Bioconductor Expression Assessment Tool for Affymetrix Oligonucleotide Arrays (affycomp)

Rafael Irizarry and Leslie Cope

October 18, 2010

Contents

1 Introduction

2 Introduction

The defining feature of oligonucleotide expression arrays is the use of several probes to assay each targeted transcript. This is a bonanza for the statistical geneticist, offering a great opportunity to create probeset summaries with specific characteristics. There are now several methods available for summarizing probe level data from the popular Affymetrix GeneChips. It is harder to identify the method best suited to a given inquiry. This package provides a *graphical tool* for summaries of Affymetrix probe level data. Plots and summary statistics offer a picture of how an expression measure performs in several important areas. This picture facilitates the comparison of competing expression measures and the selection of methods suitable for a specific investigation. The key is a benchmark dataset consisting of a dilution study and a spike-in study. Because the *truth* is known for this data, it is possible to identify statistical features of the data for which

the expected outcome is known in advance. Those features highlighted in our suite of graphs are justified by questions of biological interest, and motivated by the presence of appropriate data.

The benchmark data used is freely available to the public. The assessment data sets are the following: For the dilution study by `http://qolotus02.genelogic.com/datasets.nsf/`GeneLogic, two sources of cRNA, human liver tissue and central nervous system cell line (CNS), were hybridized to human arrays (HG-U95Av2) in a range of dilutions and proportions ?.

For the spike-in study, different cRNA fragments were added to the hybridization mixture of the arrays at different picoMolar concentrations. The cRNAs were spiked-in at a different concentration on each array (apart from replicates) arranged in a cyclic Latin square design with each concentration appearing once in each row and column. All arrays had a common background cRNA. The data can be obtained form `http://www.affymetrix.com/analysis/download_center2.affx`.

Two phenoData objects are included in the package that give more details: `dilution.phenodata` and `spikein.phenodata`.

3   What's new in version 1.2?

A new sets of assessment has been added. The wrapper is called `assessSpikeIn2` and one can send it an `ExpressionSet` with the spikein as `assessSpikeIn` but it only uses columns 1,...,13,17 and 21,...,33,37. So one only needs to get expression measures for these 28 arrays if one plans to only use `assessSpikeIn2`. The engine functions are `assessLS`, `assessSpikeInSD`, and `assessMA2`.

All spike in related assessments now work with both the HGU95A (this is the one used in version 1.1) and HGU133A. The function `read.newspikein` will read the HGU133A expression values.

4   The Image Report

If you have a file named `dilfilename.csv` with the dilution expression values and `sifilename.csv` with the spikein expression measures, say RMA, you can easily obtain the graphs and summary statistics in the image report:

```
R> library(affycomp) ##load the affy package
R> d <- read.dilution("dilfilename.csv")
R> s <- read.spikein("sifilename.csv")
R> rma.assessment <- affycomp(d, s, method.name="RMA")
```

`res` will have the necessary information to recreate the graphs without having to wait for the assessment. See below.

`affycomp` is a wrapper for other `assessAll` and `affycompTable`. Here are some examples based

```
> data(mas5.assessment)
> data(rma.assessment)
```

were created using the function `assessAll` on `ExpressionSet`s created using the RMA and MAS 5.0 methods. These are lists of lists.

```
> names(mas5.assessment)
```

```
[1] "Dilution"    "MA"            "Signal"        "FC"            "FC2"
[6] "what"        "method.name"
```

Each component is the result of a specific assessment. The names tell us what they are for. `Dilution` are the assessment based on the dilution data and can be used to create Figures 2, 3, and 4b. `MA` has the necessary information for the MA plot or Figure 1. `Signal` has the necessary information to create Figure 4a. `FC` has assessments related to fold change and can be used to create Figures 5a, 6a, and 6b. Finally `FC2` has the necessary information to create Figure 5b. The captions for these Figures will give you an idea of what they are for.

There are two kinds of plots the basic and the comparative. In the basic plots based on the expression measure being assessed are shown. In the comparative plots, the expression measure is compared to other measures, MAS 5.0 by default. Table are also automatically created with assessment statistics. Finally, a simple assessment of standard error estimates can be done. These are described in the following subsections.

4.1   Basic Plots

You can use `affycompPlot` which will automatically know what to do

Or you can use the auxiliary figure functions that will need to have a specific assessment list

4.2   Comparative Plots

You can use `affycompPlot` which will automatically know what to do

Or you can use the auxiliary figure functions that will need to have a specific assessment list

4.3   Tables

The function `tableAll` returns a matrix with assessment statistics. Once the assessment function are run all one needs to type is

```
> affycompPlot(mas5.assessment$MA)
```
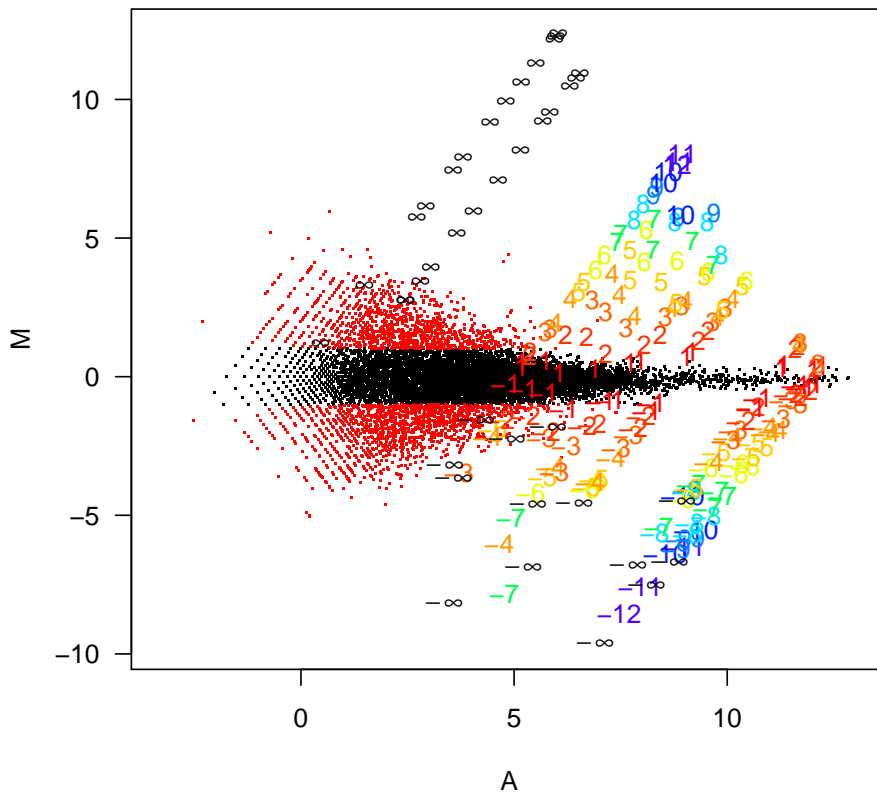
Figure 1

