2.Gene_expresion

Fay

2022-05-27

Libraries:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidyr)
library(dplyr)
library(cowplot)
library(randomForest)
```

```
## randomForest 4.7-1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

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

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
g <- read.csv("output_data/gene_expression/data_products/clean_gene_expression.csv")


# vectors for selecting gene columns
Genes <- c("IFNy", "CXCR3_bio", "IL.6", "IL.10", "IL.13", "IL.10", "IL.13", "IL1RN",
           "CASP1", "CXCL9", "IDO1", "IRGM1", "MPO", "MUC2", "MUC5AC", "MYD88",
           "NCR1", "PRF1", "RETNLB", "SOCS1", "TICAM1", "TNF")
```

Import the data:

```r
g$Parasite_challenge <- as.factor(g$Parasite_challenge)
g$Eim_MC <- as.factor(g$Eim_MC)
```

Data cleaning

```r
g <- g %>%
  dplyr::mutate(Infection = case_when(
    Parasite_challenge == "E_ferrisi" & Eim_MC == "TRUE" ~ "E_ferrisi",
    Parasite_challenge == "E_ferrisi" & Eim_MC == "FALSE" ~ "uninfected",
    Parasite_challenge == "E_falciformis" & Eim_MC == "TRUE" ~ "E_falciformis",
    Parasite_challenge == "E_falciformis" & Eim_MC == "FALSE" ~ "uninfected",
    Parasite_challenge == "uninfected" & Eim_MC == "TRUE" ~ "infected_eimeria",
    Parasite_challenge == "uninfected" & Eim_MC == "FALSE" ~ "uninfected",
    TRUE ~ ""
  ))


# how to impute delta? Replacing with 0 the ones with negative melting curve
g <- g %>%
  dplyr::mutate(Intensity = case_when(
    Eim_MC == "TRUE" ~ delta,
    Eim_MC == "FALSE" ~ 0))

g.1 <- g %>%
  dplyr::select(c(max_WL, all_of(Genes)))

# to get reproducible results we use a seed
set.seed(42)

# We want the maximum weight loss to be predicted by the data ina ll of the other columns

# iter = how many random forests are needed, in theory 6 are enough
g.imputed <- rfImpute(max_WL ~ ., data = g.1, iter = 6)
```

Imputing missing data + cleaning

```
##      |      Out-of-bag   |
## Tree |      MSE  %Var(y) |
##  300 |    26.48    62.00 |
##      |      Out-of-bag   |
## Tree |      MSE  %Var(y) |
##  300 |    27.85    65.22 |
##      |      Out-of-bag   |
## Tree |      MSE  %Var(y) |
##  300 |    27.93    65.39 |
##      |      Out-of-bag   |
## Tree |      MSE  %Var(y) |
##  300 |    27.44    64.24 |
##      |      Out-of-bag   |
## Tree |      MSE  %Var(y) |
##  300 |    28.47    66.65 |
##      |      Out-of-bag   |
## Tree |      MSE  %Var(y) |
##  300 |    28.63    67.03 |
```

```
g.imputed <- g.imputed %>% dplyr::select(-max_WL)

g <- g %>%
  dplyr::select(-all_of(Genes))

#full data set containing the imputed gene expression data
g.imputed <- cbind(g, g.imputed)
```

How many mice are in the infection planning?

```
g.imputed %>%
  filter(infection == "challenge") %>%
  group_by(Parasite_challenge) %>%
  summarize(length(EH_ID))
```

```
## # A tibble: 3 x 2
##   Parasite_challenge `length(EH_ID)`
##   <fct>                        <int>
## 1 E_falciformis                   22
## 2 E_ferrisi                       47
## 3 uninfected                      47
```

How many mice are indeed infected?

