Statistics with R for Biologists

James H. Bullard
Kasper Daniel Hansen
Margaret Taub

Berkeley, California
July 7-11, 2008

Case Study: Smith et. al. Gene-Environment Interaction in Yeast Gene Expression

Background

- In Smith et. al. the authors wish to assess the effects of yeast strain (gene) and condition (environment) on the phenotype gene expression.
- The authors have hybridized: 2 (strain) * 2 (condition) * 2 (dye). The have replicated this 3 times for a total of 24 hybridizations.
- All hybridizations were done using two-color 11k Agilent yeast arrays. All samples were hybridized against a common reference sample.
- Data was pre-processed using Agilent software to perform quality control (outlier removal) leaving a total of 4,342 "high-quality" transcripts for the "parental analysis."

"Parental Analysis"

We will focus exclusively on the first portion of their analysis. The question they wish to answer is: what genes show significant strain-condition interaction? They want to determine which genes are better described by the model:

$$\text{phenotype} \sim \text{dye} + \text{strain} + \text{condition} + \text{strain} * \text{condition}$$

As compared to:

$$\text{phenotype} \sim \text{dye} + \text{strain} + \text{condition}$$

Getting Started

Example

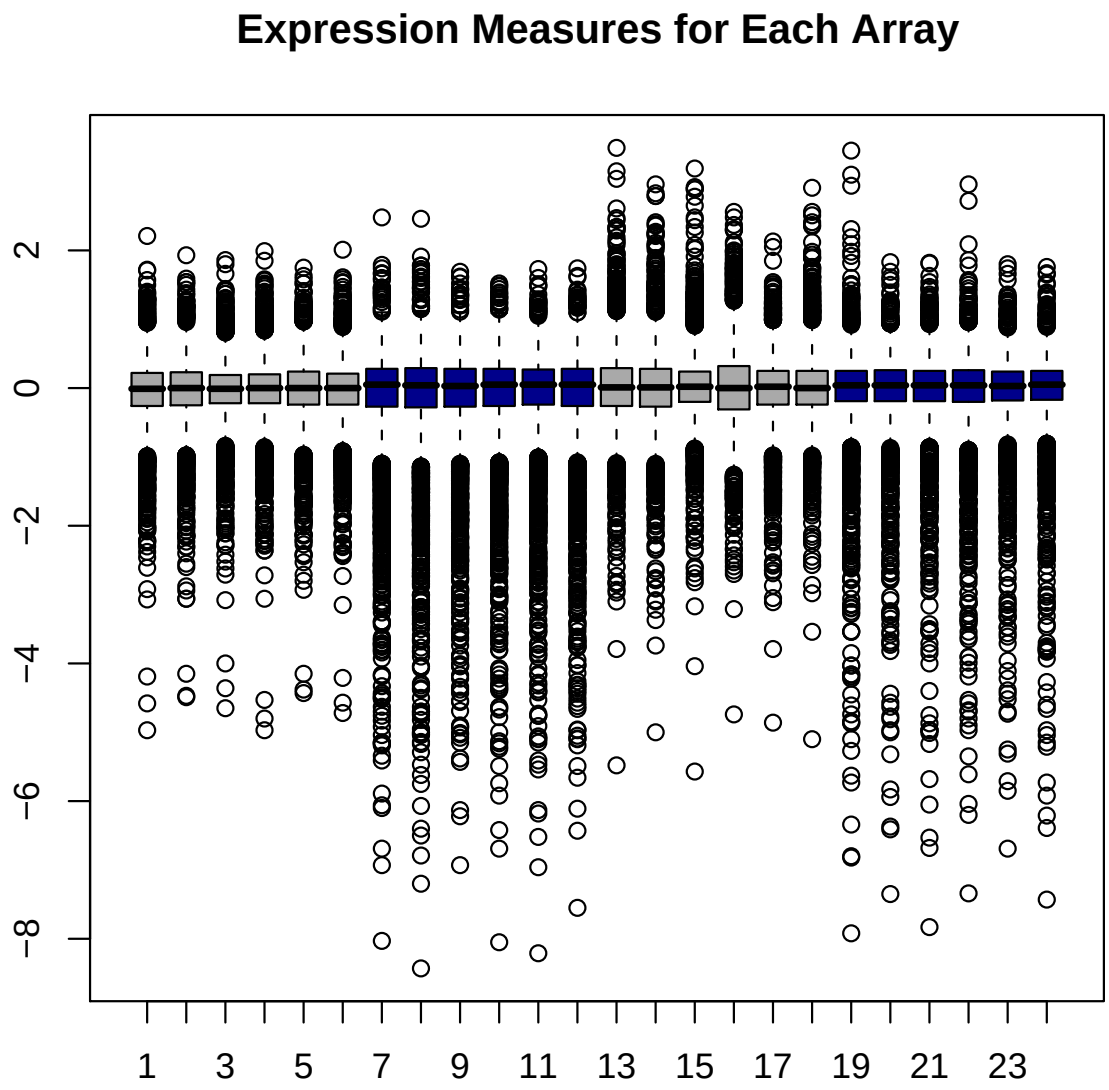
We first want to read in the data and convince ourselves that we have the same data that they have used to conduct their analysis. We have provided two .csv files from the paper to use in reproducing/understanding their analysis. The files are: smith_et_al_data.csv, smith_et_al_pvals.csv. First, read in smith_et_al_data.csv and have a look at the data. We want a matrix containing the expression measures with the row names equal to the gene names and the columns a different factor-level combination. First, make a boxplot for each microarray of all expression measures. For fun do a t-test comparing the means for each gene within a particular condition – which gene has the largest t-statistic? what does this mean? How many numbers contribute to each t-statistic? How many are "significant"? (some useful functions: expand.grid, grep)

T-tests

	glucose	ethanol
FALSE	1189	1963
TRUE	3153	2379

Table 1: Number of genes reported as differentially expressed between strains at the 5 % cutoff.

T-tests

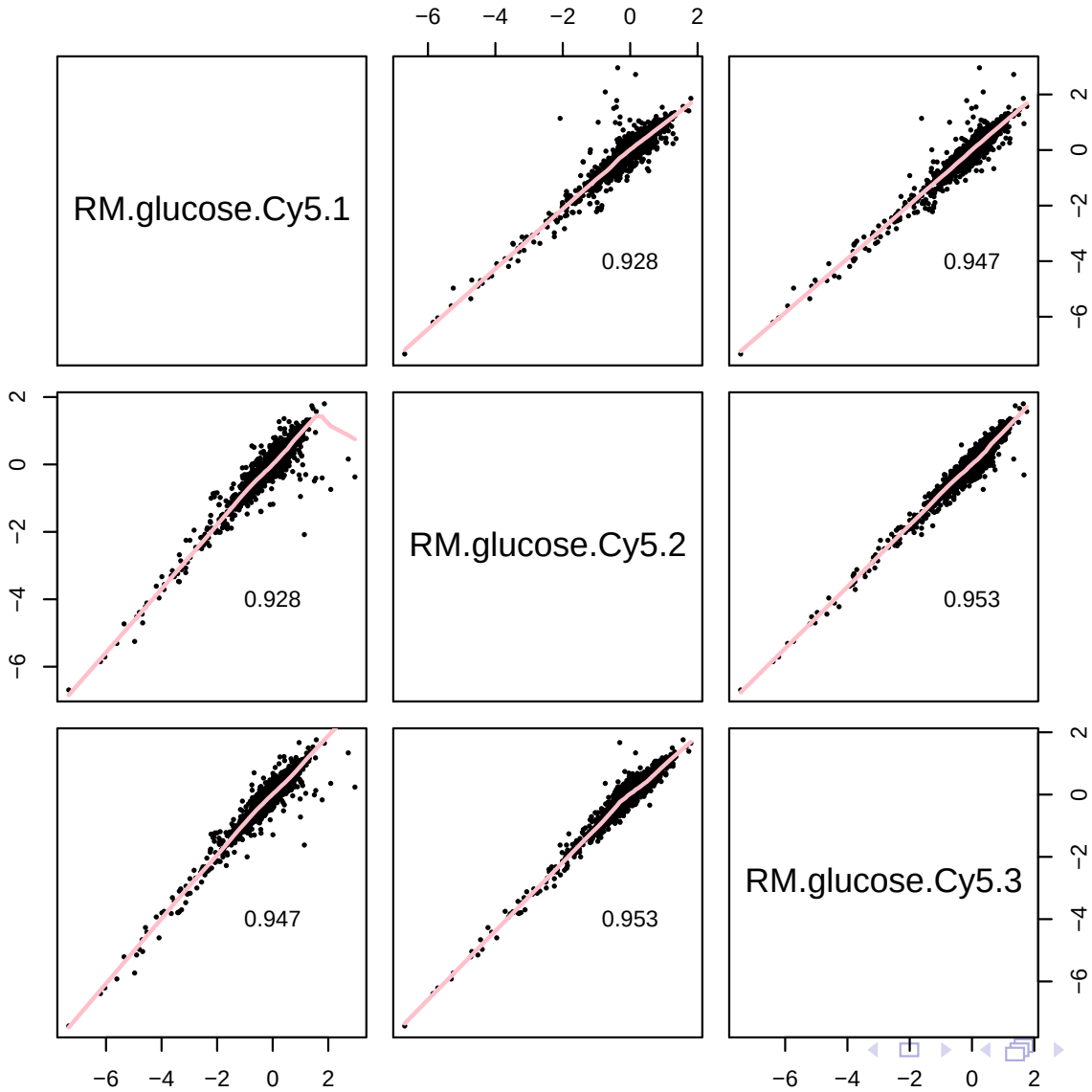

Expression Measures for Each Array
2
0
−2
−4
−6
−8
1
3
5
7
9
11
13
15
17
19
21
23

