Statistics 688 Consulting

Report for Michael Flood

Nathan Hattersley
Renee Zhang
Elmira Torabzadehkhorasani

authored by
Nathan Hattersley

Client: Michael Flood
Meeting Date: September 19, 2017
Contact:
nathanhattersley@email.arizona.edu
reneezhang@email.arizona.edu
elmira@statlab.bio5.org

0 Executive Summary

We are exploring statistical models for how an agonist affects the light response in the retinal cells of both diabetic and non-diabetic mice. You came to us with questions about the fit of your model, and sought diagnostics to resolve issues of non-normality and heterogeneity of variance. We suggest altering the conceptual model for your experiment, and re-running your analysis with both raw and log-transformed data. This new design accounts for variance from more sources and should improve the model's fit.

1 Background

The experimental design was to measure reactions to various intensities of light in the retinal cells of healthy and diabetic mice, both before and after the application of an drug agonist. After a preliminary analysis, you found that the residuals in your ANOVA model exhibited unequal variance across groups. You said you had tried a log-transform of your data, which did not correct the issue. You said you were considering non-linear and non-parametric methods of analysis, for example Naka-Rushton curves. You were also wondering how to proceed with a power analysis.

The conceptual design employed was the two-way repeated measures ANOVA. The two factors under consideration were light intensity and presence of diabetes. However, there is no random assignment of treatments in this experiment, a fact that a straightforward ANOVA design does not account for.

2 Model Considerations

From how it appears to us, there are three factors of experimental significance here. Those are diabetic/non-diabetic, light intensity, and drug treatment. In this case, it is not really a two-factor but a three-factor ANOVA that should be employed.

Also worth taking into consideration is whether or not to include "interactive" effects in the statistical model. These are effects of one independent variable on the outcome that depend on the level of another independent variable (a more in-depth explanation here). It may be worth adding interaction effects into the statistical model, as it would provide a greater level of refinement in terms of finding the specific combination of drug, diabetes, and light intensity that produces the optimal response.

In terms of the model choice, randomization is important to consider. You're not randomizing the application of treatments, which makes it more difficult to separate the variance attributable to different treatments. We can tolerate it, however, if randomization is not present—often randomization is not possible. In this case, the concept of a linear mixed model (also called split-plot) with grouped correlation effects is suitable.

In a linear mixed model, there are three types of factors: the subject, between-subject, and within-subject. The subjects are the individual retinal cells of the mice. They are considered a random sample and exhibit sampling variance, hence they are specified as a random factor. The between-subject factor is the factor that

varies from cell to cell. In this case it is diabetes. The within-subject factors are light intensity and agonist.

Ideally, the within-subject factors would be randomized. However, we can use a grouped correlation structure to account for this. We claim there is some time-dependent correlation for each subject, where one observation influences the successive observation. Such a correlation structure is ostensibly for repeated measures, but here it also captures the dependence due to the inseparability of ordering and treatment.

3 Analysis

Performing this analysis in R is relatively simple. The package *nlme* contains functions for analyzing mixed- and random-effects models. You can create a linear model object with the function:

```
model <- nlme::lme(fixed, data, random, correlation, weights, subset, method,
    na.action, control, contrasts = NULL, keep.data = TRUE)
```

Where fixed are the fixed effects, random are the random effects, and correlation is the correlation structure. In the case of our model, it would look something like:

```
model <- nlme::lme(RESPONSE ~ DRUG*DISEASE*LIGHT,
                   data = mice.dataframe,
                   random = ~1 | CELL,
                   correlation = corAR1(form=~1|CELL))
```

Where mice.dataframe is a data frame containing the variables RESPONSE, DRUG, DISEASE, LIGHT, and CELL. Functions like *anova, fixed.effects, intervals* exist for significance testing, finding effect estimates and finding confidence intervals, respectively. More regarding R analysis is included in the attached .R file and accompanying README.

Performing a power/sample size analysis of this model is more demanding, and will require careful consideration of the factors on which to focus the analysis. Informing the analysis with a contrast of interest will also increase the power of the model. You can read about contrasts here.

4 Model Assumptions

Normalization

We did not consider this during the meeting, but upon further inspection of the supplied data files, we had some questions about the normalization. It says on the supplied Statistical Questions PDF that you were normalizing by dividing by the maximum observation before treatment. However, in the Data_Diabetic vs Control Excel sheet (Norm Q and Norm Peak), it appears each response was divided by the response at light intensity 950000, regardless of drug application.

As statisticians, we would think there are two appropriate maxima to divide by: the maximum of all observations on one cell (both before and after drug application), and the maximum across every single observation. To understand this, consider that the cell maximum contains information about the entire cell, which is not a factor of interest. Information about the entire population is similarly not associated with a certain factor. However, if the normalization constant is associated with a drug treatment, then the maximum also contains information about the drug treatment. So dividing by this maximum would eliminate its information.

Another possibility often favored by statisticians is Z-score normalization, which is to subtract the overall mean and divide by overall standard deviation, i.e. $\frac{Y_i - \bar{Y}}{\sigma_Y}$ for each observation Y_i. Assuming the normality

of the observations, Z-scores approximate samples from a standard normal distribution, facilitating outlier recognition.

Violated Assumptions

Our recommendation for you regarding violations of model assumptions is to exhaust every last possibility with linear models. Generally, non-linear and non-parametric analyses are harder to understand and interpret than their linear counterparts. Below we present some options to consider within the framework of linear analysis:

- **Transformations**
  When heterogeneous variance is encountered, the first step is often to apply a log-transform to the response data. Log transforms are commonly used when data are sparsely dispersed (a classic example being animal body weights), and transformations can also correct non-normality in the data. Generally, any invertible function may be used to transform the data. You can explore a class of transformations called Box-Cox (link) for suitable transforms.
  We recommend you try the linear mixed model with raw and log-transformed responses, as the new model may change the way both dependent variables fit.

