Fitting and visualising row-linear models with `consensus`

Tim Peters

October 26, 2023

Summary

This short vignette will demonstrate how to fit a set of row-linear models in the form of an interlaboratory testing procedure (ASTM Standard E691) in a genomic context. This allows us to make comparisons of sensitivity and precision for each platform/condition the samples are measured under. We can then make broader inferences about the suitability of a given technology.

In an ideal world we would like a set of "gold standards" to validate a new technology or laboratory protocol when quantifying various genomic characteristics. The problem with this is that *all* genomic measurements are estimates, even those that are seen as more reliable, such as qPCR. Stochastic sampling of individual molecules (and their subsequent amplification) is an inescapable part of most modern laboratory protocols, which means that, inevitably, there will always be error present in the measurement. Rather than ignore this error and defining a "gold standard" (thus biasing all subsequent measurements towards that standard), the aim of this package is to encourage an empirical approach to characterising the advantages and disadvantages of a suite of candidate technological platforms, laboratory protocols or other various conditions that may influence the measurement. With this in mind, we start with some normalised gene expression data sourced from The Cancer Genome Atlas (TCGA) glioblastoma multiforme (GBM) project[1].

Firstly, make sure the package is installed:

```
if (!require("BiocManager"))
    install.packages("BiocManager")
BiocManager::install("consensus")
```

Then load package and data:

```
library(consensus)
data("TCGA")
```

We have 27 matched samples assayed across four different gene expression measurement platforms: Affymetrix-HT-HG-U133A GeneChip, Affymetrix HuEx GeneChip, Custom Agilent 244,000 feature Gene Expression Microarray and a polyA selection RNA-Seq protocol. We have selected 1000 genes at random for this test dataset.

```
sapply(mget(c("U133A", "Huex", "Agilent", "RNASeq")), dim)
```

```
##      U133A Huex Agilent RNASeq
## [1,]  1000 1000    1000   1000
## [2,]    27   27      27     27
```

```
rnames <- sapply(mget(c("U133A", "Huex", "Agilent", "RNASeq")), rownames)
head(rnames)
```

```
##      U133A   Huex    Agilent RNASeq
## [1,] "A2M"   "A2M"   "A2M"   "A2M"
## [2,] "ABCA3" "ABCA3" "ABCA3" "ABCA3"
## [3,] "ABCA4" "ABCA4" "ABCA4" "ABCA4"
## [4,] "ABCB9" "ABCB9" "ABCB9" "ABCB9"
## [5,] "ABCC3" "ABCC3" "ABCC3" "ABCC3"
## [6,] "ABCF3" "ABCF3" "ABCF3" "ABCF3"
```

```
apply(rnames[,2:ncol(rnames)], 2, function (x) all(x==rnames[,1]))
```

```
##    Huex Agilent  RNASeq
##    TRUE    TRUE    TRUE
```

```
cnames <- sapply(mget(c("U133A", "Huex", "Agilent", "RNASeq")), colnames)
head(cnames)
```

```
##      U133A                 Huex                  Agilent
## [1,] "TCGA.06.0211.01B.01" "TCGA.06.0211.01B.01" "TCGA.06.0211.01B.01"
## [2,] "TCGA.06.0190.01A.01" "TCGA.06.0190.01A.01" "TCGA.06.0190.01A.01"
## [3,] "TCGA.06.0238.01A.02" "TCGA.06.0238.01A.02" "TCGA.06.0238.01A.02"
## [4,] "TCGA.06.0645.01A.01" "TCGA.06.0645.01A.01" "TCGA.06.0645.01A.01"
## [5,] "TCGA.06.0132.01A.02" "TCGA.06.0132.01A.02" "TCGA.06.0132.01A.02"
## [6,] "TCGA.12.0618.01A.01" "TCGA.12.0618.01A.01" "TCGA.12.0618.01A.01"
##      RNASeq
## [1,] "TCGA.06.0211.01B.01"
## [2,] "TCGA.06.0190.01A.01"
## [3,] "TCGA.06.0238.01A.02"
## [4,] "TCGA.06.0645.01A.01"
## [5,] "TCGA.06.0132.01A.02"
## [6,] "TCGA.12.0618.01A.01"
```

```
apply(rnames[,2:ncol(rnames)], 2, function (x) all(x==rnames[,1]))
```

```
##    Huex Agilent  RNASeq
##    TRUE    TRUE    TRUE

rm(rnames, cnames)
```

