Cervical cancer and herpes

```r
library(inlamisclass)
library(dplyr)
library(ggplot2)
library(INLA)
```

```r
# Number of iterations for importance sampling
niter <- 100000
```

The data `cervical_cancer` consists of $n = 2044$ rows and three columns:

- $y_1, \cdots, y_n$: The cervical cancer status of a patient.
- $x_1, \cdots, x_n$: Exposure to HSV-2, measured with an accurate test, only 115 measurements available.
- $w_1, \cdots, w_n$: Exposure to HSV-2, measured with an inaccurate test.

The response $\boldsymbol{y}$ and the less accurate test result $\boldsymbol{w}$ are available for all patients, while the more accurate test result $\boldsymbol{x}$ is only available for 115 of the patients. That means that we have a validation sample of 115 samples which can be used to estimate the misclassification probabilities.

```r
validation <- filter(cervical_cancer, !is.na(x))
incomplete_data <- filter(cervical_cancer, is.na(x))
```

By looking at the data in aggregated form, it can be seen that the misclassification probabilities for the cases ($y_i = 1$) and controls ($y_i = 0$) are quite different. Therefore, we will consider two cases: Case 1, where we only consider the overall misclassification matrix, irrespective of the value of y_i, and Case 2, where we switch between the two matrices depending on the value of y_i in the modelling.

Case 1: Misclassification is considered independent of y_i

We estimate the overall misclassification matrix from the validation data:

```r
# w = 1 given x = 0
pi10 <- sum((validation$w-validation$x) == 1)/sum(validation$x==0)

# w = 0, given x = 1
pi01 <- sum(validation$w-validation$x == -1)/sum(validation$x==1)

M <- matrix(c(1-pi10, pi10, pi01, 1-pi01), byrow = TRUE, nrow = 2)
```

For the exposure model describing $\boldsymbol{x}$, we estimate the probability of exposure to HSV-2, $P(x_i = 1)$ from the validation data:

```r
p <- sum(validation$x == 1)/nrow(validation)
alpha0 <- log(p/(1-p))
```

This means that for the modelling, we will use the misclassification matrix

```r
M
```

```
##           [,1]      [,2]
## [1,] 0.7666667 0.2333333
## [2,] 0.3818182 0.6181818
```

and the exposure model $\text{logit}[E(x)] = \alpha_0 \mathbf{1}$, where α_0 is assumed to be -0.0870114.

For running the model, we will run importance sampling for the given number of iterations, and the results will be saved along with the running time:

