Analysis of the impact of non-genetical factors on immunophenotypes

Introduction

The *mmi* package was used in the article "Natural variation in immune cell parameters is preferentially driven by genetic factors," (Patin et al. 2018). The article tries to quantify the sources of variation for different functions in the immune system. It includes three different immunophenotypes: counts of immune cells per blood volume, mean flourescence intensity of surface markers per blood volume, and a few ratios between cell counts.

Determinants of a phenotype are either genetical, *i.e* related to differences in the genome of individuals, or non-genetical, intrinsic factors such as age, sex, and various cultural and demographical factors, or an interaction between the two. We focus on genetical and non-genetical factors in the article, leaving out interactions. A total of 166 immunophenotypes are analyzed, measured on 1,000 people from France stratified evenly across bins of 10 years, between 20 and 70 years old, and across sex.

Here we redo the analysis of the impact of non-genetical factor on the immunophenotypes. We do not reproduce the analysis of genetical factors, since we cannot publically release the genetics data due to privacy issues. Also, 184 of the 1000 donors included in the study did not want their data to be publically available. Therefore we only publish data from the remaining 816 donors.

The analysis is based on functionality implemented **mmi** package. We will not describe the inner workings of the package itself in any detail here, interested readers are pointed to the **mmi** R documentation.

First we load some libraries:

```
library(tidyverse)
library(glue)
library(magrittr)
library(scales)
library(mmi)
```

The immunophenotypes, stored in the variable *facs*, and the non-genetical variables, stored in the variable *ecrf*, are loaded to memory with the mmi package:

```
dim(facs)
```

```
## [1] 816 170
```

```
dim(ecrf)
```

```
## [1] 816  43
```

Included in the package is the tibble *facs_annotation* that stores the names of the immunophenotypes in the *facs* tibble, cleaned up names used in figures, and information on whether an immune parameter belongs to the adaptive or the innate immune system.

We will need a tibble including all data:

```
ecrf$SUBJID <- as.character(ecrf$SUBJID)
db <- left_join(facs, ecrf)
```

We can take a look at the distribution of age and sex in the data after removing donors that did not want their data to be public:

```
table(ecrf$Sex)
```

```
##
##   Male Female
##    399    417
```

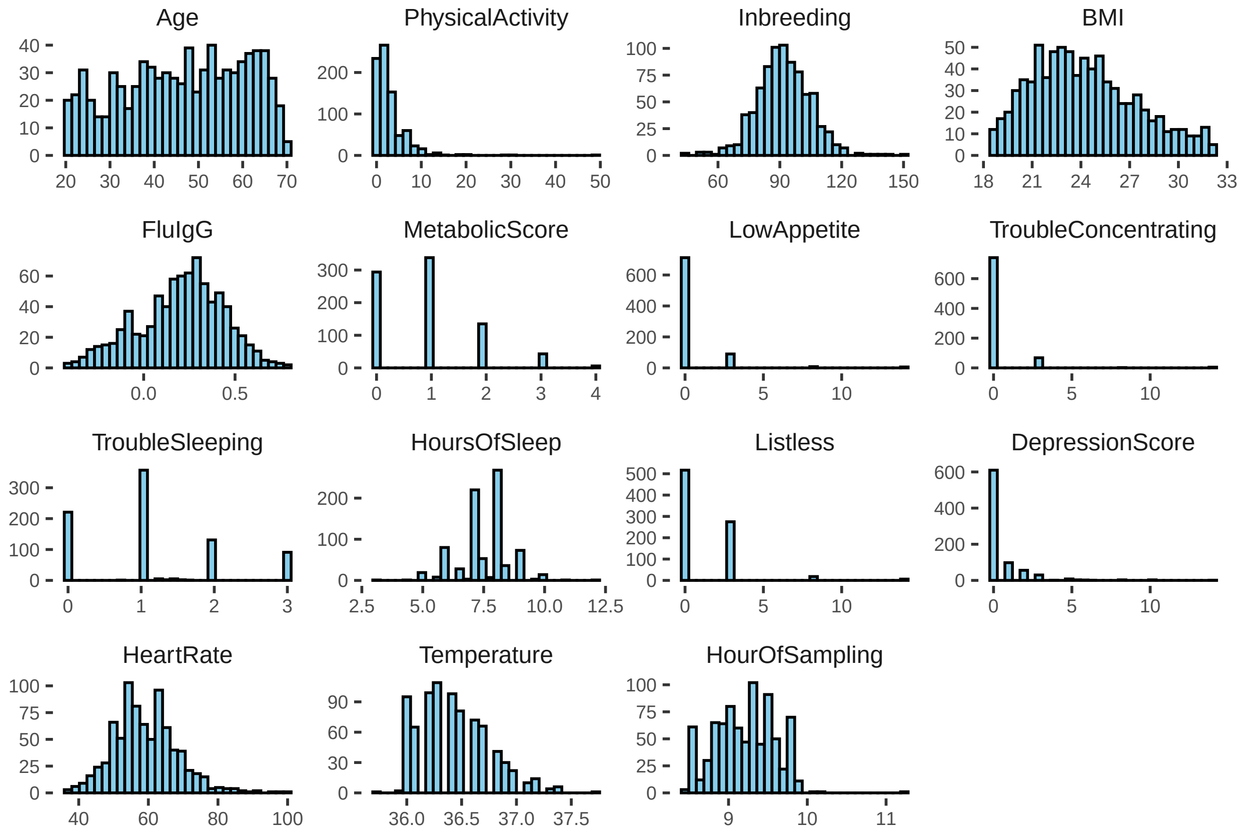
```
table(cut(ecrf$Age, breaks = c(20, 30, 40, 50, 60, 70)))
```

```
##
## (20,30] (30,40] (40,50] (50,60] (60,70]
##     122     160     170     188     176
```

Originally, each of these bins had 200 donors, so it is apparently mostly young people that are concerned with their data privacy.