Visualization

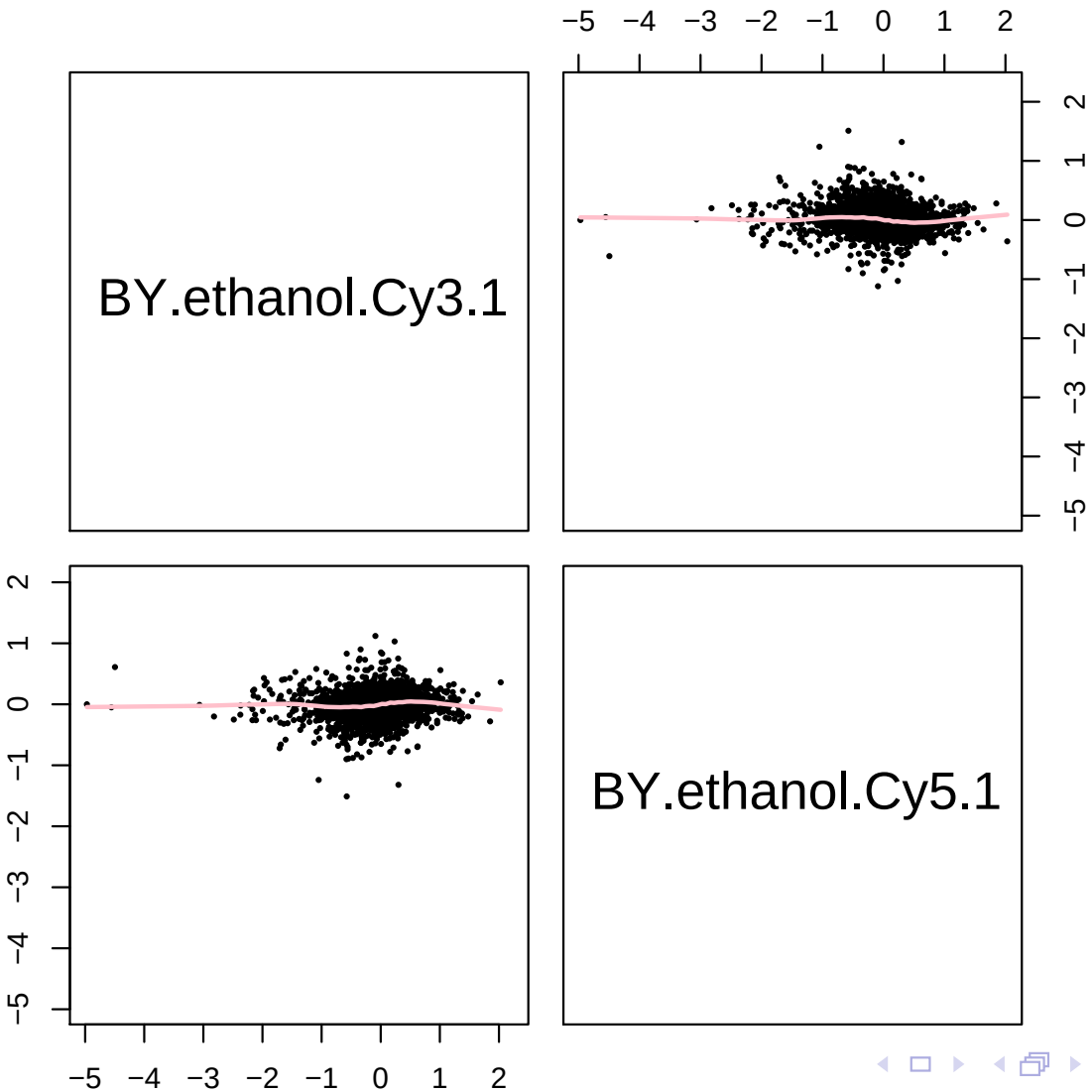
Example

1. Create a pairs plot for each of the 8 sets of three replicates.
2. Create mean difference plots comparing the replicate experiments as well as dye-swaps.
3. Create image plots of the microarray data sets (this will be an image plot where the microarrays are the columns and the genes are the rows – think about good labeling.)

Visualization



Mean-Difference Plots



Experimental Design

- In this experiment we are primarily interested in the effects of strain and condition on phenotype (gene-expression).
- In addition, we have to worry about the effect of dye on transcription (why?).
- This experiment is an example of a complete balanced design, where each factor-level occurs as at least once.
- We might imagine a simpler experiment where we have only 8 or 16 microarrays and 1 and 2 less replicates. What do the replicates give us?

Experimental Design


```
, , dye = Cy3

      condition
strain ethanol glucose
    BY       3       3
    RM       3       3

, , dye = Cy5

      condition
strain ethanol glucose
    BY       3       3
    RM       3       3
```

Statistical Models

- How do we "represent" our scientific question as a statistical question?
- We are frequently interested in the dependence of an outcome (phenotype) on a number of predictors. In this case we are interested in the effect of the predictors (strain, condition) on our phenotype and we can represent an additive dependence in the following way:

$$\text{phenotype} \sim \text{dye} + \text{strain} + \text{condition}$$

We can write this more explicitly as:

$$\text{phenotype}_j = \beta_0 + \beta_1 \text{dye}_j + \beta_2 \text{strain}_j + \beta_3 \text{condition}_j + \epsilon_j$$

Statistical Models

Here j ranges over the subjects in our experiment; the independent observational units. In our current example we have 24 expression measures (phenotype) each expression measure was obtained from an experiment conducted at a particular dye, strain, condition combination. If I tell you that dye $= 1$, strain $= 0$, and condition $= 1$, what is the phenotype?

- Here we are explicitly stating that phenotype depends on dye, strain, and condition in an additive fashion. In this model we can interpret the βs in a relatively straightforward fashion. "After fitting our model we found the value of β_2 to be equal to 4 this means that by flipping strain from 0 to 1 we can (increase | decrease) gene expression (1 | 2 | 3 | 4 | 8) times."

Statistical Models

- Why do we care about dye? What did the additivity do as I have the model stated above? What about how the model was stated in Smith et. al.?

The Linear Regression Model

- The linear regression model is one of the most common, if not, the most common way of modeling data.
- In many cases the model is not "correct" but is often very reasonable.

$$Y = X\beta + \epsilon \tag{1}$$

The linear regression model is composed of an $n \times p$ design matrix (X), an $n \times 1$ vector of outcomes (Y), a $p \times 1$ vector of parameters which we wish to estimate (generally denoted β). Linear regression finds the estimate $\hat{\beta}$ which minimizes the L_2 loss (equation: (2)).

$$L_2(\beta) = \sum_{i=1}^{n} (Y - X\beta)^2 \tag{2}$$

The Linear Regression Model

Under the following assumptions linear regression is the best linear unbiased estimator of β.

i. X and Y satisfy equation (1).

ii. The disturbance terms ϵ_i are i.i.d with mean 0 and variance σ^2.

iii. X and ϵ are independent.

Linear Models

- Statistical models in R have a special syntax (the formula syntax):

```
Y ~ X
```

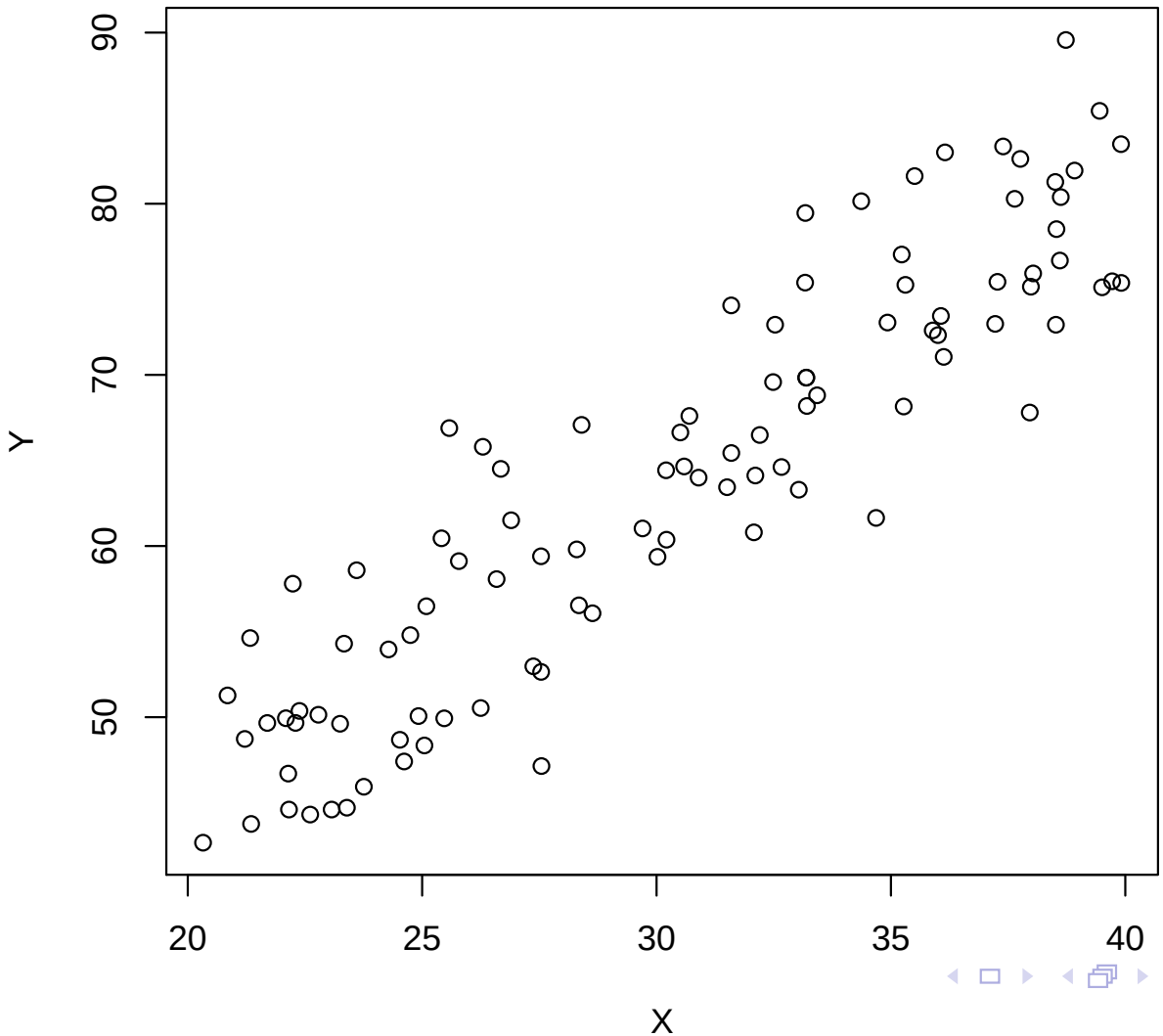
- This says that the variable Y is related to X. The formula specification is used in a variety of functions as input and depending on that function different relationships between the predictor variables (X) and the outcome variables (Y) are assumed/modeled.

