Statistical analysis of ABR thresholds using GLMMs

# Abbreviations

ABR, auditory brainstem response; AIC, Akaike information criterion; ANOVA, analysis of deviance; KO, knock-out; HET, heterozygous; WT, wild type.

# Introduction

This writing describes step-by-step processes to analyze the auditory brainstem response (ABR) thresholds. Analysis included a random effect, several fixed effects, and a response - ABR thresholds. How well the models fit was determined by the Akaike information criterion (AIC), corrected for a small sample size (AICc)1.

Note that anova refers to analysis of deviance for use in generalized linear models2. In R, anova() is different from aov(), which is the typical analysis of variance used for a response that is assumed to follow a normal distribution. In this writing, anova() was used to explore several different non-normal (non-Gaussian) distributions.

Following R packages3 were used in this analysis: tidyverse4, MuMin5, car6, lme47, and performance8. A replication of the analysis would take 4 hours on an Intel ® Core i7 Laptop with 32.0 GB RAM. The raw data are available in **Table S1**, and it can be used to perform the following analysis. R commands are written in red.

# Data extraction

ABR data were extracted from the Tucker Davis System. Thresholds were determined via MatLab custom code, and the rows and columns were transposed using OFFSET, ROUNDUP, and MOD function on Microsoft Excel for easier data handling in R. Thresholds were visually confirmed afterwards. Threshold column was combined with columns for sample ID, genotype, age, treatment, and frequency. The data were saved as a tab separated text file, as shown below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Gt | Wk | Tm | Fr | Th |
| example\_1 | 3. WT | W06 | None | F08 | 30 |
| example\_2 | 1. KO | W08 | None | F16 | 40 |
| example\_3 | 3. WT | W12 | None | F24 | 50 |
| example\_4 | 2. HET | W06 | HNK | F32 | 60 |
| example\_5 | 3. WT | W08 | HNK | F36 | 30 |
| example\_1 | 3. WT | W12 | None | F08 | 40 |
| example\_2 | 3. WT | W06 | None | F16 | 50 |

thresh.dat <- read.delim(file = "abrdata.txt ", sep = "\t", header = TRUE, stringsAsFactors = TRUE)

View(thresh.dat)

colnames(thresh.dat)

## [1] "Id" "Sx" "Gt" "Wk" "Tm" "Fr" "Th"

# Data cleaning

## Renaming variables

library(tidyverse)

newnames <- c(Id = "Id", Gt = "Genotype", wk = "Week", Tm = "Treatment", Fr = "Frequency", Th = "Threshold")

colnames(thresh.dat) <- newnames

colnames(thresh.dat)

## [1] "Id" "Genotype" "Week" "Treatment" "Frequency"

## [7] "Threshold"

## Deleting missing data

is.nan.data.frame <- function(x)

do.call(cbind, lapply(x, is.nan))

thresh.dat[is.nan(thresh.dat)] <- NA

thresh.dat <- thresh.dat[complete.cases(thresh.dat), ]

## Modifying categorical variables

The Genotype variable was cleaned and coded by stairstep contrasts9 with the levels “KO”, “Het”, and “WT” by an ascending order.

levels(thresh.dat$Genotype)

## [1] "1. KO" "2. HET" "3. WT"

thresh.dat <- thresh.dat %>%

mutate(Genotype = case\_when(

Genotype == "1. KO" ~ "KO",

Genotype == "2. HET" ~ "Het",

Genotype == "3. WT" ~ "WT"

))

ordered\_factor <- function(fact\_var) {

ord\_fact <- factor(fact\_var, ordered=TRUE)

categories <- levels(fact\_var)

n\_cat <- length(categories)

cont <- matrix(0, n\_cat, n\_cat-1)

cont[col(cont)<row(cont)] <- 1

rownames(cont) <- categories

colnames(cont) <- paste(categories[2:n\_cat], categories[1:(n\_cat-1)],

sep=" vs. ")

contrasts(ord\_fact) <- cont

return(ord\_fact)

}

thresh.dat$Genotype <- factor(thresh.dat$Genotype, c("KO", "Het", "WT"))

This was to ensure proper order. Otherwise, R would use alphabetical order of level names.

thresh.dat$Genotype <- ordered\_factor(thresh.dat$Genotype)

levels(thresh.dat$Genotype)

