# Approach 1

The most basic approach is to fit a model with a coefficient for each of the four factor combinations and then to extract the comparisons of interest as contrasts

> TS <- factor(TS, levels=c("WT.U","WT.S","Mu.U","Mu.S"))

> design <- model.matrix(~0+TS)

# Approach 2

If you do not want to find the interaction then use a nested interaction formula

> Strain <- factor(targets$Strain, levels=c("WT","Mu"))

> Treatment <- factor(targets$Treatment, levels=c("U","S"))

> design <- model.matrix(~Strain+Strain:Treatment)

> colnames(design)

[1] "(Intercept)" "StrainMu" "StrainWT:TreatmentS" "StrainMu:TreatmentS"

The first term in the model formula is an effect for Strain. This introduces an intercept column to the design matrix, which estimates the average log-expression level for wild-type unstimulated cells, and a column for Strain which estimates the mutant vs wildtype difference in the unstimulatd state

The second term in the model formula represents the interaction between stimulation and strain. Because there is no main effect for treatment in the model [no treatment term], the interaction is fitted in a nested sense. It introduces a third and a fourth column to the design matrix which represent the effect of stimulation for wild-type and for mutant mice respectively, exactly the same as the contrasts SvsUinWT and SvsUinMu defined in the previous section.

Finally, we could extract the interaction contrast `Diff` considered above by

> fit2 <- contrasts.fit(fit, c(0,0,-1,1))

> fit2 <- eBayes(fit2)

> topTable(fit2)

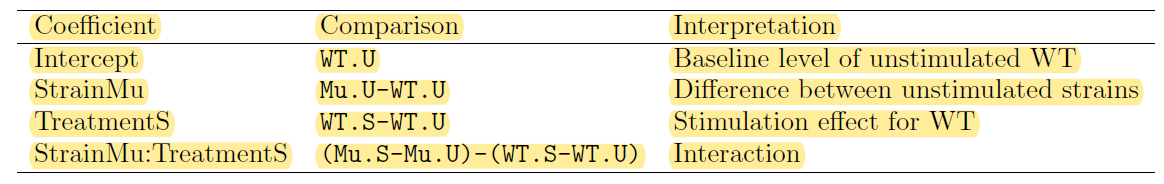
This finds genes that respond differently to the stimulus in mutant vs wild-type mice.

# Approach 3

Suppose for example that we proceed in the usual statistical way

> design <- model.matrix(~Strain\*Treatment)

This creates a design matrix which defines four coefficients with the following interpretations:



This is called the treatment-contrast parametrization. Notice that one of our comparisons of interest, Mu.S-Mu.U, is not represented and instead the comparison Mu.U-WT.U, which might not be of direct interest, is included. We need to use contrasts to extract all the comparisons of interest:

> fit <- lmFit(eset, design)

> cont.matrix <- cbind(SvsUinWT=c(0,0,1,0),SvsUinMu=c(0,0,1,1),Diff=c(0,0,0,1))

> fit2 <- contrasts.fit(fit, cont.matrix)

> fit2 <- eBayes(fit2)

The mutant stimulation effect is extracted as the sum of the third and fourth coefficients [columns] of the original model [design matrix].

# Approach 4

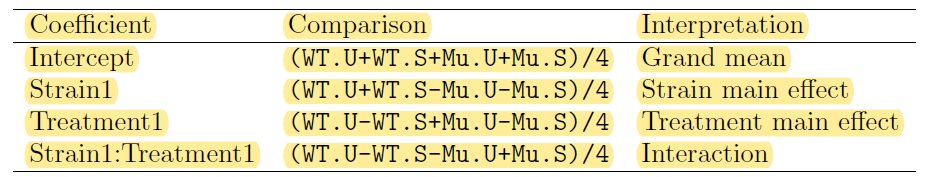
An even more classical statistical approach to the factorial experiment would be to use the sum to zero parametrization. In R this is achieved by

> contrasts(Strain) <- contr.sum(2)

> contrasts(Treatment) <- contr.sum(2)

> design <- model.matrix(~Strain\*Treatment)

This defines four coefficients with the following interpretations:



This parametrization has many appealing mathematical properties and is the classical parametrization used for factorial designs in much experimental design theory. However it defines only one coefficient which is directly of interest to us, namely the interaction. Our three contrasts of interest could be extracted using

> fit <- lmFit(eset, design)

> cont.matrix <- cbind(SvsUinWT=c(0,0,-2,-2),SvsUinMu=c(0,0,-2,2),Diff=c(0,0,0,4))

> fit2 <- contrasts.fit(fit, cont.matrix)

> fit2 <- eBayes(fit2)

The results will be identical to those for the previous three approaches.

