LANGEBIO - BIOSTATISTICS

OCTAVIO MARTÍNEZ DE LA VEGA

12. Design and Analysis of Experiments

We have spoken here of theoretical "random experiments" from which we obtain random variables. But the design and analysis of *real* field or laboratory experiments is an important area of the scientific work. On this section I will present a brief summary of the principles of the design of experiments, illustrating them with an analyses that we are currently performing.

12.1. **General Principles of the Design of Experiments.** We refer here, specifically, to experiments that will be analyzed with Statistics, but these principles, stated in an orderly manner by Fisher are of general application.

We will refer loosely to the terms "treatments" as the set of conditions that the researcher will control and impose to a set of "experimental units". An "experimental unit" is the subject or set of subjects from which we are going to take data. It can be a single subject, like a plant or a human being, or a set like a square meter of land, or a group of plants, etc. It is important to define the "experimental unit" before any other aspect of the experiment. We are going to explain the variables measured in the experiment in general as a result of the "treatments" and the experimental "error" or unexplained variation.

The main principles to design sound experiments are

- **Replication**. You must apply each treatment to more than one experimental unit. If a treatment is applied only to one experimental unit it is impossible to make statistical inference, because no measure of error or random variation will be available. The experimental units that receive the same treatment are known as "replicates" of the treatment, and the number of replicates determines the precision in the measurement of the error.
- **Randomization**. The experimental units must be assigned to the treatments in a random fashion. This avoids the possible bias that can be introduced if the "good" experimental units are assigned to a given treatment and the "bad" to another. You must use a *random* procedure to assign each experimental unit (plant, pice of land, human being, etc.) to each treatment.

Date: April 2012.

- **Blocking**. If a putative source of important variation is detected on the experimental units, the researcher needs to form "blocks" of (the most) homogeneous experimental units to apply all treatments. For example, in an experiment with rats, the genetic background of the animals could be important, thus "blocks" could be formed by by litter (of full sibs). In a field experiment, it it possible that neighbor plots have more alike conditions about soil quality, etc., thus blocks must be formed with neighbor plots.
- **Balance**. The same number of experimental units must be allocated to each treatment, that is, the same number of replicates must be obtained from each treatment (or treatment combination). This procedure warranty an optimum estimation of the error variation.
- **Factorial structure**. If treatments involve more than one "factor" at the time, then you must use all combination of factors at all levels in a single experiment. If you try one factor at the time, and later -taken into account the results of the first experiment, decide the levels of another factor, you are taking a very serious risk of not detecting interactions of the treatments. For example, assume that you want to find the optimum dose of N, P, K for a plant specie under given conditions. Assume that you have three levels of fertilization for each factor (nutriment), say "low", "medium" and "high". The correct way to proceed is to make a complete factorial experiment with 3 factors at 3 levels each, that is 3*3*3=27 treatment combinations, each replicated, say r times. Could be tempting first to try the N factor in a single experiment. Assume that you find that the best level is "high". Then you will (erroneously) proceed to make the P experiment; assume that you find then that the optimum is "high" again. However, you never explored, for example the combination N="medium", P="medium", and that one could be the real optimum!.

See the book: Fisher, R. A. (1935) *The Design of Experiments*. Oxford Science Publications.

12.2. **Continuing with our example of *Phycomyces*.** We will be using my package "TRANOVA", which can be downloaded from:

`http://datos.langebio.cinvestav.mx/~omartine/TRANOVA`

> NOTE: At this moment (April 2012) *TRANOVA* is NOT a publicly available package, thus you can download and use it only for this course. It contain unpublished data from the lab of Dr. Alfredo Herrera-Estrella. Soon it will be available to everybody but **until then** you cannot pass it to others. Treat it confidentiality.

Let's see again our simple example of *Phycomyces*. The data are in your dataset "phyco" (that you loaded into R in your first homework)

Gene	$S.R_1$	$S.R_2$	$M.R_1$	$M.R_2$
17	22 **(40)**	6 **(10)**	30 **(46)**	18 **(19)**
1661	56 **(101)**	43 **(71)**	1,440 **(2,205)**	2,163 **(2,317)**
7433	40 **(72)**	24 **(39)**	141 **(216)**	64 **(69)**
Others	551,957	608,085	651,479	931,356
Total	552,075	608,158	653,090	933,604

Note: The data are in normal text and the numbers of transcripts per million (TPM) are in **boldface**

The *experimental unit* of this experiment is the cDNA library sequenced, that is, each one of the columns. We have two treatments, say measures in Sporangiophores=S or Mycelia=M. For each treatment we have two independent *replicates* (R_1 and R_2). And the table presents measures for three genes: 17 (which we analyzed before within one treatment), 1661 and 7433. The last row, "Others" give the number of tags for all the other genes (different from 17, 1661 and 7433) that were detected in the experiment.