Notice that the dimensions, row names and column names are identical across all measurement matrices. This is required for when we construct a MultiMeasure object from this data. If this requirement is not met, an error message will tell you which matrix attributes don't match.

Now we construct the MultiMeasure:

```
tcga_mm <- MultiMeasure(names=c("U133A", "Huex", "Agilent", "RNA-Seq"),
                        data=list(U133A, Huex, Agilent, RNASeq))
tcga_mm

## MultiMeasure object with 4 platforms/conditions, 27 samples and 1000 measured loci.
```

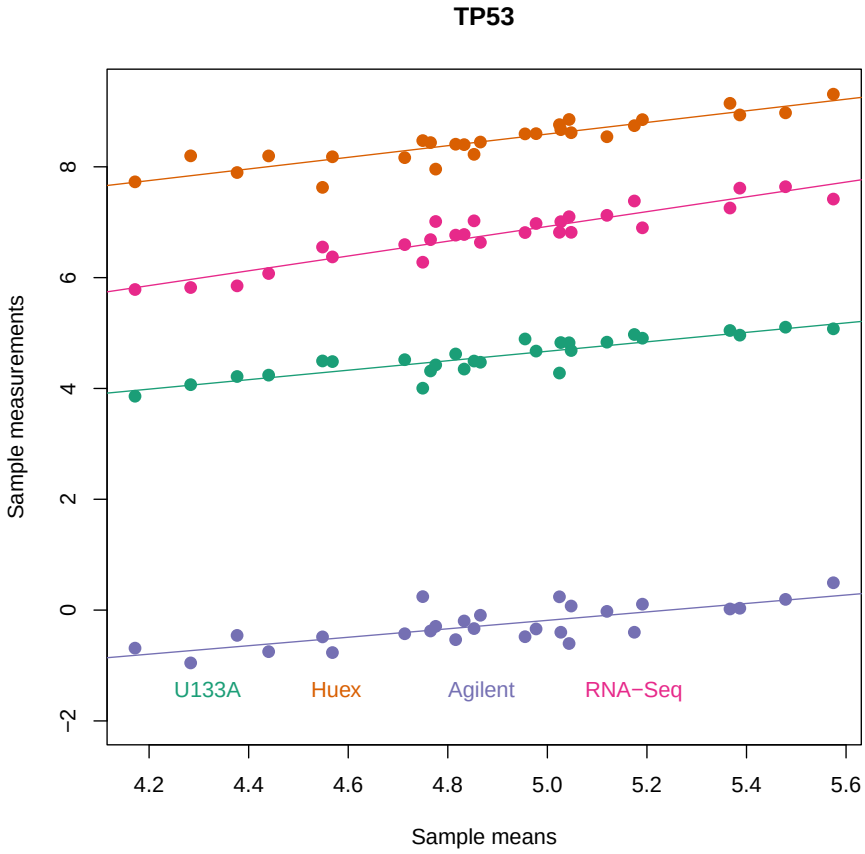
We can fit the data using the row-linear method from the ASTM standard[2]. One fit is performed for each gene represented by the matrix. The row-linear fit can be expressed in the form:

$$Z_{ij} = a_i + b_i(x_j - \bar{x}) + d_{ij} \tag{1}$$

where Z_{ij} is 4x27 matrix of measurements from sample j on platform i. a_i is row mean of the ith platform, x_j the column mean of the jth sample and $\bar{x}$ the grand mean of Z_{ij}. b_i is the slope of the regression of the sample measurements from platform i on $x_j - \bar{x}$, and d_{ij} the residual scatter about this line.