## [1] "KO" "Het" "WT"

“false factors” were found, in which numbers were recorded as if they were factors.

thresh.dat$Week <- gsub("W","",thresh.dat$Week)

thresh.dat$Week <- as.numeric(thresh.dat$Week)

thresh.dat$Frequency <- gsub("F","",thresh.dat$Frequency)

thresh.dat$Frequency <- as.numeric(thresh.dat$Frequency)

In R, factors are leveled alphabetically. Therefore, “None” was exclusively set as the reference level for Treatment.

thresh.dat$Treatment <- factor(thresh.dat$Treatment, c("None", "HNK"))

# Modeling

Given that this was a repeated measurement experiment, the analysis used mixed-level models with a random intercept by “Id”. This allowed accounting for between-sample variability. Mixed-level models (whether they be estimated with nlme, lme4, or other packages) are sensitive to inconsistent scales among variables (for example. if a variable is between 1 – 1000 and another variable is between 1-10, their scales are inconsistent, affecting model optimization). To prevent this problem, the variables “Threshold”, “Week”, and “Frequency” were scaled.

thresh.dat$Threshold <- scale(thresh.dat$Threshold, center = FALSE)

thresh.dat$Week <- scale(thresh.dat$Week)

thresh.dat$Frequency <- scale(thresh.dat$Frequency)

Modeling required the following packages:

library(MuMIn)

library(car)

library(lme4)

library(performance)

Subjects varied by genotype, and treatment. Measurements varied by week and frequency. This suggested a saturated model, in which week and frequency each could interact with genotype and treatment. However, normal distribution of the data was not guaranteed, so several commonly used distributions and links were explored for saturated models. This was done to allow the analysis to relax assumptions of normality and homoscedasticity associated with simpler types of analyses. Since a curvilinear relationship existed between frequency and threshold, models with a quadratic transformation of frequency were also explored.

It was previously observed in the **Results** section that *Sirt3-/-* mice are susceptible to early-onset hearing loss and that honokiol treatment could benefit them. Interactions among genotype, and treatment were investigated. This observational approach was important to prevent relying solely on the automated model selection.

mod1 <- lmer(Threshold ~ (Week + Frequency) \* (Genotype + Treatment) + (1 | Id), data = thresh.dat, REML = FALSE, na.action = "na.fail")

mod2 <- glmer(Threshold ~ (Week + Frequency) \* (Genotype + Treatment) + (1 | Id), data = thresh.dat, family = gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod3 <- glmer(Threshold ~ (Week + Frequency) \* (Genotype + Treatment) + (1 | Id), data = thresh.dat, family = Gamma(link = "identity"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod4 <- glmer(Threshold ~ (Week + Frequency) \* (Genotype + Treatment) + (1 | Id), data = thresh.dat, family = Gamma(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod5 <- glmer(Threshold ~ (Week + Frequency) \* (Genotype + Treatment) + (1 | Id), data = thresh.dat, family = inverse.gaussian(link = "identity"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod6 <- glmer(Threshold ~ (Week + Frequency) \* (Genotype + Treatment) + (1 | Id), data = thresh.dat, family = inverse.gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod7 <- lmer(Threshold ~ (Week + poly(Frequency, degree = 2)) \* (Genotype + Treatment) + (1 | Id), data = thresh.dat, REML = FALSE, na.action = "na.fail")

mod8 <- glmer(Threshold ~ (Week + poly(Frequency, degree = 2)) \* (Genotype + Treatment) + (1 | Id), data = thresh.dat, family = gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod9 <- glmer(Threshold ~ (Week + poly(Frequency, degree = 2)) \* (Genotype + Treatment) + (1 | Id), data = thresh.dat, family = Gamma(link = "identity"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod10 <- glmer(Threshold ~ (Week + poly(Frequency, degree = 2)) \* (Genotype + Treatment) + (1 | Id), data = thresh.dat, family = Gamma(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod11 <- glmer(Threshold ~ (Week + poly(Frequency, degree = 2)) \* (Genotype + Treatment) + (1 | Id), data = thresh.dat, family = inverse.gaussian(link = "identity"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod12 <- glmer(Threshold ~ (Week + poly(Frequency, degree = 2)) \* (Genotype + Treatment) + (1 | Id), data = thresh.dat, family = inverse.gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod13 <- lmer(Threshold ~ Week + Frequency + Genotype\*Treatment + (1 | Id), data = thresh.dat, REML = FALSE, na.action = "na.fail")

