Using farms - Factor Analysis for Robust Microarray Summarization

Djork-Arné Clevert

March 4, 2006

Contents

1 Introduction 1

2 Getting Started 1

1 Introduction

The \texttt{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. \texttt{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). \texttt{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 \texttt{weigth} = \texttt{8}, \texttt{mu} = \texttt{0}, and \texttt{scale} = \texttt{2.0} for quantile normalization and \texttt{scale} = \texttt{1.5} for cyclic loess as default. We further set the default values for the maximal EM-Steps to \texttt{cyc} = \texttt{100} and the termination criteria factor analysis to \texttt{tol} = \texttt{0.00001} if the $\lambda$-update vector has length smaller than 1. For the sake of convenience \texttt{farms} package provides three wrapper function for \texttt{affy-expresso}:

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

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

2 Getting Started

As usual, it is necessary to load the package.

\texttt{> library(farms)}
Welcome to Bioconductor

Vignettes contain introductory material. To view, simply type 'openVignette()' or start with 'help(Biobase)'. For details on reading vignettes, see the openVignette help page.

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.

```r
> 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).

```r
> 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.

Enjoy!

References