Notice that there are two sources of variation in this table; the variation **between** treatments S and M and the variation **within** treatments. This two sources of variation compose the **total** variation that we see in the table, for each one of the genes. Let's write this equation in terms of Variances:

$$V_{Tot} = V_{Bet} + V_{Wit}$$

Now, I hope you will agree that our Likelihood Ratio Test statistic, the G statistic, measures the variation or *variance* of the data, thus we must have:

$$G_{Tot} = G_{Bet} + G_{Wit}$$

We will see that in fact we can segregate these sources of variation by the simple procedure of "collapsing" the data into tables that only show one of the sources. Let's do it in R. You **must** consult the help for each one of the functions that we are going to use, use *? <name of the function>* in R.

```
> # You will be able to run this only if you installed TRANOVA.
> # In that case run the following three lines (remarked):
> library(TRANOVA)
> data(phyco.simple.example)
> phyco <- phyco.simple.example$count
> # Let's obtain the total Variance of the data:
```

```
> G.tot <- G.test(phyco)
> G.tot
       G       df  p.value
3352.569    9.000    0.000
> # To obtain the variance between treatments, let's collapse
> # the original matrix, adding the columns with the same treatment.
> phyco.T <- coll(phyco, for.col = c("S", "S", "M", "M"))
> phyco.T
            S       M
17         28      48
1661       99    3603
7433       64     205
Other 1160042 1582838
> # And let's obtain the variance between treatments
> G.bet <- G.test(phyco.T)
> G.bet
       G       df  p.value
3257.651    3.000    0.000
> # Now if our reasoning is correct, the variance within tratments
> # must be equal to:
> G.tot[1:2] - G.bet[1:2] # Why not the third component?
       G       df
94.91795  6.00000
> # Now to obtain directly the variance within treatments we need
> phyco.S <- phyco[,1:2] # Only the treatment "S"
> phyco.S
        S.R1   S.R2
17        22      6
1661      56     43
7433      40     24
Other 551957 608085
> phyco.M <- phyco[,3:4] # Only the treatment "M"
> phyco.M
        M.R1   M.R2
17        30     18
1661    1440   2163
7433     141     64
Other 651479 931359
```

```
> # And the variance "within" must be given by
> G.wit <- G.test(phyco.S)[1:2] + G.test(phyco.M)[1:2]
> G.wit # Compare with the indirect result above

       G       df
94.91795  6.00000

> # To obtain the probability of G.bet under the null hypothesis
> pchisq(q=G.wit[1], df=G.wit[2], lower.tail=FALSE)

           G
2.875631e-18

> # Lets complete G.bet with it's p-value
> G.wit <- c(G.wit, pchisq(q=G.wit[1], df=G.wit[2], lower.tail=FALSE))
> names(G.wit)[3] <- "p.value"
> # Thus we have a table:
> rbind(G.bet, G.wit, G.tot)

               G df      p.value
G.bet 3257.65099  3 0.000000e+00
G.wit   94.91795  6 2.875631e-18
G.tot 3352.56894  9 0.000000e+00
```

On doing this we took advantage of a nice property of the G statistic: It is additive (for example, Pearson's χ^2 is not). Other important property that we are going to use later is that the two components, G_{Bet} and G_{Wit} are *statistically independent*.

Which null hypotheses are testing each one of the G statistics that we just obtained?. In each case we are testing

"H_0 : The column and row criteria of classification are independent"

And this is equivalent to be testing the hypothesis that the genes are equally expressed. Thus G_{Tot} is testing "genes are equally expressed in all four libraries", G_{Bet} is testing "genes are equally expressed in the two treatments (S and M)", while G_{Wit} is testing (for the two treatments at once) "genes are equally expressed in the two replicates". For the values of the probabilities, we can see that all these three hypotheses are safely rejected (α is very small in all the three cases).

However, what we *really* want to know is if the variation between treatments is larger than the variance within treatments; that is, if we have evidence that the treatments have an effect which can be considered "larger" than the error variation that is naturally present. Thus, formally, we want to test the hypothesis:

H_{0V} : The variation between treatments is equal to the variation within treatments

versus

H_{aV} : The variation between treatments is larger than the variation within treatments

Given that the values of G had a $\chi^2(df)$ distribution and the two components, G_{Bet} and G_{Wit} are *statistically independent* we can propose the following statistic for the test of H_{0V}:

$$F = \frac{G_{Bet}/df_{Bet}}{G_{Wit}/df_{Wit}}$$

Under the null hypothesis, H_{0V}, F as the well known (Snedecor's) $\mathcal{F}$ distribution for the ratio of two variances, that is we can write

$$F \sim \mathcal{F}(df_{Bet}, df_{Wit})$$

Let's make the calculations in R

```
> # Obtaining the value of the F statistic
> phyco.F <- (G.bet[1]/G.bet[2]) / (G.wit[1]/G.wit[2])
> names(phyco.F) <- "F" # change the name that otherwise will be "G"!
> phyco.F # Let's calculate the probability under H0V:
       F
68.64141
> pf(phyco.F, df1=G.bet[2], df2=G.wit[2], lower.tail=FALSE)
           F
4.911313e-05
```

Given the very low (highly significant) value of α, say $P[F \geq 68.64141] = 4.911313e-05$ we reject the null hypothesis H_{0V} of equality of variances and conclude that there is a significant effect of the treatments on the expression of the genes.