- **Weights**
  When you look at a residual scatter-plot and see some non-random spread of residuals, we can use weights to try and correct for it.

  If your variance is unequal between groups of diabetes, light, or drug, the simplest remedial measure is to weight by the variance of the different levels. So you would give each observation a weight of $1/\sigma_j^2$, where σ_j^2 is the variance of the jth level of the group. Then in the *lme* command from Section 2, you would set the "weights" parameter to a vector the length of your dataset, which contains one weight for each observation. This technique gives *less* weight to groups with *greater* variance, giving them less influence on the fit of your model and theoretically improving the fit. There is much more nuance possible in terms of choosing weights, such as choosing $1/|\epsilon_i|$ and $1/\epsilon_i^2$, where ϵ_i is the residual of the ith observation. More on weights here.

- **Generalized Linear Models**
  When neither of the two solutions above can remedy your issue, another tack you could take is to use a generalized linear model. Without going into too much detail, generalized linear models let you assume that your response variable is characterized by some distribution *other* than the normal distribution. You can read more about GLMs here.

5 Conclusion

To get started, we suggest first doing re-normalization, and then seeing how the linear mixed model fits the data. We have included all interaction terms in the model for the sake of information. We suggest thinking about what interactions are of experimental interest, and changing the RScript to only include those factors. The code fits the proposed model from a supplied CSV, then it provides significance tests for each factors and the effect estimates. Residual and QQ plots are also plotted to a file called "out.pdf" (more on QQ plots here). The two plots should be used to assess the validity of the model assumptions.

You also asked if we could do a power analysis, but unfortunately, we do not have time in the scope of this class to do that. Please feel free to contact any of us regarding options for doing a power analysis, and we can proceed from there.

Appendix

flood.R

```
#! /usr/bin/env Rscript
## AUTHOR: Nate Hattersley
suppressWarnings(library(nlme))


## FLAG TO CONTROL EXECUTION TYPE
if (!exists("COMMANDLINE"))
    COMMANDLINE <- F
if (!exists("QUIET"))
    QUIET <- F

## FIND A FILE
if (COMMANDLINE) {
    args <- commandArgs(TRUE)
    if (length(args) < 1) {
        stop("Please supply file name")
    }
}
infile <- ifelse(COMMANDLINE, args[1], file.choose())


## OPEN FILE AND MAKE A DATA FRAME
mice.df <- read.csv(infile, colClasses =
                      c(rep("factor", 4), "numeric"))
colnames(mice.df) <- toupper(colnames(mice.df))

## CHECK DATA INTEGRITY
if (ncol(mice.df) != 5) {
    stop(
        "Incorrect number of columns. Please refer to the README for formatting instructions."
    )
}
correct <- c("CELL", "DISEASE", "DRUG", "LIGHT", "RESPONSE")
for (i in 1:5) {
    if (correct[i] != colnames(mice.df)[i]) {
        stop(
            "Incorrect header names. Please refer to the README for formatting instructions."
        )
    }
}

## CREATE THE MODEL
mice.lme <- lme(
    RESPONSE ~ DRUG * LIGHT * DISEASE,
    random = ~ 1 | CELL,
    data = mice.df,
    correlation = corAR1(form = ~ 1 | CELL)
)

if (!QUIET) {
    ## OUTPUT SUMMARIES AND PLOTS
```

```
    cat("\nSIGNIFICANCE ESTIMATES\n\n")
    anova(mice.lme)
    cat("\nFIXED EFFECTS ESTIMATES\n\n")
    fe <- fixed.effects(mice.lme)
    for (i in 1:length(fe)) {
        cat(sprintf("\t\t\t\t\t%.4f\r%s\n",
                    fe[i], names(fe)[i]))
    }


    res <- residuals(mice.lme)

    pdf("out.pdf")
    plot(mice.lme, main = "Residual Plot")
    qqnorm(res)
    qqline(res)
    invisible(dev.off())
    cat("\n\nPlots written to out.pdf.\n")
}

rm(COMMANDLINE,QUIET)
```

README

```
README for flood.R usage


SETUP
The file flood.R is an executable file and meant to be used on the command line. Doing this
    requires a Linux or Mac operating system. If the file is going to be executed from inside an R
    environment like RStudio, you must change line 8 of the file to:

COMMANDLINE <- FALSE


If you are going to execute the script from the command line, you must make your file executable.
    That can be done in Terminal with the command:

chmod +x flood.R




USAGE
The script is meant to read in a CSV file with five columns. These columns, in order, are CELL,
    DISEASE, DRUG, LIGHT, RESPONSE. The first line of the CSV file must contain these names.
    Capitalization doesnt matter. The CELL column identifies which cell the observation comes
    from, DISEASE identifies whether or not it is diabetic, DRUG identifies whether or not the
    agonist has been applied, LIGHT identifies the light intensity, and RESPONSE is the normalized
    response.


To run on the command line (preferred), open a terminal window in the folder containing the R
    file, and type:
```

```
./flood.R data.csv

Where data.csv is name of the csv file, in the same folder.



The script will run anova to obtain significance estimates on the different factors. Then it will
    run fixed.effects to calculate effects estimates for each level of each factor. Finally, it
    will plot residual and qq plots to out.pdf. You will find out.pdf in the same folder as you
    ran the script if you run it on the command line. Otherwise, you will find it in your RHOME
    directory.
```