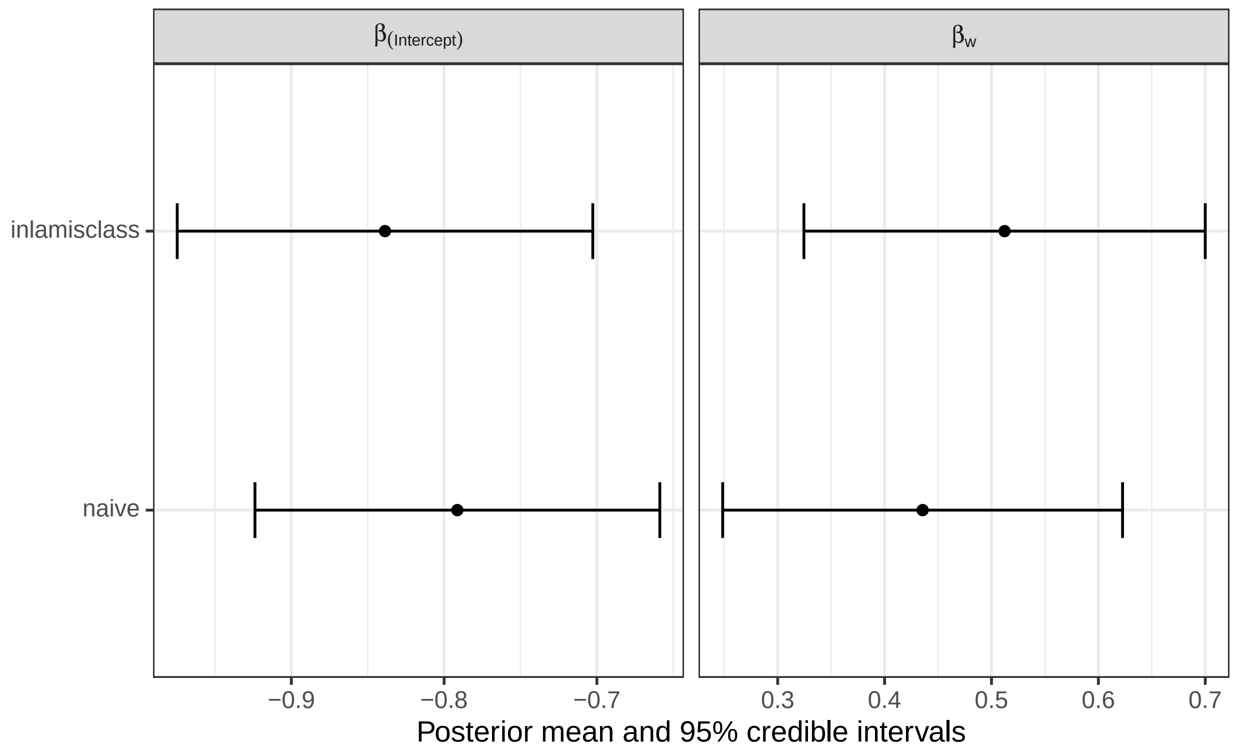
```r
start_time <- Sys.time()
case_control_model1 <- inla_is_misclass(formula_moi = y ~ w,
                                        formula_imp = w ~ 1,
                                        alpha = alpha0,
                                        MC_matrix = M,
                                        data = incomplete_data,
                                        niter = niter,
                                        family = "binomial", Ntrials = 1)
end_time <- Sys.time()

case_control_results1 <- list(runtime = end_time - start_time,
                              model = case_control_model1,
                              summary = make_results_df(case_control_model1,
                                                        niter = niter))

saveRDS(list(case_control_model = case_control_results1,
             niter = niter, rundate = Sys.time()),
        file = "results/case_control_results1.rds")
```

```r
case_control_results1 <- readRDS("results/case_control_results1.rds")
```

We fit a naive model that ignores the misclassification in $\boldsymbol{w}$ for a comparison.

```r
naive_cc <- INLA::inla(y ~ w, data = incomplete_data, family = "binomial",
                       Ntrials = 1)
```

```r
plot_compare_inlamisclass(case_control_results1$case_control_model$model,
                          naive_mod = naive_cc, plot_intercept = TRUE)
```



Case 2: Misclassification and exposure depend on y_i

We estimate the conditional misclassification matrices based on the value of $\boldsymbol{y}$:

```r
validation1 <- filter(validation, y == 1)
validation0 <- filter(validation, y == 0)

# w = 1 given x = 0, y = 1
pi10_y1 <- sum((validation1$w-validation1$x) == 1)/sum(validation1$x==0)
# w = 0, given x = 1, y = 1
pi01_y1 <- sum(validation1$w-validation1$x == -1)/sum(validation1$x==1)

# w = 1, x = 0, given y = 0
pi10_y0 <- sum((validation0$w-validation0$x) == 1)/sum(validation0$x==0)
# w = 0, x = 1, given y = 0
pi01_y0 <- sum(validation0$w-validation0$x == -1)/sum(validation0$x==1)
```

So the MC matrix for $y_i = 1$ would be

```r
M1 <- matrix(c(1-pi10_y1, pi10_y1, pi01_y1, 1-pi01_y1), byrow = TRUE, nrow = 2)
M1
```

```
##           [,1]      [,2]
## [1,] 0.8125000 0.1875000
## [2,] 0.2173913 0.7826087
```

and for $y_i = 0$:

```
M0 <- matrix(c(1-pi10_y0, pi10_y0, pi01_y0, 1-pi01_y0), byrow = TRUE, nrow = 2)
M0
```

```
##      [,1] [,2]
## [1,] 0.75 0.25
## [2,] 0.50 0.50
```

Similarly, we estimate the coefficients of the exposure model conditional on y_i:

```
p1 <- sum(validation1$x == 1)/nrow(validation1)
alpha0_1 <- log(p1/(1-p1))
p0 <- sum(validation0$x == 1)/nrow(validation0)
alpha0_0 <- log(p0/(1-p0))
```

```
c(alpha0_0 = alpha0_0, alpha0_1 = alpha0_1)
```

```
##   alpha0_0   alpha0_1
## -0.3184537  0.3629055
```

