Analysis

```r
three_lvl_dat <- readRDS("three_lvl_dat.rds")
head(three_lvl_dat)
```

```
##   site patient visit X1 X2 X3         X4         X5         X6          Y
## 1    1       1     1  1  0  0 -0.2333860  0.5258316  0.8517787   9.713744
## 2    1       1     2  0  0  0 -1.7007881 -2.3441436  0.2597312 -16.215685
## 3    1       1     3  0  1  0 -0.6558381 -0.1574295 -1.1642077  -5.654015
## 4    1       1     4  0  0  0 -0.6304705 -1.4508508 -1.4120594 -15.497097
## 5    1       1     5  0  0  1 -0.9323276 -0.3451917 -1.2283032  -7.509270
## 6    1       2     1  1  1  1  0.3219037 -1.6069905 -0.5368642  -3.596303
```

```r
H <- n_distinct(three_lvl_dat$site)
px <- 6
```

```r
source("../PEAL Engine/PEAL_Engine_RI.R")
```

```
##
## Attaching package: 'nlme'

## The following object is masked from 'package:lme4':
##
##     lmList

## The following object is masked from 'package:dplyr':
##
##     collapse
```

```r
# Preprocessing
# Calculate the number of visits per patient per site
visit_count <- three_lvl_dat %>%
  dplyr::group_by(site, patient) %>%
  dplyr::summarise(total_visits = n(), .groups = "drop")

# Reorder data (as a preprocess probably)
rearranged_data <- merge(three_lvl_dat, visit_count, by = c("site", "patient")) %>%
  arrange(site, total_visits, patient) %>%
  mutate(site = factor(site))
```

PEAL

```
# XYZ
Y <- rearranged_data$Y
X <- as.matrix(rearranged_data[, paste0("X", 1:px)])
X <- cbind(X)
Z <- list()

for(i in 1:H){
  count_mat = rearranged_data %>%
                    filter(site == i) %>%
                    group_by(site, patient) %>%
                    dplyr::summarise(n_hi = n(), .groups = 'drop')

  Z[[i]] <- (generate_Zhv_matrix(count_mat))
}

id.site <- rearranged_data$site

ShXYZ <- peal.get.summary(Y, X, Z, id.site)
```

```
fit.peal = peal.fit.RI(Y = NULL, X = NULL, Z = NULL,
                      id.site = NULL, weights = NULL,
                      pooled = F, reml = T,
                      common.s2 = T,
                      ShXYZ = ShXYZ,  # only need summary stats
                      corstr = 'independence',
                      mypar.init = NULL)
```

```
## Default mypar.init (var comp) =  1 1 1 1
## Convergence Reached and Hessian PD
##  The number of function evaluations used is 33
## Done!
```

LMM

```
fit.lmer <- lmer(
  Y ~ -1 + X1 + X2 + X3 + X4 + X5 + X6 +
       (1 | site) +                          # Site-level random intercept
       (0 + dummy(site, "1") |  site:patient) +   # Site1-specific patient RE
       (0 + dummy(site, "2") |  site:patient) +   # Site2-specific patient RE
       (0 + dummy(site, "3") |  site:patient),    # Site3-specific patient RE
  data = rearranged_data,
  control = lmerControl(
    optimizer = "optimx",
    optCtrl = list(method = "L-BFGS-B")
  )
)
```

```
## Loading required namespace: optimx
```

fixed effects

```
peal_beta = c(fit.peal$b)
lmm_beta = fixef(fit.lmer)
cbind(lmm_beta, peal_beta)
```

```
##    lmm_beta peal_beta
## X1 1.013306  1.013307
## X2 2.093979  2.093980
## X3 2.938045  2.938046
## X4 4.008884  4.008884
## X5 5.023196  5.023196
## X6 6.007726  6.007727
```