mod14 <- glmer(Threshold ~ Week + Frequency + Genotype\*Treatment + (1 | Id), data = thresh.dat, family = gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod15 <- glmer(Threshold ~ Week + Frequency + Genotype\*Treatment + (1 | Id), data = thresh.dat, family = Gamma(link = "identity"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod16 <- glmer(Threshold ~ Week + Frequency + Genotype\*Treatment + (1 | Id), data = thresh.dat, family = Gamma(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod17 <- glmer(Threshold ~ Week + Frequency + Genotype\*Treatment + (1 | Id), data = thresh.dat, family = inverse.gaussian(link = "identity"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod18 <- glmer(Threshold ~ Week + Frequency + Genotype\*Treatment + (1 | Id), data = thresh.dat, family = inverse.gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod19 <- lmer(Threshold ~ Week + poly(Frequency, degree = 2) + Genotype\*Treatment + (1 | Id), data = thresh.dat, REML = FALSE, na.action = "na.fail")

mod20 <- glmer(Threshold ~ Week + poly(Frequency, degree = 2) + Genotype\*Treatment + (1 | Id), data = thresh.dat, family = gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod21 <- glmer(Threshold ~ Week + poly(Frequency, degree = 2) + Genotype\*Treatment + (1 | Id), data = thresh.dat, family = Gamma(link = "identity"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod22 <- glmer(Threshold ~ Week + poly(Frequency, degree = 2) + Genotype\*Treatment + (1 | Id), data = thresh.dat, family = Gamma(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod23 <- glmer(Threshold ~ Week + poly(Frequency, degree = 2) + Genotype\*Treatment + (1 | Id), data = thresh.dat, family = inverse.gaussian(link = "identity"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

mod24 <- glmer(Threshold ~ Week + poly(Frequency, degree = 2) + Genotype\*Treatment + (1 | Id), data = thresh.dat, family = inverse.gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

Models with many predictors and interactions risked overfitting, so nested models were generated and compared via the second-order Akaike information criterion10. This was systematically done using the dredge function which gathered the subsets of the complex models.

dm1 <- dredge(mod1, fixed = c("Week", "Genotype", "Treatment", "Frequency"))

dm2 <- dredge(mod2, fixed = c("Week", "Genotype", "Treatment", "Frequency"))

dm3 <- dredge(mod3, fixed = c("Week", "Genotype", "Treatment", "Frequency"))

dm4 <- dredge(mod4, fixed = c("Week", "Genotype", "Treatment", "Frequency"))

dm5 <- dredge(mod5, fixed = c("Week", "Genotype", "Treatment", "Frequency"))

dm6 <- dredge(mod6, fixed = c("Week", "Genotype", "Treatment", "Frequency"))

dm7 <- dredge(mod7, fixed = c("Week", "Genotype", "Treatment", "poly(Frequency, degree = 2)"))

dm8 <- dredge(mod8, fixed = c("Week", "Genotype", "Treatment", "poly(Frequency, degree = 2)"))

dm9 <- dredge(mod9, fixed = c("Week", "Genotype", "Treatment", "poly(Frequency, degree = 2)"))

dm10 <- dredge(mod10, fixed = c("Week", "Genotype", "Treatment", "poly(Frequency, degree = 2)"))

dm11 <- dredge(mod11, fixed = c("Week", "Genotype", "Treatment", "poly(Frequency, degree = 2)"))

dm12 <- dredge(mod12, fixed = c("Week", "Genotype", "Treatment", "poly(Frequency, degree = 2)"))

dm13 <- dredge(mod13, fixed = c("Week", "Genotype", "Treatment", "Frequency"))

dm14 <- dredge(mod14, fixed = c("Week", "Genotype", "Treatment", "Frequency"))

dm15 <- dredge(mod15, fixed = c("Week", "Genotype", "Treatment", "Frequency"))

dm16 <- dredge(mod16, fixed = c("Week", "Genotype", "Treatment", "Frequency"))

dm17 <- dredge(mod17, fixed = c("Week", "Genotype", "Treatment", "Frequency"))

dm18 <- dredge(mod18, fixed = c("Week", "Genotype", "Treatment", "Frequency"))

dm19 <- dredge(mod19, fixed = c("Week", "Genotype", "Treatment", "poly(Frequency, degree = 2)"))

