Selection of CIPN Cases and Controls in N08CB

Introduction

This document describes how EORTC QLQ-CIPN20 responses (sensory items only, #1–#6, #9, #10, and #18) from NCCTG/Alliance trial N08CB were used to score patients for CIPN susceptibility and select those most and least susceptible as case and control groups for genetic association testing.

We used a Rasch-type model to obtain for each patient an estimated rate of change in symptom severity (according to the CIPN20 responses) per unit increase in cumulative oxaliplatin dose. An upward trend in symptom severity with increasing cumulative dose, observed consistently across CIPN20 sensory items, is taken to indicate CIPN susceptibility, and no trend or a downward trend are taken to indicate less or no CIPN susceptibility. Patients ranked highest based on the rate are selected as "cases", and those ranking lowest are selected as "controls".

Rasch-Type Model

For each patient, the CIPN20 responses consist of a symptom severity (1 = Not at All, 2 = A Little, 3 = Quite a Bit, or 4 = Very Much, referring to how much the patient has experienced a particular symptom) for each of the 9 sensory questionnaire items, for up to 12 cycles of chemotherapy. For subject i, CIPN20 item j, and chemotherapy cycle k, we have cumulative oxaliplatin dose received up to but not including cycle k, denoted x_{ijk}, and the patient's response, denoted Y_{ijk}. In our Rasch-type model,

$$\log \mathrm{E}(Y_{ijk}) = \log \mu_{ijk} = (\alpha_i + \gamma_j) + (\beta_i + \delta_j) x_{ijk} \tag{1}$$

where α_i and β_i are intercept and slope parameters associated with patients, and γ_j and δ_j are intercept and slope parameters associated with questionnaire items. To make the model identifiable, it can be assumed that $\gamma_1 = \delta_1 = 0$. With that constraint, the model states that each patient has a base intercept and slope corresponding to item 1 on the questionnaire. Then the intercept and slope are modified by γ_j and δ_j for $j = 2, \ldots, 6, 9, 10, 18$. We are concerned only with the patient slope parameter β_i.

We fit this model assuming $Y_{ijk} \sim \mathrm{Poisson}(\mu_{ijk})$ using the R function `glm`. Although the responses are clearly not Poisson distributed, this model is intended only to provide a ranking of patients, so we aren't concerned with achieving an exact fit.

The model is fit on all non-missing responses after excluding patients who had responses for fewer than three unique treatment cycles.

Patient Selection

From the fit of the Rasch-type model, we obtain estimates and standard errors (SEs) of the patient slope parameters. To account for uncertainty in the estimates, we rank patients based on estimate − SE (as a kind of lower bound for the slope) for the selection of cases and based on estimate + SE (as a kind of upper bound) for the selection of controls. The top and bottom quartiles in these respective rankings are selected as the cases and controls.

Implementation

We analyzed the CIPN20 responses using R version 3.4.2. R code demonstrating our analysis will be shown as below.

```
library(dplyr)
library(ggplot2)
library(tidyr)
set.seed(410)
options(stringsAsFactors = FALSE)
theme_set(theme_light())
```

The relevant variables for this analysis are the patient identifier (`Patient`), treatment cycle (`Cycle`), cumulative oxaliplatin dose in mg per m^2 body surface area (`C_dose`), and CIPN20 sensory responses (`CIPN_1`, `CIPN_2`, etc.). The data are put in long form with one row per CIPN20 response, and cumulative dose is re-scaled to avoid extremely small rate estimates and numerical stability/convergence issues.

```
cipn20 <- read.csv("data/N08CB_CIPN_responses.csv") %>%
  select(Patient, Cycle, C_dose, starts_with("CIPN")) %>%
  gather(key = "Item", value = "Response", starts_with("CIPN")) %>%
  mutate(C_dose = C_dose / 1000,
         Item = factor(Item, levels = paste("CIPN",
                                            c(1:6, 9, 10, 18),
                                            sep = "_")))
table(cipn20$Cycle)
```

```
##
##    1    2    3    4    5    6    7    8    9   10   11   12
## 3096 2979 2925 2871 2682 2619 2457 2331 2214 2025 1728 1440
```

```
summary(cipn20$C_dose)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.1664  0.3443  0.3804  0.5873  1.0098
```

```
table(cipn20$Item)
```

