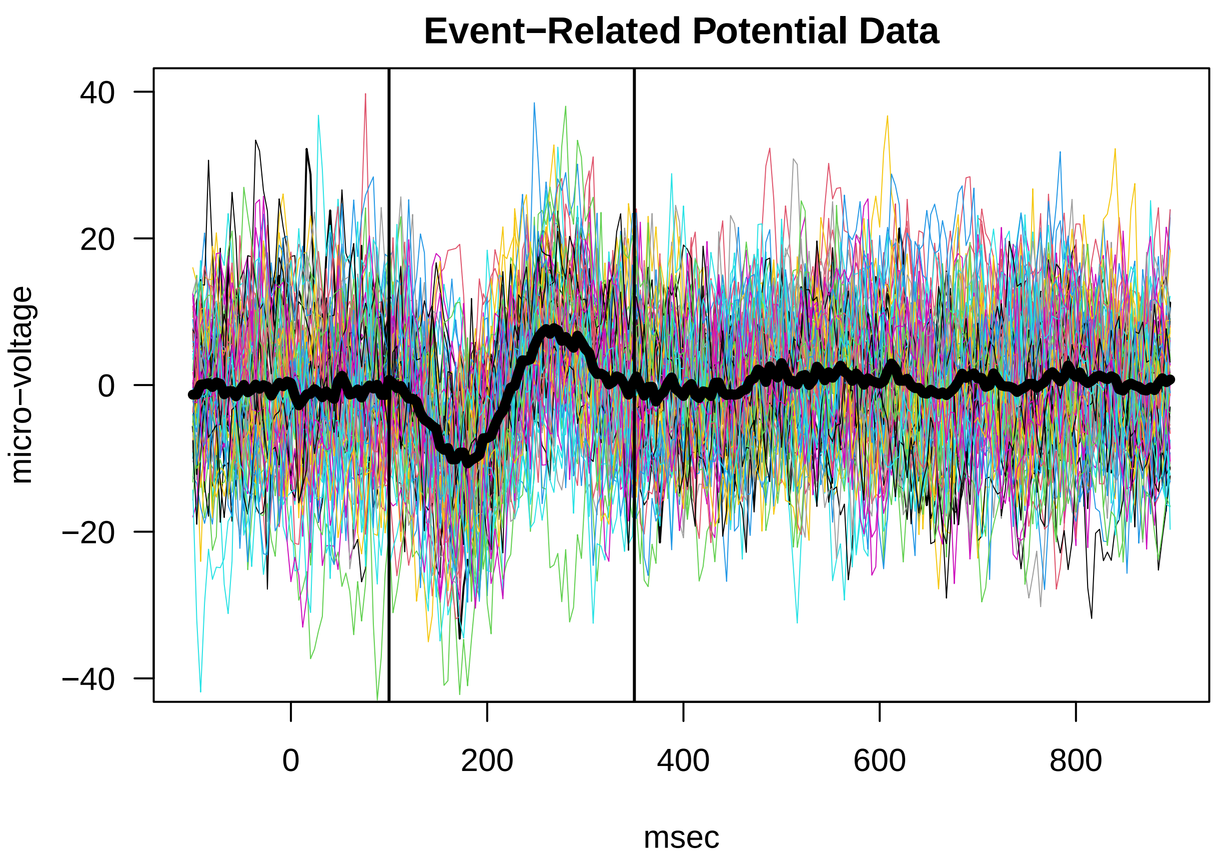
ERP Data Analysis

The raw real ERP data set is saved in `data-raw/Raw_ERP.csv`. It is also publicly available at Dr. Senturk website.