In the previous analysis we have taken the genes as a group. Of course, it is highly interesting to dissect the general effects of all genes individually into the effects of each gene. In general, for $i = 1, 2, \ldots, k$ genes we will have the following relevant hypotheses

- "H^i_{0Bet} : "The classification of treatments and *gene-i, gene-Not-i* are independent" = "There is no differential expression of between treatments for gene i"

  versus

  "H^i_{aBet} : "The classification of treatments and *gene-i, gene-Not-i* are dependent" = "There is differential expression of between treatments for gene i"
- "H^i_{0V} : "The variances between and within treatments for *gene-i* are equal = "There is no effect of the treatments in the expression of *gene-i*.

  versus

"H^i_{aV}: "The variances between treatments is larger than the variance within treatments for *gene-i* = "There is effect of the treatments in the expression of *gene-i*.

Note that to declare that a gene i is differentially expressed we must reject **both** null hypotheses: H^i_{0Bet} and H^i_{0V}, testing the first with the statistic G^i_{Bet} and the second with the statistic F^i, obtained from the table collapsed to represent only the variance given by gene i. It is important to underline that these two test are **NOT** independent, and thus some calculations are needed to obtain the conjoint probability of error of the complete procedure, say

α_T = Probability of rejecting H^i_{0Bet} and H^i_{0V} given that both are true

If we define

α_B = Probability of rejecting H^i_{0Bet} given that this hypothesis is true

and

α_F = Probability of rejecting H^i_{0V} given that this hypothesis is true

It is possible to find one of the probabilities as function of the other two (I will not give the details, but you can consult the relevant function within TRANOVA).

We are going to go through the detailed calculations for the second gene of our small example (gene 1661) and will then present the full analysis of this small dataset using TRANOVA.

```
> # Doing the analysis for the second gene as example
> phyco1661 <- coll(phyco, for.row=c("Other","1661", "Other", "Other"))
> # A 2 by 4 table; compare it with phyco
> phyco1661
```

```
       S.R1   S.R2   M.R1   M.R2
Other 552019 608115 651650 931441
1661      56     43   1440   2163
```

```
> G.tot <- G.test(phyco1661) # Total variance given by gene 1661
> G.tot
```

```
      G      df p.value
3221.67    3.00    0.00
```

```
> # Collapsing the table per treatments
> phyco1661.Bet <- coll(phyco1661, for.col=c("S","S","M","M"))
> phyco1661.Bet # Only variation between treatments
```

```
            S       M
Other 1160134 1583091
1661       99    3603
```

```
> G.bet <- G.test(phyco1661.Bet) # G between treatments
> G.wit <- G.tot[1:2] - G.bet[1:2] # G within treatments
```

```
> G.wit <- c(G.wit, pchisq(G.wit[1], df=G.wit[1], lower.tail=FALSE))
> names(G.wit)[3] <- "p.value"
> G.wit
        G        df   p.value
5.3328125 2.0000000 0.4185386
> # Thus we have a table:
> rbind(G.bet, G.wit, G.tot)
                G df   p.value
G.bet 3216.336789  1 0.0000000
G.wit    5.332812  2 0.4185386
G.tot 3221.669601  3 0.0000000
> # Obtaining the value of the F statistic
> phyco.F <- (G.bet[1]/G.bet[2]) / (G.wit[1]/G.wit[2])
> names(phyco.F) <- "F" # change the name that otherwise will be "G"!
> phyco.F # Let's calculate the probability under H0V:
       F
1206.244
> pf(phyco.F, df1=G.bet[2], df2=G.wit[2], lower.tail=FALSE)
           F
0.0008279901
```

Note that for this second gene (gene 1661) both statistics, G_{Bet} and F are highly significant (probabilities ≈ 0 and thus we can affirm that this gene present a differential expression between the two treatments. Let's see the table in transcripts per million, that is easier to interpret

```
> # Let's see the tables in Transcript Per Million
> round(counts2tpm(phyco)) # phyco in TPM
        S.R1   S.R2   M.R1   M.R2
17        40     10     46     19
1661     101     71   2205   2317
7433      72     39    216     69
Other 999786 999880 997533 997595
> # Now, adding per treatment
> temp <- round(coll(counts2tpm(phyco), for.col=c("S","S","M","M")))
> temp
            S       M
17         50      65
1661      172    4522
```

```
7433      112     284
Other 1999666 1995129
> # the "Fold change"
> temp[1:3,2]/temp[1:3,1] # M / S
       17      1661      7433
 1.300000 26.290698  2.535714
```

Note, from the last row of R output that the second gene (1661) has a large fold change of more than 26 times more estimated expression in the "M" treatment compared with the "S" treatment. The first gene (17) has a modest fold change of only 1.3 and the third gene (7433) a relatively large fold change of 2.5 However, this last gene appear to be highly variable (see first table of the R output above).