Formulas Continued

We'll construct a tiny example to see how the pieces fit together in the model.

```
> N <- 100
> X <- runif(N, 20, 40)
> Y <- 3 + 2 * X + rnorm(N, mean = 0,
+     sd = 5)
> plot(Y ~ X)
```

Formulas Continued



Formulas Continued

Now suppose we would like to fit a linear model to the data. In R this is as simple as:

```
> myFit <- lm(Y ~ X)
```

1. What is the class of `myFit`?
2. How can we extract the estimates: $\hat{\beta}$ from this object?
3. What are the functions which are specialized for this class (hint methods)?

```
> coefficients(summary(myFit))
```

```
             Estimate Std. Error
(Intercept) 8.102548 2.62603974
X           1.838198 0.08486113
              t value     Pr(>|t|)
(Intercept)  3.085463 2.641767e-03
X           21.661246 3.820746e-39
```

Formulas: Syntax

- A comprehensive reference of how to specify different formulas in R can be found at: formulas

```
Y ~ M
    Y is modeled as M.
M_1 + M_2
    Include M_1 and M_2.
M_1 - M_2
    Include M_1 leaving out terms of M_2.
M_1 : M_2
    The tensor product of M_1 and M_2. If both terms
    are factors, then the subclasses factor.
M_1 * M_2
    M_1 + M_2 + M_1:M_2.
```

Formulas: Syntax

`I(M)`

> `Insulate M. Inside M all operators have their normal arithmetic meaning, and that term appears in the model matrix.`

This was lifted right from that page!

Understanding Model Formulas: Factors

- Parameterization of the model can be quite tricky. Here we need to understand what happens with the factors dye, strain, and condition in the formula in order to fully appreciate what the βs represent. We just want to skim the surface here. Lets step back and look at a simple example.
- Does this code work:

```
> genotype <- sample(c("AA", "AB",
+     "BB"), size = 100, replace = TRUE)
> cholesterol <- 160 + 3 * genotype +
+     rnorm(100)
```

- So what I really need to do is the following:

Understanding Model Formulas: Factors


```
> genotype <- sample(c("AA", "AB",
+     "BB"), size = 100, replace = TRUE)
> genotype <- factor(genotype)
> designMatrix <- model.matrix(~genotype,
+     data = genotype)
> head(designMatrix)
```

	(Intercept)	genotypeAB	genotypeBB
1	1	0	1
2	1	1	0
3	1	0	1
4	1	0	0
5	1	1	0
6	1	0	1

Understanding Model Formulas: Factors

- What we have done is convert the "factors" into "dummy" variables so that we can do some matrix algebra on them. What happened to genotype AA?
- Now we can simulate some data quite simply:

```
> cholesterol <- designMatrix %*%
+      c(160, -40, -20) + rnorm(100,
+      sd = 10)
```

- In summary, when we have factors we code them as dummy variables and we drop one of the levels – this level becomes the baseline which we compare the resulting coefficients against. In the example above having genotype BB makes your cholesterol how much higher than having genotype AA?

Understanding Model Formulas: Factors

- We can choose which factor gets dropped by reordering the levels:

```
> levels(genotype) <- c("BB", "AB",
+     "AA")
```

- Additionally, if we fit the model with: -1 then we maintain all of the factors - in this simple example the coefficients βs will represent the genotype means, when we keep the intercept in then the βs represent the change from moving from the excluded genotype (AA) to a particular genotype.

Fitting the Model

- The next step is that we want to "fit" the model, this means we want to estimate the parameters.
- Again, we fit the model using least squares.

```
> lm(cholesterol ~ genotype)

Call:
lm(formula = cholesterol ~ genotype)

Coefficients:
(Intercept)   genotypeAB   genotypeBB
     160.54       -41.29       -20.31

> lm(cholesterol ~ genotype - 1)
```

Fitting the Model


```
Call:
lm(formula = cholesterol ~ genotype - 1)

Coefficients:
genotypeAA  genotypeAB  genotypeBB
     160.5       119.2       140.2
```

Assessing Model Parameters: Significance

- The next step is to decide whether or not a parameter is a good "predictor" of our outcome. At this point we have to discuss "statistical significance."
- In our cholesterol example the two genotypes are wildly significant – what does this mean?

```
> round(coefficients(summary(lm(cholesterol ~
+     genotype)))[, -(1:2)], 200)
```

	t value	Pr(>|t|)
(Intercept)	99.662125	1.606217e-99
genotypeAB	-18.539477	1.194242e-33
genotypeBB	-9.181818	7.839062e-15

Assessing Model Parameters: F-statistic

In the summary matrix of the model fit we have a p-value and a t-statistic for each term in the fit. This t-statistic is for the test that the coefficient in front of term is 0. It is often much more sensible to test a block of variables together. In this example, what would it mean to accept the test that $\beta_{AB} = 0$, but reject the test $\beta_{BB} = 0$? What we are interested in testing then is: $\beta_{AB} = \beta_{BB} = 0$ if we reject this test then we keep the terms in the model, otherwise we might conclude: "there is not sufficient evidence to reject the null hypothesis that genotype has an affect on cholesterol."

We can compute this using the following R functions: aov or anova on the model fit (the object returned by lm, try summary on the fits.

Assessing Model Parameters: F-statistic


```
             Df  Sum Sq Mean Sq F value
genotype      2 27685.7 13842.9  172.09
Residuals    97  7802.6    80.4
             Pr(>F)
genotype    < 2.2e-16
Residuals
```

Back to The Smith et. al. Dataset

$$\text{phenotype}_j = \beta_0 + \beta_1 \text{dye}_j + \beta_2 \text{strain}_j + \beta_3 \text{condition}_j + \epsilon_j$$

- What do each of the β coefficients represent?
- What kind of variables are dye, condition, and strain?

```
    strain condition dye
V2      BY   ethanol Cy3
V5      BY   ethanol Cy5
V8      BY   glucose Cy3
V11     BY   glucose Cy5
V14     RM   ethanol Cy3
V17     RM   ethanol Cy5
V20     RM   glucose Cy3
V23     RM   glucose Cy5
```

Assessing Model Parameters: Simulation

Now we want to simulate data to solidify some of the concepts above. First, we need some "predictor" variables. We are going to use the design used in Smith et. al. Here we can use the function model.matrix to help us generate the data. After constructing the design matrix we are going to use this to generate some outcome variables. Use the following formula:

$$\text{phenotype} = 1 + \beta_{strain}\text{strain} + \beta_{condition}\text{condition} + \epsilon \quad (3)$$

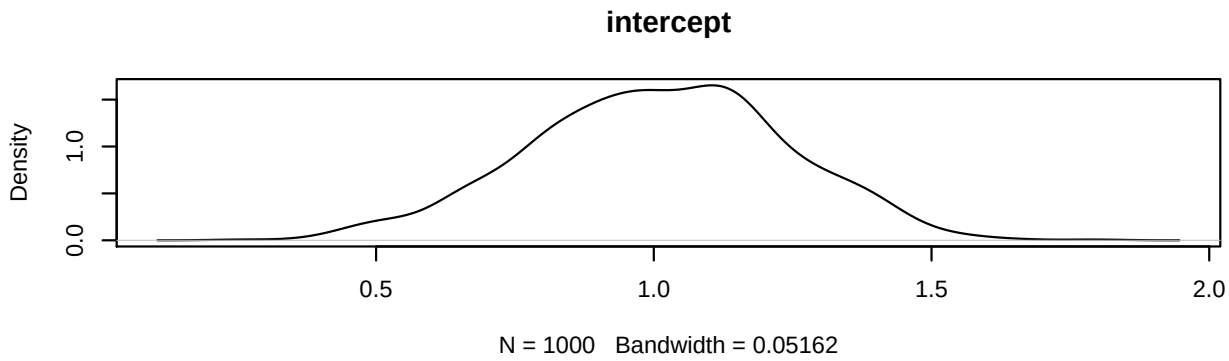
$\beta_{strain} = -.5$, $\beta_{condition} = -.95$, and $\epsilon \sim N(0, .5)$ to start. We will want to change our error distribution after we get the hang of it, but for now lets keep it simple. After we have constructed a data set, fit the model:

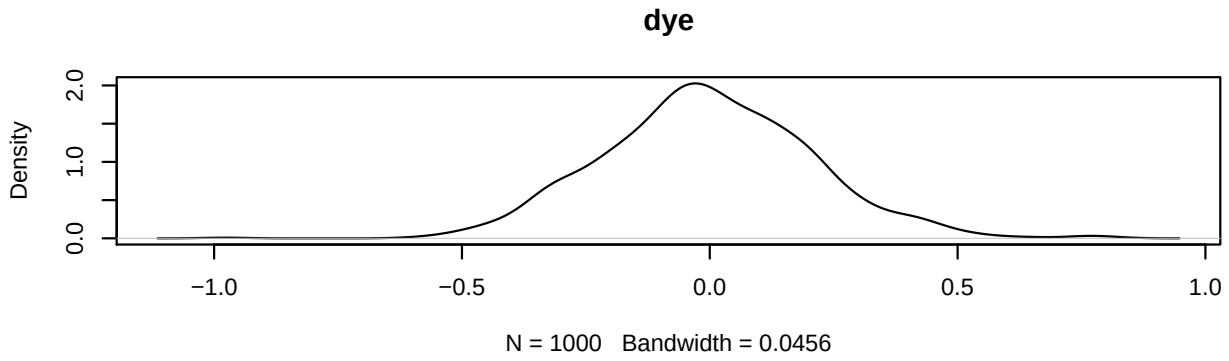
$$\text{phenotype} \sim \text{dye} + \text{strain} * \text{condition} \quad (4)$$

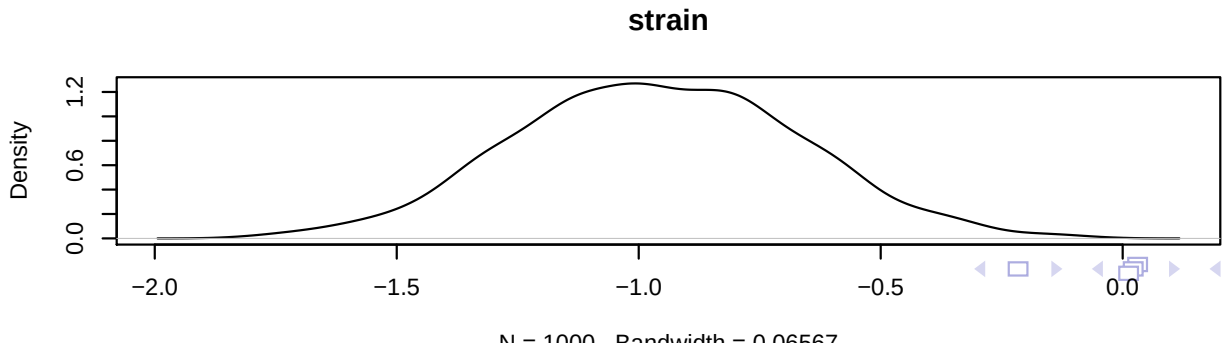
Assessing Model Parameters: Simulation

Interpret the output. After you have fit the model one time on your simulated data set we want to generate 1000 data sets and fit the model on each of these data sets. This should help us understand some of the assumptions and results of the linear model.