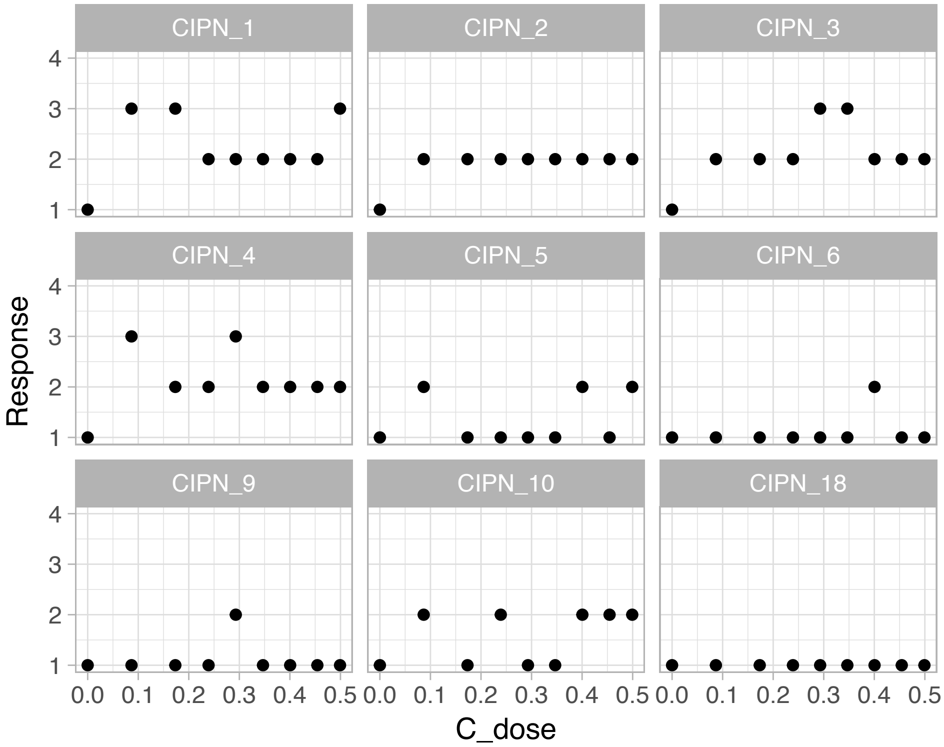
```
##
##  CIPN_1  CIPN_2  CIPN_3  CIPN_4  CIPN_5  CIPN_6  CIPN_9 CIPN_10 CIPN_18
##    3263    3263    3263    3263    3263    3263    3263    3263    3263
```

```
table(cipn20$Response)
```

```
##
##     1     2     3     4
## 20899   5394   1683    502
```

An example of CIPN20 responses for one patient are shown below.

```
example_pt <- sample(unique(cipn20$Patient), 1)
ggplot(cipn20 %>% filter(Patient == example_pt)) +
  geom_point(aes(x = C_dose, y = Response)) +
  scale_y_continuous(breaks = 1:4, limits = c(1, 4)) +
  facet_wrap(~ Item)
```



We exclude all missing CIPN20 responses and all patients who had responses for fewer than three unique treatment cycles.

```
cipn20 <- cipn20 %>%
  filter(!is.na(Response))
num_cycles <- cipn20 %>%
  group_by(Patient) %>%
  summarize(Cycles = length(unique(Cycle))) %>%
  filter(Cycles >= 3)
cipn20 <- cipn20 %>%
  filter(Patient %in% num_cycles$Patient)
```

The Rasch-type model is fit as follows.

```
fit <- glm(Response ~ (Patient + Item) * C_dose,
           family = quasipoisson(link = "log"),
           data = cipn20)
```

Next, we extract the patient rate (slope) estimates and SEs. One patient serves as a reference patient. The reference patient's slope is given by the “base” slope estimate (the coefficient on `C_dose`). For the remaining patients, the slope estimate for a given patient represents the difference between that patient's and the reference patient's slopes. First we obtain slope estimates and SEs for those non-reference patients.