Figure 1: The MA plot shows log fold change as a function of mean log expression level. A set of 14 arrays representing a single experiment from the Affymetrix spike-in data are used for this plot. A total of 13 sets of fold changes are generated by comparing the first array in the set to each of the others. Genes are symbolized by numbers representing the nominal $\log_2$ fold change for the gene. Non-differentially expressed genes with observed fold changes larger than 2 are plotted in red. All other probesets are represented with black dots.

```
> affycomp.figure2(mas5.assessment$Dilution)
```
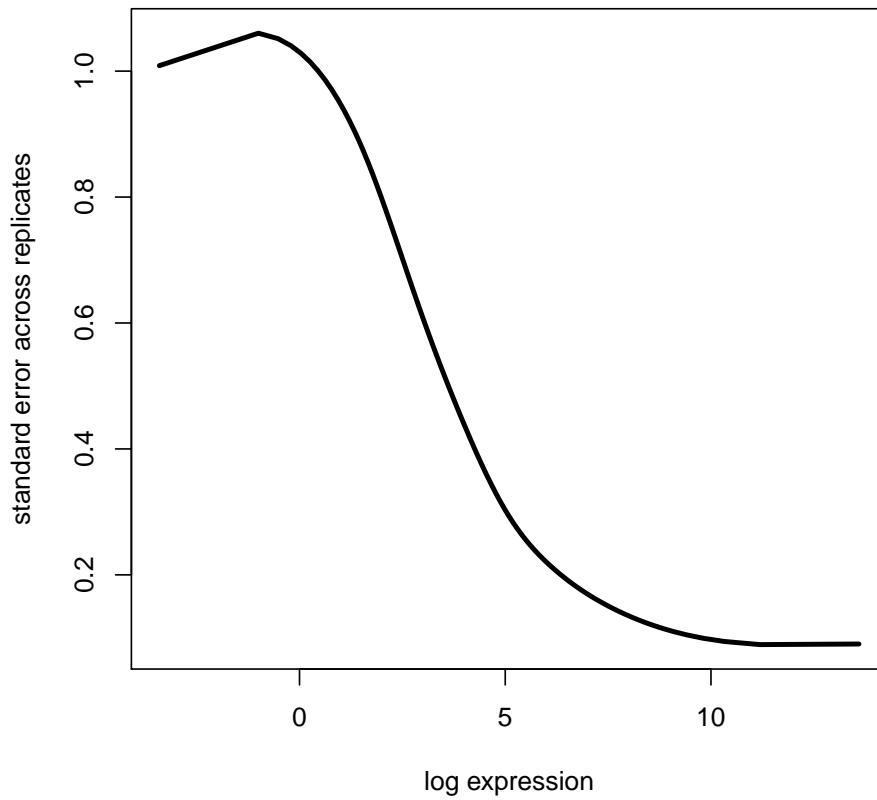
Figure 2

Figure 2: For each gene, and each experimental condition, we calculate the mean log expression and the observed standard deviation across 5 replicates. The resulting scatterplot is smoothed to generate a single curve representing mean standard deviation as a function of mean log expression.

```
> affycomp.figure3(mas5.assessment$Dilution)
```
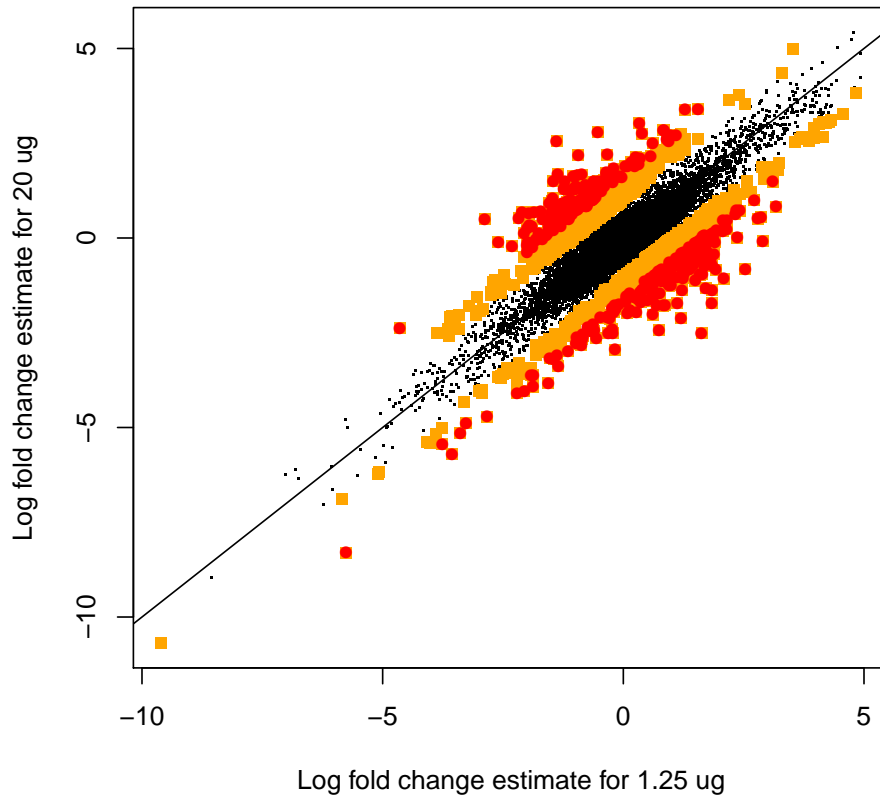
Figure 3