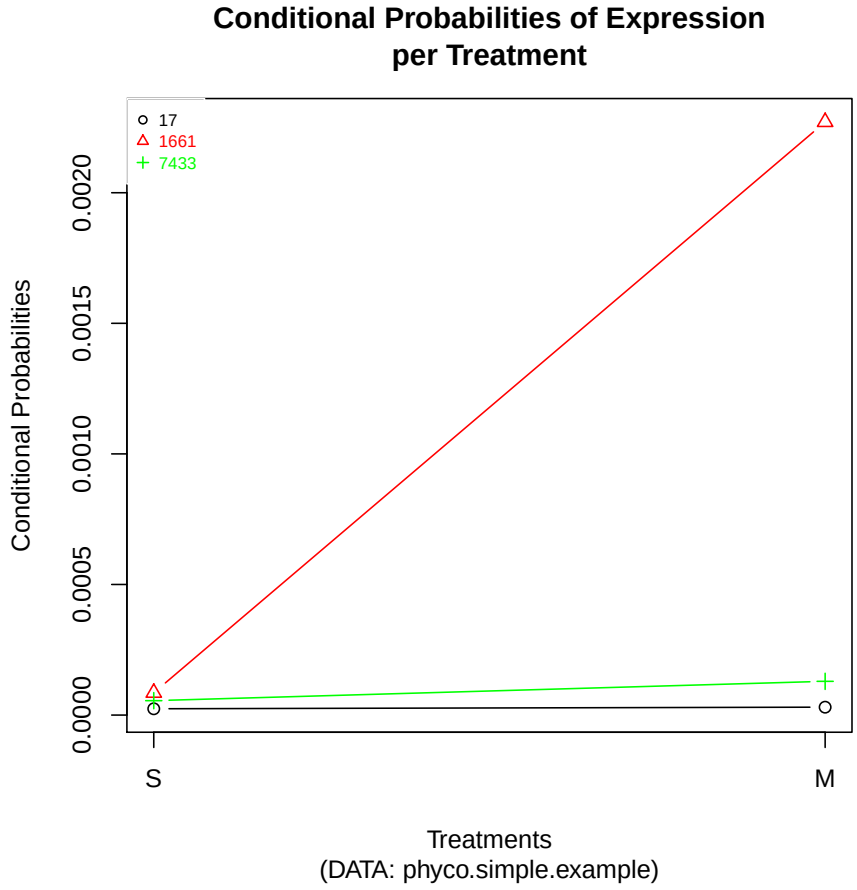
The complete analysis of this small example can be performed in my package *TRANOVA* with just one step. Let's do it

```
> # I will let ALL the parameters at defaults except
> # offset=TRUE because the last row of our table is an "offset"
> phyco.t <- tranova(phyco, design=c("S","S","M","M"), offset=TRUE)
> names(phyco.t) # See the names of this list
[1] "Genes"           "PerGene"         "Corr.Fact"
[4] "Matrices"        "summary.PerGene" "Call"
[7] "DataSummary"     "ValuesOfAlpha"
> phyco.t # See all results in the object
$Genes
                           G df          P.G        F
BetweenTreatments 3257.65099  3 0.000000e+00 68.64141
WithinTreatments    94.91795  6 2.875631e-18       NA
Total             3352.56894  9 0.000000e+00       NA
                          P.F
BetweenTreatments 4.911313e-05
WithinTreatments            NA
Total                       NA

$PerGene
            G.t df.t        P.G.t          G.b df.b
17     21.09882    3 1.004166e-04    0.9193465    1
1661 3221.88665    3 0.000000e+00 3216.5699323    1
7433  109.58346    3 1.348888e-23   40.1617157    1
            P.G.b       G.w df.w        P.G.w
17   3.376466e-01 20.179476    2 4.150329e-05
```

```
1661 0.000000e+00  5.316722    2 7.006297e-02
7433 2.337852e-10 69.421749    2 8.418965e-16
                F          P.F       F.corr     P.F.corr
17   9.111698e-02 0.7912576808 9.111698e-02 0.7912576808
1661 1.209982e+03 0.0008254351 1.209982e+03 0.0008254351
7433 1.157036e+00 0.3946122694 1.157036e+00 0.3946122694
     Sign.T
17        0
1661      1
7433      0

$Corr.Fact
 Between    Total
1.000072 1.000067

$Matrices
$Matrices$Original
        S.R1   S.R2   M.R1   M.R2
17        22      6     30     18
1661      56     43   1440   2163
7433      40     24    141     64
Other 551957 608085 651479 931359

$Matrices$Treatments
            S       M
17         28      48
1661       99    3603
7433       64     205
Other 1160042 1582838


$summary.PerGene
   N.genes    S.G.bet   %S.G.bet        S.F       %S.F
   3.00000    2.00000   66.66667    1.00000   33.33333
  S.F.corr    %F.corr  S.tranova %S.tranova
   1.00000   33.33333    1.00000   33.33333

$Call
$Call$To_tranova
tranova(x = phyco, design = c("S", "S", "M", "M"), offset = TRUE)
```

```
$Call$To_tranova.replicates
tranova.replicates(x = x, design = design, genes = genes, alpha.G.b = alpha.genes.G.b,
    alpha.F = alpha.F, Bonferroni = Bonferroni, give.matrices = give.matrices,
    offset = offset, old.call = the.call)


$DataSummary
$DataSummary$Rows
    N.genes   N.u.genes %Redundancy
          4           4           0

$DataSummary$Design
$DataSummary$Design$Treatments
   N.exp.units   N.treatments Avg.replicates
             4              2              2

$DataSummary$Design$ReplicatesPerTreatment
S M
2 2



$ValuesOfAlpha
 alpha.G.b    alpha.F    alpha.T
0.01666667 0.01666667 0.00201207

attr(,"class")
[1] "tranova.replicates"
> # A plot of the conditional probability of expression per
> # treatment.
> cond.prob.plot(phyco.t$Matrices$Treatments, where="topleft")
                S           M
17   2.413308e-05 3.025158e-05
1661 8.532769e-05 2.270759e-03
7433 5.516133e-05 1.291995e-04
> title(main="Conditional Probabilities of Expression\nper Treatment",
+ sub="(DATA: phyco.simple.example)")
```

```
                S            M
17   2.413308e-05 3.025158e-05
1661 8.532769e-05 2.270759e-03
7433 5.516133e-05 1.291995e-04
```



