R Notebook

LM

26062019

Modelling driver mutation burden in normal endometrium

Markdown file to document methods used in the analysis of the driver mutation burden in normal endometrium.

Load Libraries

```
library(tidyverse)
library(magrittr)
library(lme4)
library(lmerTest)
library(rlang)
library(knitr)
library(kableExtra)
library(pbkrtest)
```

Load in data files

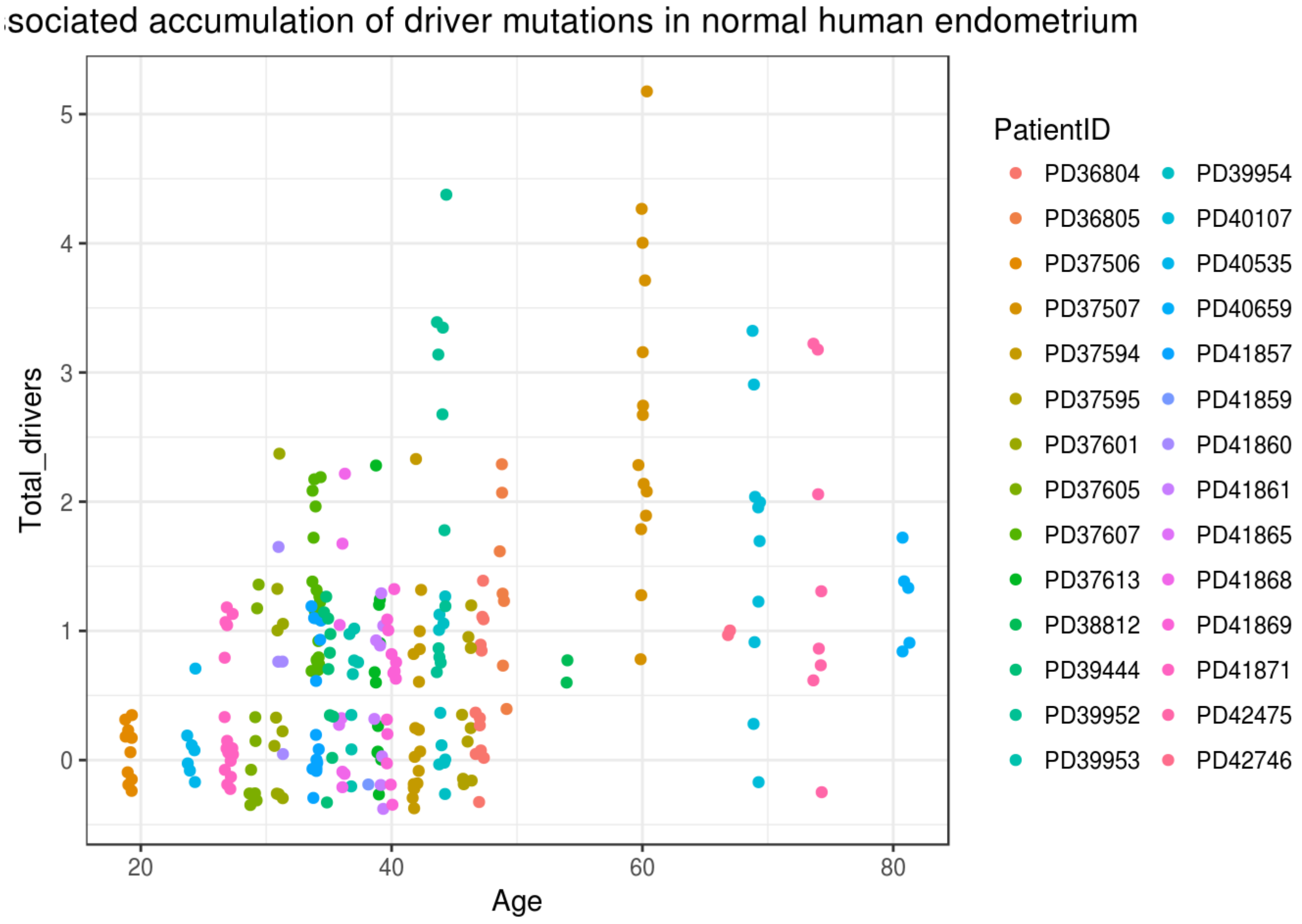
Load in sample level data for the 28 donors with associated meta-data, including Body Mass Index (BMI), Parity and Cohort (sample source).

```
endom_burden <- read.csv("Endometrium_for_model_26062019.csv", stringsAsFactors = F, na.strings = c("", "NA", "Unknown", "Uncertain"))
# Samples per patient
endom_burden %>% group_by(PatientID) %>%  count(PatientID) %>%  rename(`Sample count` = n) %>% arrange(desc(`Sample count`)) %>%  kable() %>%  kable_styling(bootstrap_options = c("striped", "condensed"), full_width = F, position = "left")
```