Figure 3: This plot, using the GeneLogic dilution data, shows the sensitivity of fold change calculations to total RNA abundance. Average log fold-changes between liver and CNS for the lowest concentration and the highest in the dilution data set are computed. Orange and red color is used to denote genes with $M_{6g} - M_{1g}$ bigger than $\log_2(2)$ and $\log_2(3)$ respectively. The rest are denoted with black.

```
> par(mfrow = c(2, 1))
> affycomp.figure4a(mas5.assessment$Signal)
> affycomp.figure4b(mas5.assessment$Dilution)
```
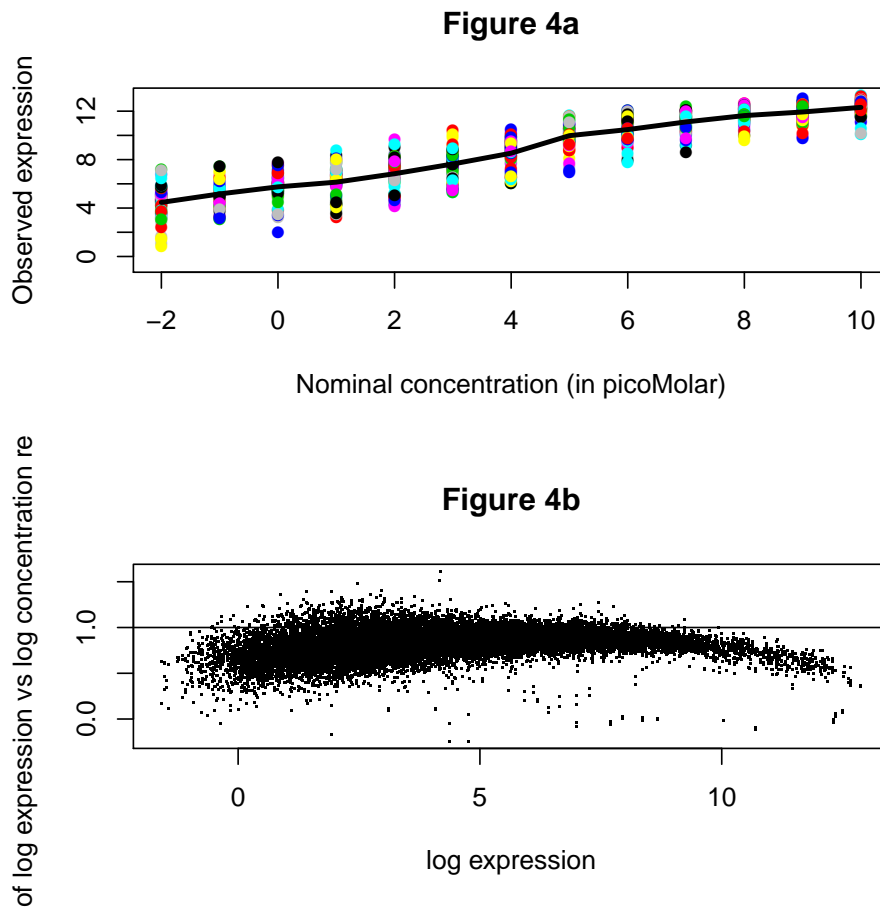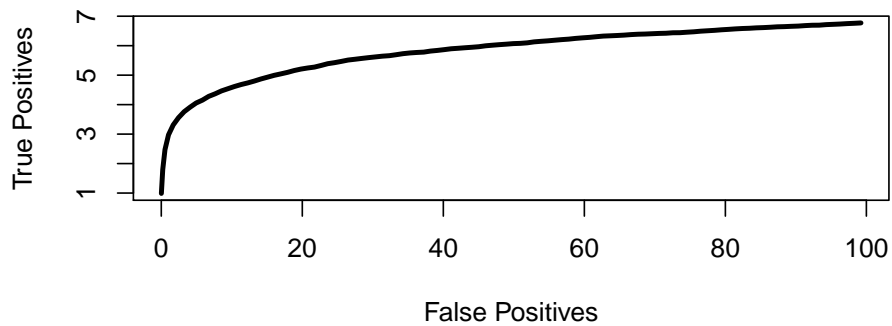
Figure 4a

Figure 4b

Figure 4: a) Average observed log_2 intensity plotted against nominal log_2 concentration for each spiked-in gene for all arrays in Affymetrix spike-In experiment. b) For the GeneLogic dilution data, log expression values are regressed against their log nominal concentration. The slope estimates are plotted against average log intensity across all concentrations.

```
> par(mfrow = c(2, 1))
> affycomp.figure5a(mas5.assessment$FC)
> affycomp.figure5b(mas5.assessment$FC)
```
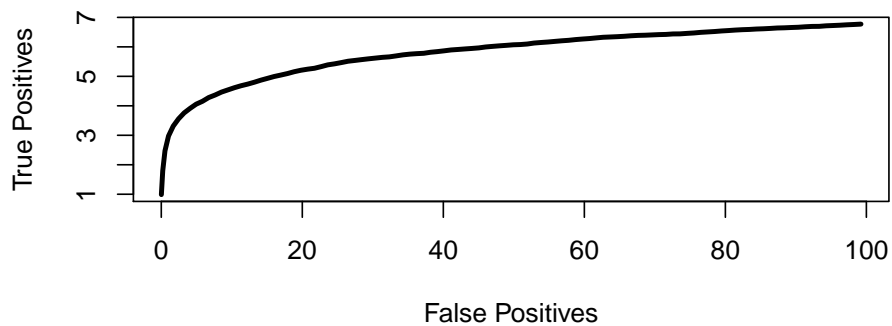
Figure 5a