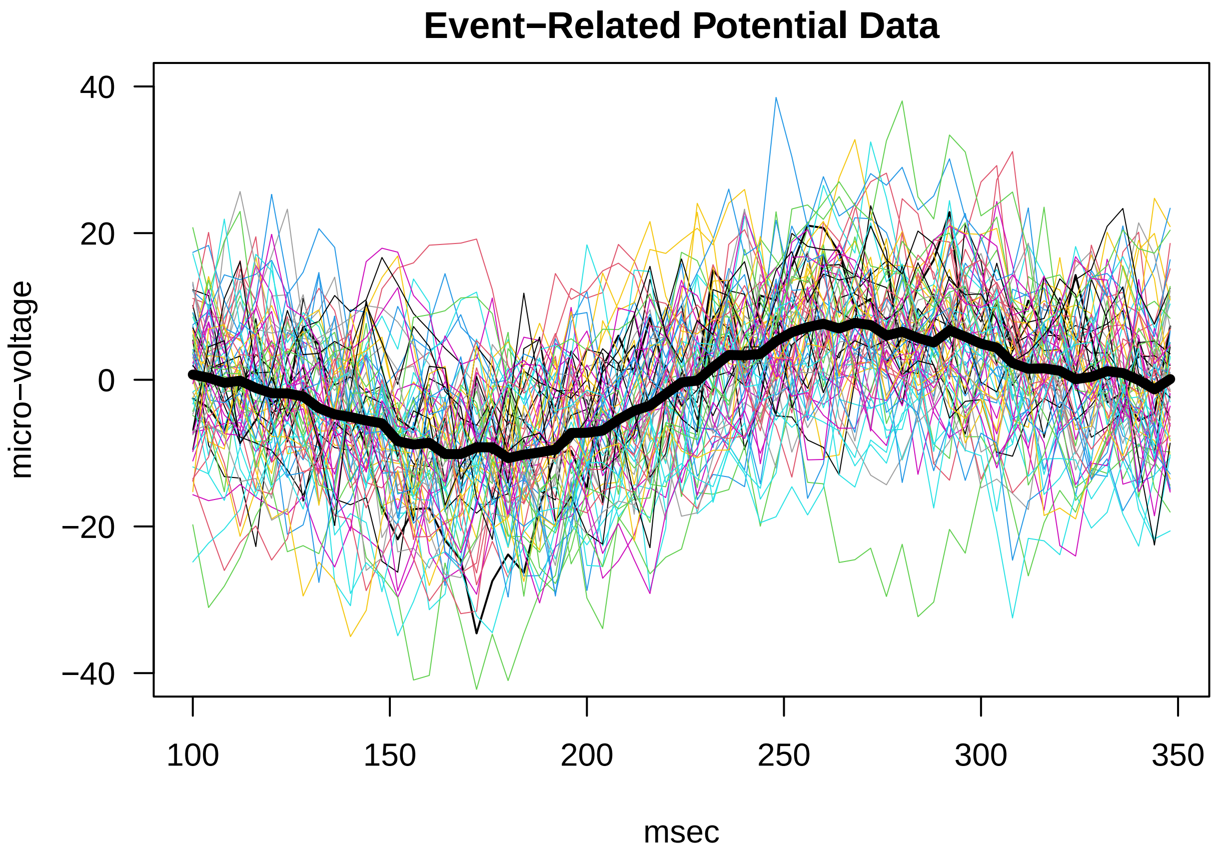
```
dataset <- read.csv("../data-raw/Raw_ERP.csv", header = TRUE)
names(dataset) <- c("id", "group", "region", "electrode", "condition",
                    "trial", seq(-100, 896, 4))
TT <- 250
erp_data <- dataset[, 7:(TT+6)]
```

The data are plotted and shown in Figure 4 in the paper. The focused time interval is $[100, 350]$ msec.

```
par(mar = c(4, 4, 2, 1))
plot(seq(-100, 896, 4), erp_data[1, ], type = "l", ylim = c(-40, 40),
     xlab = "msec", ylab =  "micro-voltage",
     main = "Event-Related Potential Data", las = 1)
for (i in 2:70) {
    lines(seq(-100, 896, 4), erp_data[i, ], col = i, lwd = 0.5)
}
lines(seq(-100, 896, 4), apply(erp_data, 2, mean), col = 1, lwd = 5)
abline(v = c(100, 350), lwd = 1.5)
```



```r
plot(seq(100, 350, 4), erp_data[1, as.character(seq(100, 350, 4))],
     type = "l", ylim = c(-40, 40),
     xlab = "msec", ylab =  "micro-voltage",
     main = "Event-Related Potential Data", las = 1)
for (i in 2:70) {
    lines(seq(100, 350, 4), erp_data[i, as.character(seq(100, 350, 4))], col = i, lwd = 0.5)
}
lines(seq(100, 350, 4), apply(erp_data[, as.character(seq(100, 350, 4))], 2, mean), col = 1, lwd = 5)
```




```r
## time bound index
lwr_idx <- which(names(erp_data) == 100)
upr_idx <- which(names(erp_data) == 348)
bd <- lwr_idx:upr_idx
msec_idx <- seq(-100, 896, 4)
n_trial <- 1:72

## sample size
n_erp <- length(seq(-100, 896, 4)[bd])
```