The *mmi* package includes the function *plot_list* that conveniently plots histograms of variables in a data frame (actually it plots variables in a list, hence the name, but data frames are lists). We can use it to look at the numerical non-genetical variables:

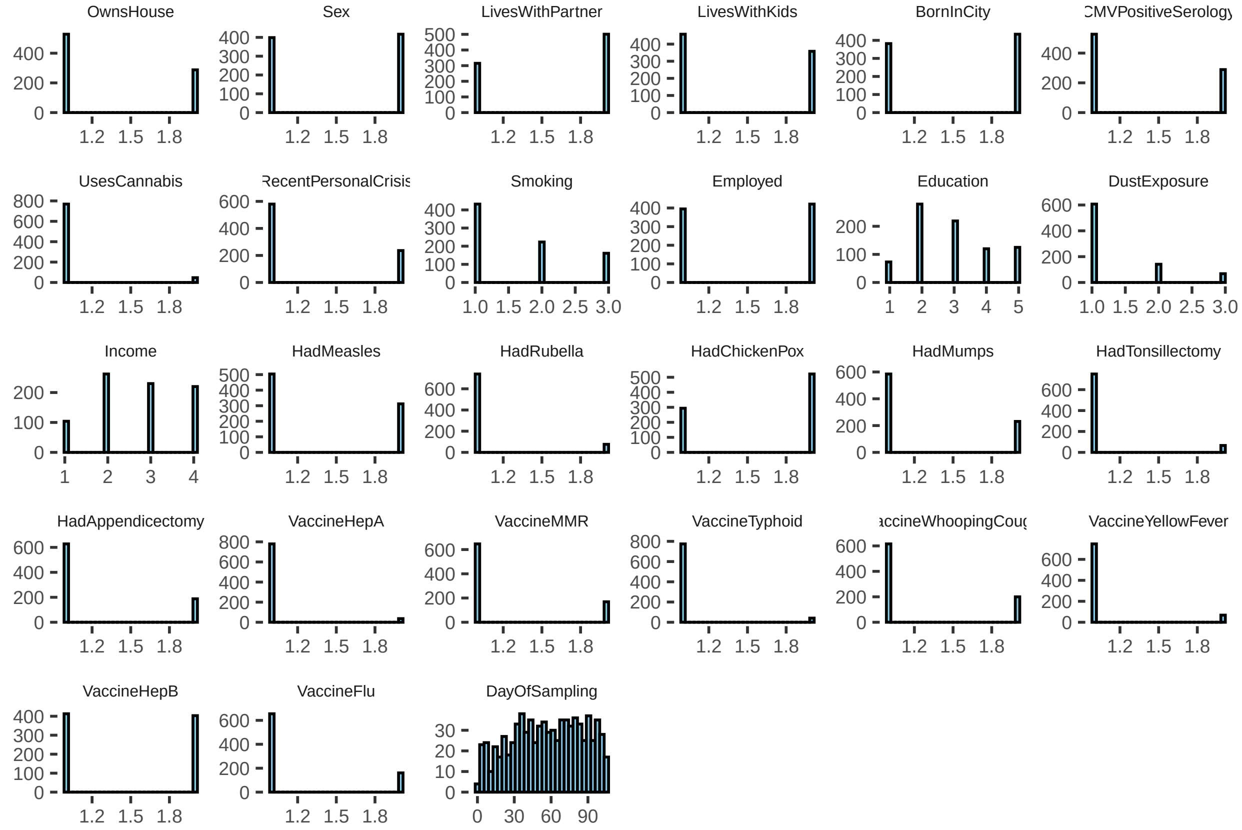
```
plot_list(keep(ecrf, is.numeric), facet_title_size = 9)
```



The *MetabolicScore* variable is a composite variable designed to measure an individuals metabolic health, described in (Thomas et al. 2015). The *HourOfSampling* is the hour of the day when the blood sample for the measurement was drawn. The other variables are usual intrinsic factors like age, typical clinical variables like body temperature and heart rate, antibodies for flu (*FluIgG*), and variables related to mental health. The variable *DepressionScore* is a composite variable similar to *MetabolicScore*, that summarizes the other mental health variables.

We can take a look at the categorical variables by first converting their values to numeric:

```
plot_list(keep(select(ecrf, -SUBJID), function(x) !is.numeric(x)), facet_title_size = 6)
```



There are some socio-economic variables like level of education and income, and variables related to vaccination and childhood diseases. The variables *Sex*, *CMVInfection*, which states if an individual is seropositive for the cytomegalovirus (https://en.wikipedia.org/wiki/Cytomegalovirus), and *Smoking* are key variables in the study. The *DayOfSampling* variable is what day the blood was drawn for the sample.