Figure 5b

Figure 5: A typical identification rule for differential expression filters genes with fold change exceeding a given threshold. This figure shows average ROC curves which offer a graphical representation of both specificity and sensitivity for such a detection rule. a) Average ROC curves based on comparisons with nominal fold changes ranging from 2 to 4096. b) As a) but with nominal fold changes equal to 2.
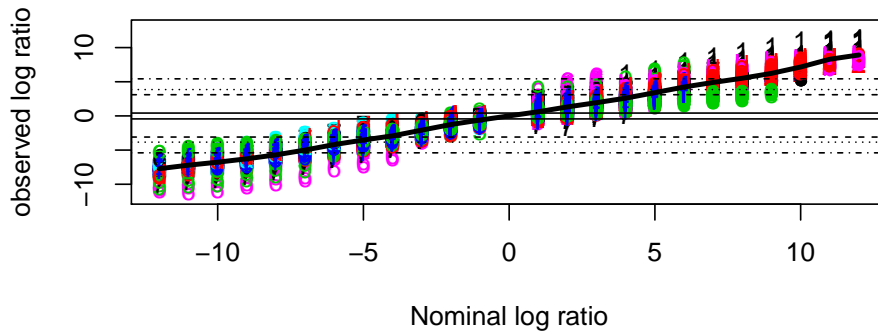
```
> par(mfrow = c(2, 1))
> affycomp.figure6a(mas5.assessment$FC)
> affycomp.figure6b(mas5.assessment$FC)
```
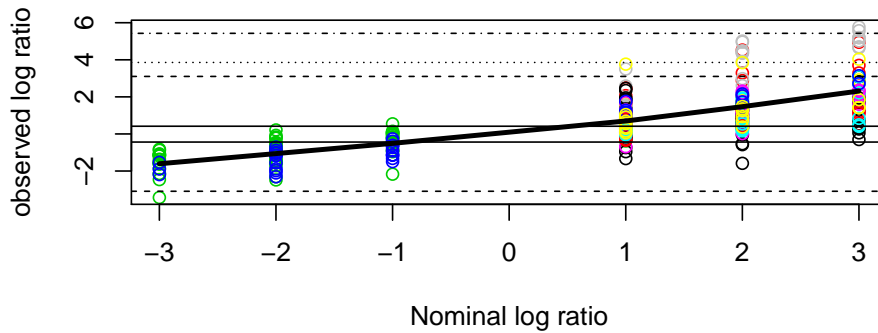
Figure 6a

Figure 6b

Figure 6: a) Observed log fold changes plotted against nominal log fold changes. The dashed lines represent highest, 25th highest, 100th highest, 25th percentile, 75th percentile, smallest 100th, smallest 25th, and smallest log fold change for the genes that were not differentially expressed. b) Like a) but the observed fold changes were calculated for spiked in genes with nominal concentrations no higher than 2pM.

```
> par(mfrow = c(2, 1))
> affycompPlot(mas5.assessment$MA, rma.assessment$MA)
```
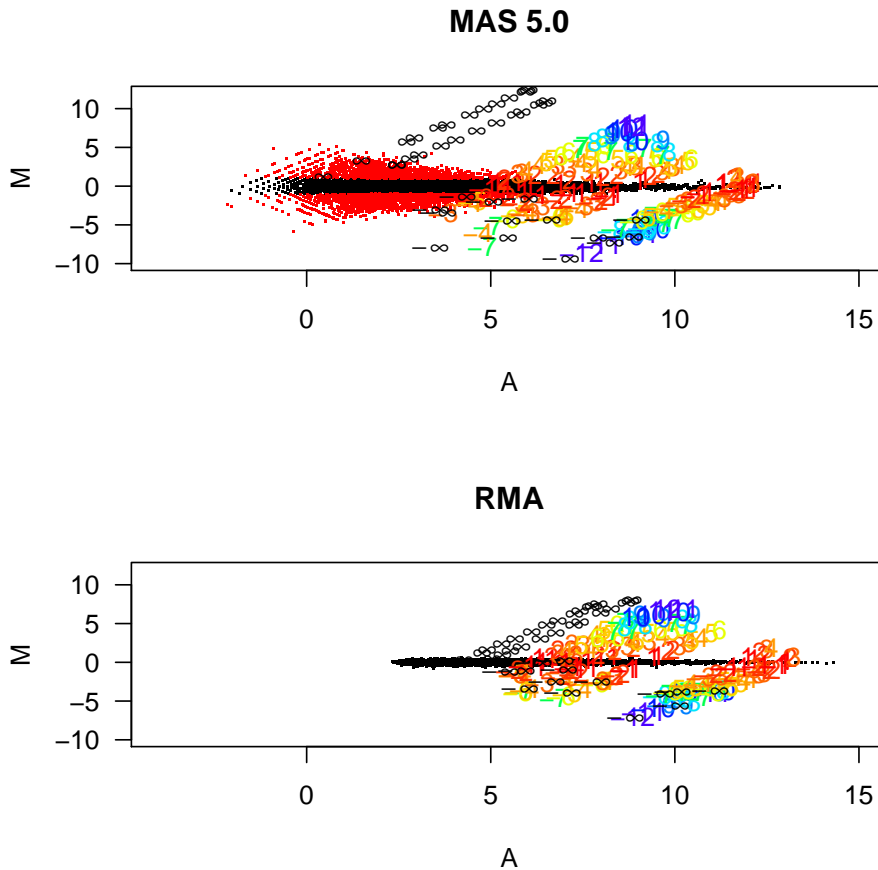