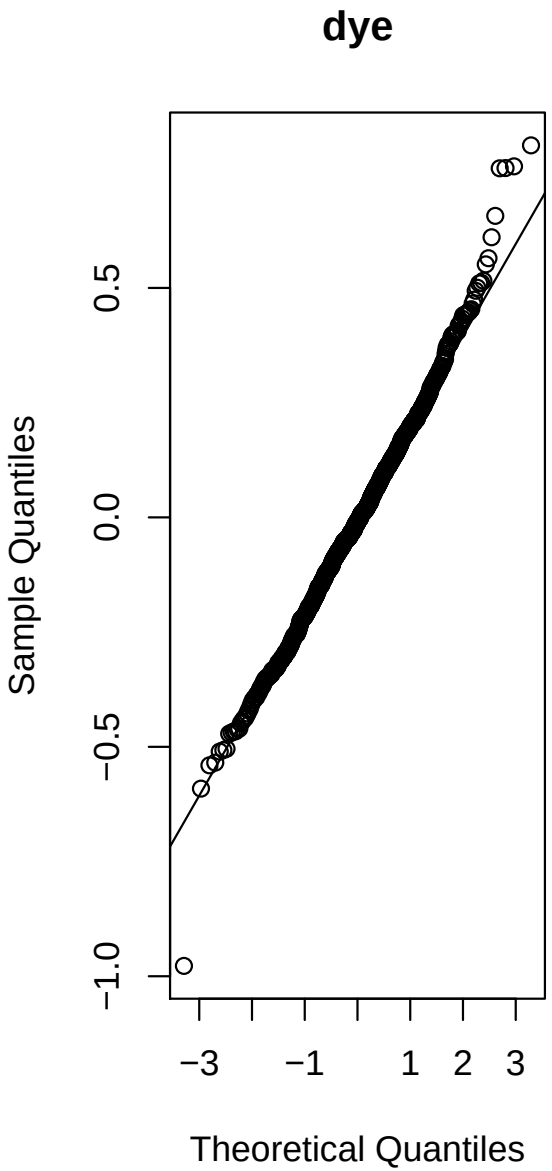
Distribution of our Estimates

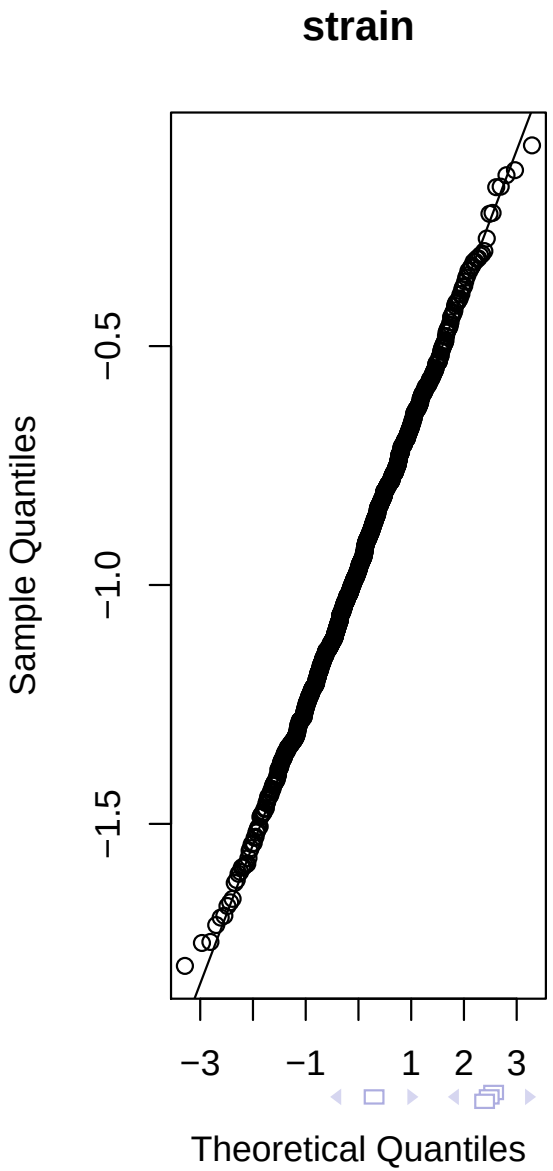




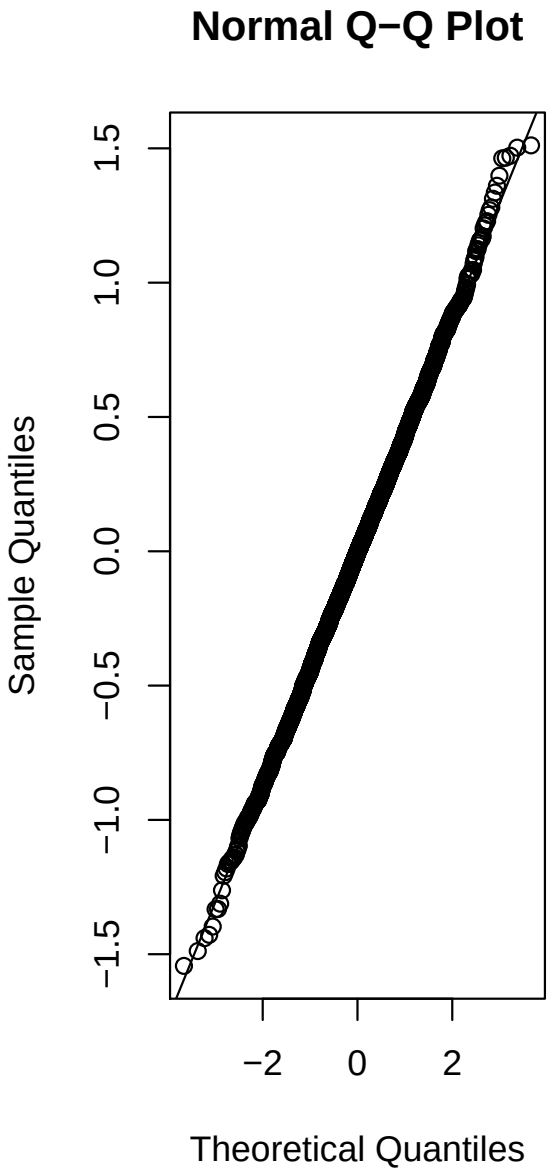


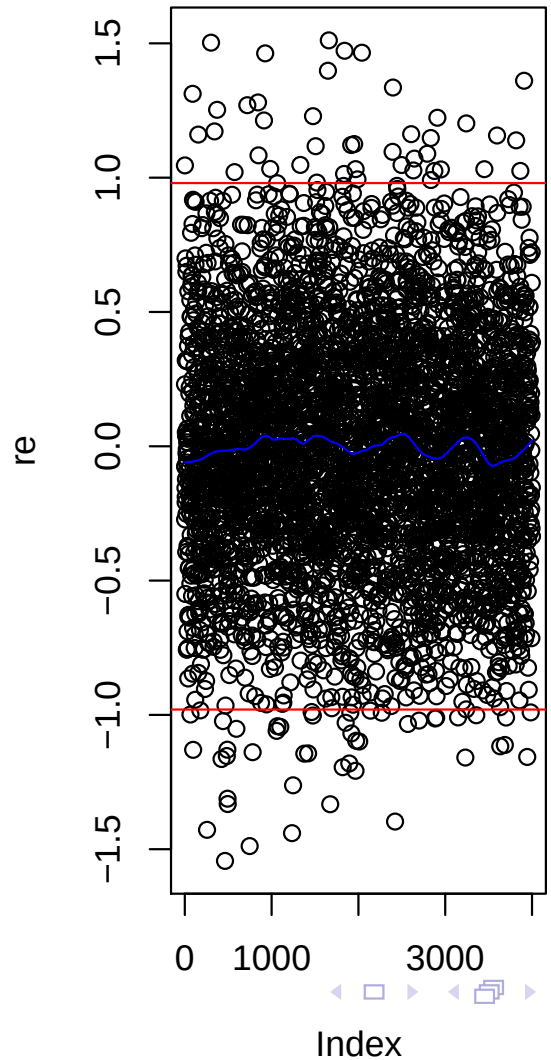
Distribution of our Estimates



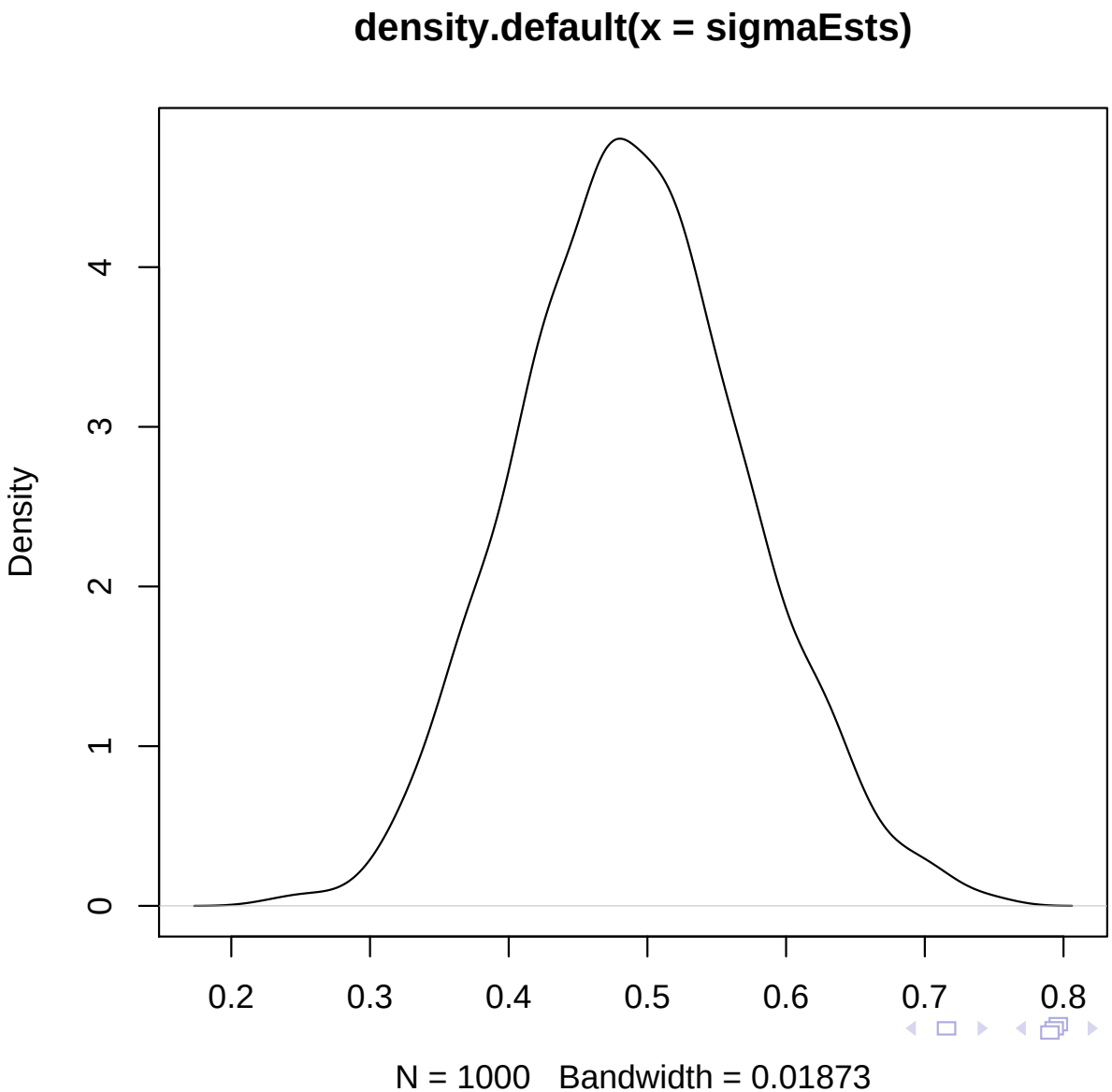


Residual Distribution

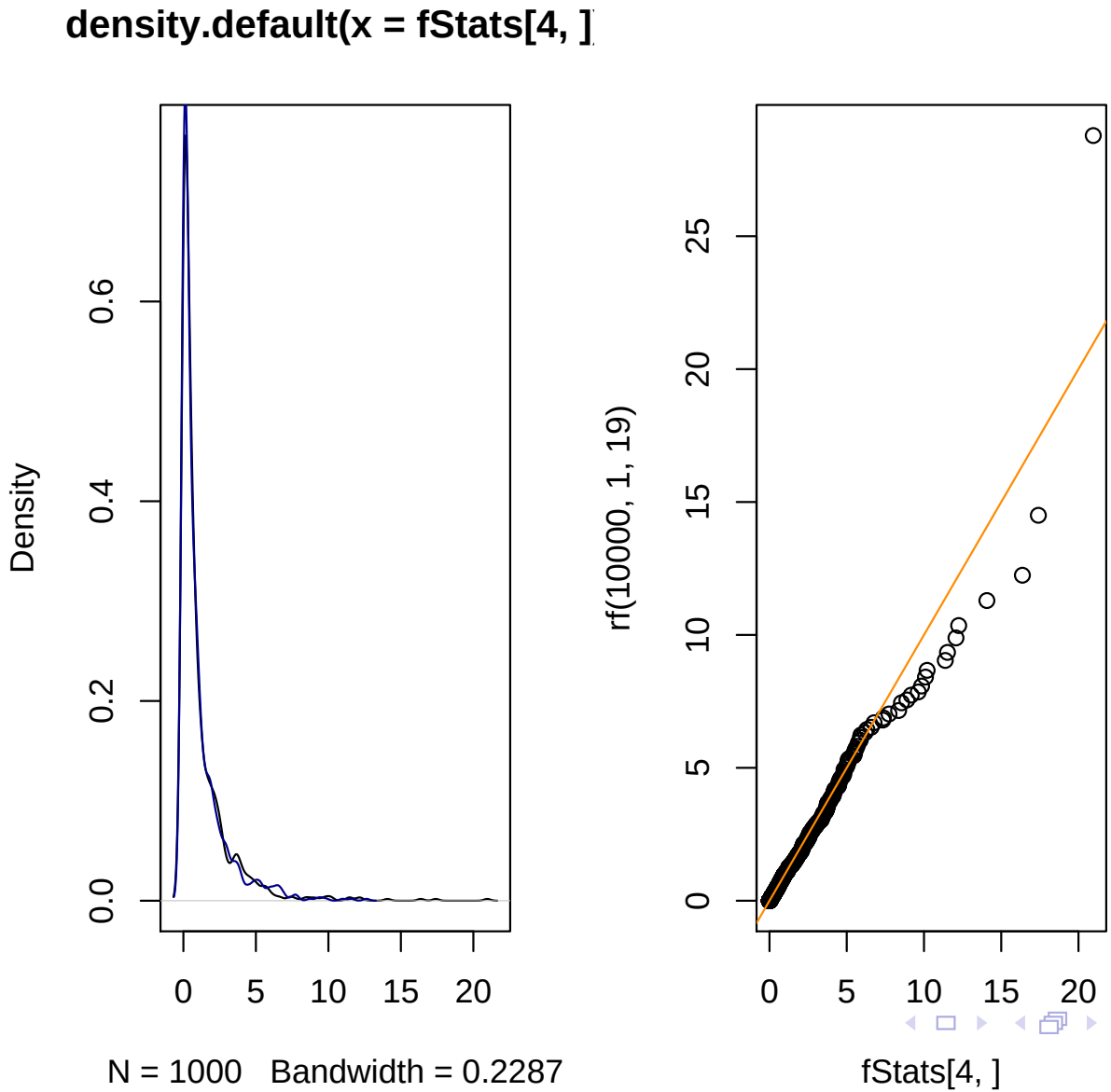




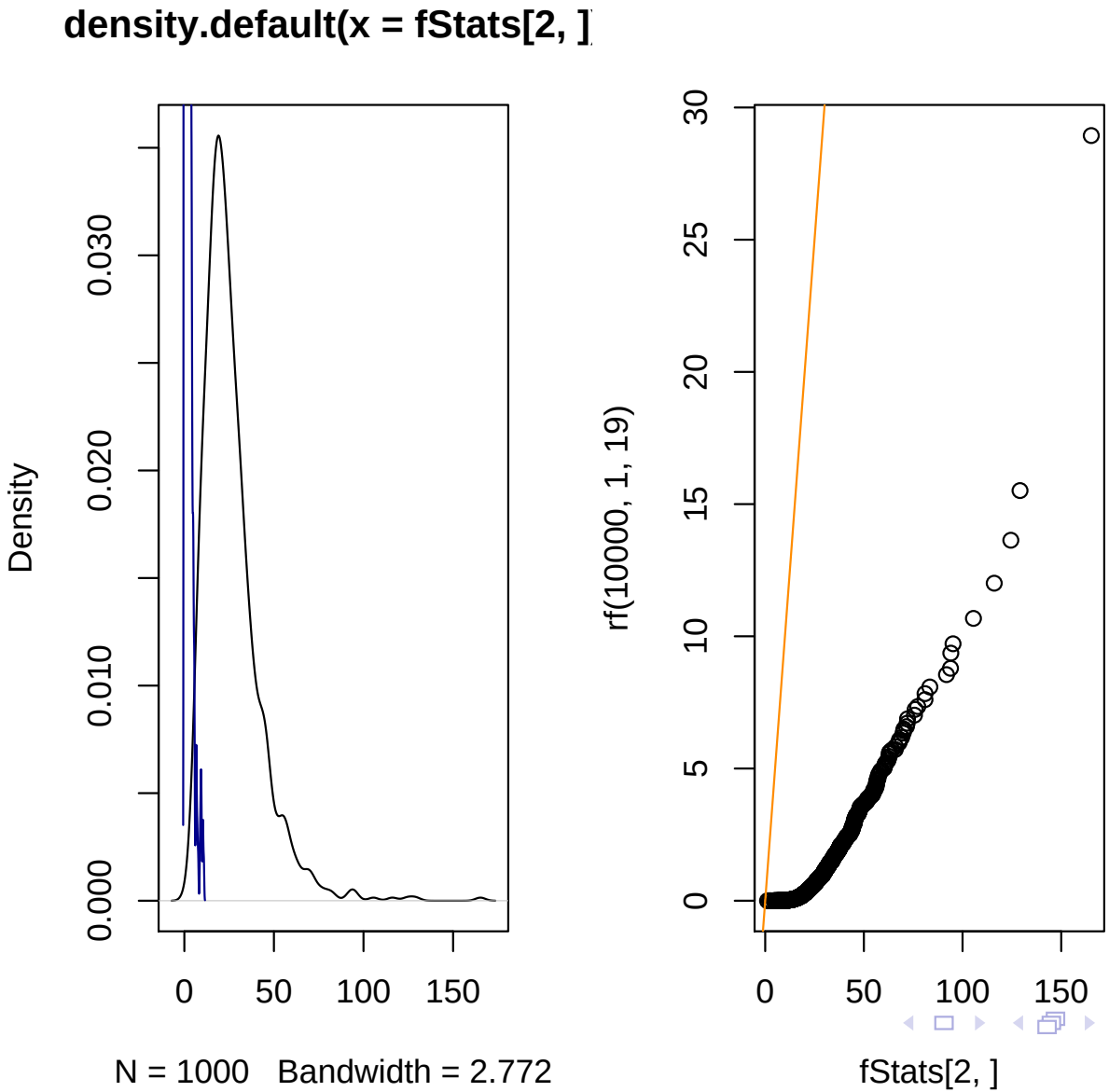
Estimating σ



Null Distribution: F, Interaction Term



Null Distribution: F, Strain Term



Testing

- How many false positives did we commit when we looked at the F-test for the inclusion of dye and the interaction in the model?

```
> sum(fStats[1, ] > qf(0.95, 1, 19))/1000

[1] 0.05

> sum(fStats[4, ] > qf(0.95, 1, 19))/1000

[1] 0.054
```

- How many false negatives did we commit when we tested whether or not the main effects of strain and condition were significant?

```
> sum(fStats[2, ] < qf(0.95, 1, 19))/1000

[1] 0.003
```

Testing


```
> sum(fStats[3, ] < qf(0.95, 1, 19))/1000
[1] 0.351
```

Go back and change the error distribution used to simulate the 1000 data sets. Choose something with larger variance, such as a T distribution with less than 6 degrees of freedom. If you have time then go back and generate the data with a "dye" effect and then exclude that term when you fit the model.