The following code shows how the posterior density of latency is computed, and reproduces Figure 6 when ERP waveform is averaged over two trials.

```r
library(dgp)
library(emulator)
```

```
## Loading required package: mvtnorm
```

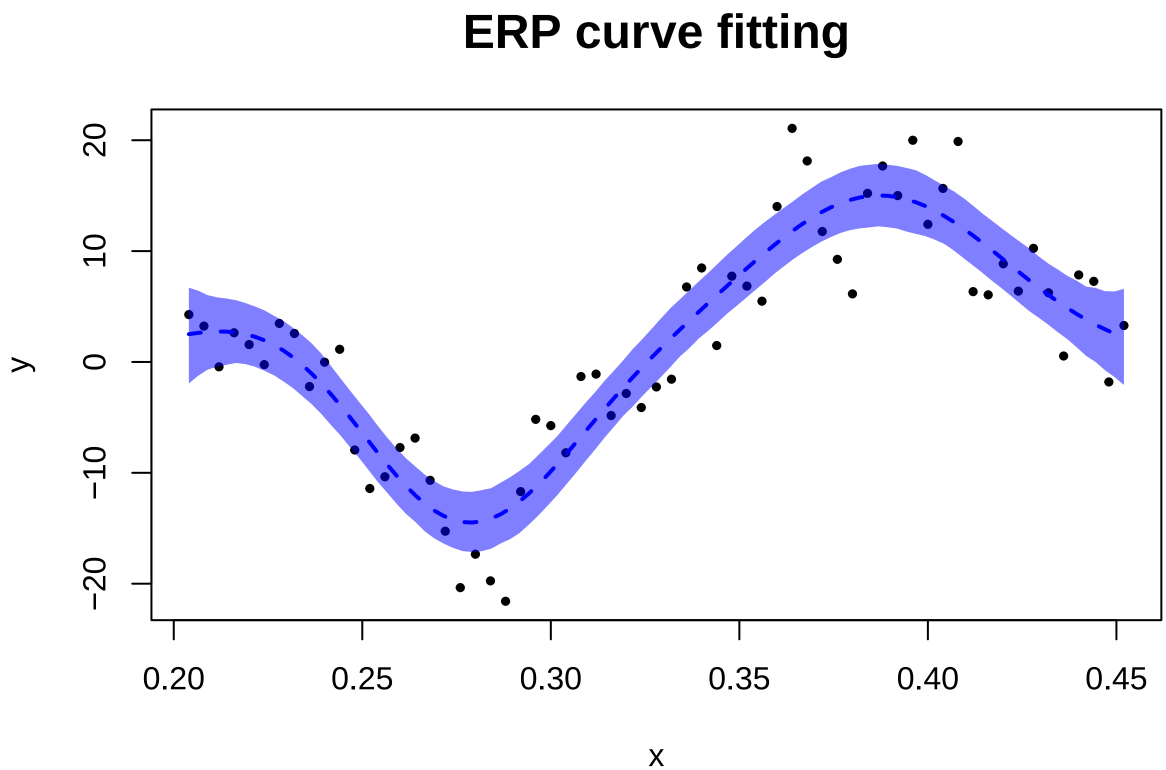
```
################################################
### average two trials
################################################
## ERP waveform averaged over 2 trials
k <- 1
erp <- apply(as.matrix(erp_data)[(2*k-1):(2*k), ], 2, mean)

## time points
x <- seq(0.004, 1, length = 250)

H0 <- outer(x[bd], x[bd], FUN = function(x1, x2) (x1 - x2))
x_a <- min(x[bd])
x_b <- max(x[bd])
grid_t <- seq(x_a, x_b, length.out = 400)

log_post <- rep(0, length(grid_t))
log_post33 <- rep(0, length(grid_t))

x_test <- seq(x_a, x_b, length.out = 100)
x_test_msec <- seq(min(seq(-100, 896, 4)[bd]), max(seq(-100, 896, 4)[bd]),
                   length.out = 100)

## Optimizing hyperparameters
gp_res <- Rsolnp::solnp(pars = c(.5, .5, .5), fun = log_mar_lik_gp,
                        LB = c(0.0001, 0.0001, 0.0001),
                        UB = c(1 / 0.0001, 1 / 0.0001, 1 / 0.0001),
                        control = list(TOL = 1e-5, trace = 0),
                        y = erp[bd], H0 = H0)

n <- length(x[bd])
lam <- gp_res$par[1] ^ 2 / (n * gp_res$par[2] ^ 2)
Kff <- se_ker(H0 = H0, tau = 1, h = gp_res$par[3])
A <- Kff + diag((n * lam), n)

## Posterior distribution of local extrema
for (i in 1:length(grid_t)) {
    log_post[i] <- log_post_t_theory(t = grid_t[i], y = erp[bd], x = x[bd],
                                     Kff = Kff, A = A, lambda = lam,
                                     h = gp_res$par[3], sig2 = gp_res$par[1] ^ 2,
                                     shape1 = 1, shape2 = 1,
                                     a = x_a, b = x_b)
    log_post33[i] <- log_post_t_theory(t = grid_t[i], y = erp[bd], x = x[bd],
                                       Kff = Kff, A = A, lambda = lam,
                                       h = gp_res$par[3], sig2 = gp_res$par[1] ^ 2,
                                       shape1 = 3, shape2 = 3,
                                       a = x_a, b = x_b)
}
post_prob <- exp(log_post - max(log_post))
post_prob33 <- exp(log_post33 - max(log_post33))
```

