Quick start guide for using the *LVSmiRNA* package in R

Stefano Calza, Suo Chen and Yudi Pawitan

May 3, 2016

Contents

1 Overview

This document provides a brief guide to the *LVSmiRNA* package, which is a package for normalization of microRNA (miRNA) microarray data.

There are four components to this package. These are: i) Reading in data. ii) Identify a subset of miRNAs with the smallest array-to-array variation, called LVS miRNAs. iii) Normalization using the selected reference set. iv) Summarization

The LVS normalization method Calza et al. [2007] builds upon the fact that the data-driven housekeeping miRNAs that are the least variant across samples might be a good reference set for normalization. The total information extracted from probe-level intensity data of all samples is modeled as a function of array and probe effects using robust linear fit (*rlm*) Huber [1964]. The method selects miRNAs according to the array-effect statistic and residual standard deviation (SD) from the model. The modified LVS also incorporates a

more complex *joint* model to identify the LVS. Instead of assuming constant residual variation in *rlm*, the dispersion parameters are modeled as a function of array and probe effects Lee et al. [2006]. The advantage of LVS normalization is that it is robust against violation of standard assumptions in most methods: the majority of features do not vary between samples and the proportion of up and down regulated expression are approximately equal.

2 Getting started

To load the **LVSmiRNA** in your R session, type

```
> library(LVSmiRNA)
>
```

2.1 Example data: Spike-in Agilent chip

We demonstrate the functionality of this R package using miRNA expression data from a spike-in experiment Willenbrock et al. [2009] which is included as part of the package. The input file **Comparison_Array.txt** is a text file containing a header row, names of the samples in one column called 'Sample'.

1. To begin, users will have to save the relevant image processing output files and the file containing samples descriptions (e.g. **Comparison Array.txt** in a directory. The example data can be downloaded from the author website (`http://www.med.unibs.it/~calza/software/examples.tar.gz`).

2. Read the samples description file (e.g. `Comparison_Array.txt`) into R.

   ```
   > dir.files <- system.file("extdata", package="LVSmiRNA")
   > taqman.data <- read.table(file.path(dir.files,"Comparison_Array.txt"),header=TRUE,as.is=TRUE)
   >
   ```

3. Read in the raw intensities data. Modify the `FileName` column in `taqman.data` to match the position of the exemple files. The current values assume that the files are located in the working directory.

   ```
   > here.files <- "some/path/to/files"
   > ## NOT RUN
   ```

```
> MIR <- read.mir(taqman.data,path=here.files)
>
```

4. A binary version of the data is already available in the package.

```
> data("MIR-spike-in")
>
```

5. Identify LVS miRNAs: calculate residual variance and array-to-array variability which is measured by a χ^2 statistic for the raw data fitted by either standard rlm or joint model.

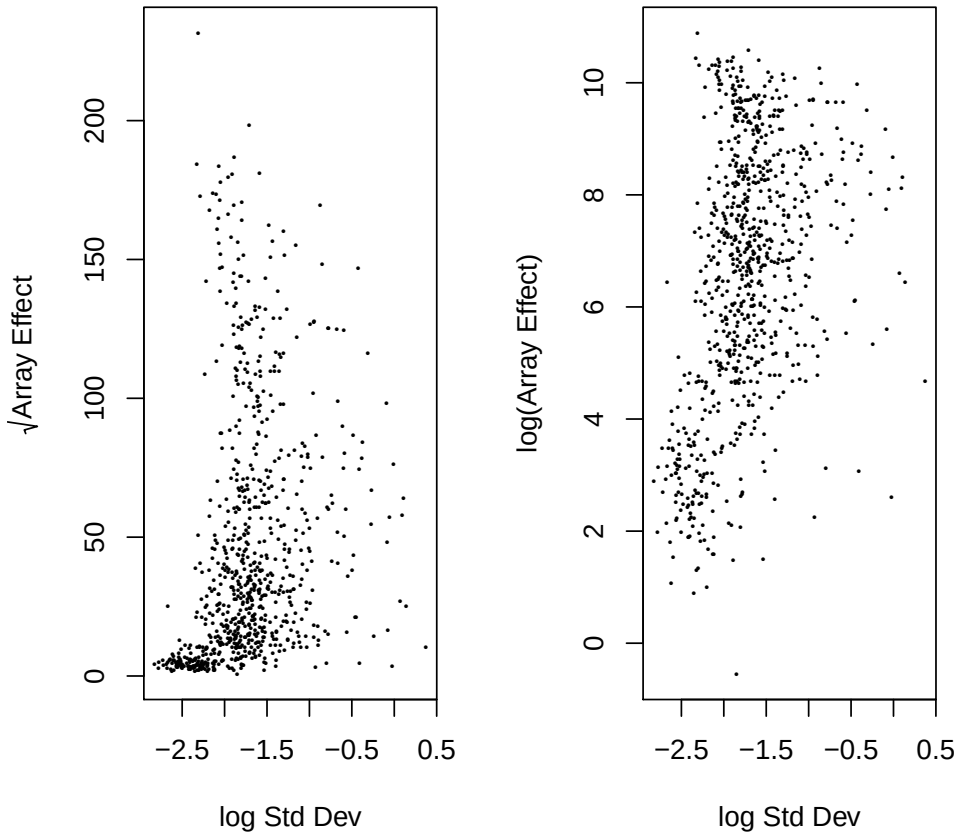
```
> MIR.RA <- estVC(MIR)
>
```

6. Again for semplicity the object can be directly loaded from the package

```
> data("MIR_RA")
>
```

7. Make a scatter plot to visualize the relationship between the square-root or logarithm of array effect versus the logrithm of the residual SD, called the 'RA-plot'.

```
> plot(MIR.RA)
>
```