1. What happens to our estimates?
2. What happens to the distribution of our estimates?
3. What about the distribution of our residuals?
4. What about the distribution of our test-statistic (F-statistic)?
5. What happens to the p-values, do we commit more false positives and false negatives or fewer?

ANOVA

- Analysis of Variance models are linear models with categorical predictors.
- Our last example was an ANOVA model with three factors taking on two distinct levels each. Factors can have as many discreet levels as they want, but the more levels and factors the more data you want to estimate parameters.
- In the Smith et. al. paper the statistical model which they fit is given by:

$$\text{phenotype} \sim \text{dye} + \text{strain} * \text{condition} \tag{5}$$

  Here I have written the model in terms of R's notation rather than the notation in the paper.
- Is this the "full" model?

Permutation Tests

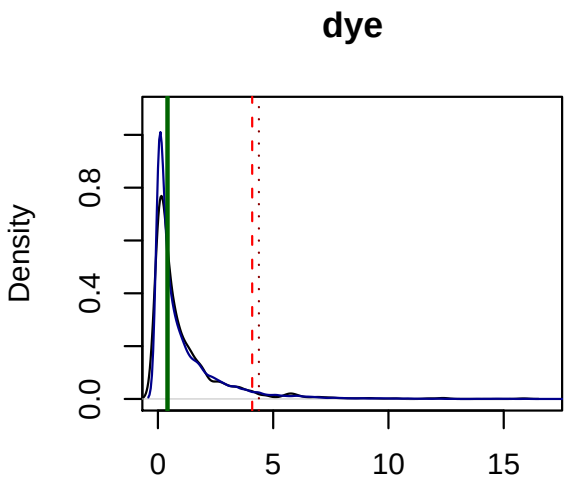
- In Smith et. al. they perform a permutation test instead of the F-test which we performed above - A permutation test allows us to construct the null distribution directly.
- As we saw above we found it relatively difficult to construct an example where the choice of an (independent) error structure induced lots of false positives.
- With a permutation test we are going to shuffle our predictors and then recompute an F-statistic, we are going to use the permutation distribution of F-statistics to test against.

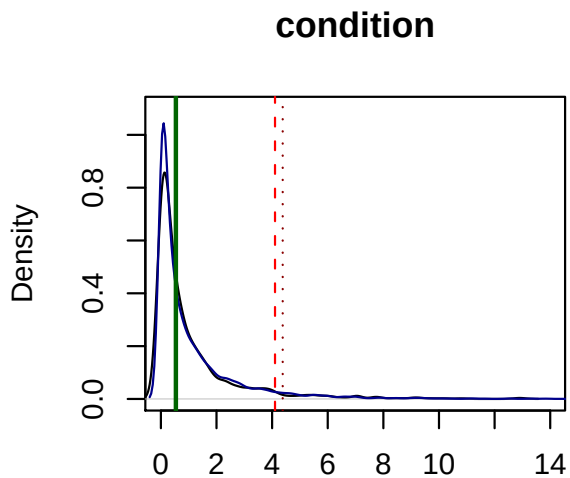
Permutation Tests

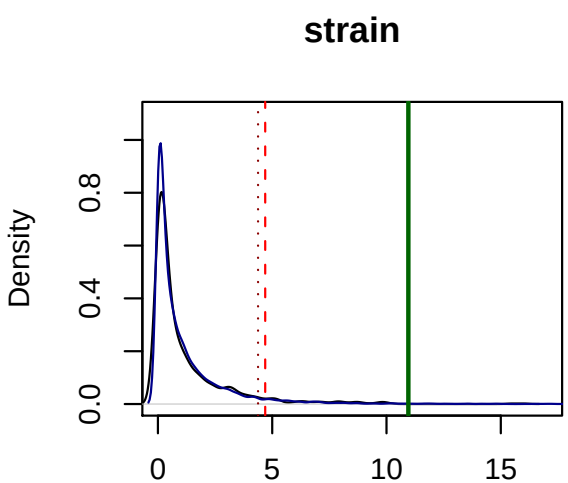
Example

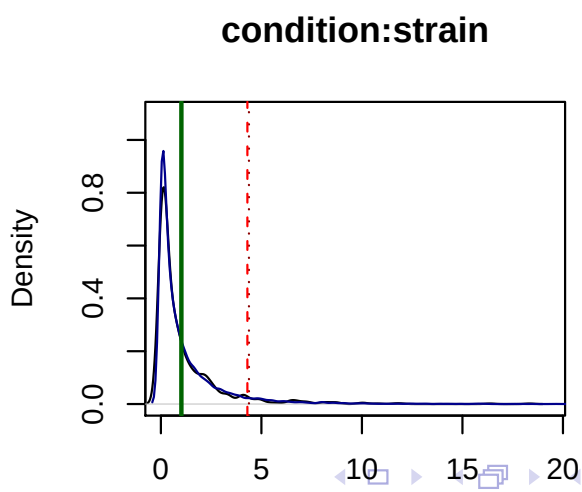
Using our "model.matrix" from above and normal errors simulate one data set. From this "simulated" data set construct a permutation null distribution for the F-statistics. Each F-statistic is a measure of how much evidence there is to include the term in the model as compared to the full model. Under the null distribution and some assumptions about the error distribution (ϵ is IID with normal errors and has mean 0) this F-statistic should be F distributed. After constructing a permutation distribution using the F-statistics test the observed F-statistic against this distribution.

Permutation Tests









Real Data Analysis

We now want to put all of the pieces together and redo the analysis done in Smith et. al. This comprises a number of steps and we want to try to do these in order.

- Read in the file "smith_et_al_pvals.csv" and then make sure you can reproduce the results from the paper, ie. how many genes were significant at the cutoffs they report? Now, compute a FDR controlled cutoff from the pvalues reported in this file. They reported a FDR cutoff of .03 can you recover this?
- Fit the ANOVA for the three genes in figure 1C. Make sure you get the same results. The genes names are: "YCR040W", "YDR343C", "YLR174W", ie. reproduce the plots.

Real Data Analysis

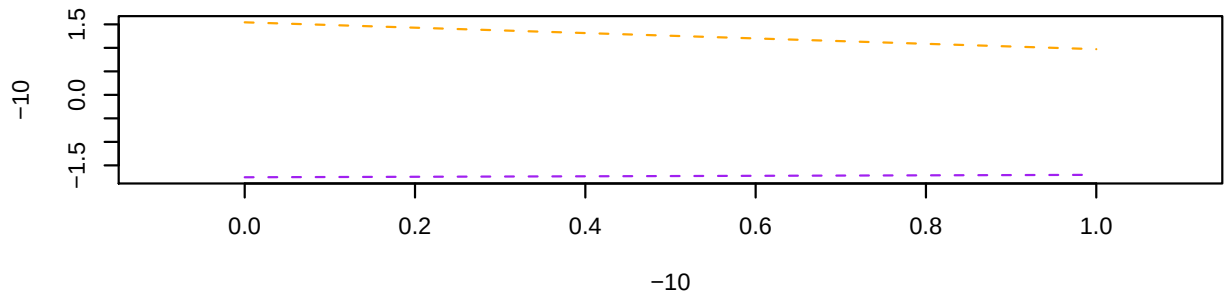
- Now fit the ANOVA for all genes. Compute an F-statistic using R for the test of whether the effect strain, condition, and strain*condition is 0. Compare your pvalues from this test with the pvalues they reported. Use an FDR correction for our pvalues and then make a table of their rejections and our rejections. (see aov and anova)
- Investigate the standardized residuals from each model fit and assess normality. Answer the question: How non-normal is my data? Look at the standardized residuals by microarray, do you notice anything?
- Finally, perform a permutation test on a subset of the genes. Choose a subset which contains genes we have accepted and rejected. See if the permutation test has changed our conclusion.

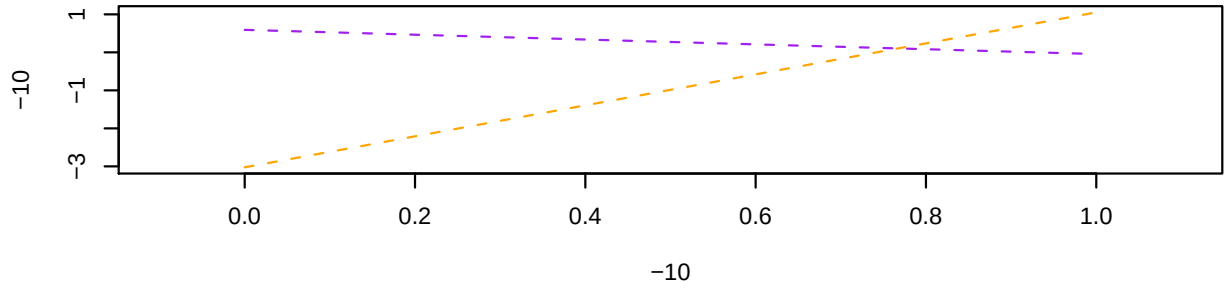
Recapitulation

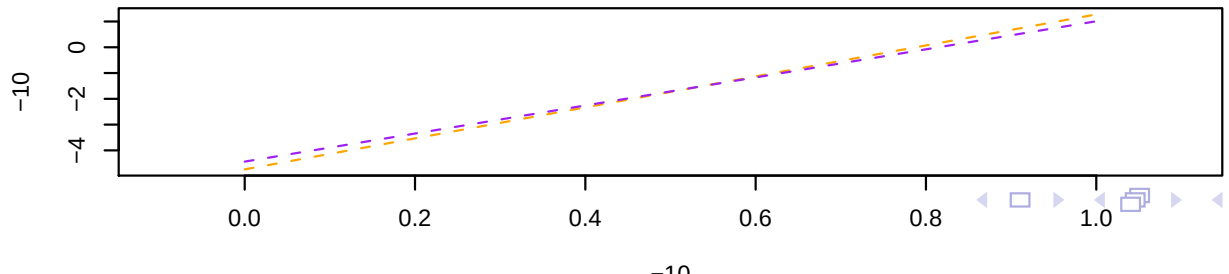
Here we produce the number of genes with significant interaction as in the paper - This demonstrates that we have probably read in the data correctly.

```
[1] 2037
```