Firstly, let's visualise one of these fits, from a well-known gene, TP53:

```
plotOneFit(tcga_mm, "TP53", brewer.pal(n = 4, name = "Dark2"))
```



This gene is generally concordant across platforms, since the regression lines are fairly parallel and the residuals don't fall too far away.

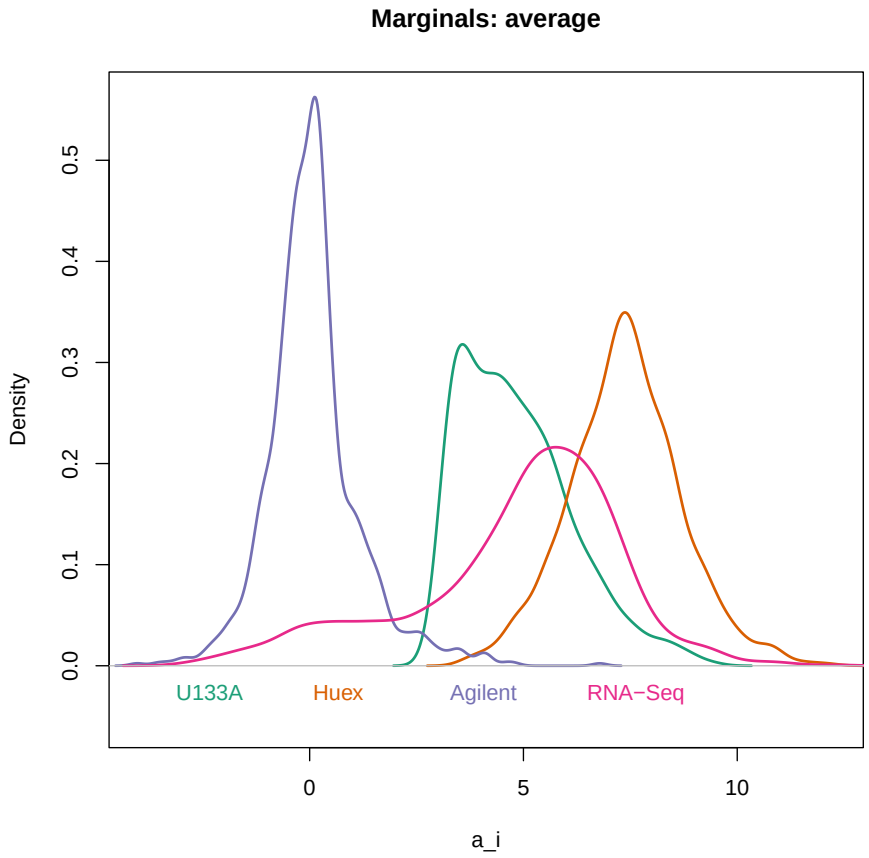
Now to perform the fitting. This will create a S4 class object of type ConsensusFit.

```
fit <- fitConsensus(tcga_mm)
fit

## ConsensusFit object with 4 platforms/conditions and 1000 measured loci.
```