PatientID	Sample count
PD37607	19
PD37594	17
PD41871	17
PD37507	14
PD41857	14
PD36804	13
PD41869	13
PD37613	11
PD39952	11

PatientID	Sample count
PD37506	10
PD37601	10
PD39444	10
PD39954	10
PD40107	10
PD37595	9
PD37605	9
PD39953	8
PD41861	8
PD42475	8
PD36805	7
PD40535	7
PD41868	6
PD40659	5
PD41860	4
PD38812	2
PD41865	2
PD42746	2
PD41859	1

```
# Look at the raw data
  endom_burden %>% ggplot(aes(Age, Total_drivers, colour = PatientID)) +
  geom_jitter() +
  theme(plot.title = element_text(size = 8)) +
  ggtitle("Age-associated accumulation of driver mutations in normal human endometrium") +
  theme(plot.title = element_text(size = 14)) + theme_bw() +theme(plot.title = element_text
(hjust = 0.5))
```



Fit a mixed-effect model to estimate driver mutation rates

To account for the non-independent sampling per patient we use a generalized linear mixed effects model with Poisson distribution. We also use a random slope with fixed intercept as most women will start menarche at a similar age (~13 years), but to account for the potential differences in the rates at which mutations were acquired in different individuals due to variation in parity, contraception and other factors.

We test features that can have an effect on mutation burden or are modulate endometrial cancer risk:

- Age
- Read depth & VAF (‘Vafdepth’)
- BMI
- Parity
- Cohort

We use backwards elimination to define the final model

Define full model and drop each fixed effect in turn

```
# Combine read depth and median sample depth (Seq_X) as 'Vafdepth'
  endom_burden %<>%  mutate(Vafdepth = Seq_X*SampleMedianVAF)

# Make BMI and Parity numeric
  endom_burden %<>%  mutate(BMI.QC = as.numeric(BMI))
  endom_burden %<>%  mutate(Parity.QC = as.numeric(Parity))

# Exclude cases without Parity data
  endom_burden.qc <- endom_burden %>% filter(!is.na(Parity.QC))

# Define the full model containing all features
  full_glmer_model = glmer(Total_drivers ~ Age + Vafdepth + BMI.QC + Parity.QC + Cohort +(Age - 1|PatientID), data=endom_burden.qc, family = poisson(link = "log"), control =   glmerControl(optimizer="bobyqa", optCtrl = list(maxfun = 100000)))

  print(summary(full_glmer_model))
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: Total_drivers ~ Age + Vafdepth + BMI.QC + Parity.QC + Cohort +
##     (Age - 1 | PatientID)
##    Data: endom_burden.qc
## Control:
## glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))
##
##      AIC      BIC   logLik deviance df.resid
##    483.6    514.6   -232.8    465.6      222
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.2757 -0.7002 -0.1361  0.5323  2.0615
##
## Random effects:
##  Groups    Name Variance  Std.Dev.
##  PatientID Age  4.832e-05 0.006951
## Number of obs: 231, groups:  PatientID, 25
##
## Fixed effects:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -1.937221   0.728279  -2.660  0.00781 **
## Age                     0.031603   0.011826   2.672  0.00753 **
## Vafdepth                0.044643   0.028273   1.579  0.11434
## BMI.QC                 -0.006626   0.023231  -0.285  0.77547
## Parity.QC              -0.259493   0.113226  -2.292  0.02192 *
## CohortPost-mortem       0.242012   0.917639   0.264  0.79199
## CohortTAH               0.153797   0.424937   0.362  0.71741
## CohortTransplant donor  0.304985   0.280186   1.089  0.27637
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Age    Vfdpth BMI.QC Prt.QC ChrtP- ChrTAH
## Age         -0.493
## Vafdepth    -0.311  0.087
## BMI.QC      -0.626 -0.136 -0.275
## Parity.QC   -0.271 -0.211 -0.003  0.371
## ChrtPst-mrt  0.300 -0.502  0.000 -0.013 -0.264
## CohortTAH    0.243 -0.275  0.115 -0.225 -0.197  0.281
## ChrtTrnspld  0.305 -0.450  0.045 -0.167 -0.210  0.412  0.388
```

```
# "user" parametric boot function as defined in drop1.merMod help example
  PBSumFun <- function(object, objectDrop, ...) {
    pbnames <- c("stat", "p.value")
    r <- if (missing(objectDrop)) {
      setNames(rep(NA, length(pbnames)), pbnames)
    } else {
      pbtest <- PBmodcomp(object, objectDrop, nsim = nsim, ref = NULL, seed=12345, details
 = 0)
      unlist(pbtest$test[2, pbnames])
    }
    attr(r, "method") <- c("Parametric bootstrap via pbkrtest package")
    r
  }
# Drop each fixed effect from model and test significance
# Use 1000 samples to form the reference distribution
nsim <- 1000
drop1(full_glmer_model, test = "user", sumFun = PBSumFun)
```

```
## Single term deletions
##
## Model:
## Total_drivers ~ Age + Vafdepth + BMI.QC + Parity.QC + Cohort +
##     (Age - 1 | PatientID)
## Method:
## Parametric bootstrap via pbkrtest package
##
##
##             stat p.value
## <none>
## Age       6.7178 0.05277
## Vafdepth  2.4586 0.14317
## BMI.QC    0.0821 0.83577
## Parity.QC 5.3143 0.08761
## Cohort    1.1445 0.85466
```