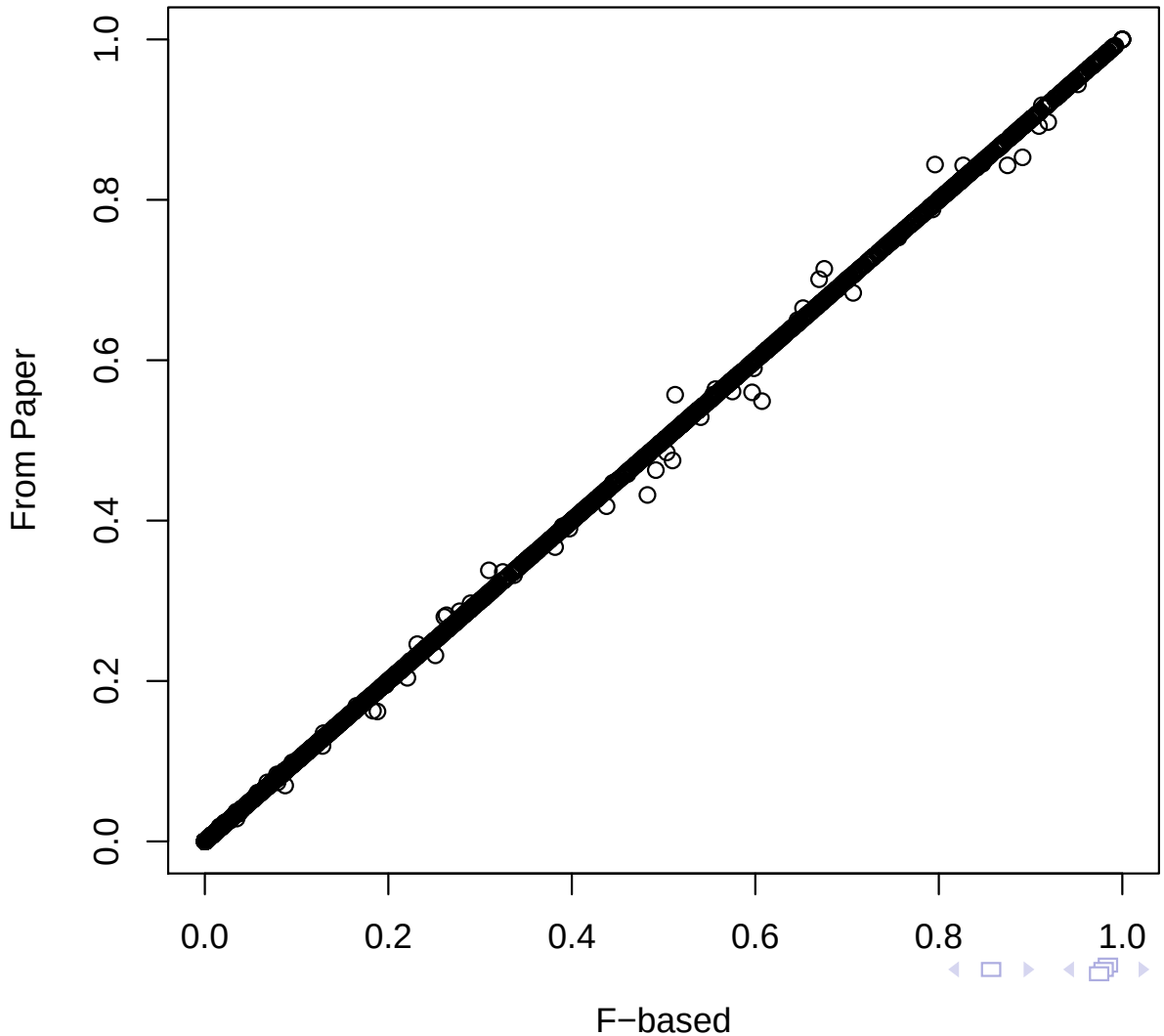
Recapitulation (Figure 1C)







P-Values



P-values

Here we make the table of their rejections and our rejections based on an F-test. What do we find? Is our test more or less conservative.

	F.based	fromWeb	Freq
1	FALSE	FALSE	2305
2	TRUE	FALSE	0
3	FALSE	TRUE	121
4	TRUE	TRUE	1916

Table 2:

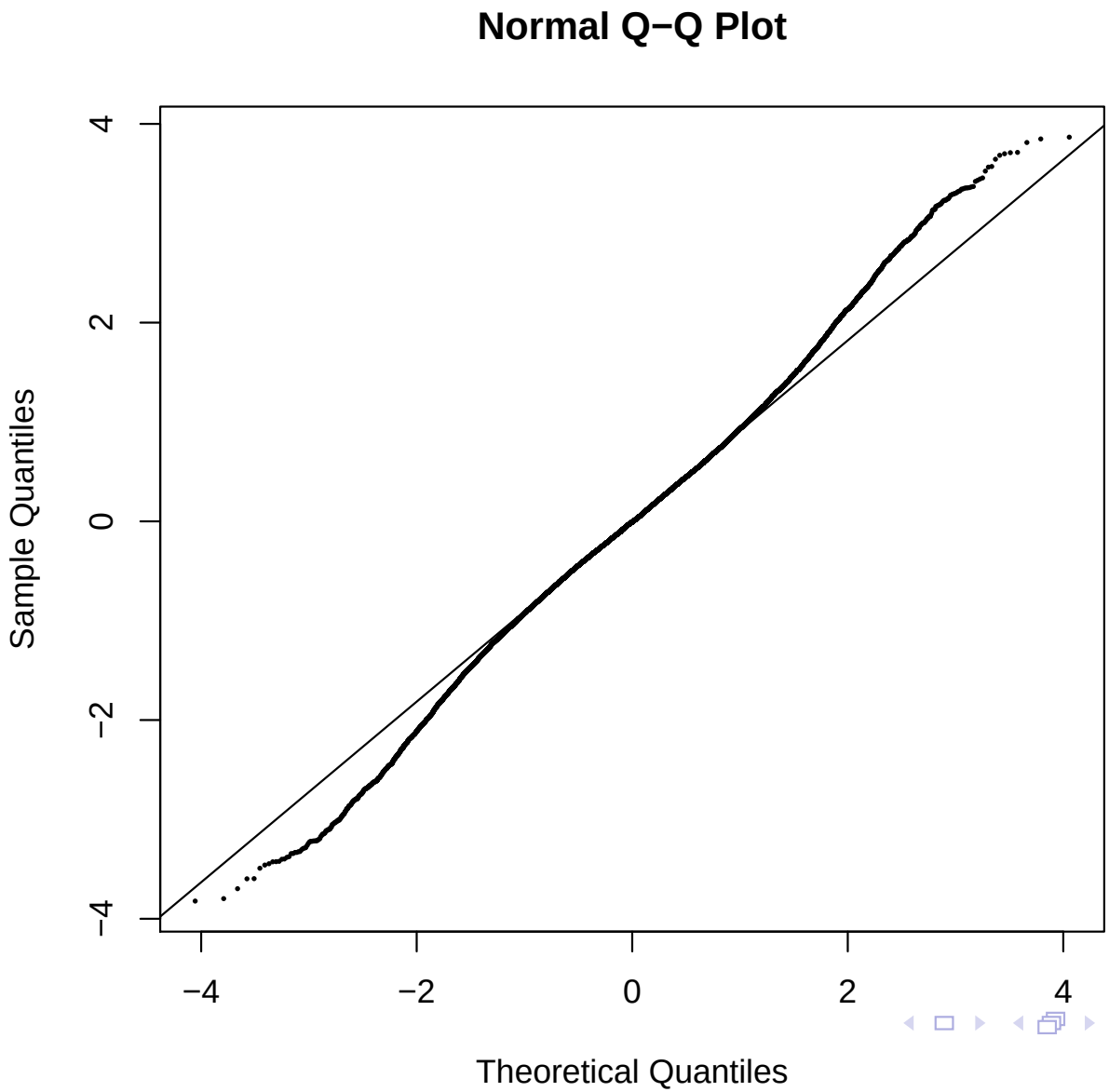
P-values: Theirs

Now we are going to check when we do the FDR on their reported p-values we get the same FDR adjusted cutoff. Do we?

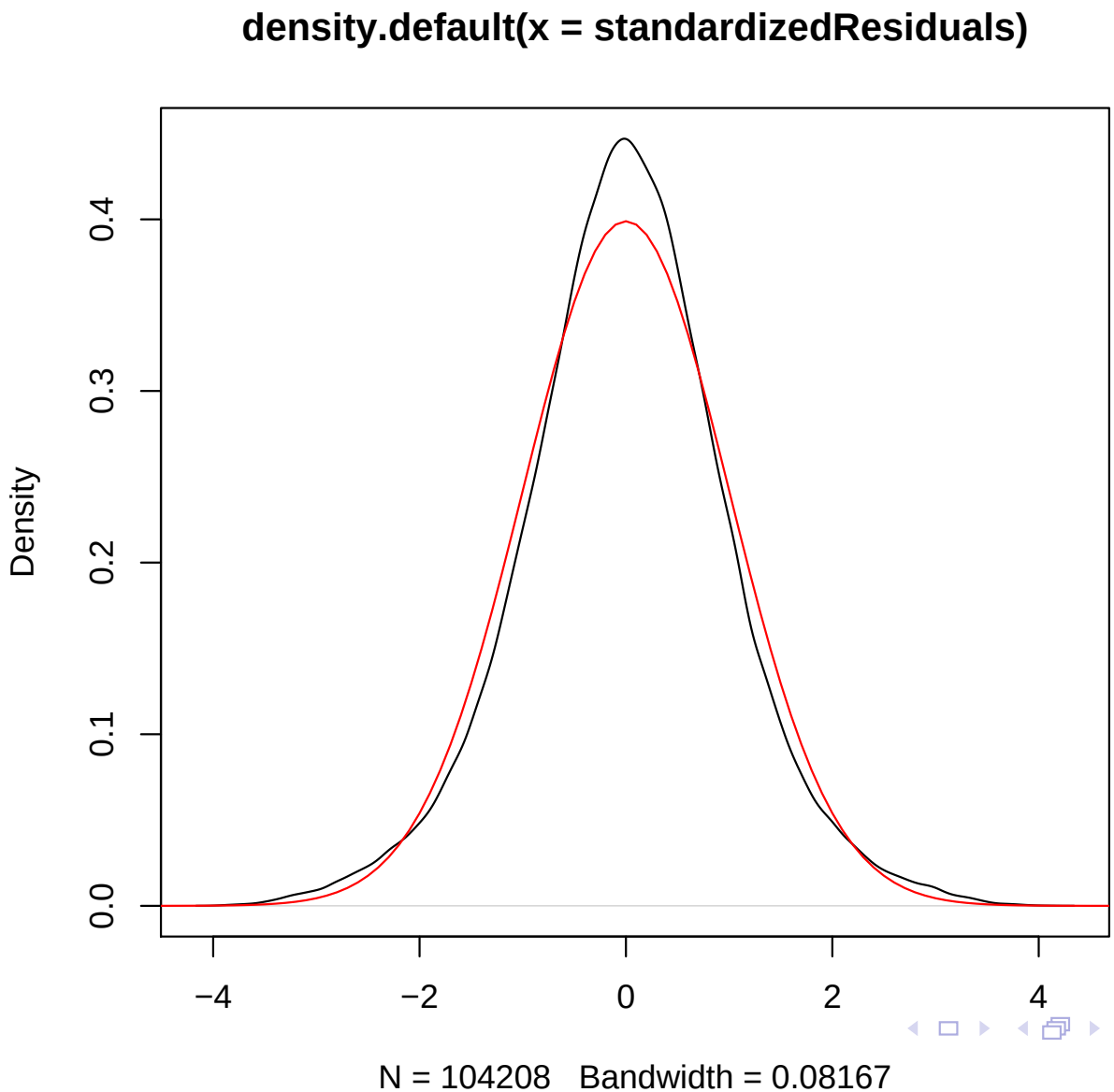
```
> sInteractionPvals <- sort(interactionPval)
> cuts <- 0.05 * (1:nGenes)/nGenes
> cutoff <- max(sInteractionPvals[sInteractionPvals <
+     cuts])
> cutoff
```

```
[1] 0.022
```