From these analyses we can conclude that:

- a) There is a significant effect of treatments (S versus M) in the expression of the genes as a group.
- b) That, individually, only one of the genes (the gene 1661) can be considered as differentially expressed.

12.3. **The Full *Phycomyces* dataset and its analysis.** The data that we just analyzed is a small subset of the full experiment with *Phycomyces* carried out in the lab of Dr. Alfredo Herrera-Estrella (manuscript in preparation). The experiment has

as main goal to detect the effect of three factors in the transcriptome of *Phycomyces*: Genotype, Structure and Light as well as their interactions. For each factor, two levels were used and three biological replicates were obtained by sequencing independently the c-DNA libraries. All libraries were sequenced in the Solid 4 platform and the unique mapping of the tags was performed using GeneSifterTM net software (http://www.geospiza.com/). This resulted into a total of $2 * 2 * 2 * 3 = 24$ independent libraries. The number of gene tags per library varied from 457,665 to 1,069,453 with an average of 710,084 tags per library. The total number of genes detected was 11,897 generating a data matrix of dimension 11,897 x 24, containing 285,528 cells. The mean number of tags per cell was 60 with a maximum counting of 37,650 tags in a single cell.

```
> # A first look at the data
> data(phyco) # The full Phycomyces data
> names(phyco) # This is a list with two components
[1] "count"  "design"
> class(phyco$count) # The type of structure that contains the counts
[1] "matrix"
> attributes(phyco$count)$dim # The dimension of the matrix
[1] 11897    24
> # Let's see the names of the columns:
> attributes(phyco$count)$dimnames[[2]]
 [1] "MSP.R1" "MSP.R2" "MSP.R3" "MSA.R1" "MSA.R2"
 [6] "MSA.R3" "MMP.R1" "MMP.R2" "MMP.R3" "MMA.R1"
[11] "MMA.R2" "MMA.R3" "WSP.R1" "WSP.R2" "WSP.R3"
[16] "WSA.R1" "WSA.R2" "WSA.R3" "WMP.R1" "WMP.R2"
[21] "WMP.R3" "WMA.R1" "WMA.R2" "WMA.R3"
> # The codes within the three characters of the names are for:
> # 1st: <Genotype>; M = Mutant, W = Wild type,
> # 2nd: <Structure>; S = Sporangiophores, M = Mycelia,
> # 3dt: <Light>; A = Absent, P = Present
> # and then .R<Number of replicate>. Thus for example:
> # "WSP.R2" means G=Wild type, S=Sporangiophores, P=Light Present; Replicate 2
> # Let's see the design vector for the experiment:
> phyco$design
 [1] "MSP" "MSP" "MSP" "MSA" "MSA" "MSA" "MMP" "MMP"
 [9] "MMP" "MMA" "MMA" "MMA" "WSP" "WSP" "WSP" "WSA"
[17] "WSA" "WSA" "WMP" "WMP" "WMP" "WMA" "WMA" "WMA"
```

```
> # A summary of the first three libraries counts'
> summary(phyco$count[,1:3])
     MSP.R1           MSP.R2            MSP.R3
 Min.   :   0.0   Min.   :   0.00   Min.   :   0.00
 1st Qu.:  13.0   1st Qu.:  14.00   1st Qu.:  15.00
 Median :  35.0   Median :  36.00   Median :  41.00
 Mean   :  52.8   Mean   :  53.42   Mean   :  62.75
 3rd Qu.:  73.0   3rd Qu.:  74.00   3rd Qu.:  86.00
 Max.   :1397.0   Max.   :1868.00   Max.   :1672.00
> # And a small part of phyco$count
> phyco$count[1:5,1:6]
  MSP.R1 MSP.R2 MSP.R3 MSA.R1 MSA.R2 MSA.R3
1     25     29     39     19     32     22
2      5      3      4      3      0      1
3      6     20      8     92    171     47
4     23     16     30     11     17     11
5     40     47     51     18     29     26
```