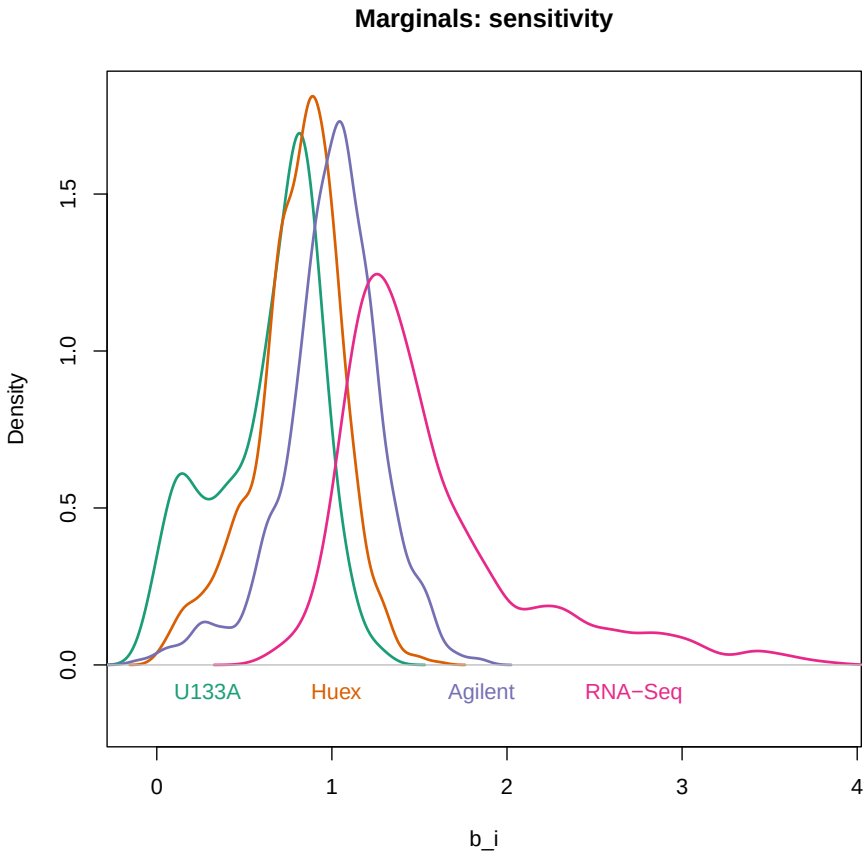
Once this is done, we might be interested in seeing what the distributions of some parameters from Equation (1) look like over all 1000 genes. First, let's see what the distribution of the averages (a_is) are, which serve as dynamic ranges of each platform:

```
plotMarginals(fit, "average", brewer.pal(n = 4, name = "Dark2"))
```



Then we can see which platforms have the greatest sensitivity (b_i) to changes in gene expression:

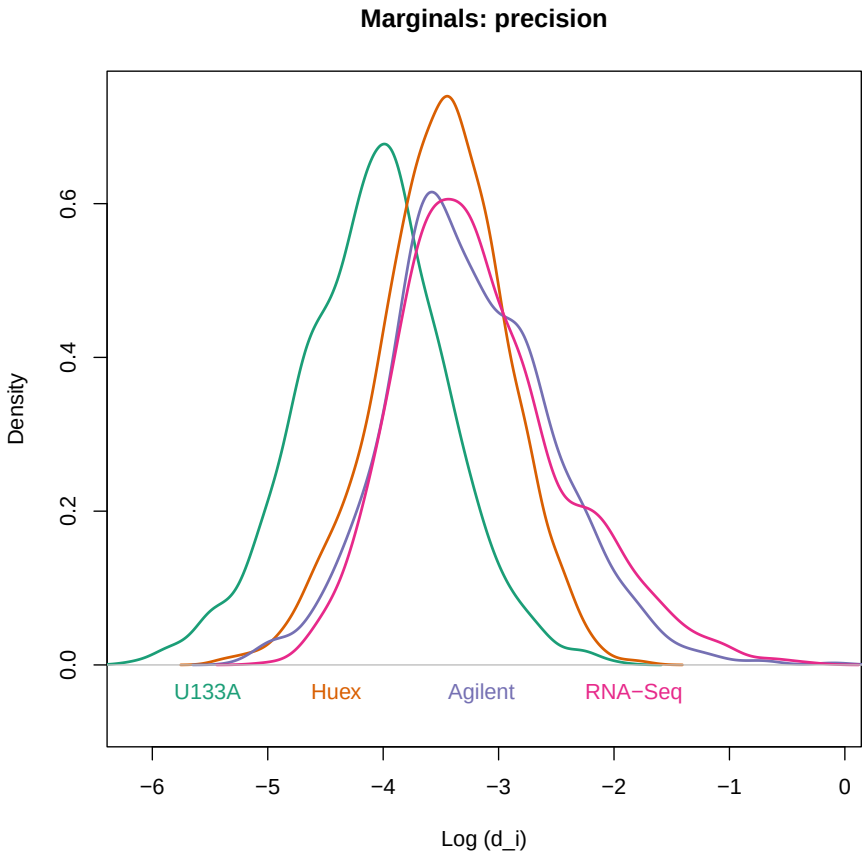
```
plotMarginals(fit, "sensitivity", brewer.pal(n = 4, name = "Dark2"))
```



