

# Lab: Analysis of complex microarray experimental designs

Martin Morgan

10 January, 2007

## 1 Part I: An initial analysis

Recall: two-color microarray experiment.

- 5 cell lines PEC32, PEC34, PEC36, PEC39, PEC40.
- 2 drugs (SFN and HGF) each present at two levels (Low and High).
- Each chip has 15488 spots representing 2 technical replicates of each of 7744 genes.

### 1.1 Understanding the design

Evaluate the following code to read a description of the experimental design into R:

```
> dataDir <- system.file("extdata", package = "HGFSFN")
> library("limma")
> targets <- readTargets("ArrayDesc", path = dataDir)
> factors <- targets[["Cy3"]]
> exptlDesign <- data.frame(cellLine = factor(sub("PEC([0-9]+)_.*",
+ "\\1", factors)), SFN = factor("Low", levels = c("Low",
+ "High")), HGF = factor("Low", levels = c("Low",
+ "High")))
> exptlDesign[grepl("SFN", targets[["Cy3"]]), "SFN"] <- "High"
> exptlDesign[grepl("HGF", targets[["Cy3"]]), "HGF"] <- "High"
```

1. Notice that `targets[["Cy3"]]` has different treatments, while `targets[["Cy5"]]` is always labeled “control”. This indicates that the data are from a two-color array with a **common reference design**:

treatments are always assayed with Cy3, and are always directly compared to control samples assayed with Cy5.

2. Use R commands like `show`, `table` and `summary` to explore the experimental design described by `exptlDesign`.
3. How many factors, and how many treatment levels of each factor, are present in this design?
4. How much replication is there? Does this limit the analyses that can be performed, and hence the biological questions asked?
5. It is not apparent from the design data, but each chip (row of `exptlDesign`) contains two replicates of each gene. How might this influence the design and statistical power of the experiment?

Note: the results of the following steps can be loaded with

```
> library("HGFSFN")
> data("M")
```

but feel free to recreate the data as follows:

- Read the data into your session of R:

```
> RG <- read.maimages(targets[["FileName"]], path = dataDir,
+   source = "genepix", wt.fun = wtflags())
```

- Normalize the data, e.g., using `vsn` normalization; the `strata` argument is meant to accommodate the duplicate spots present on the array.

```
> MA <- normalizeBetweenArrays(RG, "vsn", strata = rep(1:2,
+   each = nrow(RG)/2))
```

- Build an `ExpressionSet` object from MA:

```
> library(convert)
> M <- as(MA, "ExpressionSet")
> sampleNames(M) <- targets[["Cy3"]]
> featureNames(M) <- MA[["genes"]][["ID"]]
```

With the expression set read from disk or created as above, remove rows with suspect descriptions:

```

> metaD <- read.csv(file.path(dataDir, "PEDB_ARRAY_annotations.csv"),
+   as.is = TRUE)
> dropRows <- metaD[["Description"]] %in% c("EMPTY",
+   "Failed Sequencing")
> M <- M[!dropRows, ]

```

## 1.2 Model matrix

Create a model matrix describing the experimental design. Here is the necessary code:

```

> X <- model.matrix(~SFN * HGF, exptlDesign)

```

- What are the names of the columns of  $X$ ?
- Interpret each column in terms of how the entries influence the inclusion of specific effects. For any row in the model matrix, how would you know whether the corresponding sample included the High treatment of SNF?
- Interpret the interaction terms of the model matrix. When would you expect to see a 1 to indicate the presence of an interaction?
- From `exptlDesign`, determine how many replicates of each treatment are present. Is there enough data to estimate an interaction term?

Consider the model matrix created from including just cell line in the analysis:

```

> XcellLine <- model.matrix(~cellLine, exptlDesign)

```

- Interpret the columns of this model matrix, e.g., what is the ‘standard’ that samples are being compared to? What does it mean when there is a 1 in the second or third column of the matrix?