Batch effects

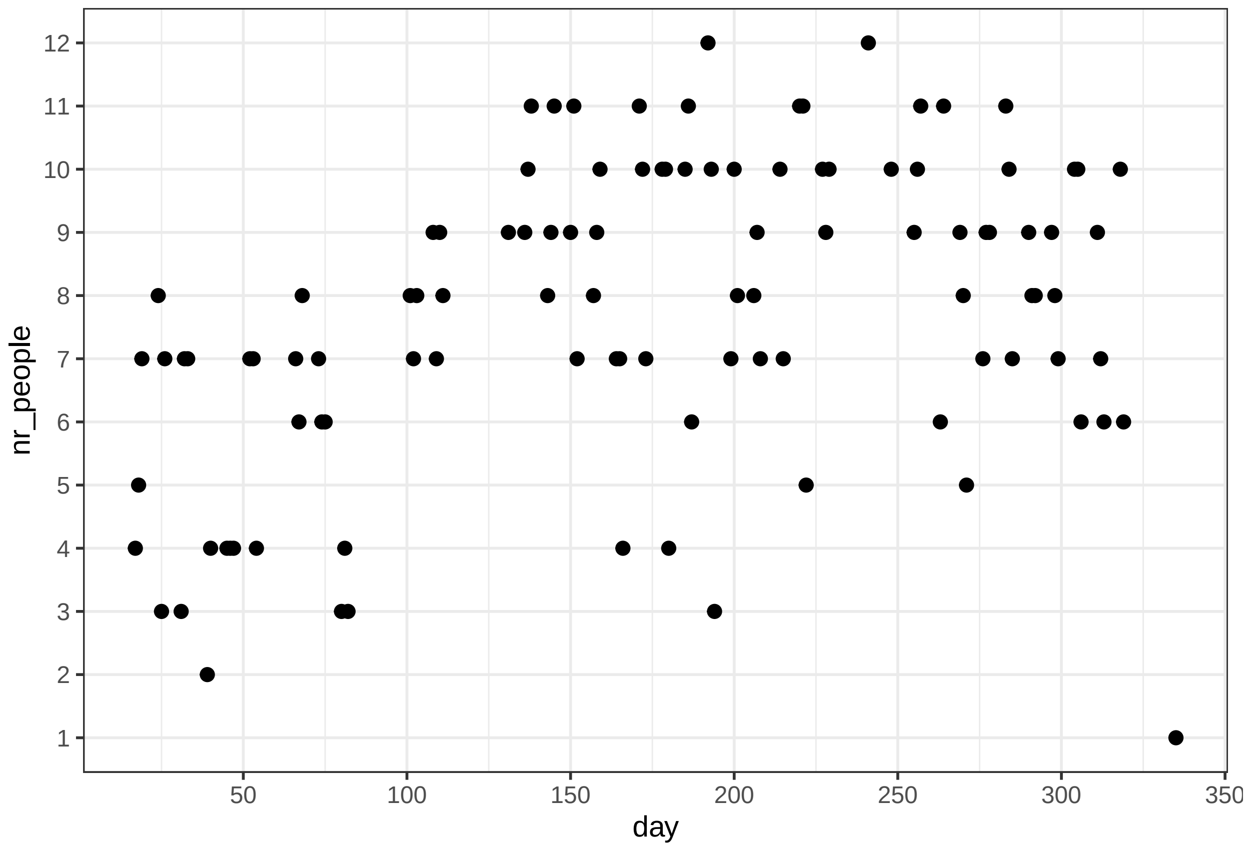
Let us look a little closer on the *DayOfSampling* variable. To see how many donors had blood drawn at a particular day (counted from the first day) we can for instance do:

```
people_each_day <- as.numeric(table(ecrf$DayOfSampling))
c(mean(people_each_day), 1.96 * sd(people_each_day))
```

```
## [1] 7.698113 4.802294
```

```
plt_frame <- tibble(day = as.numeric(levels(ecrf$DayOfSampling)),
                    nr_people = people_each_day)

ggplot(plt_frame, aes(x = day, y = nr_people)) +
  geom_point(size = 2) +
  scale_x_continuous(breaks = pretty_breaks(n = 10)) +
  scale_y_continuous(breaks = seq(1, 12, 1), minor_breaks = NULL) +
  theme_bw()
```



So around 7.7 ± 4.80 blood samples are drawn each day. To see if there is a batch effect across day of blood draw we can estimate the intraclass correlation coefficient. This is the correlation among observations within the same day (assuming the correlation is the same across days). If this parameter is large it means that the variation between days is larger then the variation within days, indicating a batch effect, since donors where randomly assigned to day of blood draw. The intraclass coefficient can also be interpreted as the proportion of the total variability attributed to variability between days. To compute it we must first setup the models. In this case we only want the day of sampling as a random effect, and we want to build a model for each immunophenotype:

```
spec <- specify(facs_annotation$FACS.NAME, model = "lmm", treatments =  "(1|DayOfSampling)")
fam <- make_fam(spec, db)
intraclass <- prop_var(fam)
select(intraclass, response, prop_var) %>% arrange(desc(prop_var))
```

```
## # A tibble: 166 x 2
##    response                             prop_var
##    <chr>                                   <dbl>
##  1 MFI_CCR7_in_CD8bpos_EM.panel1            88.6
##  2 MFI_CCR7_in_CD4pos_EM.panel1             88.4
##  3 MFI_CCR7_in_CD8bpos_EMRA.panel1          88.3
##  4 MFI_CCR7_in_CD8bpos_CM.panel1            86.4
##  5 CD19_MFI_in_Bcells.panel6                83.2
##  6 MFI_NEUTROPHILS_FceRI.panel7             83.1
##  7 MFI_CCR7_in_CD4pos_CM.panel1             77.4
##  8 MFI_CD16_in_CD14hi_mono.panel5      76.3
##  9 MFI_CCR7_oin_CD4pos_EMRA.panel1     74.2
## 10 MFI_HLADR_in_CD56hi.panel4          70.7
## # ... with 156 more rows
```