dm20 <- dredge(mod20, fixed = c("Week", "Genotype", "Treatment", "poly(Frequency, degree = 2)"))

dm21 <- dredge(mod21, fixed = c("Week", "Genotype", "Treatment", "poly(Frequency, degree = 2)"))

dm22 <- dredge(mod22, fixed = c("Week", "Genotype", "Treatment", "poly(Frequency, degree = 2)"))

dm23 <- dredge(mod23, fixed = c("Week", "Genotype", "Treatment", "poly(Frequency, degree = 2)"))

dm24 <- dredge(mod24, fixed = c("Week", "Genotype", "Treatment", "poly(Frequency, degree = 2)"))

dm1$call <- attr(dm1, "model.calls"); dmd1 <- data.frame(dm1); dmd1 <- dmd1[, c("AICc", "df", "call")]

dm2$call <- attr(dm2, "model.calls"); dmd2 <- data.frame(dm2); dmd2 <- dmd2[, c("AICc", "df", "call")]

dm3$call <- attr(dm3, "model.calls"); dmd3 <- data.frame(dm3); dmd3 <- dmd3[, c("AICc", "df", "call")]

dm4$call <- attr(dm4, "model.calls"); dmd4 <- data.frame(dm4); dmd4 <- dmd4[, c("AICc", "df", "call")]

dm5$call <- attr(dm5, "model.calls"); dmd5 <- data.frame(dm5); dmd5 <- dmd5[, c("AICc", "df", "call")]

dm6$call <- attr(dm6, "model.calls"); dmd6 <- data.frame(dm6); dmd6 <- dmd6[, c("AICc", "df", "call")]

dm7$call <- attr(dm7, "model.calls"); dmd7 <- data.frame(dm7); dmd7 <- dmd7[, c("AICc", "df", "call")]

dm8$call <- attr(dm8, "model.calls"); dmd8 <- data.frame(dm8); dmd8 <- dmd8[, c("AICc", "df", "call")]

dm9$call <- attr(dm9, "model.calls"); dmd9 <- data.frame(dm9); dmd9 <- dmd9[, c("AICc", "df", "call")]

dm10$call <- attr(dm10, "model.calls"); dmd10 <- data.frame(dm10); dmd10 <- dmd10[, c("AICc", "df", "call")]

dm11$call <- attr(dm11, "model.calls"); dmd11 <- data.frame(dm11); dmd11 <- dmd11[, c("AICc", "df", "call")]

dm12$call <- attr(dm12, "model.calls"); dmd12 <- data.frame(dm12); dmd12 <- dmd12[, c("AICc", "df", "call")]

dm13$call <- attr(dm13, "model.calls"); dmd13 <- data.frame(dm13); dmd13 <- dmd13[, c("AICc", "df", "call")]

dm14$call <- attr(dm14, "model.calls"); dmd14 <- data.frame(dm14); dmd14 <- dmd14[, c("AICc", "df", "call")]

dm15$call <- attr(dm15, "model.calls"); dmd15 <- data.frame(dm15); dmd15 <- dmd15[, c("AICc", "df", "call")]

dm16$call <- attr(dm16, "model.calls"); dmd16 <- data.frame(dm16); dmd16 <- dmd16[, c("AICc", "df", "call")]

dm17$call <- attr(dm17, "model.calls"); dmd17 <- data.frame(dm17); dmd17 <- dmd17[, c("AICc", "df", "call")]

dm18$call <- attr(dm18, "model.calls"); dmd18 <- data.frame(dm18); dmd18 <- dmd18[, c("AICc", "df", "call")]

dm19$call <- attr(dm19, "model.calls"); dmd19 <- data.frame(dm19); dmd19 <- dmd19[, c("AICc", "df", "call")]

dm20$call <- attr(dm20, "model.calls"); dmd20 <- data.frame(dm20); dmd20 <- dmd20[, c("AICc", "df", "call")]

dm21$call <- attr(dm21, "model.calls"); dmd21 <- data.frame(dm21); dmd21 <- dmd21[, c("AICc", "df", "call")]

dm22$call <- attr(dm22, "model.calls"); dmd22 <- data.frame(dm22); dmd22 <- dmd22[, c("AICc", "df", "call")]

dm23$call <- attr(dm23, "model.calls"); dmd23 <- data.frame(dm23); dmd23 <- dmd23[, c("AICc", "df", "call")]

dm24$call <- attr(dm24, "model.calls"); dmd24 <- data.frame(dm24); dmd24 <- dmd24[, c("AICc", "df", "call")]

dmd.sum <- rbind(dmd1, dmd2, dmd3, dmd4, dmd5, dmd6, dmd7, dmd8, dmd9, dmd10, dmd11, dmd12, dmd13, dmd14, dmd15, dmd16, dmd17, dmd18, dmd19, dmd20, dmd21, dmd22, dmd23, dmd24)

dmd.sum <- dmd.sum[order(dmd.sum$AICc),]

head(dmd.sum)