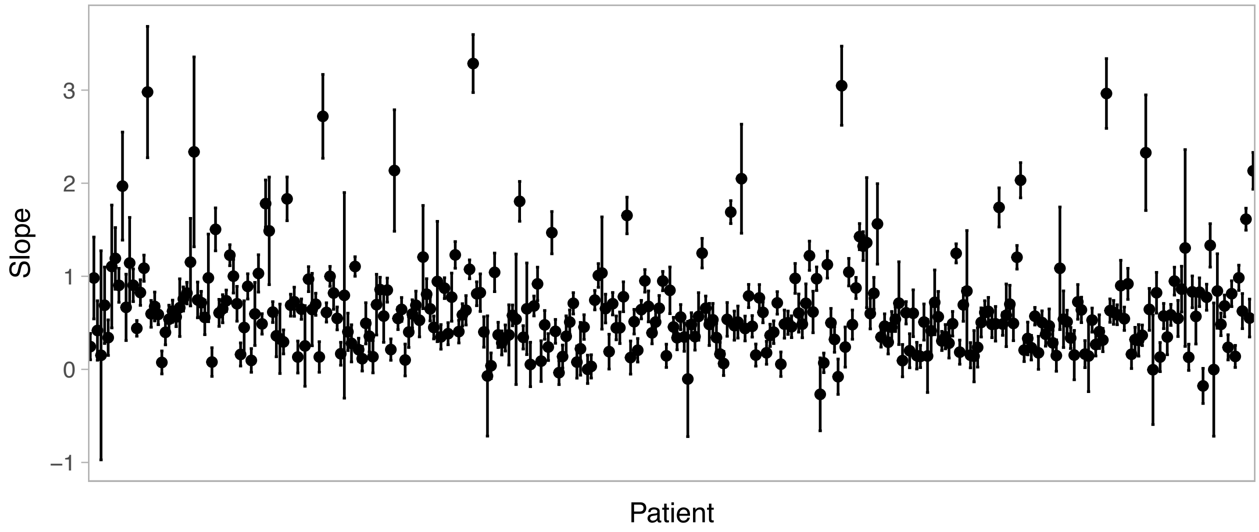
```r
# Get names of patient slope parameters.
slope_params <- grep("^Patient[^:]+:C_dose$",
                     names(coef(fit)),
                     value = TRUE)
# Form table of estimates and SEs.
est <- data.frame(
  Patient = substr(slope_params, nchar("Patient") + 1,
                   regexpr(":", slope_params) - 1),
  Slope = coef(fit)["C_dose"] + coef(fit)[slope_params],
  SE = sqrt(diag(vcov(fit))[slope_params] +
              diag(vcov(fit))["C_dose"] +
              2 * vcov(fit)[slope_params, "C_dose"])
)
```

Then we obtain the reference patient's estimate and SE.

```r
ref_patient <- setdiff(unique(cipn20$Patient), est$Patient)
est <- rbind(data.frame(Patient = ref_patient,
                        Slope = coef(fit)["C_dose"],
                        SE = sqrt( diag(vcov(fit))["C_dose"])),
             est)
```

The rate estimates ± 1 SE are plotted below.

```r
ggplot(est) +
  geom_point(aes(x = Patient, y = Slope)) +
  geom_errorbar(aes(x = Patient, ymin = Slope - SE, ymax = Slope + SE)) +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
```