To obtain the fitted curve and uncertainty bands, we can use the function `get_pred_ci_gp()`.

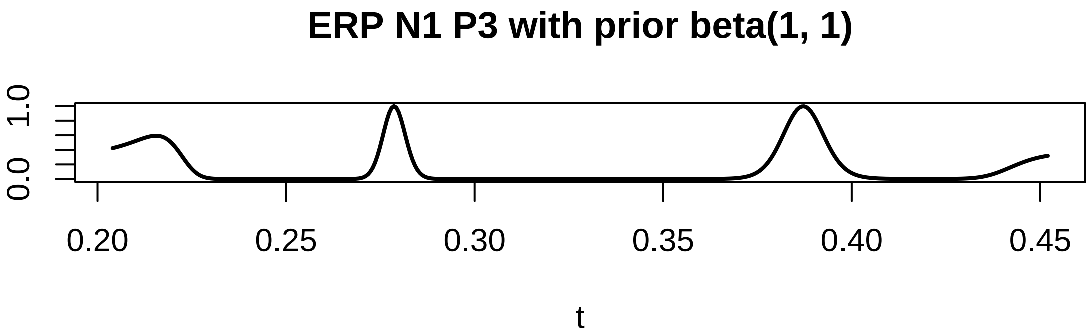
```
pred <- get_pred_ci_gp(eb_par = gp_res$par, x = x[bd], x_test = x_test,
                       y = erp[bd])
```

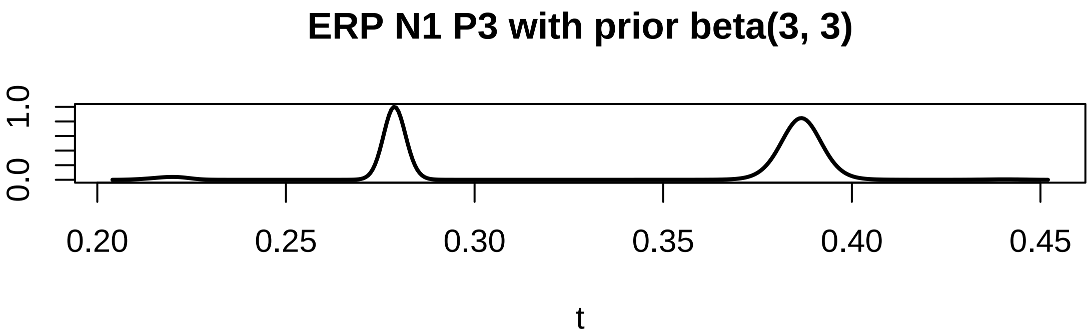
The function `plot_pred_gp_f_y()` plots the fitted waveform as well as the uncertainty intervals.

```
## Curve fitting plotting
plot_pred_gp_f_y(x = x[bd], y = erp[bd],
                 x_test = x_test,
                 mu_test = pred$mu_test,
                 CI_Low_f = pred$ci_low,
                 CI_High_f = pred$ci_high,
                 ylim = range(erp[bd]),
                 xlim = c(x_a, x_b), is.der.line = FALSE,
                 is.true.fcn = FALSE, cex = 0.5,
                 plot.type = "p", col.poly = rgb(0, 0, 1, 0.5),
                 pred_lwd = 2, title = "ERP curve fitting", is.legend = FALSE)
```




```
## Posterior density plotting
par(mfrow = c(2, 1))
plot(grid_t, post_prob, ylab = "", xlab = "t", type = 'l',
     main = paste0("ERP N1 P3 with prior beta(1, 1)"), ylim = c(0, 1), lwd = 2)
plot(grid_t, post_prob33, ylab = "", xlab = "t", type = 'l',
     main = paste0("ERP N1 P3 with prior beta(3, 3)"), ylim = c(0, 1), lwd = 2)
```






```
## HPD interval
(erp_hpd <- get_hpd_interval_from_den(post_prob, grid_t = grid_t,
                                      target_prob = 0.95))
```

```
## $no_cluster
## [1] 4
##
## $ci_lower
## [1] 0.2040000 0.2723709 0.3755489 0.4395689
##
## $ci_upper
## [1] 0.2251328 0.2848020 0.3991679 0.4520000
##
## $prob_value
## [1] 0.9596019
##
## $den_value
## [1] 0.1012871
```

```
## estimated number of stationary points
(erp_map <- get_map(post_den = post_prob, grid_t = grid_t, hpdi = erp_hpd))
```

```
## [1] 0.2158095 0.2785865 0.3873584 0.4520000
```