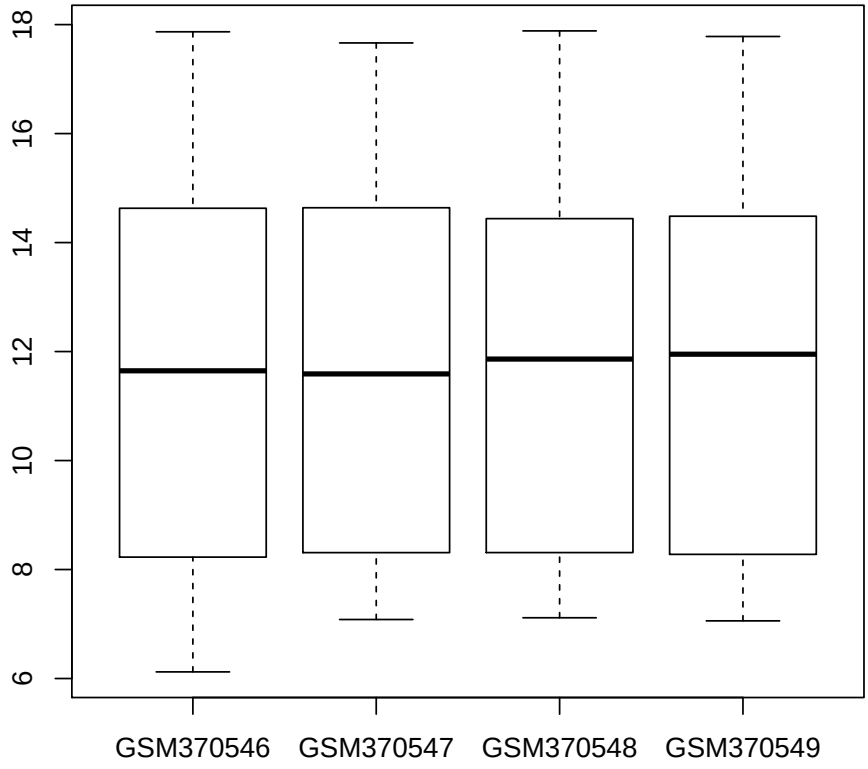


8. Normalization: perform *lvs* normalization based on `MIR.RA`. The default procedure will first summarize data (based on "rlm" method) and then normalize.

```
> MIR.lvs <- lvs(MIR,RA=MIR.RA)
>
```

9. Now have a look at the box plot of data after normalization.

```
> boxplot(MIR.lvs)
>
```



10. Summarization can also be performed without normalization, using different methods.

```
> ex.1 <- summarize(MIR,RA=MIR.RA,method="rlm")
> ex.2 <- summarize(MIR,method="medianpolish")
>
```

11. If no `RA` argument is supplied to the function `lvs`, the computation performed by `estVC` will be carried on within `lvs`. As this is the most computationally intensive step in the procedure we stringly suggest to perform it using `estVC` and saving the result for following steps.

3 Parallel computation

The bottle neck in `LVSmiRNA` in terms of speed is the computation of Array & probes effects (function `estVC`). Therefore `LVSmiRNA` allows for paralle computation using either `multicore` or `snow`.

To use either packages the user must load them and set up the cluster (if needed) manually.

3.1 Using `multicore`

The package `multicore` requires minimum user intervention. The only option is the choice of the number of cores to use. By default `multicore` would use all the available. Setting a different number can be done in the `options`.

```
> require(multicore)
> options(cores=8)
> MIR.RA <- estVC(MIR)
>
```

3.2 Using `snow`

The package `snow` requires the user to set up a cluster object manually. The user must choose the number of clusters and the type. See `?makeCluster` gor more details.

Here is an example using "SOCK" cluster type.

```
> require(snow)
> cl <- makeCluster(8,"SOCK")
> MIR.RA <- estVC(MIR,clName=cl)
> stopCluster(cl)
>
```

And here an example using "MPI". This would load the `Rmpi` package.

```
> cl <- makeCluster(8,"MPI")
> MIR.RA <- estVC(MIR,clName=cl)
> stopCluster(cl)
>
```

References

S Calza, D Valentini, and Y Pawitan. Normalization of Oligonucleotide Arrays Based on the Least-variant Set of Genes. *BMC Bioinformatics*, 140:5–9, 2007.

P. J. Huber. Robust Estimation of a Location Parameter. 35:73–101, 1964.

Y Lee, J Nelder, and Y Pawitan. *Generalized Linear Models with Random Effects*. Chapman and Hall, 2006.

Hanni Willenbrock, Jesper Salomon, K I M Bundvig Barken, Finn Cilius Nielsen, and Thomas Litman. Quantitative miRNA expression analysis: Comparing microarrays with next-generation sequencing. *RNA*, 15:2028–2034, 2009. doi: 10.1261/rna.1699809.454.