Figure 1: The MA plot shows log fold change as a function of mean log expression level. A set of 14 arrays representing a single experiment from the Affymetrix spike-in data are used for this plot. A total of 13 sets of fold changes are generated by comparing the first array in the set to each of the others. Genes are symbolized by numbers representing the nominal $\log_2$ fold change for the gene. Non-differentially expressed genes with observed fold changes larger than 2 are plotted in red. All other probesets are represented with black dots.

```
> affycomp.compfig2(list(mas5.assessment$Dilution, rma.assessment$Dilution),
+      method.names = c("MAS 5.0", "RMA"))
```
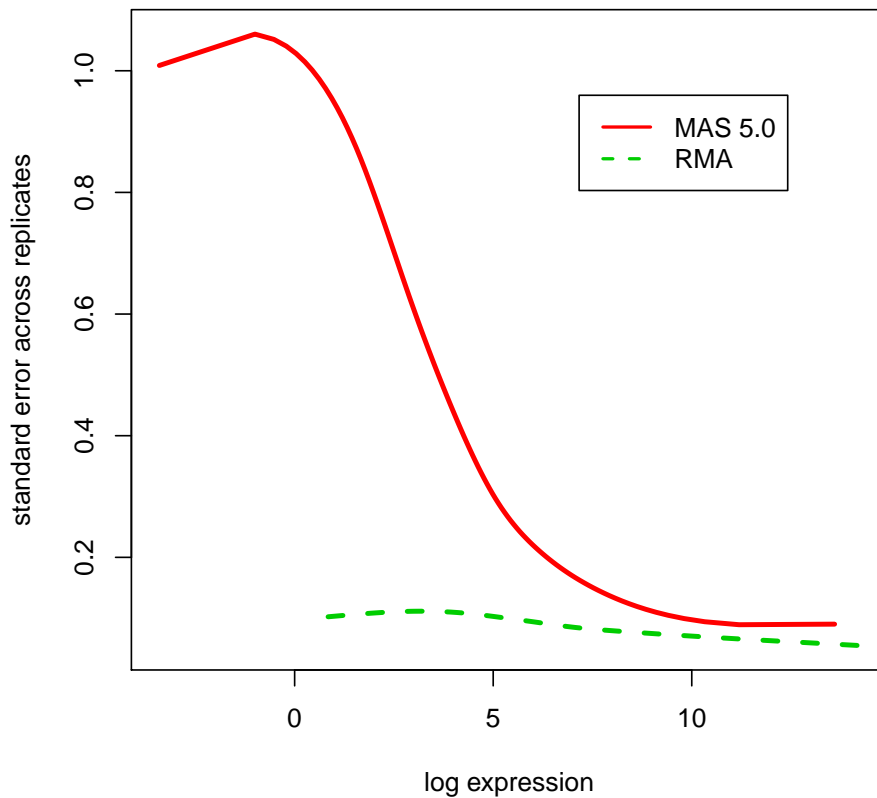
Figure 2

Figure 2: For each gene, and each experimental condition, we calculate the mean log expression and the observed standard deviation across 5 replicates. The resulting scatterplot is smoothed to generate a single curve representing mean standard deviation as a function of mean log expression.

```
> affycomp.compfig3(list(mas5.assessment$Dilution, rma.assessment$Dilution),
+     method.names = c("MAS 5.0", "RMA"))
```
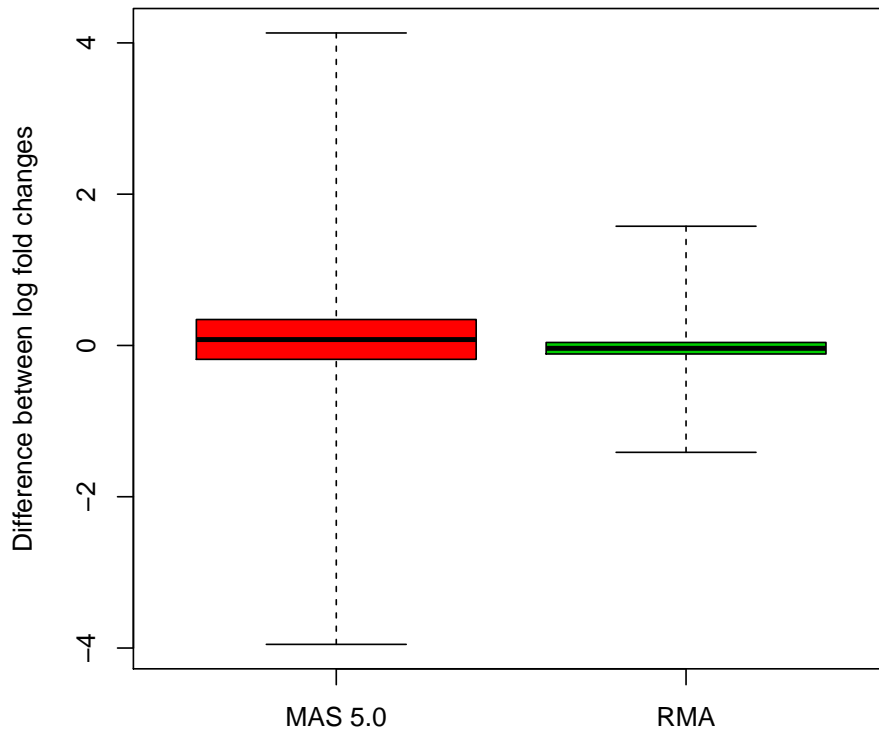
Figure 3