Clearly, RNA-Seq is the most sensitive, followed by the Agilent array, then Huex and finally U133A. Interestingly, U133A has a second mode at 0, which is indicative of a subset of genes that do not show any response to expression change on this platform. The marginals of the averages for this platform show a right skew, which contributes to this phenomenon.

Let's plot the precision (d_i) marginals, remembering that higher values mean lower precision:

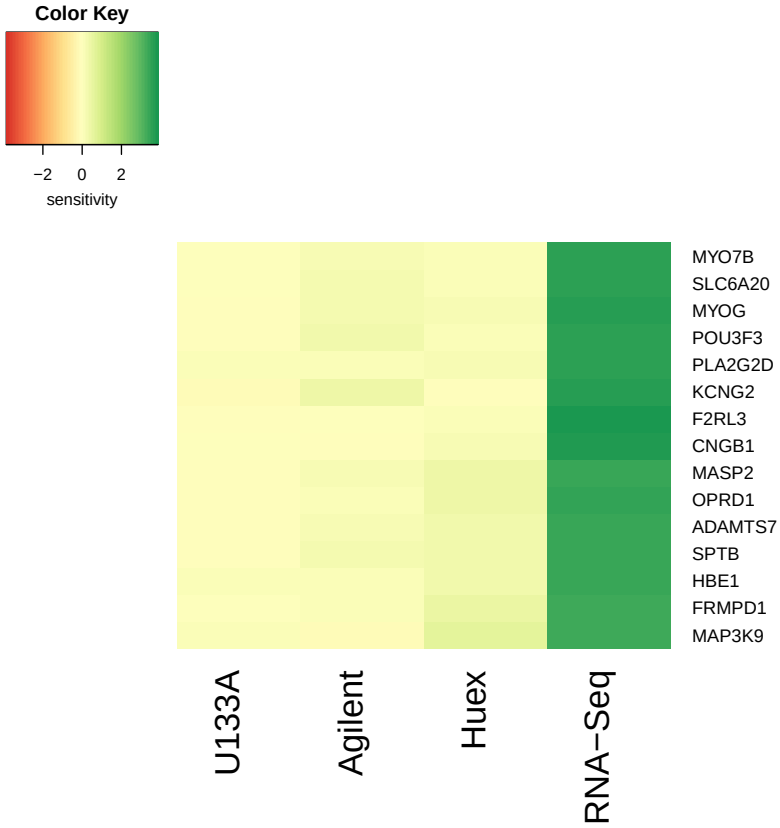
```
plotMarginals(fit, "precision", brewer.pal(n = 4, name = "Dark2"))
```



All platforms except U133A are generally similar in their precision, suggesting that, from a platform design perspective, there may have been a trade-off between risk and reward in detecting changes in gene expression on U133A that has now been surmounted by more recent technologies.