But this is the same as regression model:

```
exp_glm <- glm(x~y, family = "binomial", data = validation)$coef
alphas <- exp_glm["(Intercept)"] + c(0, exp_glm["y"])
alphas
```

```
##                      y
## -0.3184537  0.3629055
```

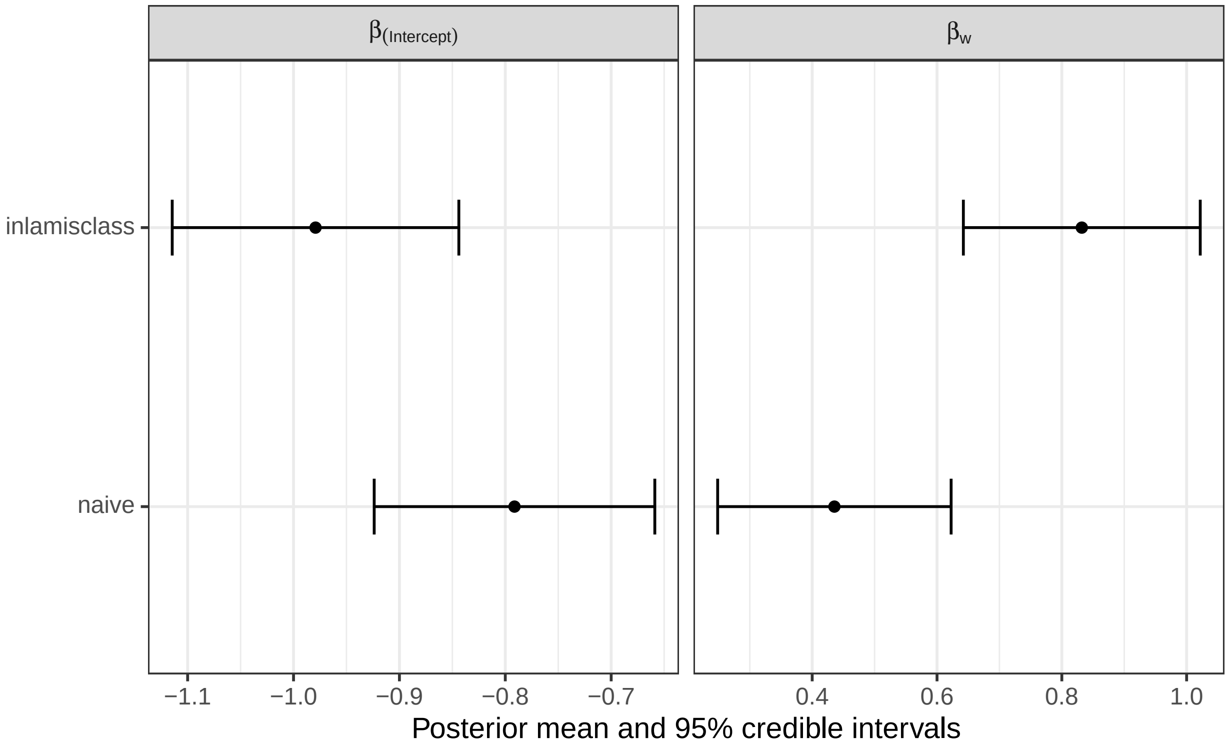
So we use the exposure/imputation model $\text{logit}[E(\boldsymbol{x} \mid \boldsymbol{y})] = \alpha_0\mathbf{1} + \alpha_y\boldsymbol{y}$, with fixed values for the coefficients, $\alpha_0 = -0.3185$ and $\alpha_y = 0.3629$.

```
start_time <- Sys.time()
case_control_model2 <- inla_is_misclass(formula_moi = y ~ w,
                                        formula_imp = w ~ y,
                                        alpha = alphas,
                                        MC_matrix = list(MC_1 = M1, MC_0 = M0),
                                        data = incomplete_data,
                                        niter = niter,
                                        conditional = "y",
                                        family = "binomial", Ntrials = 1)
end_time <- Sys.time()

case_control_results2 <- list(
  runtime = end_time - start_time,
  model = case_control_model2,
  summary = make_results_df(case_control_model2, niter = niter))

saveRDS(list(case_control_model = case_control_results2,
             niter = niter, nburnin = 0, rundate = Sys.time()),
        file = "results/case_control_results2.rds")
```

```r
case_control_results2 <- readRDS("results/case_control_results2.rds")
```

