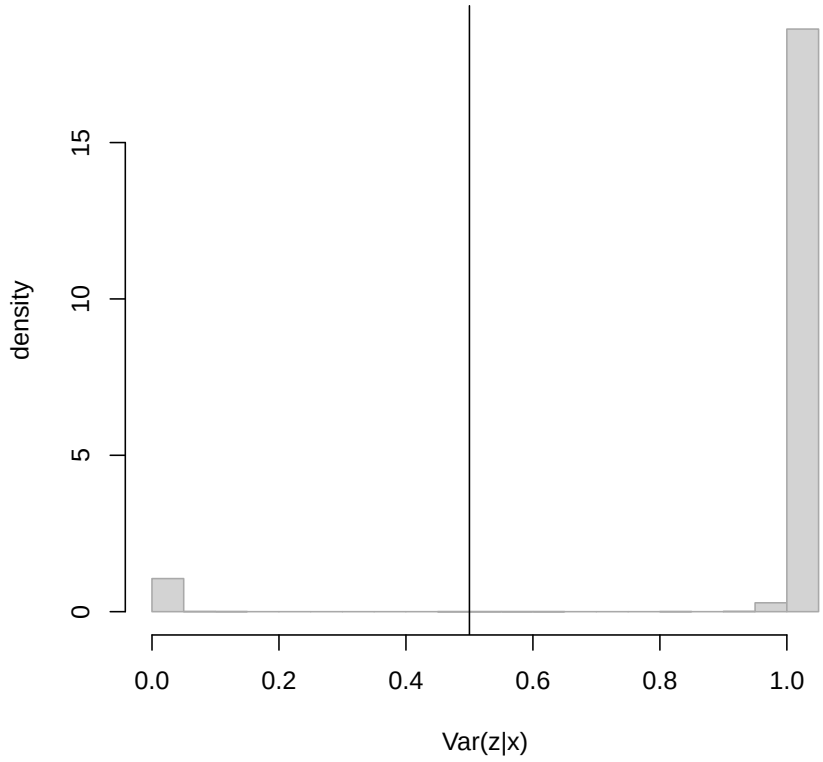

Figure 1: Histogram of $\mathbf{var}(\mathbf{z}|\mathbf{x})$ for the dilution data set, that is provided in affy.

3.2 Laplacian I/NI call

In contrast to the previous section, we will now apply the Laplacian-FARMS to summarize the data.

```
> eset<-qFarms(Dilution, laplacian=TRUE)
```

```
background correction: none
normalization: quantiles
PM/MM correction : pmonly
expression values: farms
background correcting...done.
normalizing...done.
12625 ids to be processed
|                    |
|####################|

> INIs <- INIcalls(eset)  # apply I/NI calls
> summary(INIs)

Summary
Informative probe sets      : 22.28%
Non-Informative probe sets  : 77.72%

> I_data <- getI_Eset(INIs) # affybatch containing only  informative probe sets
> I_data

ExpressionSet (storageMode: lockedEnvironment)
assayData: 2813 features, 4 samples
  element names: exprs, se.exprs
protocolData: none
phenoData
  sampleNames: 20A 20B 10A 10B
  varLabels: liver sn19 scanner
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation: hgu95av2
```

Enjoy!

References

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

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

Hochreiter, S., Clevert, D.-A., and Obermayer, K. (2006). A new summarization method for affymetrix probe level data. *Bioinformatics*, page btl033.

Talloen, W., Clevert, D.-A., Hochreiter, S., Amaratunga, D., Bijnens, L., Kass, S., and Goehlmann, H. W. (2007). I/ni-calls for the exclusion of non-informative genes: a highly effective filtering tool for microarray data. *Bioinformatics*, page btm478.

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

```
>   plot(INIs)  # draws a density plot of I/NI-calls
```

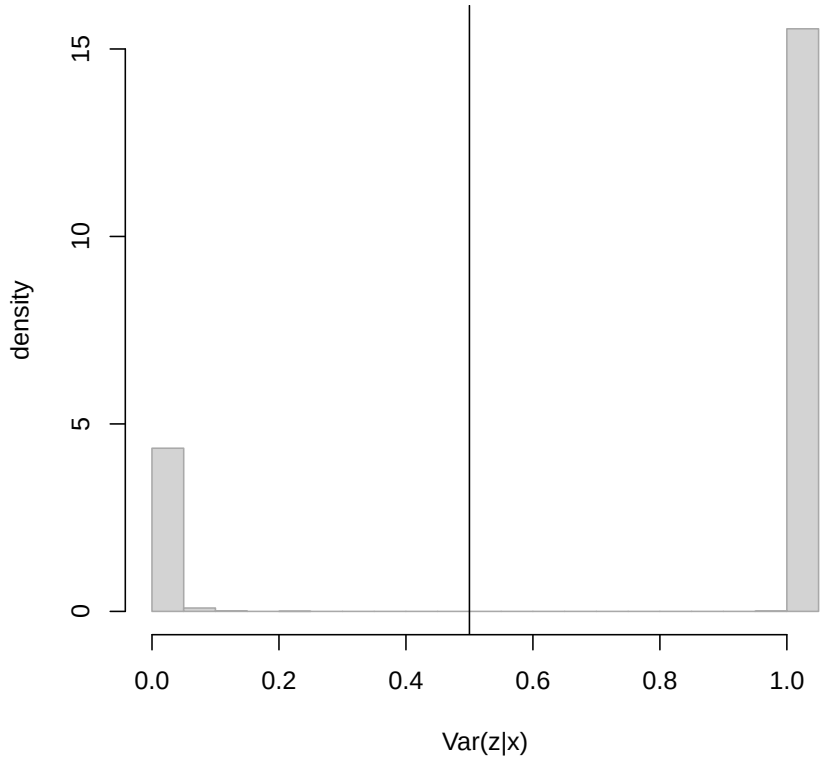

Figure 2: Histogram of $\mathbf{var}(\mathbf{z}|\mathbf{x})$ for the dilution data set, that is provided in affy.