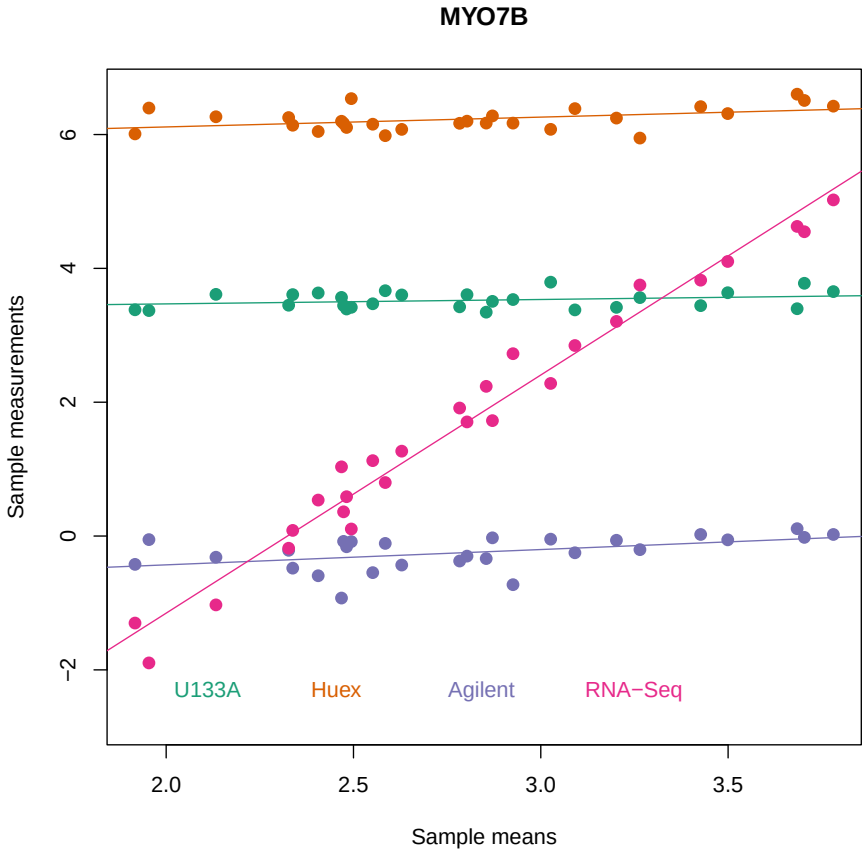
Finally, we are interested in the gene loci whose measurements are the most *dis*cordant across platforms. These are most easily visualised on a heatmap. Like with `plotMarginals`, we can choose to plot discordance in terms of average, sensitivity or precision.

```
plotMostDiscordant(fit, "sensitivity", 15)
```



Clearly, the governing feature of the most sensitivity-discordant genes is that RNA-Seq explains the vast majority of change in gene expression. To see what a row-linear fit looks like for one of these genes, we again use `plotOneFit`.

```
plotOneFit(tcga_mm, "MYO7B", brewer.pal(n = 4, name = "Dark2"))
```



We have seen that the row-linear model is able to provide us with a method of assessing the measurement quality of various transcriptomic platforms, that acts as an alternative to a "gold standard". The model need not be restricted to gene expression either - platforms assessing other measurements such as DNA methylation are just as applicable. The method can also be used to assess competing laboratory protocols, given a suite of matched aliquots of material are provided across three or more variations on such processes.

```
sessionInfo()

## R version 4.3.1 (2023-06-16)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Ventura 13.6.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib
```

```
##
## locale:
## [1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] consensus_1.20.0   RColorBrewer_1.1-3 knitr_1.43
##
## loaded via a namespace (and not attached):
##  [1] compiler_4.3.1     gplots_3.1.3       tools_4.3.1        KernSmooth_2.23-22
##  [5] highr_0.10         caTools_1.18.2     xfun_0.39          matrixStats_1.0.0
##  [9] bitops_1.0-7       gtools_3.9.4       evaluate_0.21
```

References

[1] Verhaak, R. G. W., Hoadley, K. A., Purdom, E., Wang, V., Qi, Y., Wilkerson, M. D., ..., Cancer Genome Atlas Research Network. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer Cell*, 2010, **17**(1), 98-110.

[2] Mandel, J. Analyzing Interlaboratory Data According to ASTM Standard E691. In *Quality and Statistics: Total Quality Management* (pp. 59-59-12), 1994. 100 Barr Harbor Drive, PO Box C700, West Conshohocken, PA 19428-2959: ASTM International.