## Treating blocking variable as a random factor

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **day** | **genotype** | **generation** |  | **Subject** | **Condition** | **Tissue** |
| 1 | control | gen 2 |  | 1 | Diseased | A |
| 1 | control | gen 3 |  | 1 | Diseased | B |
| 1 | test | gen 2 |  | 2 | Diseased | A |
| 1 | test | gen 3 |  | 2 | Diseased | B |
| 2 | control | gen 2 |  | 3 | Diseased | A |
| 2 | control | gen 3 |  | 3 | Diseased | B |
| 2 | test | gen 2 |  | 4 | Normal | A |
| 2 | test | gen 3 |  | 4 | Normal | B |
| 3 | control | gen 2 |  | 5 | Normal | A |
| 3 | control | gen 3 |  | 5 | Normal | B |
| 3 | test | gen 2 |  | 6 | Normal | A |
| 3 | test | gen 3 |  | 6 | Normal | B |

If we only wanted to compare the two tissue types, we could do a paired-samples comparison.

If we only wanted to compare diseased to normal, we could do an ordinary two-group comparison.

Since we need to make comparisons both within and between subjects, it is necessary to treat Patient as a random effect. This can be done in limma using the duplicateCorrelation function.

The two experimental factors Condition and Tissue could be handled in many ways. Here we will assume that it is convenient to join the two into a combined factor:

> Treat <- factor(paste(targets$Condition,targets$Tissue,sep="."))

> design <- model.matrix(~0+Treat)

> colnames(design) <- levels(Treat)

Then we estimate the correlation between measurements made on the same subject:

> corfit <- duplicateCorrelation(eset, design, block = targets$Subject)

> corfit$consensus

Then this inter-subject correlation is input into the linear model fit:

> fit <- lmFit(eset, design, block = targets$Subject, correlation = corfit$consensus)

Now we can make any comparisons between the experimental conditions in the usual way.

[**manual calculations**](http://www.flutterbys.com.au/stats/ws/ws11.html#Q1.2)

* we can predict multiple new y values from multiple new x values by multiplying the matrix of coefficients [aka parameters] with the matrix of contrasts [actual values of the variables].
* Each column of the contrast matrix contains new values of the variables of interest.
* For a model with two variables, each column in the contrast matrix has a new value for the second variable (the first variable is the intercept, therefore its value is fixed at 1).

|  |  |
| --- | --- |
| Coefficient matirx | Contrast matrix |
| |  |  | | --- | --- | | 51.933 | 0.8114 | | |  |  |  |  | | --- | --- | --- | --- | | 1.000 | 1.000 | 1.000 | ... | | 25.000 | 27.273 | 29.545 | ... | |

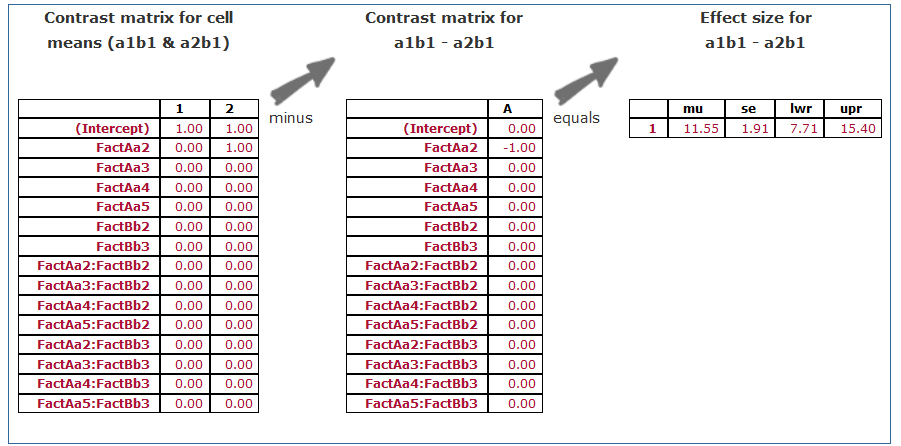
* Calculate (derive) the standard errors of the predicted yield values for each of the new Fertilizer concentrations.

vc <- vcov(model\_name) #extracts variance-covariance matrix corresponding for the coefs

se <- sqrt(diag(cc %\*% vc %\*% t(cc))) #calculates se by using vc matrix and contrast matrix cc

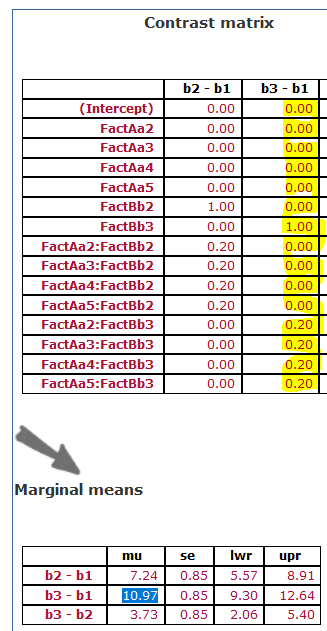
**Effect sizes - for simple main effects**

For example, to derive the effect size of a1 vs a2 for the b1 level of Factor B:



**Effect sizes - for marginal effects (no interactions)**

Effect sizes of FactB marginal means



### Simple main effects - effects associated with one factor at specific levels of another factor

But what do we do in the presence of an interaction then - when effect sizes of marginal means will clearly be underrepresenting the patterns. The presence of an interaction implies that the effect sizes differ according to the combinations of the factor levels.