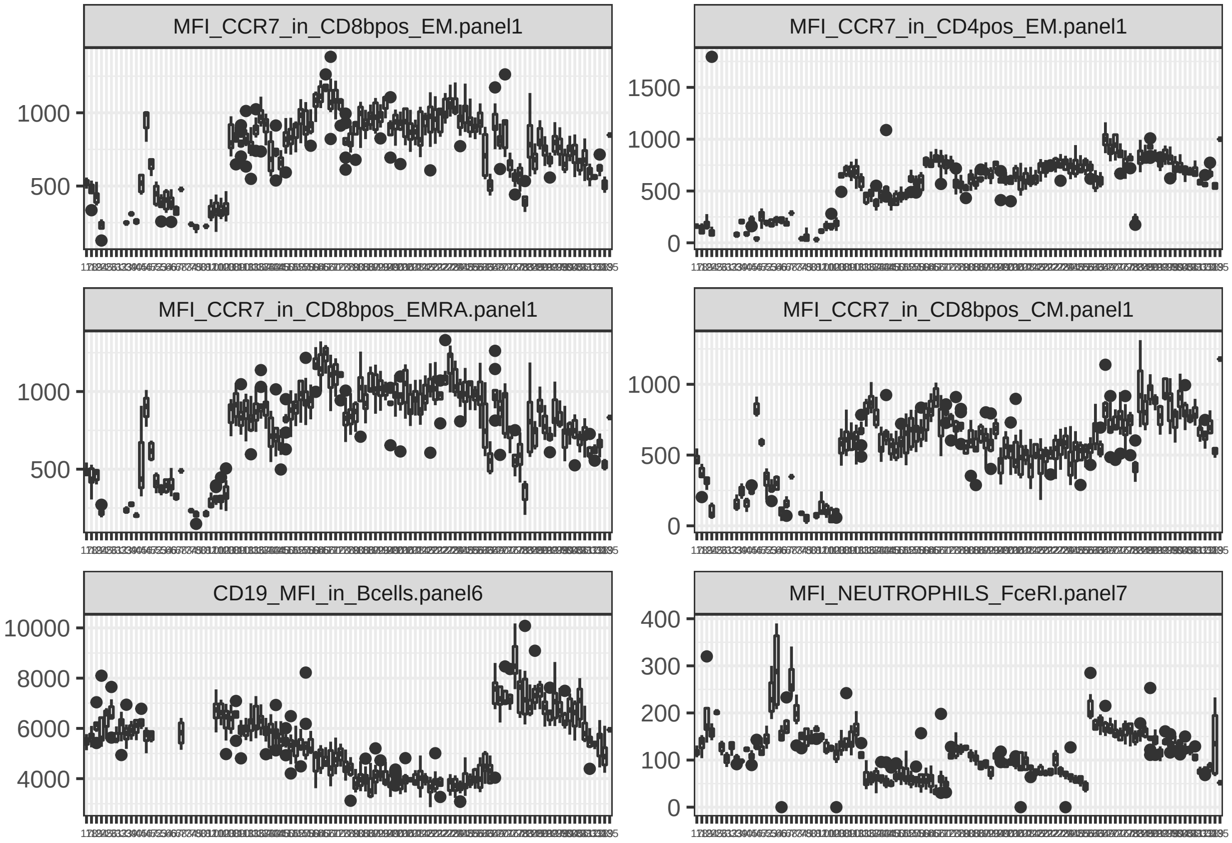
Apparently, day of blood draw has a huge effect on some immunophenotypes, particularly MFIs. To get a sense of how day of blood draw affects the measurements we can plot immunophenotype values across days. Let us take a look at the 6 immunophenotypes with the largest intraclass coefficient.

```
resp <- intraclass %>%
  arrange(desc(prop_var)) %>%
  slice(1:6) %$% response

plt_frame <- db[c("DayOfSampling", resp)] %>%
  gather(key = "Pheno", value = "Value", -DayOfSampling)

plt_frame$Pheno <- factor(plt_frame$Pheno, levels = resp)

plt_frame %>%
  ggplot(aes(x = factor(DayOfSampling), Value)) +
  geom_boxplot() +
  theme_bw() +
  theme(axis.text.x = element_text(size = 4),
        axis.title = element_blank(),
        strip.text = element_text(size = 8)) +
  facet_wrap(~ Pheno, scales = "free", ncol = 2)
```



Inference

We will fit a model for each response variable and each treatment variable, for a total of 5080 models. The models will be fit by the **mmi** package using functions from the **lme4** package. We will use age, sex, and the hour of blood draw as covariates for all models, and we add CMV infection status as a covariate for absolute counts and ratios. We adjust for day of blood-draw with a random effect. We log-transform the outcome in all cases. We regard all 5080 models as one multiple testing family, and we use the false discovery rate as error rate (Benjamini and Hochberg 1995). We test the hypothesis that the linear regression parameter for the treatment variable on a particular response variable is zero using the kenward-rogers approximation of the F-test (Kenward and Roger 1997). We construct confidence intervals using the implementation in the **lme4** package, which uses the profile likelihood (Bates et al. 2014). We only construct and show confidence intervals for parameters that are significant on the 0.01 level. To correct for this selection we widen the confidence intervals using the false coverage rate (FCR) procedure (Benjamini and Yekutieli 2005). The confidence intervals are constructed to have an FCR of 0.01.