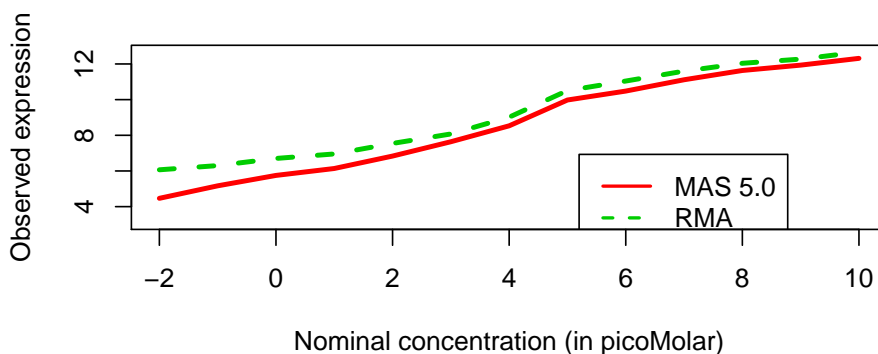Figure 3: This plot, using the GeneLogic dilution data, shows the sensitivity of fold change calculations to total RNA abundance. Average log fold-changes between liver and CNS for the lowest concentration and the highest in the dilution data set are computed. Orange and red color is used to denote genes with $M_{6g} - M_{1g}$ bigger than $\log_2(2)$ and $\log_2(3)$ respectively. The rest are denoted with black.

```
> par(mfrow = c(2, 1))
> affycomp.compfig4a(list(mas5.assessment$Signal, rma.assessment$Signal),
+       method.names = c("MAS 5.0", "RMA"))
> affycomp.compfig4b(list(mas5.assessment$Dilution, rma.assessment$Dilution),
+       method.names = c("MAS 5.0", "RMA"))
```
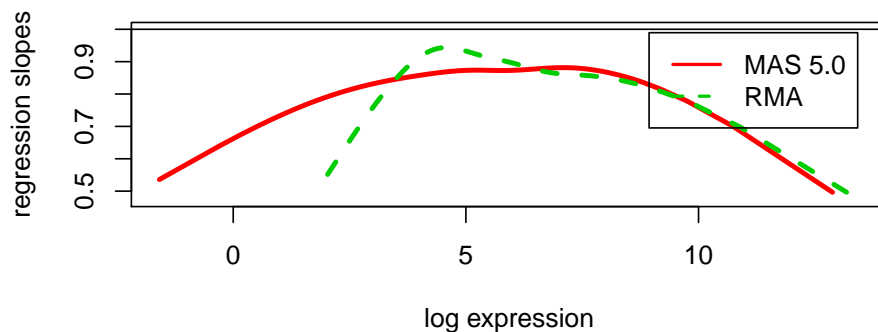

Figure 4: a) Average observed log_2 intensity plotted against nominal log_2 concentration for each spiked-in gene for all arrays in Affymetrix spike-In experiment. b) For the GeneLogic dilution data, log expression values are regressed against their log nominal concentration. The slope estimates are plotted against average log intensity across all concentrations.

```
> par(mfrow = c(2, 1))
> affycomp.compfig5a(list(mas5.assessment$FC, rma.assessment$FC),
+     method.names = c("MAS 5.0", "RMA"))
> affycomp.compfig5b(list(mas5.assessment$FC2, rma.assessment$FC2),
+     method.names = c("MAS 5.0", "RMA"))
```

Figure 5a
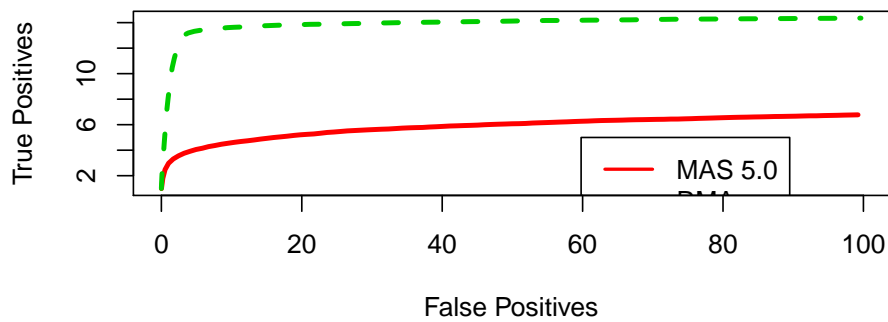
Figure 5b

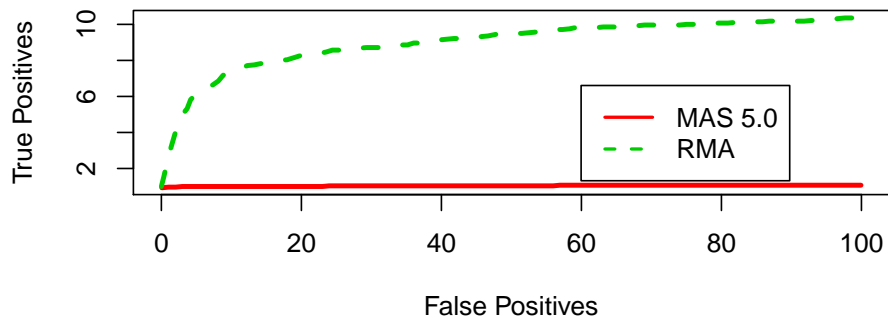Figure 5: A typical identification rule for differential expression filters genes with fold change exceeding a given threshold. This figure shows average ROC curves which offer a graphical representation of both specificity and sensitivity for such a detection rule. a) Average ROC curves based on comparisons with nominal fold changes ranging from 2 to 4096. b) As a) but with nominal fold changes equal to 2.

```
> par(mfrow = c(2, 2))
> affycomp.figure6a(mas5.assessment$FC, main = "a) MAS 5.0", ylim = c(-12,
+     12))
> affycomp.figure6a(rma.assessment$FC, main = "a) RMA", ylim = c(-12,
+     12))
> affycomp.figure6b(mas5.assessment$FC, main = "b) MAS 5.0", ylim = c(-6,
+     6))
> affycomp.figure6b(rma.assessment$FC, main = "b) RMA", ylim = c(-6,
+     6))
```