For example, lets explore the effects of b1 vs b3 at each of the levels of Factor A

ES se lwr upr Comp

X.b3...b1. 11.391382 1.909191 7.649436 15.13333 b3 - b1

X.b3...b1..1 15.156703 1.909191 11.414757 18.89865 b3 - b1

X.b3...b1..2 9.703497 1.909191 5.961552 13.44544 b3 - b1

X.b3...b1..3 11.745450 1.909191 8.003504 15.48740 b3 - b1

X.b3...b1..4 6.851351 1.909191 3.109406 10.59330 b3 - b1

Compare each level of FactB at each level of FactA

ES se lwr upr FactA Comp

b2 - b1 5.313463582 1.909191 1.5715182 9.05540892 a1 b2 - b1

b3 - b1 11.391381750 1.909191 7.6494364 15.13332709 a1 b3 - b1

b3 - b2 6.077918168 1.909191 2.3359728 9.81986351 a1 b3 - b2

b2 - b1 1 6.074033717 1.909191 2.3320884 9.81597906 a2 b2 - b1

b3 - b1 1 15.156702589 1.909191 11.4147573 18.89864793 a2 b3 - b1

b3 - b2 1 9.082668872 1.909191 5.3407235 12.82461421 a2 b3 - b2

b2 - b1 2 9.693900659 1.909191 5.9519553 13.43584600 a3 b2 - b1

b3 - b1 2 9.703497500 1.909191 5.9615522 13.44544284 a3 b3 - b1

b3 - b2 2 0.009596841 1.909191 -3.7323485 3.75154218 a3 b3 - b2

b2 - b1 3 4.569607646 1.909191 0.8276623 8.31155298 a4 b2 - b1

b3 - b1 3 11.745449699 1.909191 8.0035044 15.48739504 a4 b3 - b1

b3 - b2 3 7.175842053 1.909191 3.4338967 10.91778739 a4 b3 - b2

b2 - b1 4 10.552754396 1.909191 6.8108091 14.29469973 a5 b2 - b1

b3 - b1 4 6.851350940 1.909191 3.1094056 10.59329628 a5 b3 - b1

b3 - b2 4 -3.701403456 1.909191 -7.4433488 0.04054188 a5 b3 - b2

* What is the relationship between main effects and marginal means or marginal mean effect sizes?

The marginal mean effect sizes are the main effects.

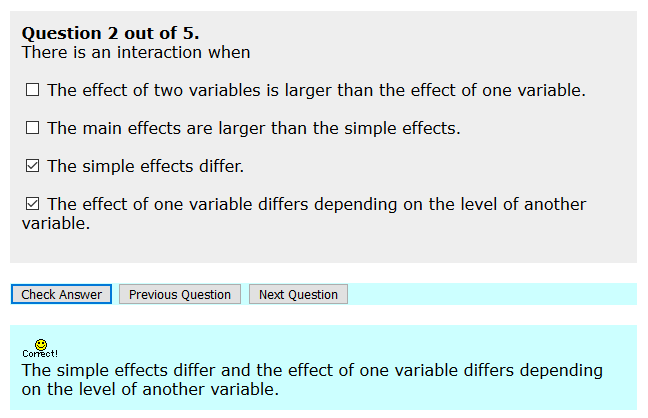
* Marginal means can be calculated after including ~~or exclusing~~ interaction, they are identical to sample means
* Similarly, marginal mean effect sizes can be calculated after including interaction

<http://onlinestatbook.com/2/analysis_of_variance/multiway.html>

A main effect of an independent variable is the effect of the variable averaging over the levels of the other variable(s). It is convenient to talk about main effects in terms of marginal means. A marginal mean for a level of a variable is the mean of the means of all levels of the other variable. For example, the marginal mean for the level "Obese" is the mean of "Girlfriend Obese" and "Acquaintance Obese." Table 1 shows that this marginal mean is equal to the mean of 5.65 and 6.15, which is 5.90. Similarly, the marginal mean for the level "Typical" is the mean of 6.19 and 6.59, which is 6.39. The main effect of Weight is based on a comparison of these two marginal means.

In contrast to a main effect, which is the effect of a variable averaged across levels of another variable, the simple effect of a variable [also called simple main effect] is the effect of the variable at a single level of another variable.

Recall that there is an interaction when the effect of one variable differs depending on the level of another variable. This is equivalent to saying that there is an interaction when the simple effects differ.



<https://stats.stackexchange.com/questions/198918/are-simple-main-effects-different-from-separate-tests-performed-at-each-level?rq=1>

If there is a significant interaction in ANOVA, we do not interpret the main effects and instead calculate the simple main effects. I wanted to ask - How is this different from just performing the separate t-tests at each level? I know that the output is largely the same, but is there a difference in power, or some other statistical difference?

when you do a bunch of separate t-tests, each test will use its own variance estimate. Each of this estimates will have lower degrees of freedom, so lower precision, than a common (pooled) estimate.

If an assumption of equal variances is (approximately) true, this will lose information, and yes, lose power. In some cases this power loss could be quite large.