Since we log-transform the data we need to make sure that there all observations are positive. Cell counts and MFIs should be positive but for some rare phenotypes there could be zeros. As long as there are not too many zeros its fine to just a cell to all observations and model the X + 1 variable.

```
facs_with_zero <- map_lgl(facs, ~ any(. <= 0, na.rm = TRUE))
sum(facs_with_zero)
```

```
## [1] 43
```

```
facs_with_zero <- names(facs[facs_with_zero])
facs_no_zeros <- facs
facs_no_zeros[facs_with_zero] <- facs_no_zeros[facs_with_zero] + 1
db_no_zeros <- left_join(facs_no_zeros, ecrf)
```

To perform the analysis we first create the specifications. The specify function specifies models for each combination of elements in the responses and treatments vectors supplied to the function. For each combination it adjusts for controls specified in the controls string. The response, treatment, and control terms should be given as string in formula syntax. The function understands **lme4** random effect syntax. It understands log and log10 transformations given in formula syntax. The package will backtransform estimated parameters and confidence intervals. To specify log transformed outcomes we can use the glue package.

```
str_mfis <- facs_annotation$FACS.NAME[grepl("^MFI_", facs_annotation$FACS.NAME)]
str_counts_and_ratios <- facs_annotation$FACS.NAME[grepl("^N_", facs_annotation$FACS.NAME)]
response_mfi <- glue("log({str_mfis})")
response_counts <- glue("log({str_counts_and_ratios})")
response_mfi[1:5]
```

```
## log(MFI_CD16_in_CD16hi_of_NKnew.panel4)
## log(MFI_CD16_of_CD56hi_of_NKnew.panel4)
## log(MFI_CD69_in_CD16hi.panel4)
## log(MFI_CD69_in_CD56hi.panel4)
## log(MFI_CD8a_in_CD16hi.panel4)
```

```
response_counts[1:5]
```

```
## log(N_CD16hi.panel4)
## log(N_CD56hi.panel4)
## log(N_CD69posCD16hi.panel4)
## log(N_CD69posCD56hi.panel4)
## log(N_CD8aposCD16hi.panel4)
```

```
controls_mfis <- "HourOfSampling + Sex + Age + (1|DayOfSampling)"
controls_counts <- "HourOfSampling + Sex + Age + CMVPositiveSerology + (1|DayOfSampling)"
```

```
# Define treatmets
str_treatments <- names(ecrf)
str_treatments <- str_treatments[!str_treatments %in%
                                   c("SUBJID", "HourOfSampling", "DayOfSampling")]
spec_mfis <- specify(response_mfi,
                     str_treatments,
                     controls = controls_mfis,
                     model = "lmm")


str_counts_and_ratios <- facs_annotation$FACS.NAME[grepl("^N_", facs_annotation$FACS.NAME)]
spec_counts <- specify(response_counts,
                       str_treatments,
                       controls = controls_counts,
                       model = "lmm")

spec <- c(spec_mfis, spec_counts)
spec
```

```
## spec_fam  object of length  5080
## Contains:
##  5080 spec_lmm objects
```

```
fam <- make_fam(spec, db_no_zeros)
fam
```

```
## fam  object of length  5080
## Contains:
##  5080 mmi_lmm objects
```

Note that this procedure is quite memory intensive. Next we do the hypothesis tests using *ft* and construct FCR-adjusted confidence intervals. Such confidence intervals can be constructed by the function *select_confidence*. This takes some time to run.

```
hyp <- ft(fam)
selected_confs <- select_confidence(fam, hyp, thresh = 0.01, level = 0.99)

# Add some metadata
selected_confs <- left_join(selected_confs, hyp) %>%
  left_join(facs_annotation, by = c("response" = "FACS.NAME")) %>%
  select(-response, -model_id) %>%
  rename(response = FACS.DESC)
```

There are 113 parameters significant on the 0.01 level. Printing and arranging, we see that the strongest signals are found for CMV infection status, age, and smoking:

```
left_join(hyp, facs_annotation, by = c("response" = "FACS.NAME")) %>%
  select(-response, -test, -Type, -model_id) %>%
  rename(response = FACS.DESC) %>%
  arrange(FDR)
```