To replicate the result of the grand ERP waveworm averaged over 72 trials, simply replace `erp <- apply(as.matrix(erp_data)[(2*k-1):(2*k), ], 2, mean)` with `erp <- apply(erp_data, 2, mean)`, and run the code.

```
##################################################
### average across all trials (Figure 5 in the paper)
##################################################
## ERP waveform averaged over 72 trials
```

```r
erp <- apply(erp_data, 2, mean)

## Optimizing hyperparameters
gp_res <- Rsolnp::solnp(pars = c(.5, .5, .5), fun = log_mar_lik_gp,
                        LB = c(0.0001, 0.0001, 0.0001),
                        UB = c(1 / 0.0001, 1 / 0.0001, 1 / 0.0001),
                        control = list(TOL = 1e-5, trace = 0),
                        y = erp[bd], H0 = H0)

n <- length(x[bd])
lam <- gp_res$par[1] ^ 2 / (n * gp_res$par[2] ^ 2)
Kff <- se_ker(H0 = H0, tau = 1, h = gp_res$par[3])
A <- Kff + diag((n * lam), n)

## Posterior distribution of local extrema
for (i in 1:length(grid_t)) {
    log_post[i] <- log_post_t_theory(t = grid_t[i], y = erp[bd], x = x[bd],
                                     Kff = Kff, A = A, lambda = lam,
                                     h = gp_res$par[3], sig2 = gp_res$par[1] ^ 2,
                                     shape1 = 1, shape2 = 1,
                                     a = x_a, b = x_b)
    log_post33[i] <- log_post_t_theory(t = grid_t[i], y = erp[bd], x = x[bd],
                                       Kff = Kff, A = A, lambda = lam,
                                       h = gp_res$par[3], sig2 = gp_res$par[1] ^ 2,
                                       shape1 = 3, shape2 = 3,
                                       a = x_a, b = x_b)
}
post_prob <- exp(log_post - max(log_post))
post_prob33 <- exp(log_post33 - max(log_post33))
```

To obtain the fitted curve and uncertainty bands, we can use the function `get_pred_ci_gp()`.

```r
pred <- get_pred_ci_gp(eb_par = gp_res$par, x = x[bd], x_test = x_test,
                       y = erp[bd])
```

The function `plot_pred_gp_f_y()` plots the fitted waveform as well as the uncertainty intervals.

```r
## Curve fitting plotting
plot_pred_gp_f_y(x = x[bd], y = erp[bd],
                 x_test = x_test,
                 mu_test = pred$mu_test,
                 CI_Low_f = pred$ci_low,
                 CI_High_f = pred$ci_high,
                 ylim = range(erp[bd]),
                 xlim = c(x_a, x_b), is.der.line = FALSE,
                 is.true.fcn = FALSE, cex = 0.5,
                 plot.type = "p", col.poly = rgb(0, 0, 1, 0.5),
                 pred_lwd = 2, title = "Figure 5: ERP curve fitting (avged across 72 trials)", is.legend
```

Figure 5: ERP curve fitting (avged across 72 trials)