random effects

```
summ.lmer <- summary(fit.lmer)
lmm_sigma = c(site = sqrt(summ.lmer$varcor$site),
                diag(sqrt(summ.lmer$varcor$site.patient.2)),
                diag(sqrt(summ.lmer$varcor$site.patient.1)),
                diag(sqrt(summ.lmer$varcor$site.patient)),
               residual = summ.lmer$sigma)

peal_sigma = c(sqrt(fit.peal$V), sqrt(fit.peal$s2))
cbind(lmm_sigma, peal_sigma)
```

```
##                  lmm_sigma peal_sigma
## site             0.1094358  0.1094553
## dummy(site, "1") 0.9705763  0.9705814
## dummy(site, "2") 0.9562770  0.9562715
## dummy(site, "3") 0.8411524  0.8411468
## residual         0.7070875  0.7070876
```

RS

```
source("../PEAL Engine/PEAL_Engine_RS.R")
```

```
# Create formula for design matrix
# fixed effects part
formula <- ~ X1 + X2 + X3 + X4 + X5 + X6 -1
X <- model.matrix(formula, data = rearranged_data)

# RS part
formula_RS <- ~ X6 - 1
X_RS <- model.matrix(formula_RS, data = rearranged_data)
```

```
Z <- construct_Z_list(rearranged_data, colnames(X_RS))

m_h_all = (rearranged_data %>% group_by(site) %>% dplyr::summarise(n_distinct(patient)))[,2]

ShXYZ <- peal.get.summary(Y, X, Z, id.site, m_h_all = m_h_all)
```

```
fit.peal = peal.fit.RS(Y = NULL, X = NULL, Z = NULL,
                      id.site = NULL, weights = NULL,
                      pooled = F, reml = T,
                      common.s2 = T,
                      ShXYZ = ShXYZ,  # only need summary stats
                      corstr = 'independence',
                      mypar.init = NULL)
```

```
## Default mypar.init (var comp) =  1 1 1 1 1 1 1
## Convergence Reached and Hessian PD
##  The number of function evaluations used is  45
## Done!
```

```
fit.lmer <- lmer(
  Y ~ -1 + X1 + X2 + X3 + X4 + X5 + X6 +
       (1 | site) +
       (0 + dummy(site, "1")|| site:patient) +  (0 +  covar_composite_1 || site:patient) +
       (0 + dummy(site, "2")|| site:patient)  + (0 +  covar_composite_2 || site:patient) +
       (0 + dummy(site, "3")|| site:patient)  + (0 +  covar_composite_3 || site:patient),
  data = rearranged_data %>%
    mutate(
      covar_composite_1 = dummy(site, "1") * X6,
      covar_composite_2 = dummy(site, "2") * X6,
      covar_composite_3 = dummy(site, "3") * X6
    ),
  control = lmerControl(
    optimizer = "optimx",
    optCtrl = list(method = "L-BFGS-B")
  )
)
```

```
lmer.mod_RS_summ <- summary(fit.lmer)
lmer_beta <- fixef(fit.lmer)

lmer_sigma = c(sqrt(lmer.mod_RS_summ$varcor$site),
                diag(sqrt(lmer.mod_RS_summ$varcor$site.patient.5)),
                 diag(sqrt(lmer.mod_RS_summ$varcor$site.patient.3)),
                 diag(sqrt(lmer.mod_RS_summ$varcor$site.patient.1)),
                diag(sqrt(lmer.mod_RS_summ$varcor$site.patient.4)),
                diag(sqrt(lmer.mod_RS_summ$varcor$site.patient.2)),
                diag(sqrt(lmer.mod_RS_summ$varcor$site.patient)),
               lmer.mod_RS_summ$sigma)
names(lmer_sigma) <- c("site_RI_sigma", paste0("pt_sigma_RI_bysite", 1:H),paste0("pt_sigma_RS_bysite",
```