AICc df

149 -845.0174 16

169 -842.1429 18

69 -835.5525 15

89 -835.3456 17

1411 -823.8736 16

1611 -821.6232 18

Call

**#149**

glmer(formula = Threshold ~ Genotype + poly(Frequency, degree = 2) + , Treatment + Week + (1 | Id) + Genotype:poly(Frequency, degree = 2) + , poly(Frequency, degree = 2):Treatment + Treatment:Week, data = thresh.dat, , family = Gamma(link = "log"), control = glmerControl(optimizer = "bobyqa", , optCtrl = list(maxfun = 1e+05)), na.action = "na.fail")

**#169**

glmer(formula = Threshold ~ Genotype + poly(Frequency, degree = 2) + , Treatment + Week + (1 | Id) + Genotype:poly(Frequency, degree = 2) + , Genotype:Week + poly(Frequency, degree = 2):Treatment + Week:Treatment, , data = thresh.dat, family = Gamma(link = "log"), control = glmerControl(optimizer = "bobyqa", , optCtrl = list(maxfun = 1e+05)), na.action = "na.fail")

**#69**

glmer(formula = Threshold ~ Genotype + poly(Frequency, degree = 2) + , Treatment + Week + (1 | Id) + Genotype:poly(Frequency, degree = 2) + , poly(Frequency, degree = 2):Treatment, data = thresh.dat, , family = Gamma(link = "log"), control = glmerControl(optimizer = "bobyqa", , optCtrl = list(maxfun = 1e+05)), na.action = "na.fail")

**#89**

glmer(formula = Threshold ~ Genotype + poly(Frequency, degree = 2) + , Treatment + Week + (1 | Id) + Genotype:poly(Frequency, degree = 2) + , Genotype:Week + poly(Frequency, degree = 2):Treatment, data = thresh.dat, , family = Gamma(link = "log"), control = glmerControl(optimizer = "bobyqa", , optCtrl = list(maxfun = 1e+05)), na.action = "na.fail")

**#1411**

glmer(formula = Threshold ~ Genotype + poly(Frequency, degree = 2) + , Treatment + Week + (1 | Id) + Genotype:poly(Frequency, degree = 2) + , poly(Frequency, degree = 2):Treatment + Treatment:Week, data = thresh.dat, , family = inverse.gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", , optCtrl = list(maxfun = 1e+05)), na.action = "na.fail")

**#1611**

glmer(formula = Threshold ~ Genotype + poly(Frequency, degree = 2) + , Treatment + Week + (1 | Id) + Genotype:poly(Frequency, degree = 2) + , Genotype:Week + poly(Frequency, degree = 2):Treatment + Week:Treatment, , data = thresh.dat, family = inverse.gaussian(link = "log"), , control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05)), , na.action = "na.fail")

…

Typically, the model with the lowest AICc is selected with this sort of comparison, unless a more parsimonious model (fewer effects) is present with an AICc within 2 of the lowest11. Although there were models with lower AICc values, some failed to converge due to high gradient values. Therefore, we selected the best-converged model (#1411).

#1411

Threshold.mod <- glmer(formula = Threshold ~ Genotype + poly(Frequency, degree = 2) + Treatment + Week + (1 | Id) + Genotype:poly(Frequency, degree = 2) + poly(Frequency, degree = 2):Treatment + Treatment:Week, data = thresh.dat, family = inverse.gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05)), na.action = "na.fail")

Statistics were estimated for the model, including analysis of deviance (ANOVA) and Nakagawa’s marginal pseudo-*R2* 12.