## 1.3 Model fit and differentially expressed genes

As in the lecture, we’ll first fit a simplified model that ignores `cellLine` and uses only one of each duplicate spot. Here is the code to select just the first half of the data:

```

> Mh <- M[1:(nrow(M)/2), ]

```

*limma* takes a two-step approach to fitting models: the actual fit (`lmFit`), and application of methods to assess evidence for differential expression (`eBayes`). Here is the code to perform an initial analysis:

```
> fit <- lmFit(Mh, X)
> fitE <- eBayes(fit)
```

Notice that the model matrix is included as part of `lmFit`, and that arguments in `eBayes` that we do not use influence how much correction is involved.

- Use `class` to figure out the class of `fit`, and consult the help page for this class (hint: `class?MArrayLM`) to figure out what sorts of information you now have available. Do the same for `fitE`.
- Explore the first few columns of the `coefficients` element of `fit`. What are these coefficients?
- Consult the help page `?decideTests` and use this method to summarize the results of your analysis. How many features are significantly over-expressed in the SFN High treatment?

```
> summary(decideTests(fitE))
```

	(Intercept)	SFNHigh	HGFHigh	SFNHigh:HGFHigh
-1	1858	0	0	0
0	2500	6598	6598	6598
1	2240	0	0	0

## 1.4 Fitted and residual values

A model can always be fit, how do we know if the fit is any good? We start by looking at fitted and residual values.

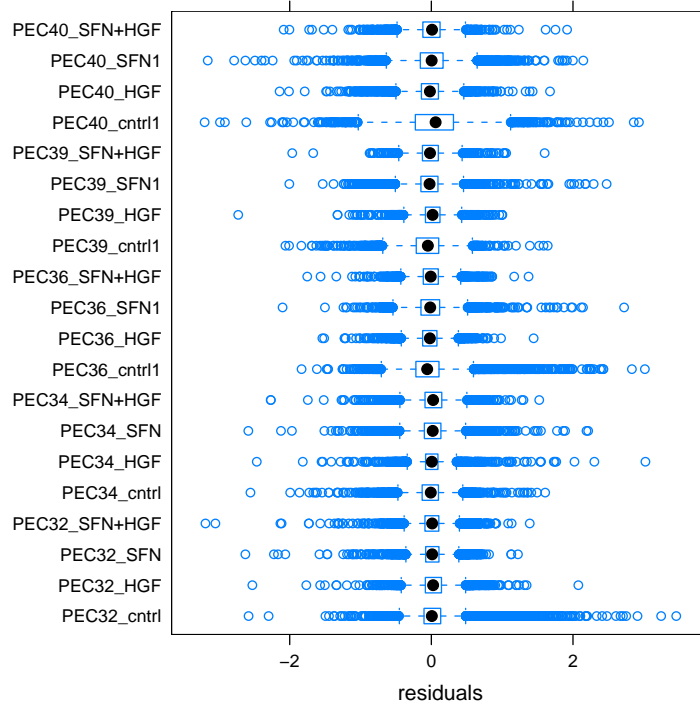
- Consult the help page `?fitted.MArrayLM` and `?residual.MArrayLM`. Calculate a matrix of fitted and of residual values. What do the rows and columns represent?
- What is a fitted and a residual value?
- For a gene that is differentially expressed, would you expect fitted values to differ between levels of a treatment? What about residuals?

The following code builds a data frame with a column for observed, fitted, and residual expression values, and additional columns summarizing aspects of the experimental design; because the fitted and residual values are based on duplicated spot numbers, we average the duplicates

```
> Mh <- (exprs(M)[1:(nrow(M)/2), ] + exprs(M)[1:(nrow(M)/2),  
+      ])/2  
> observedVals <- as.vector(Mh)  
> fittedVals <- as.vector(fitted(fitE))  
> residualVals <- as.vector(residuals(fitE, Mh))  
> df <- data.frame(sampleId = rep(sampleNames(M),  
+   each = nrow(Mh)), uniqueId = featureNames(M)[1:(nrow(M)/2)],  
+   SFN = rep(exptlDesign[["SFN"]], each = nrow(Mh)),  
+   HGF = rep(exptlDesign[["HGF"]], each = nrow(Mh)),  
+   observed = observedVals, fitteds = fittedVals,  
+   residuals = residualVals)
```