```
g.imputed %>%
  filter(infection == "challenge") %>%
  group_by(Infection) %>%
  summarize(length(EH_ID))
```

```
## # A tibble: 4 x 2
##   Infection        `length(EH_ID)`
##   <chr>                      <int>
## 1 E_falciformis                 22
## 2 E_ferrisi                     39
## 3 infected_eimeria               9
## 4 uninfected                    46
```

I gues mice got mixed up here?

Splitting data into training and testing sets Splitting between training and testing: - Assess model performance on unseen data - Avoid over-fitting

```
#select the relevant columns:
g.imputed <- g.imputed %>%
  dplyr::select(c(max_WL, Infection, Intensity, all_of(Genes)))

# split data into training and test

set.seed(123) # this will help us reproduce this random assignment

# in this way we can pick the random numbers

training.samples <- g.imputed$max_WL%>%
  createDataPartition(p = .7, # this is the partiicition! In this case 0.7 = training data and 0.3 = te
                      list = FALSE) # we don't want to get a list in return

train.data <- g.imputed[training.samples, ] #include all the randomly selected rows
test.data <- g.imputed[-training.samples, ]
```

```
#train the model
model <- randomForest(max_WL ~., data = train.data, proximity = TRUE)
```

Building the model

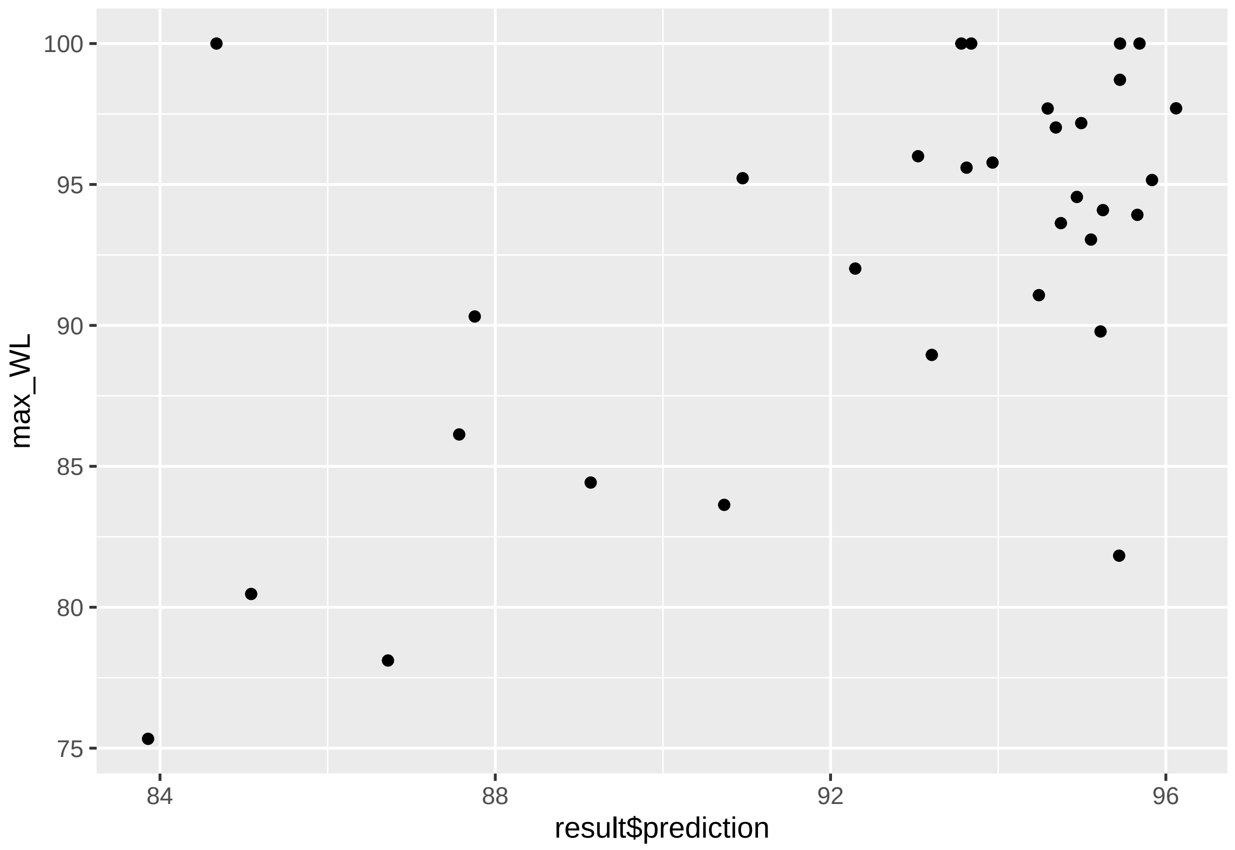
```
predictions <- predict(model, test.data)