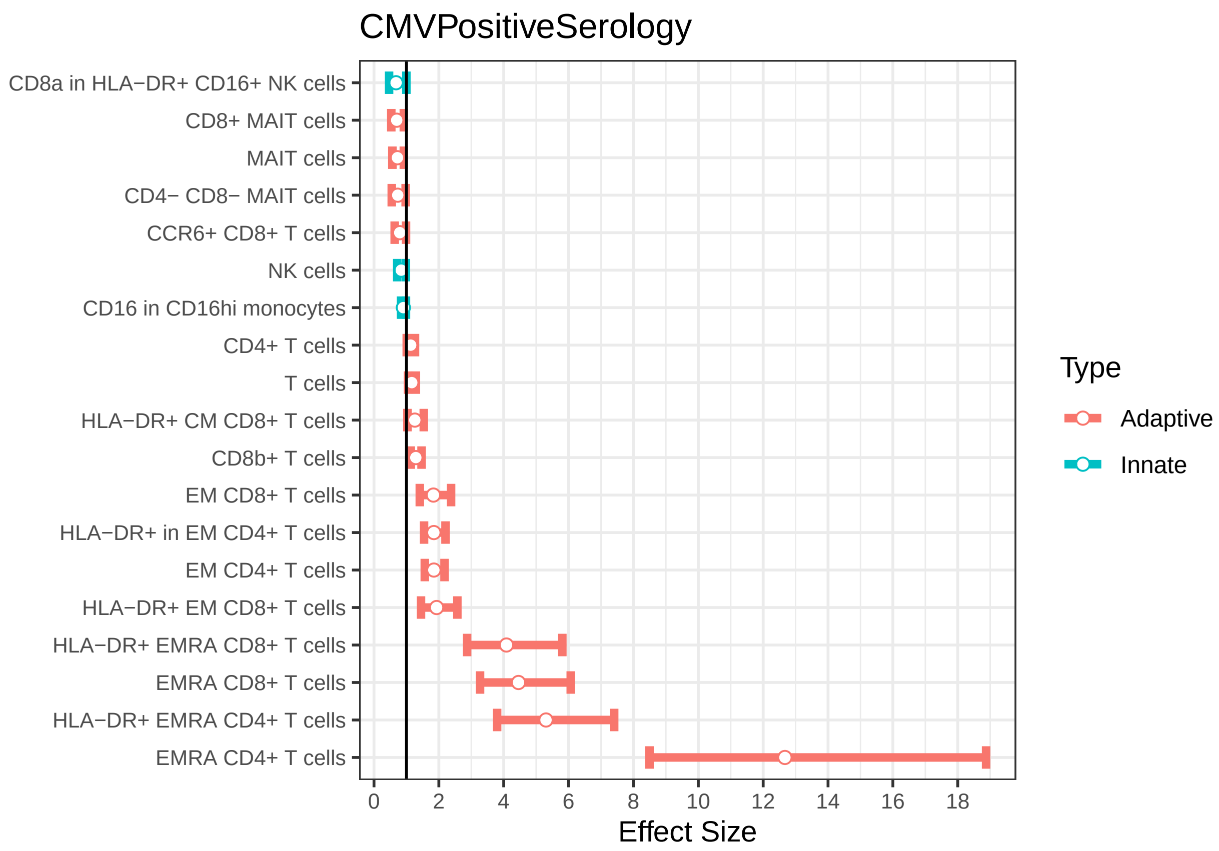
```
## # A tibble: 5,080 x 5
##    variable                    p model      FDR response
##    <chr>                   <dbl> <chr>    <dbl> <chr>
##   1 CMVPositiveSerology 8.07e-92 lmm   4.10e-88 EMRA CD4+ T cells
##   2 Age                 3.37e-80 lmm   8.57e-77 naive CD8+ T cells
##  3 CMVPositiveSerology 1.18e-62 lmm   2.01e-59 HLA-DR+ EMRA CD4+ T cells
##  4 CMVPositiveSerology 2.14e-59 lmm   2.71e-56 EMRA CD8+ T cells
##  5 CMVPositiveSerology 3.64e-43 lmm   3.70e-40 HLA-DR+ EMRA CD8+ T cells
##  6 CMVPositiveSerology 1.17e-38 lmm   9.91e-36 EM CD4+ T cells
##  7 CMVPositiveSerology 2.51e-33 lmm   1.82e-30 HLA-DR+ in EM CD4+ T cells
##  8 Age                 2.71e-31 lmm   1.72e-28 CD4- CD8- MAIT cells
##  9 Age                 6.84e-31 lmm   3.86e-28 CD8b- CD4- T cells
## 10 Age                 5.39e-28 lmm   2.74e-25 naive Treg
## # ... with 5,070 more rows
```

Finally, we plot the confidence intervals of the significant parameters, like we do in the paper. Here, we color the effects based on if the immunophenotype is a part of the adaptive or the innate immune system. Since we have to do this four times it is convenient to define a function that sets up the plot:

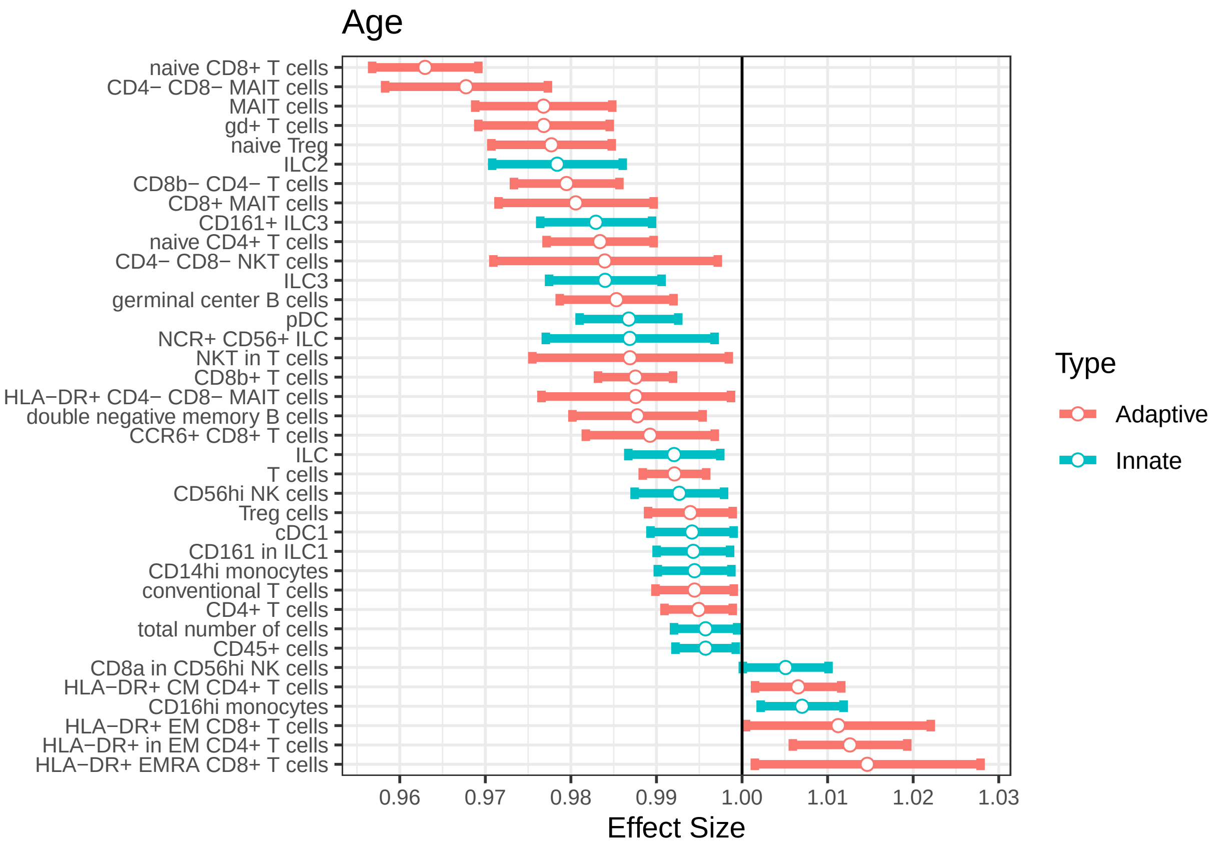
```
plot_effects <- function(frame, title) {
  plt_frame %>%
    ggplot(aes(x = reorder(response, desc(est)), y = est,
               ymin = lower, ymax = higher, colour = Type)) +
    geom_errorbar(width = 0.6, size = 1.5) +
    geom_point(size = 2, shape = 21, fill = "white") +
    scale_y_continuous(breaks = pretty_breaks(10)) +
    ylab("Effect Size") +
    geom_hline(aes(yintercept = 1), colour = "black") +
    coord_flip() +
    ggtitle(title) +
    xlab(NULL) +
    theme_bw() +
    theme(axis.text = element_text(size = 8))
}
```