Here is a plot summarizing residuals by sample:

```
> library(lattice)  
> print(bwplot(sampleId ~ residuals, df))
```



- Remembering that residuals should be independent of the sample from which they are from, is there any obvious indication from the residuals that the fit is somehow inadequate?
- Can you plot, for a subset of genes, the observed expression level of individual samples? Does this plot look consistent with the results of `decideTests`?

## 2 Part II: Technical replicates

The main novelty explored in this part of the lab is to include technical replicates. We also include cell lines in the model.

### 2.1 Expanded model matrix

We start with a simple modification: including cell lines as part of the experimental design. This represents a change to the formula describing the experiment, and hence to the model matrix:

```
> Xfull <- model.matrix(~cellLine + SFN * HGF, expt1Design)
```

- Although our formula includes `cellLine`, it does not include the interaction between `cellLine` and the other factors. Why is that?
- Investigate `Xfull` as you did in the first part of the lab. What does a 1 represent in a column corresponding to a cell line? to the `SFN` factor? For an interaction?
- The formulation in `Xfull` treats cell lines as so-called **fixed effects**, implying that they are somehow inherently interesting (we want to say something explicit about, say, cell line 32). In reality, though, these are meant to represent random samples from a population of individuals, and hence are **random effects**. As an advanced exercise, use the R package `nlme` to fit an appropriate model to the data. Start with just a single spot, and then include duplicate correlations and comparison of all probes in the analysis. In which ways does this analysis differ from the one performed by `limma`?

## 2.2 Incorporating duplicate spots

A classical ANOVA approach to technical replicates would include the repeated measures as part of the model matrix. As noted in the lecture, `limma` takes a different approach, using shared correlations across spots within an array to get an overall consensus estimate. The central functionality for this is as follows:

```
> dupCor <- duplicateCorrelation(exprs(M), design = Xfull,  
+   ndups = 2, spacing = nrow(M)/2)  
> dupFit <- lmFit(M, design = Xfull, ndups = 2,  
+   spacing = nrow(M)/2, correlation = dupCor[["consensus.correlation"]])  
> dupFitE <- eBayes(dupFit)
```

- Use the help page for `duplicateCorrelation` to understand the arguments provided. What additional data is available in `dupCor`?
- Notice the similarity of the `lmFit` and `eBayes` function calls to the version without duplicate correlations.

The general sense is that it is relatively easy to fit increasingly complicated models!

### 2.3 Results and their interpretation

The moment of truth! What are the consequences of including duplicated spots and cell lines for our results?

- Use `decideTests` to summarize significance of results so far. How many genes increase expression in the presence of SFN? How many decrease expression?
- Use the techniques and plotting methods of the previous part of the lab to investigate residuals in terms of samples. Is there any indication of problems here?
- The object `dupDecided`, below, is a matrix with 0, -1, or 1 indicating which features and coefficients are different, and how. Use this information to produce xy-plots, like those in the first part of the lab, of the expression levels from each sample for the genes showing differential expression with respect to SFN. Are these plots consistent with the statistical analysis?
- Graphically investigate the relationship between residuals and SFN treatment level. Are there any obvious causes for concern here?

```
> summary(dupDecided <- decideTests(dupFitE))
```

```
(Intercept) cellLine34 cellLine36 cellLine39 cellLine40
-1          1701         54         495         278         649
0           2762        6466        5942        6175        5392
1           2135         78         161         145         557
  SFNHigh HGFHigh SFNHigh:HGFHigh
-1         52         0             0
0        6519        6598            6598
1         27         0             0
```

Finally, how would you proceed to identify the genes associated with differentially expressed (e.g., up regulated) features?