fixed effects

```
peal_beta = c(fit.peal$b)
lmm_beta = fixef(fit.lmer)
cbind(lmm_beta, peal_beta)
```

```
##    lmm_beta peal_beta
## X1 1.025053  1.025055
## X2 2.104015  2.104018
## X3 2.936898  2.936901
## X4 3.999226  3.999225
## X5 5.029522  5.029522
## X6 6.012435  6.012434
```

random effects

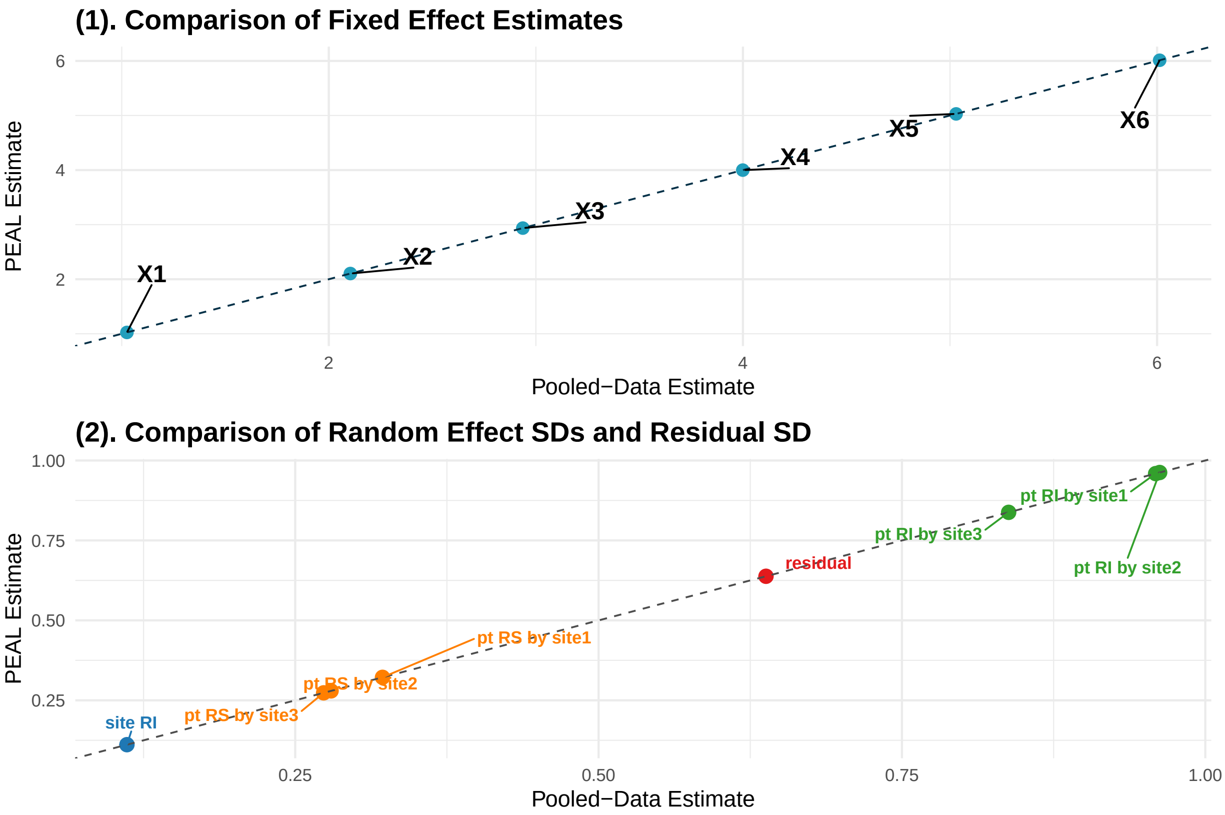
```
summ.lmer <- summary(fit.lmer)
lmm_sigma = c(sqrt(lmer.mod_RS_summ$varcor$site),
                diag(sqrt(lmer.mod_RS_summ$varcor$site.patient.5)),
                 diag(sqrt(lmer.mod_RS_summ$varcor$site.patient.3)),
                 diag(sqrt(lmer.mod_RS_summ$varcor$site.patient.1)),
                diag(sqrt(lmer.mod_RS_summ$varcor$site.patient.4)),
                diag(sqrt(lmer.mod_RS_summ$varcor$site.patient.2)),
                diag(sqrt(lmer.mod_RS_summ$varcor$site.patient)),
               lmer.mod_RS_summ$sigma)
H = 3
names(lmm_sigma) <- c("site RI", paste0("pt RI by site", 1:H),paste0("pt RS by site", 1:H), "residual")

peal_sigma = c(sqrt(fit.peal$V), sqrt(fit.peal$s2))
cbind(lmm_sigma, peal_sigma)
```

```
##                  lmm_sigma peal_sigma
## site RI          0.1112932  0.1113393
## pt RI by site1 0.9590522  0.9590176
## pt RI by site2 0.9623781  0.9624495
## pt RI by site3 0.8379339  0.8378841
## pt RS by site1 0.3219149  0.3219251
## pt RS by site2 0.2795292  0.2795442
## pt RS by site3 0.2733985  0.2733912
## residual         0.6379580  0.6379587
```

```
library(ggplot2)
library(tibble)
library(patchwork)
```

```
##
## Attaching package: 'patchwork'
## The following object is masked from ’package:MASS’:
##
##     area
```

```
library(ggrepel)

# Panel A: Beta
df_beta <- tibble(
  dlmm = as.numeric(peal_beta),
  lmer = as.numeric(lmm_beta),
  coef = names(lmm_beta)
)

p_beta <- ggplot(df_beta, aes(x = lmer, y = dlmm, label = coef)) +