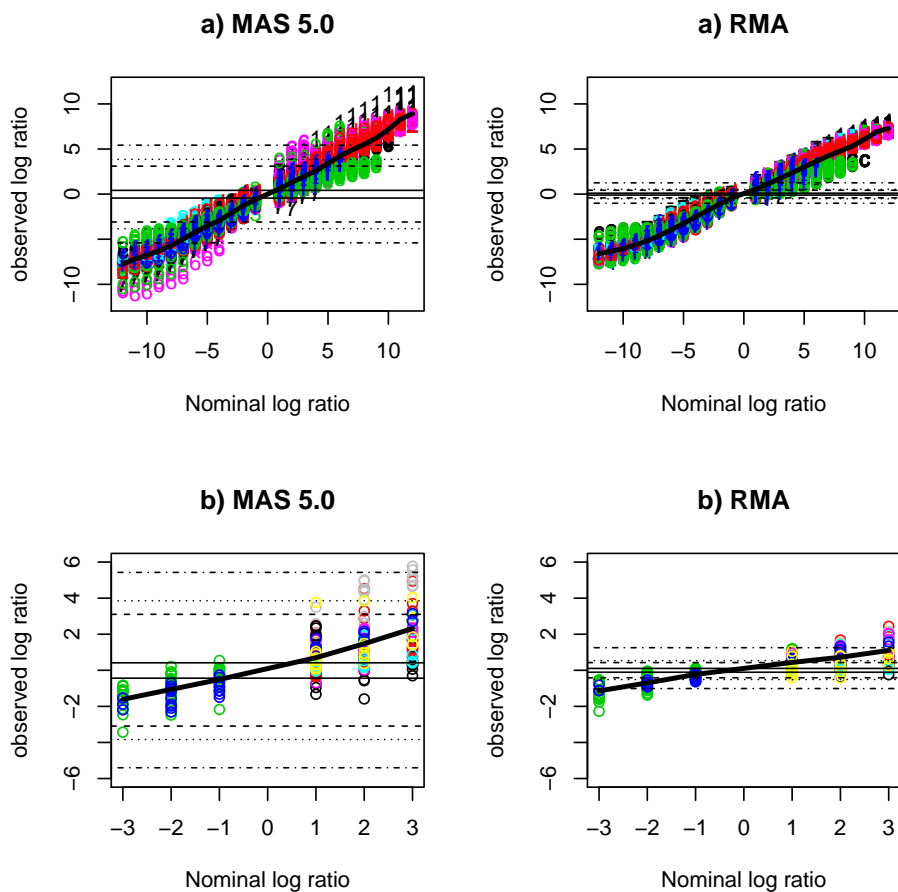Figure 6: a) Observed log fold changes plotted against nominal log fold changes. The dashed lines represent highest, 25th highest, 100th highest, 25th percentile, 75th percentile, smallest 100th, smallest 25th, and smallest log fold change for the genes that were not differentially expressed. b) Like a) but the observed fold changes were calculated for spiked in genes with nominal concentrations no higher than 2pM.

```
> data(rma.assessment)
> data(mas5.assessment)
> tableAll(rma.assessment, mas5.assessment)


                              RMA        MAS 5.0
Median SD                0.08811999 2.920239e-01
R2                       0.99420626 8.890008e-01
1.25v20 corr             0.93645083 7.297434e-01
2-fold discrepancy      21.00000000 1.226000e+03
3-fold discrepancy       0.00000000 3.320000e+02
Median slope             0.86631340 8.474941e-01
Signal detect slope      0.62537111 7.058227e-01
Signal detect R2         0.80414899 8.565416e-01
AUC (FP<10)              0.57774758 2.171241e-01
AUC (FP<15)              0.62731941 2.379589e-01
AUC (FP<25)              0.68984189 2.696331e-01
AUC (FP<100)             0.82066051 3.557341e-01
AFP, call if fc>2       15.84156379 3.108992e+03
ATP, call if fc>2       11.97942387 1.281893e+01
IQR                      0.30801579 2.655135e+00
Obs-intended-fc slope    0.61209902 6.932507e-01
Obs-(low)int-fc slope    0.35950904 6.471881e-01
FC=2, AUC (FP<10)        0.30344013 6.160714e-02
FC=2, AUC (FP<15)        0.34324269 6.190476e-02
FC=2, AUC (FP<25)        0.40009470 6.232830e-02
FC=2, AUC (FP<100)       0.54261364 6.508575e-02
FC=2, AFP, call if fc>2  1.00000000 3.072179e+03
FC=2, ATP, call if fc>2  1.71428571 3.714286e+00
```

The function `affycompTable` make a minimal table (that is also more informative).

```
> affycompTable(rma.assessment, mas5.assessment)


                          RMA      MAS.5.0 whatsgood Figure
Median SD            0.08811999 2.920239e-01         0      2
R2                   0.99420626 8.890008e-01         1      2
1.25v20 corr         0.93645083 7.297434e-01         1      3
2-fold discrepancy  21.00000000 1.226000e+03         0      3
3-fold discrepancy   0.00000000 3.320000e+02         0      3
Signal detect slope  0.62537111 7.058227e-01         1     4a
Signal detect R2     0.80414899 8.565416e-01         1     4a
Median slope         0.86631340 8.474941e-01         1     4b
AUC (FP<100)         0.82066051 3.557341e-01         1     5a
```

```
AFP, call if fc>2       15.84156379 3.108992e+03       0    5a
ATP, call if fc>2       11.97942387 1.281893e+01      16    5a
FC=2, AUC (FP<100)       0.54261364 6.508575e-02       1    5b
FC=2, AFP, call if fc>2  1.00000000 3.072179e+03       0    5b
FC=2, ATP, call if fc>2  1.71428571 3.714286e+00      16    5b
IQR                      0.30801579 2.655135e+00       0     6
Obs-intended-fc slope    0.61209902 6.932507e-01       1    6a
Obs-(low)int-fc slope    0.35950904 6.471881e-01       1    6b
```