omni.anov <- anova(update(Threshold.mod, ~ 1 + (1 | Id)), Threshold.mod)

marg.anov <- **Anova**(Threshold.mod, type = "III")  
omni.r2 <- r2(Threshold.mod)$R2\_marginal

marg.r2s <- omni.r2 - c(r2(glmer(formula = Threshold ~

poly(Frequency, degree = 2) +

Treatment +

Week +

poly(Frequency, degree = 2):Treatment+

Treatment: Week +

(1 | Id)

, data = thresh.dat, family = inverse.gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))))$R2\_marginal,

r2(glmer(formula = Threshold ~

Genotype +

Treatment +

Week +

Treatment: Week +

(1 | Id)

, data = thresh.dat, family = inverse.gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))))$R2\_marginal,

r2(glmer(formula = Threshold ~

Genotype +

poly(Frequency, degree = 2) +

Week +

Genotype:poly(Frequency, degree = 2) +

(1 | Id)

, data = thresh.dat, family = inverse.gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))))$R2\_marginal,

r2(glmer(formula = Threshold ~

Genotype +

poly(Frequency, degree = 2) +

Treatment +

Genotype:poly(Frequency, degree = 2) +

poly(Frequency, degree = 2):Treatment+

(1 | Id)

, data = thresh.dat, family = inverse.gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))))$R2\_marginal,

r2(glmer(formula = Threshold ~

Genotype +

poly(Frequency, degree = 2) +

Treatment +

Week +

poly(Frequency, degree = 2):Treatment+

Treatment: Week +

(1 | Id)

, data = thresh.dat, family = inverse.gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))))$R2\_marginal,

r2(glmer(formula = Threshold ~

Genotype +

poly(Frequency, degree = 2) +

Treatment +

Week +

Genotype:poly(Frequency, degree = 2) +

Treatment: Week +

(1 | Id)

, data = thresh.dat, family = inverse.gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))))$R2\_marginal,

r2(glmer(formula = Threshold ~

Genotype +

poly(Frequency, degree = 2) +

Treatment +

Week +

Genotype:poly(Frequency, degree = 2) +

poly(Frequency, degree = 2):Treatment+

(1 | Id)

, data = thresh.dat, family = inverse.gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))))$R2\_marginal

)

omni.sum <- cbind(omni.anov[2,6:8], omni.r2)

#adjust according to the number of fixed effects

rownames(omni.sum) <- "Model"

colnames(omni.sum) <- c("Chisq", "Df", "Pr(>Chisq)", "R2")

marg.anov$R2 <- c(NA, marg.r2s) # Fixed length mismatch

marg.sum <- marg.anov

colnames(marg.sum) <- c("Chisq", "Df", "Pr(>Chisq)", "R2")

# Model summary

summary(Threshold.mod, correlation = TRUE)

A screenshot of a computer program

AI-generated content may be incorrect.

In order to add the comparison of WT vs. KO (currently missing above), the genotype factor was temporarily re-leveled, so that KO was the reference group.

thresh.dat$Genotype <- factor(as.character(thresh.dat$Genotype), ordered = FALSE)

thresh.dat$Genotype <- relevel(thresh.dat$Genotype, ref = "KO")  
thresh.dat$Genotype <- factor(thresh.dat$Genotype)

thresh.dat$Genotype <- relevel(thresh.dat$Genotype, ref = "KO")

Threshold.mod <- glmer(Threshold ~ Genotype + poly(Frequency, degree = 2) + Treatment + Week + Genotype:poly(Frequency, degree = 2) + poly(Frequency, degree = 2):Treatment + Treatment:Week + (1 | Id), data = thresh.dat, family = inverse.gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

summary(Threshold.mod, correlation = TRUE)

A screenshot of a computer

AI-generated content may be incorrect.

Note that the family was “inverse gaussian”. This means that the distribution of ABR thresholds was assumed to be the inverse normal distribution. This was a highly right-skewed distribution, and the response was related to this distribution in a logarithmic fashion (log link), so, it was regarded as even more skewed **(Fig. S2)**.

The correlations of fixed primary effects were examined (this part was too large to be displayed here, but it will be shown in R). None of the independent predictors correlated highly, indicating that multicollinearity was not a problem.

Attempting to interpret the coefficients was not done, because there was no reasonable way to directly interpret them, given multiple interactions.

# Model Statistics

print(omni.sum, digits = 4)

Chisq Df Pr(>Chisq) R2

Model 821.3 13 3.747e-167 0.1881

print(marg.sum, digits = 4)

A screenshot of a computer

AI-generated content may be incorrect.