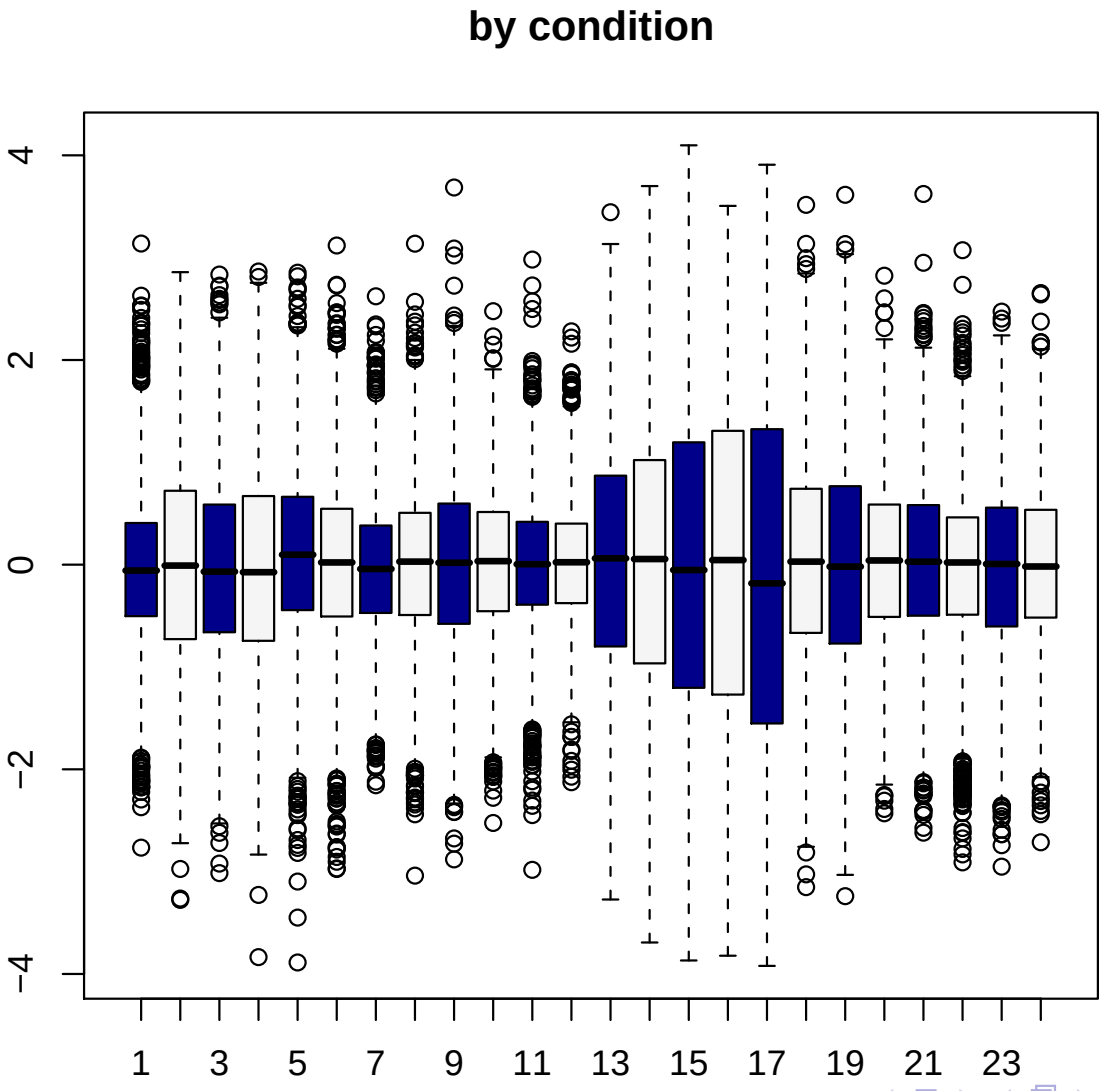
Residuals Analysis



Residuals Continued



Residuals by Array



Permutation Test

We are going to do the permutation test using LIMMA. Practically speaking this just turns out to involve a lot less thinking and is much faster. We then want to make a comparison between their results. For your analysis it probably makes sense to just do the permutation test on one gene so that you can understand just how it works. Also, we can do the permutation test using larger numbers of permutations – the more permutations we do the more accurate our pvalues are. There are too many details to go into here with respect to LIMMA.

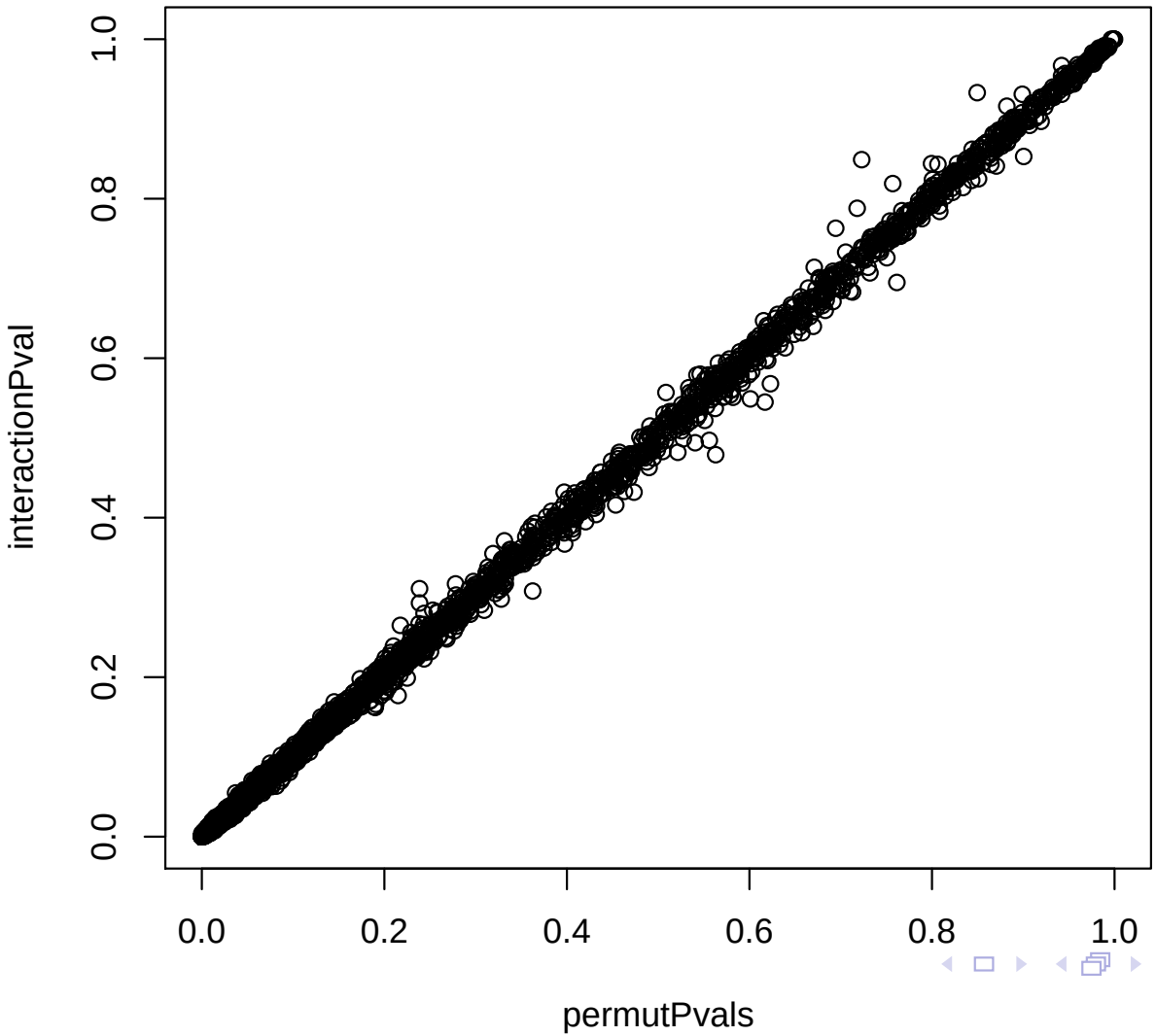
LIMMA


```
> require(limma)
> computeF <- function(ED) {
+     limmaFitFull <- lmFit(marrayData,
+         model.matrix(~dye + strain +
+             condition + condition:strain,
+             data = ED))
+     limmaFitSub <- lmFit(marrayData,
+         model.matrix(~dye + strain +
+             condition, data = ED))
+     ((limmaFitSub$sigma^2 * limmaFitSub$df) -
+         (limmaFitFull$sigma^2 *
+             limmaFitFull$df))/limmaFitFull$sigma^2
+ }
> observedFstats <- computeF(ED)
```

LIMMA


```
> nPermutations <- 2000
> permutFstats <- sapply(1:nPermutations,
+     function(i) {
+         ED[, 1:2] <- ED[sample(1:nrow(ED)),
+             1:2]
+         computeF(ED)
+     })
> permutPvals <- rowMeans(permutFstats >
+     observedFstats)
> sPermutPvals <- sort(permutPvals)
> cuts <- 0.05 * ((1:nGenes)/nGenes)
> permutCutoff <- max(sPermutPvals[sPermutPvals <
+     cuts])
```

Visualizing the Output



Visualizing the Output