

Ebola_GP82_analysis.r

max

Fri Jan 5 17:57:53 2018

```
library(lme4)

## Loading required package: Matrix
library(ggplot2)
#####
source("diehl_aux.r")
#####
## Data preparation
raw <- read.csv("../data/GP82_raw_data.csv", header = TRUE) ## "all" data
EVD_GP82 <- read.csv("../data/EVD_case_fatality_GP82.csv", header = TRUE) ## clean data
coord_data <- read.csv("../data/coord_data_locations.csv", header = TRUE)
pop_level <- read.csv("../data/population_level.csv", header = TRUE)
EVD_GP82 <- merge(EVD_GP82, coord_data, by = "location")
EVD_GP82 <- merge(EVD_GP82, pop_level, by = "location")
EVD_GP82 <- subset(EVD_GP82, country.y == "GIN") ## restrict attention to Guinea
xtabs(~ Outcome + GP82, EVD_GP82)

##           GP82
## Outcome    A  V
##   died      45 93
##   survived  23 18

nrow(EVD_GP82)

## [1] 179

#####
### Exploratory statistical analysis (ESA)

## Question 1: Do Ct values according to EBOV variant ?
# All data available
all_A <- raw$Assay.CT.Value[raw$GP82 == "A"]
sum(!is.na(all_A)) ## valid observations

## [1] 97

all_V <- raw$Assay.CT.Value[raw$GP82 == "V"]
sum(!is.na(all_V))

## [1] 216

t.test(all_A, all_V)

##
## Welch Two Sample t-test
##
## data:  all_A and all_V
## t = 0.69147, df = 157.85, p-value = 0.4903
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
## -0.6943069 1.4423291
## sample estimates:
## mean of x mean of y
## 20.05938 19.68537

# Just the data we'll use for further analyses
filtered_A <- EVD_GP82$Assay.CT.Value[EVD_GP82$GP82 == "A"]
sum(!is.na(filtered_A))

## [1] 68

filtered_V <- EVD_GP82$Assay.CT.Value[EVD_GP82$GP82 == "V"]
sum(!is.na(filtered_V))

## [1] 111

t.test(filtered_A, filtered_V)

##
## Welch Two Sample t-test
##
## data: filtered_A and filtered_V
## t = 1.8356, df = 112.02, p-value = 0.06907
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.09118178 2.38791734
## sample estimates:
## mean of x mean of y
## 20.17765 19.02928

#####
pdf("../plots/Ct_per_genotype.pdf")
ggplot(EVD_GP82) +
  geom_boxplot(aes(x = GP82, y = Assay.CT.Value, fill = GP82), colour = c("grey80", "black")) +
  scale_fill_manual(values = c("black", "red")) +
  theme_bw()
dev.off()

## pdf
## 2

pdf("../plots/Ct_per_genotype_and_outcome.pdf")
ggplot(EVD_GP82) +
  geom_boxplot(aes(x = Outcome, y = Assay.CT.Value, fill = Outcome)) +
  facet_grid(~GP82)
dev.off()

## pdf
## 2

#### Now let's look at odds ratios
(CrossTab_gp82 <- xtabs(~ GP82 + Outcome , EVD_GP82) )

##      Outcome
## GP82 died survived
##      A      45      23
##      V      93      18
```

```
summary(CrossTab_gp82)
```

```
## Call: xtabs(formula = ~GP82 + Outcome, data = EVD_GP82)
## Number of cases in table: 179
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 7.403, df = 1, p-value = 0.006511
```

```
a <- CrossTab_gp82["V", "died"]
b <- CrossTab_gp82["V", "survived"]
c <- CrossTab_gp82["A", "died"]
d <- CrossTab_gp82["A", "survived"]
SqrT <- sqrt(1/a + 1/b + 1/c + 1/d)
(rawOR <- (a*d)/(b*c) )
```

```
## [1] 2.640741
```

```
alpha <- .95
( lwrOR <- exp(log(rawOR) + qnorm((1 - alpha)/2) * SqrT) )
```

```
## [1] 1.295563
```

```
( uprOR <- exp(log(rawOR) + qnorm((1 + alpha)/2) * SqrT) )
```

```
## [1] 5.382609
```

```
## Relative risk
m.RR <- log({a/(a+b)} / {c/(c + d)})
sdRR <- sqrt( (1/a + 1/c) - (1/(a + b) + 1/(c+d)))
Z <- qnorm(.975)
CI.RR <- c( exp(m.RR - Z*sdRR), exp(m.RR + Z*sdRR))
c(exp(m.RR), CI.RR)
```

```
## [1] 1.266066 1.048446 1.528856
```

```
### Ok, but what happens if we try to correct for other variables?
```

```
### First, some data preparation
```

```
forFit <- EVD_GP82
```

```
forFit$Assay.CT.Value <- -standz(EVD_GP82$Assay.CT.Value) ## invert sign because of lower CT = higher v
```

```
forFit$pdensMN <- standz(EVD_GP82$pdensMN)
```

```
forFit$geconMN <- standz(EVD_GP82$geconMN)
```

```
forFit$ttt50kMN <- standz(EVD_GP82$ttt50kMN)
```

```
forFit$cumCases <- standz(log(EVD_GP82$cumCases))
```

```
forFit$GP82 <- ifelse(EVD_GP82$GP82 == "V", 1, 0) ## GP82-A as baseline
```

```
forFit$Outcome <- ifelse(EVD_GP82$Outcome == "died", 1, 0)
```

```
simple_glm <- glm(Outcome ~ Assay.CT.Value + GP82, data = forFit, family = "binomial")
summary(simple_glm)
```

```
##
```

```
## Call:
```