Now we can produce the plots. Note that these will not be exactly the same as the plots in the paper because in the paper we also control for genome-wide significant SNPs, and also because of the observations removed due to privacy reasons:

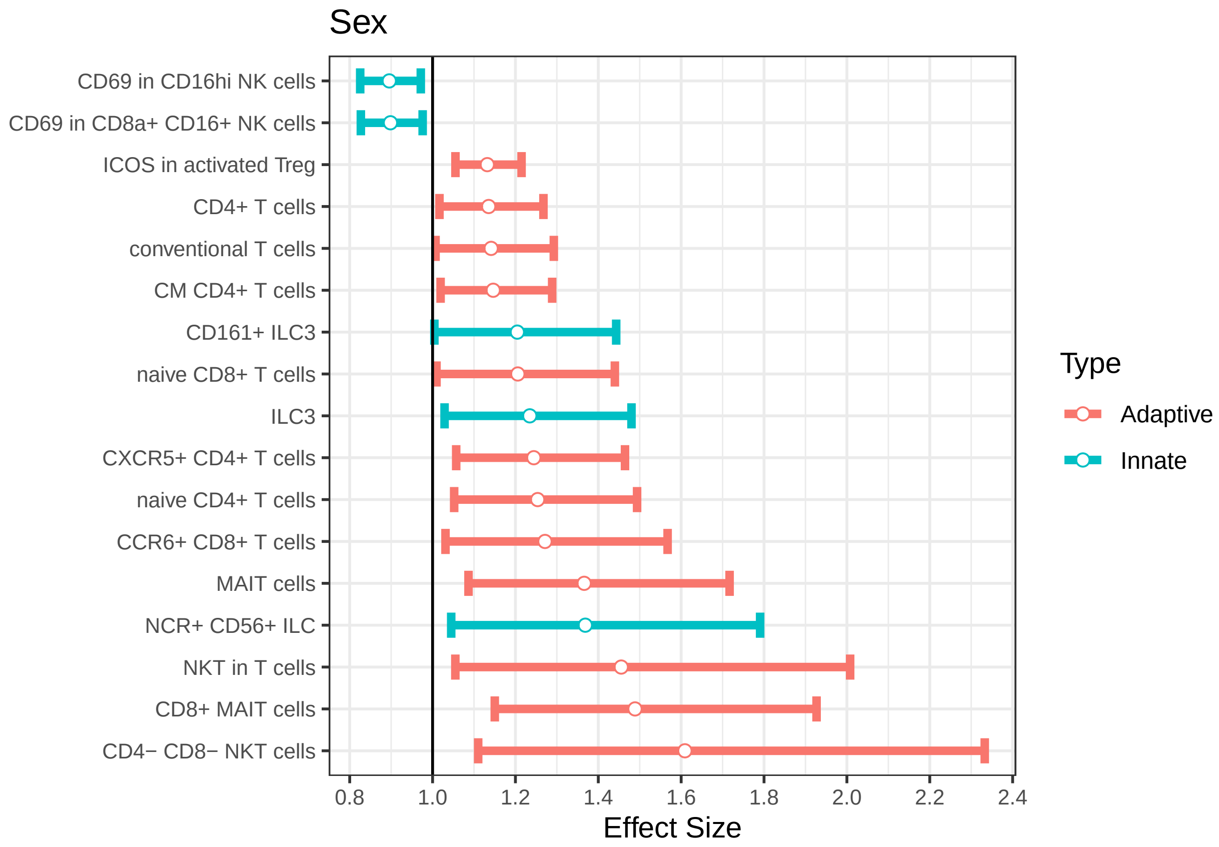
```
plt_frame <- selected_confs %>% filter(variable == "CMVPositiveSerology")
plot_effects(plt_frame, "CMVPositiveSerology")
```