Remove feature with the largest P > 0.05 to make reduced model 1

```
# Remove Cohort from the full model
  reduced1_glmer_model <- update(full_glmer_model, ~ . -Cohort, control=glmerControl(optimi
zer="bobyqa", optCtrl = list(maxfun = 100000)))
# Drop each fixed effect from the model and test significance
  drop1(reduced1_glmer_model, test = "user", sumFun = PBSumFun)
```

```
## Single term deletions
##
## Model:
## Total_drivers ~ Age + Vafdepth + BMI.QC + Parity.QC + (Age -
##     1 | PatientID)
## Method:
## Parametric bootstrap via pbkrtest package
##
##
##              stat p.value
## <none>
## Age       10.8137 0.00326
## Vafdepth   2.3500 0.13436
## BMI.QC     0.0160 0.91478
## Parity.QC  4.7712 0.06361
```

Remove next feature with the largest P > 0.05 to make reduced model 2

```
# Remove BMI from the above model
  reduced2_glmer_model <- update(reduced1_glmer_model, ~ . -BMI.QC, control=glmerControl(op
timizer="bobyqa", optCtrl = list(maxfun = 100000)))
# Drop each fixed effect from the model and test significance
  drop1(reduced2_glmer_model, test = "user", sumFun = PBSumFun)
```

```
## Single term deletions
##
## Model:
## Total_drivers ~ Age + Vafdepth + Parity.QC + (Age - 1 | PatientID)
## Method:
## Parametric bootstrap via pbkrtest package
##
##
##              stat  p.value
## <none>
## Age       10.8621 0.002105
## Vafdepth   2.4033 0.137539
## Parity.QC  5.0721 0.037190
```

Remove next feature with the largest P > 0.05 to make reduced model 3

```
# Remove Vafdepth from the above model
  reduced3_glmer_model <- update(reduced2_glmer_model, ~ . -Vafdepth, control=glmerControl
(optimizer="bobyqa", optCtrl = list(maxfun = 100000)))
# Drop each fixed effect from model and test significance
  drop1(reduced3_glmer_model, test = "user", sumFun = PBSumFun)
```

```
## Single term deletions
##
## Model:
## Total_drivers ~ Age + Parity.QC + (Age - 1 | PatientID)
## Method:
## Parametric bootstrap via pbkrtest package
##
##
##              stat  p.value
## <none>
## Age       10.3793 0.003125
## Parity.QC  5.8943 0.019348
```

Define the final model

```
# Define the final model keeping only the significant features  (P < 0.05)

  final_glmer_model <- reduced3_glmer_model

# Print summary for the final model
  print(summary(final_glmer_model))
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: Total_drivers ~ Age + Parity.QC + (Age - 1 | PatientID)
##    Data: endom_burden.qc
## Control: 
## glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 1e+05))
##
##      AIC      BIC   logLik deviance df.resid 
##    477.1    490.9   -234.6    469.1      227 
##
## Scaled residuals: 
##     Min      1Q  Median      3Q     Max 
## -1.2451 -0.6912 -0.1927  0.6225  2.0057 
##
## Random effects:
##  Groups    Name Variance  Std.Dev.
##  PatientID Age  5.987e-05 0.007738
## Number of obs: 231, groups:  PatientID, 25
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -1.643601   0.391387  -4.199 2.68e-05 ***
## Age          0.035460   0.009878   3.590 0.000331 ***
## Parity.QC   -0.253115   0.102227  -2.476 0.013285 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr) Age   
## Age       -0.930       
## Parity.QC  0.204 -0.440
```

```
# Estimate confidence intervals using "likelihood profile" method
  confint.merMod(final_glmer_model, method = "profile")
```

```
## Computing profile confidence intervals ...
```

```
##                    2.5 %      97.5 %
## .sig01       0.002577037  0.01361534
## (Intercept) -2.493282376 -0.87980304
## Age          0.015388799  0.05650318
## Parity.QC   -0.463678195 -0.05087779
```