12.4. **An important sideline: Multiple Testing.** Note that in our dataset we have a matrix of 11,897 rows (genes) and using TRANOVA we will be performing 11,897 test, one for each gene. How this affects our probability of Type I Error, α?.

12.4.1. *Example:* "I don't like this result" *(matarili-rili-ron).* Let's put a different example. Take 100 numbers from the Standard Normal Distribution, that is, $X_i \sim \mathcal{N}(\mu = 0, \sigma^2 = 1)$, and divide them in two arbitrary groups of the same size: $n = 50$, say a and b. Now perform a test for $H_0 : \mu_a = \mu_b$ *versus* $H_A : \mu_a \neq \mu_b$. We can carry out, for example, the very well known Student's t-test, fixing an acceptable probability of error Type I, for example $\alpha = 0.05$ (having a 95% of confidence). What do you expect? Well of course you **know** that $H_0 : \mu_a = \mu_b$ is true (because all the 100 numbers came from the same distribution). Let's try it in R:

```
> my.num <- rnorm(100)
> summary(my.num) # A quick analysis of the full data
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-3.01100 -0.75880 -0.05547 -0.10740  0.57190  2.97600
> # 50 numbers at random from 1 to 100.
> my.sample.a <- sort(sample(x=c(1:100), size=50, replace = FALSE))
> my.sample.a
```

```
 [1]   2   3   4  10  12  13  14  15  17  18  19  24  25
[14]  27  28  30  31  34  35  36  39  41  43  44  45  47
[27]  48  49  54  55  56  61  62  63  64  68  71  72  73
[40]  76  77  79  80  81  87  88  91  95  99 100
```

```
> my.sample.b <- setdiff(c(1:100), my.sample.a) # The rest
> my.sample.b
```

```
 [1]  1  5  6  7  8  9 11 16 20 21 22 23 26 29 32 33 37
[18] 38 40 42 46 50 51 52 53 57 58 59 60 65 66 67 69 70
[35] 74 75 78 82 83 84 85 86 89 90 92 93 94 96 97 98
```

```
> # Let's divide my.num in the two (random) groups:
> my.group.a <- my.num[my.sample.a] # First group
> my.group.b <- my.num[my.sample.b] # Second group
> # And let's perform the test at 95% confidence level
> my.test <- t.test(my.group.a, my.group.b)
> my.test # Prints the result
```

```
        Welch Two Sample t-test

data:  my.group.a and my.group.b
t = -0.2936, df = 86.152, p-value = 0.7697
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.4556829  0.3383888
sample estimates:
  mean of x   mean of y
-0.13675175 -0.07810468
```

```
> my.test$p.value # Prints the p.value for the test
```

```
[1] 0.7697431
```

The "more likely" result (95%) of this test is that the null hypothesis $H_0 : \mu_a = \mu_b$ was not rejected. But now assume that for some strange reason you say *"I do not like the results of this test, let's try again (with the same data)"*.

This of course make's not sense, because if you repeat "many" times this process then, eventually, you will get (erroneously) "Significant" results, just because you selected too many "small" numbers in one group and "large" in the other!. In fact, in average this will happens around 5% of the times (because you used $\alpha = 0.05$). I am so sure that it will happen sooner rather than later, that I will embark by laptop in a (potentially) infinite loop, of which it will go out ONLY when the test is rejected:

```
> my.prob <- 0.05 # The threshold to reject
> how.many.tests <- 1 # We have carried out one test already
```

```
> # NOTE: I will be using the SAME data: my.num
> while(my.test$p.value > my.prob){ # A potentially infinite loop
+ how.many.tests <- how.many.tests + 1 # Increase this
+ my.sample.a <- sort(sample(x=c(1:100), size=50, replace = FALSE)) # A "new" sa
+ my.sample.b <- setdiff(c(1:100), my.sample.a) # The rest
+ my.group.a <- my.num[my.sample.a] # First group
+ my.group.b <- my.num[my.sample.b] # Second group
+ my.test <- t.test(my.group.a, my.group.b)
+ }
> # Now, eventually this loop ended after:
> how.many.tests
[1] 10
> my.test # Prints the result

        Welch Two Sample t-test

data:  my.group.a and my.group.b
t = 2.5804, df = 96.487, p-value = 0.01137
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1151574 0.8827207
sample estimates:
 mean of x  mean of y
 0.1420413 -0.3568977
> my.test$p.value # Prints the p.value for the test
[1] 0.01137463
> summary(my.group.a); summary(my.group.b)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.8410 -0.5207  0.2788  0.1420  0.7252  2.9760
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-3.0110 -1.0380 -0.1276 -0.3569  0.4446  1.5620
```