```
## glm(formula = Outcome ~ Assay.CT.Value + GP82, family = "binomial",
##      data = forFit)
```

```
##
```

```
## Deviance Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -2.2077  0.2806  0.4394  0.6210  1.8405
##
```

```
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.0189    0.3046   3.345 0.000824 ***
## Assay.CT.Value  2.0395    0.4308   4.734 2.2e-06 ***
## GP82           0.7400    0.4028   1.837 0.066199 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 192.65  on 178  degrees of freedom
## Residual deviance: 156.94  on 176  degrees of freedom
## AIC: 162.94
##
## Number of Fisher Scoring iterations: 4

getOdds(simple_glm, hier = FALSE)

## Waiting for profiling to be done...

##               est      2.5 %    97.5 %
## (Intercept)    2.770133 1.5597542  5.187539
## Assay.CT.Value  7.686444 3.4580497 18.918033
## GP82           2.095999 0.9492025  4.647809

## (corrected) relative risks
simple.pois.glm <- glm(Outcome ~ Assay.CT.Value + GP82, data = forFit, family = "poisson")
exp(cbind(coefficients(simple.pois.glm), confint(simple.pois.glm)))

## Waiting for profiling to be done...

##               2.5 %    97.5 %
## (Intercept)    0.665769 0.4891752 0.8805533
## Assay.CT.Value  1.654133 1.1447092 2.4290820
## GP82           1.196355 0.8425497 1.7240292

NewCt <- seq(min(forFit$Assay.CT.Value), max(forFit$Assay.CT.Value), length.out = 100)
mut <- c(0, 1)
newDat <- expand.grid(NewCt, mut)
names(newDat) <- c("Assay.CT.Value", "GP82")
Pred.logit <- predict(simple_glm, newdata = newDat, se.fit = TRUE)
invlogit <- function(x) exp(x)/(1 + exp(x))
alpha <- 0.95
InvLogitPred <- data.frame(lwr = invlogit(Pred.logit$fit + qnorm((1-alpha)/2) * Pred.logit$se.fit) ,
                          mean = invlogit(Pred.logit$fit),
                          upr = invlogit(Pred.logit$fit + qnorm((1 + alpha)/2) * Pred.logit$se.fit))
toPlot_preds <- data.frame(newDat, InvLogitPred)
toPlot_preds$GP82 <- ifelse(toPlot_preds$GP82 == 1, "V", "A")

pdf("../plots/predicted_fatality_rates.pdf")
ggplot(toPlot_preds, aes(x = Assay.CT.Value, y = mean)) +
  # ggtitle("Predicted fatality rates") +
  geom_ribbon(aes(ymin = lwr, ymax = upr, fill = GP82), alpha = .2) +
  scale_x_continuous("Transformed Ct", expand = c(0, 0)) +
  scale_y_continuous("Fatality rate", expand = c(0, 0)) +
  scale_fill_manual(values = c("black", "red")) +
  geom_line(aes(colour = GP82), size = 1) +
```

```

scale_color_manual(values = c("black", "red")) +
guides(fill = FALSE) +
theme_bw()
dev.off()

## pdf
## 2

model_of <- glmer(Outcome ~ Assay.CT.Value + GP82 + (cumCases|location),
                  data = forFit, family = "binomial")
summary(model_of)

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: Outcome ~ Assay.CT.Value + GP82 + (cumCases | location)
## Data: forFit
##
##      AIC      BIC    logLik deviance df.resid
##  167.5    186.7    -77.8    155.5      173
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.5246  0.1808  0.2943  0.4408  2.1712
##
## Random effects:
## Groups Name Variance Std.Dev. Corr
## location (Intercept) 0.2539  0.5039
## cumCases 0.4845  0.6961  1.00
## Number of obs: 179, groups: location, 17
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.9824    0.3862   2.544  0.0110 *
## Assay.CT.Value  2.1619    0.4935   4.381 1.18e-05 ***
## GP82           0.7973    0.4561   1.748  0.0804 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) A.CT.V
## Assay.CT.V1  0.252
## GP82         -0.668 -0.078

getOdds(model_of)

## Computing profile confidence intervals ...

## Warning in if (parm == "theta_") {: the condition has length > 1 and only
## the first element will be used

## Warning in if (parm == "beta_") {: the condition has length > 1 and only
## the first element will be used

##              est      2.5 %    97.5 %
## (Intercept)  2.670811 1.1461366 5.921091
## Assay.CT.Value 8.687964 3.5723089 25.581957

```

```
## GP82          2.219620 0.9045236  5.671575
RR <- ranef(model_of, condVar = TRUE)
names(RR) <- " "
names(RR[[1]]) <- c("Location-specific intercept", "coefficient of cumulative number of cases")

pdf("../plots/varying_intercepts.pdf")
lattice::dotplot(RR)

## $` `
dev.off()

## pdf
## 2
```