Again, we plot the results from this model together with the estimates from the model that does not account for misclassification.

```r
plot_compare_inlamisclass(case_control_results2$case_control_model$model,
                          naive_mod = naive_cc, plot_intercept = TRUE)
```



Comparing the cases

We plot both cases together, along with the naive model, to compare.

```r
plot_compare_inlamisclass(
  list(case_control_results1$case_control_model$model,
       case_control_results2$case_control_model$model),
  naive_mod = naive_cc, plot_intercept = TRUE,
  num_inlamisclass_models = 2, niter = case_control_results2$niter)
```

Figure for article

```r
results_nondiff <- make_results_df(case_control_results1$case_control_model$model)$moi
results_diff <- make_results_df(case_control_results2$case_control_model$model)$moi

results_naive <- naive_cc$summary.fixed
```

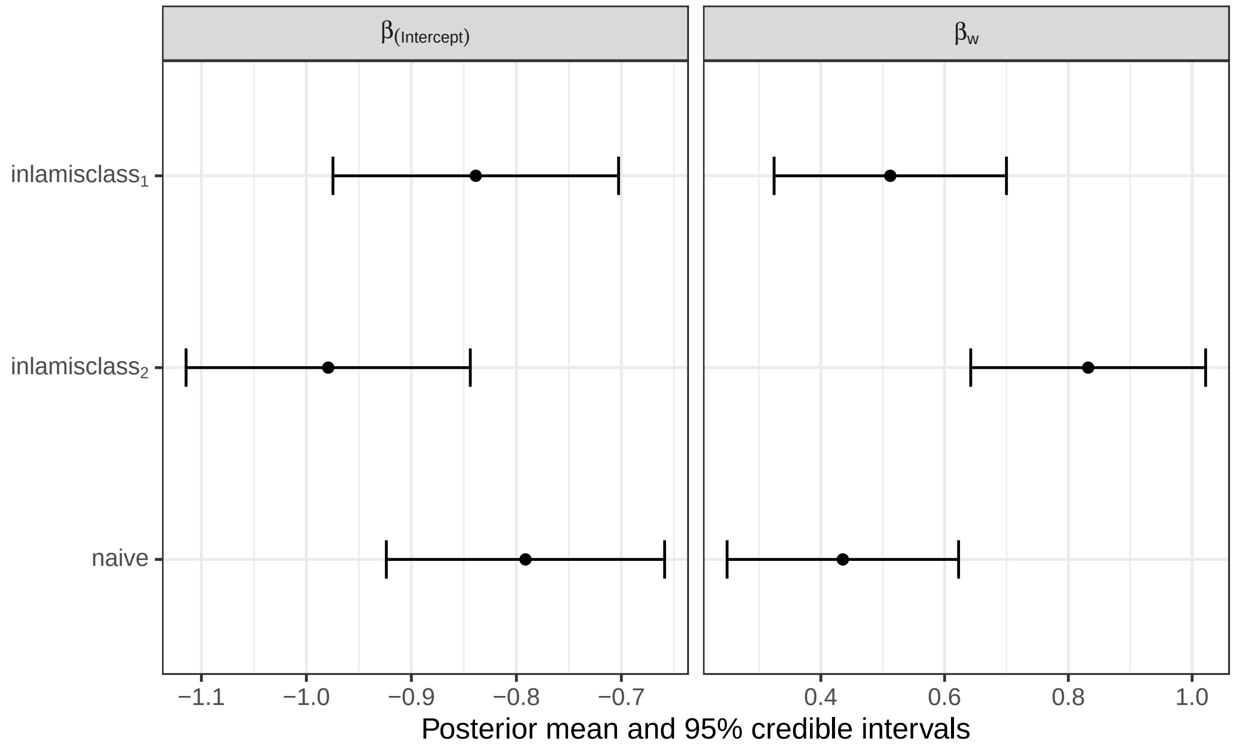
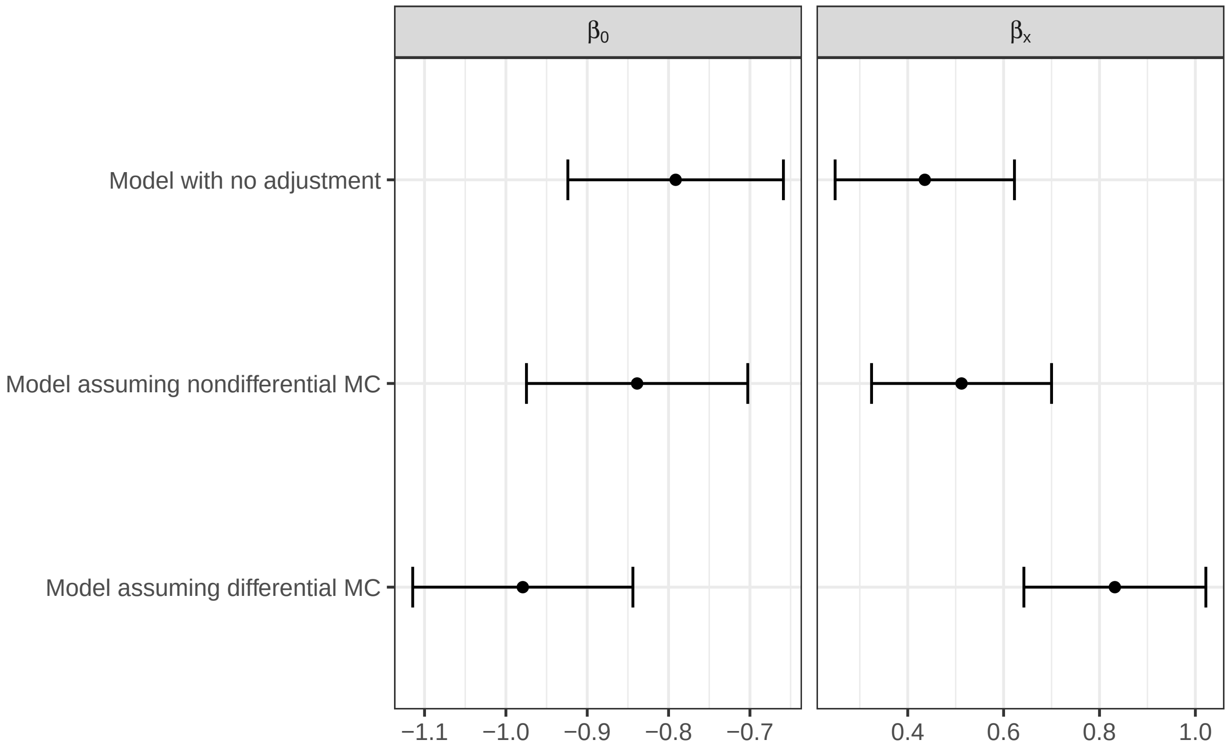


Figure 1: Results from both cases along with the naive model. `inlamisclass_1` corresponds to case 1 and `inlamisclass_2` corresponds to case 2.

```
results_naive$variable <- rownames(results_naive)

all_res <- dplyr::bind_rows(nondiff = results_nondiff, diff = results_diff, Naive = results_naive, .id 
all_res$labels <- paste0("beta", "[", c(0, "x"), "]")
all_res$Model <- factor(all_res$Model, levels = c("Naive", "nondiff", "diff"))
all_res$Model <- plyr::revalue(all_res$Model,
                               c("Naive" = "Model with no adjustment",
                                 "nondiff" = "Model assuming nondifferential MC",
                                 "diff" = "Model assuming differential MC"))

ggplot(all_res, aes(y = Model)) +
  geom_point(aes(x = mean)) +
  geom_errorbarh(aes(xmin = .data$"0.025quant", xmax = .data$"0.975quant"), height = .2) +
  scale_y_discrete(limits = rev) +
  facet_wrap(vars(labels), scales = "free_x", labeller = label_parsed) +
  theme_bw() +
  theme(axis.title = element_blank())
```



```r
ggsave("figures/case_control.pdf", height = 2.3, width = 7)
```