In the case of the *Phycomyces* experiment (as in any RNA-Seq) experiment we will be doing n test, where n is the number of genes detected. These tests are not completely independent, because we re-arrange the same data into n tables (in fact we have $n - 1$ degrees of freedom from the rows). In any case we need to correct our value of α to have a small number of **false positives**; i.e., cases for which we claim significance when it was really the random error the one that caused the artefactual result. The most stringent of the procedures designed with this aim is the *Bonferroni's correction*. See, for example:

http://en.wikipedia.org/wiki/Bonferroni_correction

If n tests will be performed, then Bonferroni's correction consist into taking in all the individual tests a probability of Error Type I equal to

$$\alpha' = \frac{\alpha}{n}$$

This guarantee that the "experimental wise" error, or proportion of false positives will be not larger than α.

Now, lest return to the analysis of the *Phycomyces* experiment. We need to decide which value of Error Type I, α_T, we are willing to accept. Assume that we are willing to take a **maximum** proportion of false positives of 10%, that is we want $\alpha'_T = 0.1$. Then, using Bonferroni's correction we set the error in each individual test in $\alpha_T = 0.01/11,897 = 8.40548e-07$. Now, into TRANOVA we need to set the values of α_G and α_F, the values of error in the G_{bet} and F tests respectively. There are various strategies for this, but let assume that we agree to restrain the error probabilities to be equal to each other, that is

$$\alpha_G = \alpha_F$$

then we can calculate them in R

```
> # We have 11,897 genes and 8 treatments with 3 replicates each.
> # Now for the G_bet test of each gene we will have df=(2-1)*(8-1)=7
> # and for the G_wit we will have df=(2-1)*8*(3-1)=16
> # Note that the total df=(2-1)(24-1)= 23 = 7+16
> # Now calculate the alphas:
> equal.alphas(alpha.T=0.01/11897, df.bet=7, df.wit=16, give.error=T)

      alphas    abs.error calc.alpha.T
3.385764e-05 1.430045e-16 8.405480e-07
```

Thus, we conclude that setting $\alpha_G = \alpha_F = 3.385764e{-}05$ we obtain $\alpha_T = 0.01/11897$ and thus a maximum of false positives of 0.1 or 10%.

Now let's go back to the analysis of the dataset. We can perform it in R.

```
> # Run the following two lines without the remarks "#"
> # phyco.tranova <- tranova(phyco$count, design=phyco$design,
> # alpha.genes.G.b=3.385764e-05, alpha.F=3.385764e-05, Bonferroni=FALSE)
>
> # We set the alpha levels at the value that we predeterminated
> # and Bonferroni=FALSE because we have already take that into account
> # In my machine it took around 4.3 minutes to run.
> names(phyco.tranova) # Names of the results.
```

```
[1] "Genes"           "PerGene"         "Corr.Fact"
[4] "Matrices"        "summary.PerGene" "Call"
[7] "DataSummary"     "ValuesOfAlpha"
```

```
> phyco.tranova$ValuesOfAlpha
```

```
   alpha.G.b      alpha.F      alpha.T
3.385764e-05 3.385764e-05 8.405481e-07
```

First let't see the **global** analysis, taking all genes as a group

```
> phyco.tranova$Genes
```

```
                          G     df P.G        F P.F
BetweenTreatments   9771292  83272   0 17.14625   0
WithinTreatments    1302581 190336   0       NA  NA
Total              11073873 273608   0       NA  NA
```

From this TRANOVA table we can see that the value of the F statistic (17.14625) is highly significant, as well as the G_{Bet} and thus we conclude that there is significant differential expression of the genes involved in the experiment with regard with the treatments; i.e., at least two of the treatments give differential expression of the genes.

Now we must determine how each one of the factors of the experiment influenced the results. Again, this can be done at global level (genes as a group) or, individually, gene by gene.

In particular, think about the *variance* between treatments. It is possible to segregate the variance, G_{bet} in the global experiment (which includes all 2*2*2=8 treatments) into components that are explained by each one of the factors and their interactions. In particular, consider that

$$E[G_{Bet}] = \sigma^2_G + \sigma^2_S + \sigma^2_L + \sigma^2_{GxS} + \sigma^2_{GxL} + \sigma^2_{SxL} + \sigma^2_{GxSxL}$$

while the expected value of $E[G_{Wit}]$ is

$$E[G_{Wit}] = \sigma^2_\varepsilon$$

the unexplained variation or "error variance".

In the equation above for $E[G_{Bet}]$ the sub-indexes denote the source of variation, say G = Genotype, S = Structure and L = Light, and we have main effects (G, S and L) as well as second order interactions (GxS, GxL and SxL) and the third order interaction $GxSxL$. It is possible to segregate these sources of variation performing analyses in which we take into account only the relevant source. Performing various analyses, ignoring in turn one or more of the factors, but using all the data, it it possible to segregate each source of variation by using linear functions of the

expectation of G_{Bet} in the various analyses performed. We will give an example here.