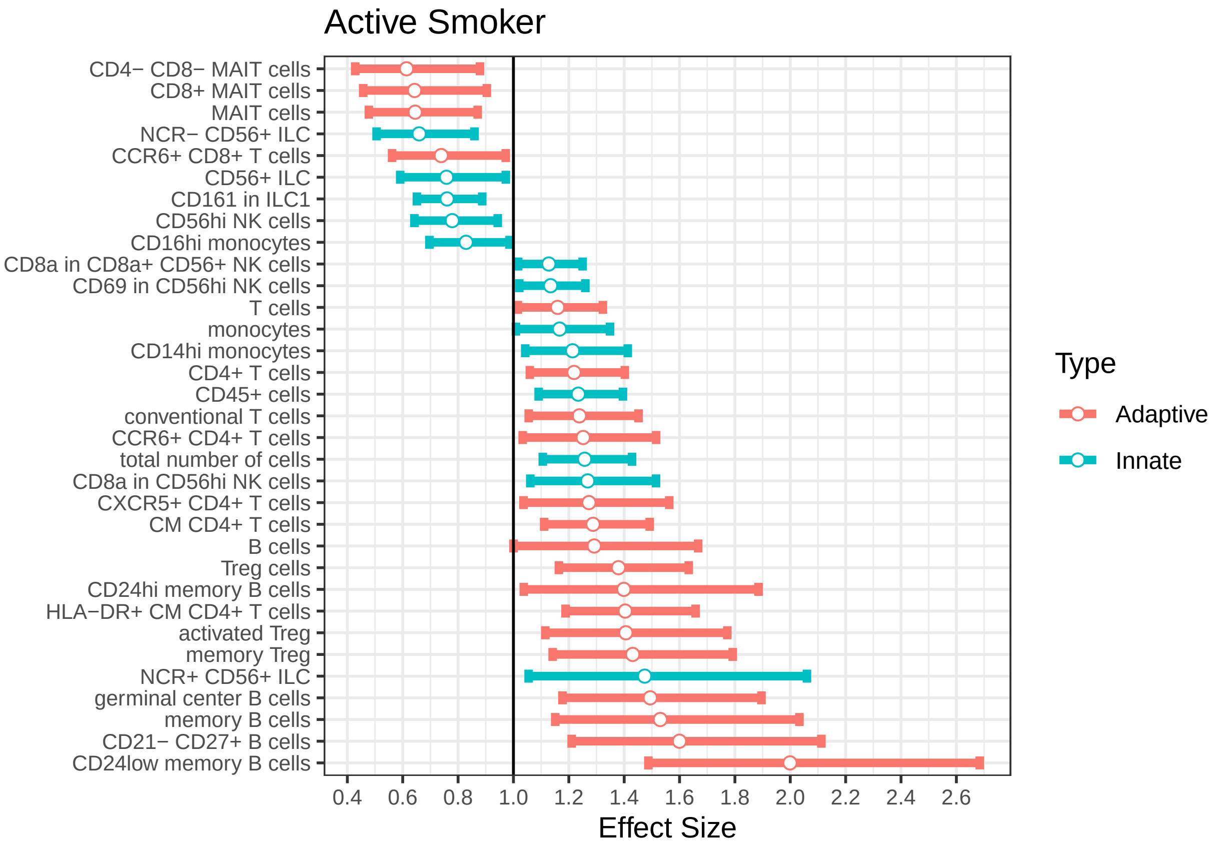
```
plt_frame <- selected_confs %>% filter(variable == "Age")
plot_effects(plt_frame, "Age")
```



```
plt_frame <- selected_confs %>% filter(variable == "Sex")
plot_effects(plt_frame, "Sex")
```



```
plt_frame <- selected_confs %>% filter(variable == "Smoking",
                                       levels == "SmokingActive")
plot_effects(plt_frame, "Active Smoker")
```



References

Bates, Douglas, Martin Maechler, Ben Bolker, Steven Walker, et al. 2014. "Lme4: Linear Mixed-Effects Models Using Eigen and S4." *R Package Version* 1 (7): 1–23.

Benjamini, Yoav, and Yosef Hochberg. 1995. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society. Series B (Methodological)* 57 (1): 289–300.

Benjamini, Yoav, and Daniel Yekutieli. 2005. "False Discovery Rate-Adjusted Multiple Confidence Intervals for Selected Parameters." *Journal of the American Statistical Association* 100 (469): 71–81.

Kenward, Michael G, and James H Roger. 1997. "Small Sample Inference for Fixed Effects from Restricted Maximum Likelihood." *Biometrics*, 983–97.

Patin, Etienne, Milena Hasan, Jacob Bergstedt, Vincent Rouilly, Valentina Libri, Alejandra Urrutia, Cécile Alanio, et al. 2018. "Natural Variation in the Parameters of Innate Immune Cells Is Preferentially Driven by Genetic Factors." *Nature Immunology* 19 (3): 302–14. https://doi.org/10.1038/s41590-018-0049-7.

Thomas, Stéphanie, Vincent Rouilly, Etienne Patin, Cécile Alanio, Annick Dubois, Cécile Delval, Louis-Guillaume Marquier, et al. 2015. "The Milieu Intérieur Study – an Integrative Approach for Study of Human Immunological Variance." *Clinical Immunology* 157 (2): 277–93. https://doi.org/https://doi.org/10.1016/j.clim.2014.12.004.