The above table describes the omnibus ANOVA (analysis of deviance) and R2 for the entire model. The model is overall significant with p-value being less than 0.05.

The following table was generated to summarize the fixed effects. This table allows comparison between genotypes, treatments, and their interactions with frequency and time. By looking at Estimate, Standard Error, z-value, and p-value for each term, it is possible to interpret the contribution of predictors in the model.

thresh.dat$Genotype <- as.character(thresh.dat$Genotype)

thresh.dat$Genotype <- factor(thresh.dat$Genotype, levels = c("KO", "Het", "WT"))

Threshold.mod <- glmer(

Threshold ~ Genotype + poly(Frequency, degree = 2) + Treatment + Week + Genotype:poly(Frequency, degree = 2) + poly(Frequency, degree = 2):Treatment + Treatment:Week + (1 | Id), data = thresh.dat, family = inverse.gaussian(link = "log"), control = glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e5)), na.action = "na.fail")

library(broom.mixed)

coef\_table <- tidy(Threshold.mod, effects = "fixed")

print(coef\_table, digits = 3)

Table S1. Fixed effect estimates from the final GLMM (KO as reference)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Term** | **Estimate** | **Std. Error** | **z** | ***p*-value** |
| (Intercept) | –0.020 | 0.036 | –0.56 | 0.578 |
| Genotype: Het vs KO | –0.070 | 0.048 | –1.46 | 0.145 |
| Genotype: WT vs KO | –0.114 | 0.045 | –2.51 | 0.012 |
| poly(Freq)₁ | 7.61 | 0.367 | 20.7 | < 10⁻⁹⁵ |
| poly(Freq)₂ | 5.16 | 0.349 | 14.8 | < 10⁻⁴⁹ |
| Treatment: HNK | 0.009 | 0.050 | 0.18 | 0.856 |
| Week | 0.029 | 0.004 | 6.58 | 4.65×10⁻¹¹ |
| Het × poly(Freq)₁ | –1.23 | 0.489 | –2.52 | 0.012 |
| WT × poly(Freq)₁ | –1.01 | 0.436 | –2.31 | 0.021 |
| Het × poly(Freq)₂ | 0.113 | 0.465 | 0.243 | 0.808 |
| WT × poly(Freq)₂ | –0.140 | 0.418 | –0.336 | 0.737 |
| poly(Freq)₁ × HNK | –3.03 | 0.459 | –6.60 | 4.2×10⁻¹¹ |
| poly(Freq)₂ × HNK | –1.57 | 0.444 | –3.53 | 4.2×10⁻⁴ |
| HNK × Week | –0.0258 | 0.0094 | –2.76 | 0.0058 |

# References

1. Lee H, Ghosh SK. Performance of Information Criteria for Spatial Models. *J Stat Comput Simul* **79**, 93-106 (2009).

2. McCullagh PN, J. *Generalized Linear Models (Chapman & Hall/CRC Monographs on Statistics and Applied Probability)*, 2 edn. Chapman & Hall/CRC (1989).

3. R Core Team. R: A language and environment for statistical computing Vienna, Austria: R Foundataion for Statistical Computing.).

4. Wickham H*, et al.* Welcome to the Tidyverse. *Journal of Open Source Software* **4**, 1686 (2019).

5. Barton K. Package 'MuMIn'.).

6. Fox JW, S; Price, B. Package 'car'.). 3.1-3 edn (2024).

7. Bates D, Mächler M, Bolker B, Walker S. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software* **67**, 1 - 48 (2015).

8. Daniel Ludecke MSB-S, Indrajeet Patil, Philip Waggoner, Dominique Makowski. performance: An R Package for Assessment, Comparison and Testing of Statistical Models. *Journal of Open Source Software* **6**, 3139 (2021).

9. Gullickson A. Better contrasts for ordinal variables in R.) (2020).

10. Kenneth P. Burnham DRA. *Model Selection and Multimodel Inference: A practical information theoretic approach*, 2 edn. Spring New York (2002).

11. Greenwood M. *Intermediate Statistics with R*. Montana State University (2022).

12. Nakagawa S, Johnson PCD, Schielzeth H. The coefficient of determination R(2) and intra-class correlation coefficient from generalized linear mixed-effects models revisited and expanded. *J R Soc Interface* **14**, (2017).