  geom_point(color = "#219ebc", size = 3) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "#023047") +
  geom_text_repel(size = 5, max.overlaps = Inf, box.padding = 2, fontface = "bold") +
  labs(
    x = "Pooled-Data Estimate", y = "PEAL Estimate",
    title = "(1). Comparison of Fixed Effect Estimates"
  ) +
  theme_minimal(base_size = 13) + theme(plot.title = element_text(face = "bold"))

# Panel B: Sigma
# Create df_sigma with grouping for color
df_sigma <- tibble(
  dlmm = as.numeric(peal_sigma),
  lmer = as.numeric(lmm_sigma),
  component = names(lmm_sigma),
  type = c(
    "Site RI", rep("Pt RI", H),  rep("Pt RS", H), "Residual"
  )
)

type_colors <- c(
  "Site RI" = "#1f78b4",     # Blue
  "Pt RI" = "#33a02c",       # Green
  "Pt RS" = "#ff7f00",       # Orange
  "Residual" = "#e31a1c"     # Red
)


# Sigma plot with color
p_sigma <- ggplot(df_sigma, aes(x = lmer, y = dlmm, label = component, color = type)) +
  geom_point(size = 3.5) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "gray30") +
  geom_text_repel(size = 3.5, max.overlaps = Inf, box.padding = 1.2, fontface = "bold") +
  labs(
    x = "Pooled-Data Estimate",
    y = "PEAL Estimate",
    title = "(2). Comparison of Random Effect SDs and Residual SD",
    color = "Component Type"
  ) +
  scale_color_manual(values = type_colors) +
  theme_minimal(base_size = 13) +
  theme(
    legend.position = "none",
    plot.title = element_text(face = "bold")
  )

# Combine with patchwork
compare_plot <- p_beta / p_sigma +
  plot_annotation(
    title = ""
  )

compare_plot
```


(1). Comparison of Fixed Effect Estimates
PEAL Estimate
X1
X2
X3
X4
X5
X6
Pooled−Data Estimate
(2). Comparison of Random Effect SDs and Residual SD
PEAL Estimate
site RI
pt RS by site3
pt RS by site2
pt RS by site1
residual
pt RI by site3
pt RI by site1
pt RI by site2
Pooled−Data Estimate