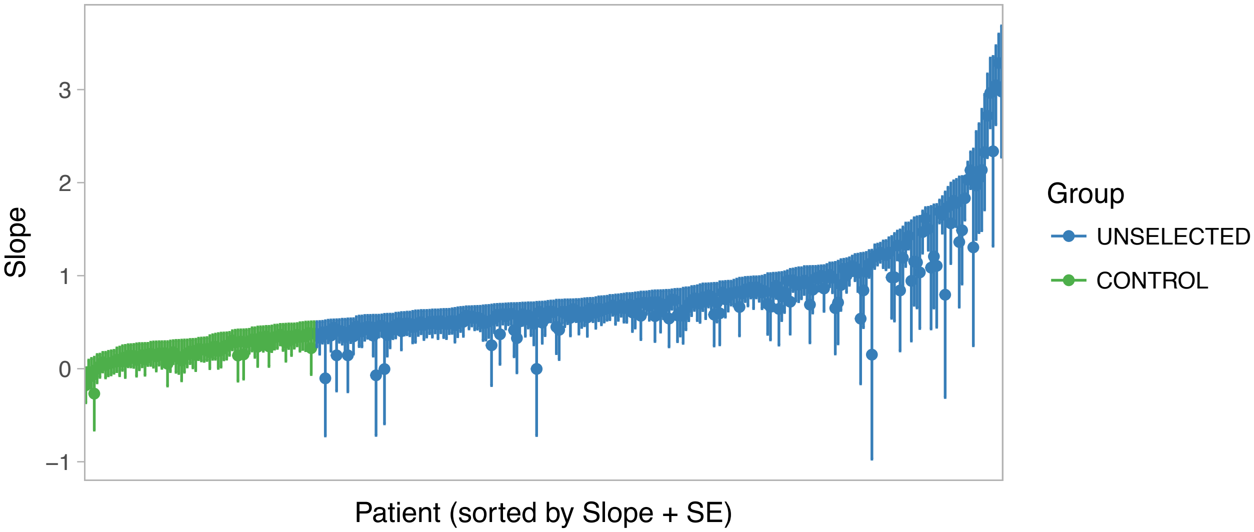


Patients in the lowest 25% of slope-plus-SE values are selected as controls.

```
n_per_group <- round(nrow(est) / 4)
n_per_group
```

```
## [1] 82
```

```
est$Group <- "UNSELECTED"
est$Group <- factor(est$Group, levels = c("CASE", "UNSELECTED", "CONTROL"))
# Re-order to select controls.
est <- est %>%
  arrange(Slope + SE)
est$Group[1:n_per_group] <- "CONTROL"
ggplot(est %>%
         mutate(Patient = factor(Patient, levels = est$Patient))) +
  geom_point(aes(x = Patient, y = Slope, color = Group)) +
  geom_errorbar(aes(x = Patient, ymin = Slope - SE, ymax = Slope + SE,
                    color = Group)) +
  scale_x_discrete(name = "Patient (sorted by Slope + SE)") +
  scale_color_manual(values = c("#377eb8", "#4daf4a")) +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
```

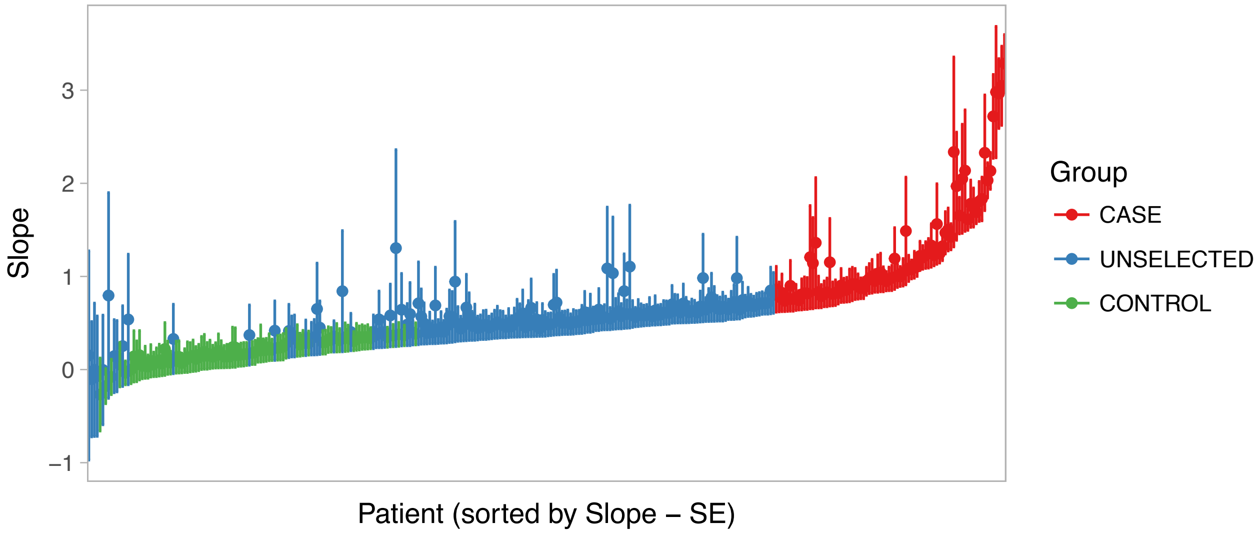


Patients in the highest 25% of slope-minus-SE values are selected as cases, making sure that none of these patients has already been selected as a control.

```
# Re-order to select cases.
est <- est %>%
  arrange(desc(Slope - SE))
# Check for patients already selected as controls.
any(est$Group[1:n_per_group] == "CONTROL")
```

```
## [1] FALSE
```

```
est$Group[1:n_per_group] <- "CASE"
ggplot(est %>%
         mutate(Patient = factor(Patient, levels = rev(est$Patient)))) +
  geom_point(aes(x = Patient, y = Slope, color = Group)) +
  geom_errorbar(aes(x = Patient, ymin = Slope - SE, ymax = Slope + SE,
                    color = Group)) +
  scale_x_discrete(name = "Patient (sorted by Slope - SE)") +
  scale_color_brewer(palette = "Set1") +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
```



A final visualization of the case and control selection follows.

```
ggplot(est) +
  geom_point(aes(x = Slope, y = Patient, color = Group)) +
  geom_errorbarh(aes(x = Slope,
                     xmin = Slope - SE,
                     xmax = Slope + SE,
                     y = Patient,
                     color = Group)) +
  scale_color_brewer(palette = "Set1") +
  theme(axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
```