4.4   Standard deviation assessment

The package also contains a simple tool to assess standard error estimates. For this to work the `ExpressionSet` object used for the assessment must have standard error estimates for the dilution data. We include two examples in the package.

```
> data(rma.sd.assessment)
> data(lw.sd.assessment)
> tableAll(rma.sd.assessment, lw.sd.assessment)

                   RMA      dChip
IQR of log ratio 0.910003 1.1918510
Correlation      0.364145 0.7925533
```

For the SD assessment, there are also comparison plots as well as simple plots.
There are also comparison plots as well as simple plots.
Enjoy!

```
> affycompPlot(lw.sd.assessment, rma.sd.assessment)
```
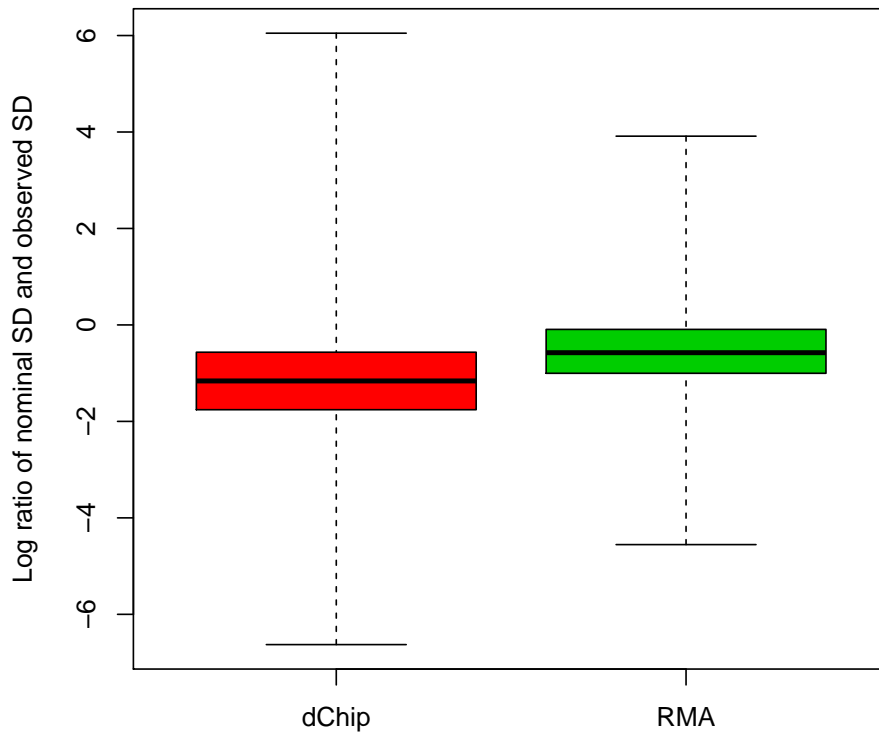
Figure 7

Figure 7: Using the replicates from the dilution data, we calculate the mean predicted variance for each gene, tissue and dilution by squaring the estimated standard error. The usual sample variance $s^2_{tdg} = \sum_r (y_{tdrg} - y_{td \cdot g})^2/4$ are calculated as well. These boxplots are of the log ratios of the predicted and observed variance.

```
> affycompPlot(lw.sd.assessment)
```
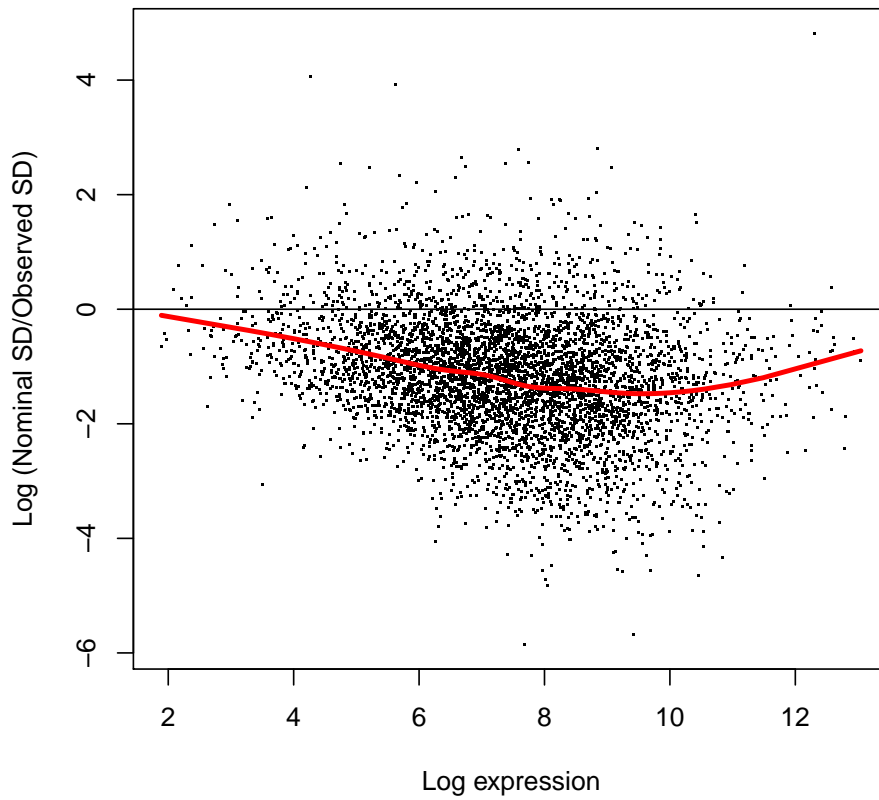
Figure 7

Figure 7: Using the replicates from the dilution data, we calculate the mean predicted variance for each gene, tissue and dilution by squaring the estimated standard error. The usual sample variance $s_{tdg}^2 = \sum_r (y_{tdrg} - y_{td \cdot g})^2 / 4$ are calculated as well. These boxplots are of the log ratios of the predicted and observed variance.