To "isolate" the factor "Genotype" we can run the analysis considering as treatments only the variation in genotype, that is "re-labeling" the treatments as

```
> # Let's take only the first letter of the names of the treatments
> # (Remember: This letter indicates the "Genotype" = "M" or "W")
> Genotype <- substring(phyco$design, 1, 1)
> Genotype
 [1] "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "W"
[14] "W" "W" "W" "W" "W" "W" "W" "W" "W" "W" "W" "W"
> # Now, lets perform the analysis using this vector as "design"
> # phyco.tranova.G <- tranova(phyco$count, design=Genotype)
> # (I ignore all other parameters)
> phyco.tranova.G$Genes # The "Genes" Table (including G.bet)
                           G     df P.G         F P.F
BetweenTreatments   316262.1  11896   0 0.6467762   1
WithinTreatments  10757610.8 261712   0        NA  NA
Total             11073872.9 273608   0        NA  NA
```

Now we have a value for the estimated $E[G_{Bet}]$ when only the genotype is varying; we reason that **in this case**

$$E[G_{Bet}] = \sigma^2_G$$

because only that factor has been taken into account. We can realize other analyses, considering in turn only one factor or pair of factors.

In fact, consider the following table of expectations of G_{Bet} in each analyses:

Analysis (i)	Factors	$E[G_{Bet}]_i$
1	G	σ^2_G
2	S	σ^2_S
3	L	σ^2_L
4	GS	$\sigma^2_G + \sigma^2_S + \sigma^2_{GxS}$
5	GL	$\sigma^2_G + \sigma^2_L + \sigma^2_{GxL}$
6	SL	$\sigma^2_S + \sigma^2_L + \sigma^2_{SxL}$
7	GSL	$\sigma^2_G + \sigma^2_S + \sigma^2_L + \sigma^2_{GxS} + \sigma^2_{GxL} + \sigma^2_{SxL} + \sigma^2_{GxSxL}$

Now, see that there are linear functions of the estimates of $E[G_{Bet}]_i$ in the analyses that allow us to estimate each one of the components; calling i to the $i-th$ row of the previous table, then

Linear function of $E[G_{Bet}]_i$	Estimated component
G_1	σ^2_G
S_2	σ^2_S
L_3	σ^2_L
$GS_4 - G_1 - S_2$	σ^2_{GS}
$GL_5 - G_1 - L_3$	σ^2_{GL}
$SL_6 - S_2 - L_3$	σ^2_{SL}
$GSL_7 - GS_4 - GL_5 - SL_6 + G_1 + S_2 + L_3$	σ^2_{GSL}

Making the corresponding analyses, we can obtain estimates of each one of the variance components. The following table presents these estimates as well as the percentages of variation which can be attributed to each factor and interaction of factor.

Component	Estimate	*df*	F	$\%V$
σ^2_G	316,262	11,896	4	3
σ^2_S	7246,089	11,896	89	74
σ^2_L	526,251	11,896	6	5
σ^2_{GxS}	359,848	11,896	4	4
σ^2_{GxL}	504,370	11,896	6	5
σ^2_{SxL}	345,767	11,896	4	4
σ^2_{GxSxL}	472,705	11,896	6	5
Error	1,302,581	190,336		
Total	11,073,873	273,608	17	100

From the previous table we can see that, by percentage, the most important factor in the experiment is S=Structure which causes around 74% of the variance. Interestingly, even the high order interaction GxSxL accounts for around 5% of the variance in differential expression. That means that the phenomenon is not completely lineal; the changes in the expression of the genes need to be explained not only by the addition of the individual effect of the genes, but also for the "epistatic" interaction of (possibly) several loci.

The analyses performed at level of genes, gave a huge matrix with 11,897 rows (the genes) and many columns, showing the results of the distinct tests performed and estimates of each one of the seven effects of the factors and interactions. There is a mine of information there, and of course methods of *data mining* are needed to obtain **biological knowledge** from these results.

12.4.2. *Homework.*

- 1. Program a function to simulate credible RNA-Seq experiments with at least 2 treatments and 2 replicates and a fixed number of genes, say g. Notice

that for each gene you must input at least two parameters: The average expression frequency and the difference (if any) of expression frequency between treatments. You could find useful the function "construct.pseudo.replicates" of TRANOVA.

- 2. Run the function that you programed with at least: 2 treatments and 2 replicates per treatment with g=100 genes and null effect of the treatment. Perform a full analysis and interpretation with TRANOVA.
- 3. Run the function that you programed with at least: 2 treatments and 2 replicates per treatment with g=100 genes and a not null effect of the treatment for 50% of the genes. Perform a full analysis and interpretation with TRANOVA.

I hope you have enjoyed the lectures and learned something useful.