result <- test.data
result['prediction'] <- predictions


#add the results to a data frame containing test data and the prediction
result <- cbind(g[row.names(result), ], result$prediction)
```

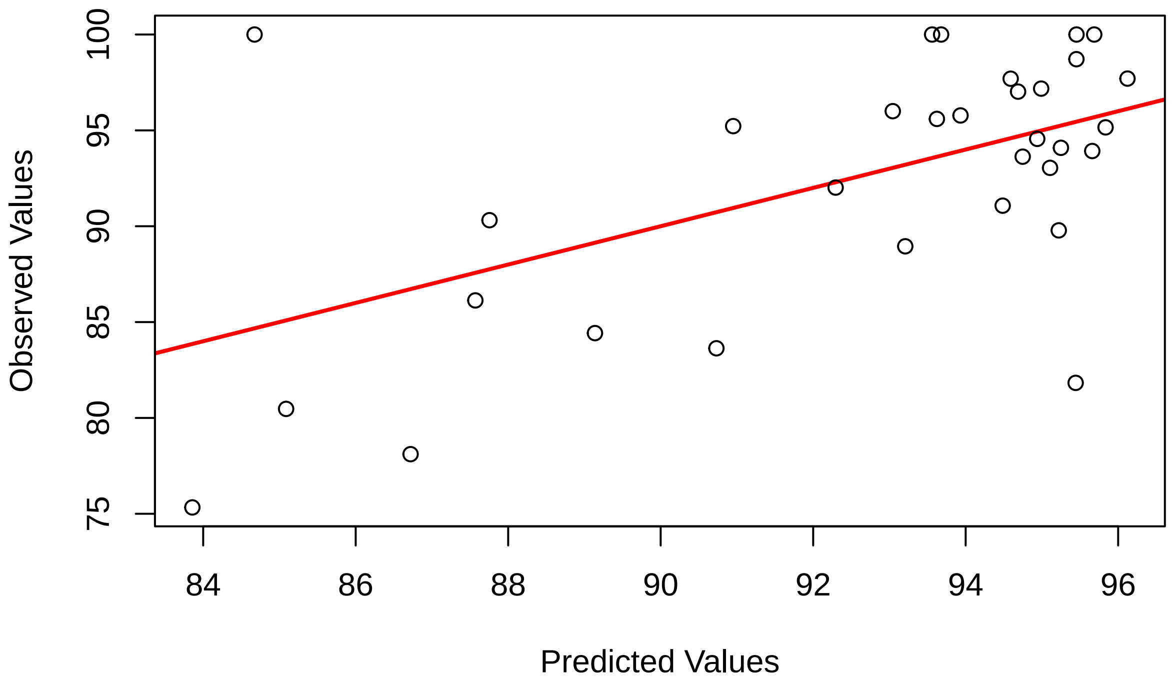
Making predictions

```
result   %>%
  ggplot() +
  geom_point(aes(x = `result$prediction`, y = max_WL))
```



Visualizations

```
plot(y =result$max_WL, x = result$`result$prediction`,
     # Draw plot using Base R
     xlab = "Predicted Values",
     ylab = "Observed Values",
     abline(a = 0,                                          # Add straight line
       b = 1,
       col = "red",
       lwd = 2))
```


Observed Values
Predicted Values
75
80
85
90
95
100
84
86
88
90
92
94
96