Mice_imputation.rmd

Fay

2022-11-01

Load libraries

```
library(mice)
```

```
##
## Attaching package: 'mice'

## The following object is masked from 'package:stats':
##
##     filter

## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
library(tidyr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v ggplot2 3.4.1      v dplyr   1.0.10
## v tibble  3.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## v purrr   0.3.5
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks mice::filter(), stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(VIM)
```

```
## Loading required package: colorspace
## Loading required package: grid
## VIM is ready to use.
##
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues
##
## Attaching package: 'VIM'
##
## The following object is masked from 'package:datasets':
##
##     sleep
```

```
library(fitdistrplus)
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
##
## Loading required package: survival
```

```
library(fitur)
```

```
##
## Attaching package: 'fitur'
##
## The following object is masked from 'package:purrr':
##
##     rdunif
```

```
library(visdat)
library(DESeq2)
```

```
## Loading required package: S4Vectors
## Loading required package: stats4
## Loading required package: BiocGenerics
##
## Attaching package: 'BiocGenerics'
##
## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union
##
## The following objects are masked from 'package:mice':
##
##     cbind, rbind
##
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
##
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min
##
##
## Attaching package: 'S4Vectors'
##
## The following objects are masked from 'package:dplyr':
##
##     first, rename
##
```

```
## The following object is masked from 'package:tidyr':
##
##     expand
##
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
##
## Loading required package: IRanges
##
## Attaching package: 'IRanges'
##
## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice
##
## The following object is masked from 'package:purrr':
##
##     reduce
##
## Loading required package: GenomicRanges
## Loading required package: GenomeInfoDb
## Loading required package: SummarizedExperiment
## Loading required package: MatrixGenerics
## Loading required package: matrixStats
##
## Attaching package: 'matrixStats'
##
## The following object is masked from 'package:dplyr':
##
##     count
##
##
## Attaching package: 'MatrixGenerics'
##
## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars
##
## Loading required package: Biobase
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
##
##
## Attaching package: 'Biobase'
##
## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians
##
## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

Load data

Import data

```
hm <- read.csv("output_data/2.1.norm_MICE_data_set.csv")
```

I only include GAPDH as a housekeeping gene, as PPIB is missing in a large number

```
# Vectors for selecting genes
#Lab genes
# The measurements of IL.12 and IRG6 are done with an other assay and will
#ignore for now
Gene_lab   <- c("IFNy", "CXCR3", "IL.6", "IL.13", # "IL.10",
                "IL1RN","CASP1", "CXCL9", "IDO1", "IRGM1", "MPO",
                "MUC2", "MUC5AC", "MYD88", "NCR1", "PRF1", "RETNLB", "SOCS1",
                "TICAM1", "TNF") #"IL.12", "IRG6")

Genes_wild   <- c("IFNy", "CXCR3", "IL.6", "IL.13",# "IL.10",
                  "IL1RN","CASP1", "CXCL9", "IDO1", "IRGM1", "MPO",
                  "MUC2", "MUC5AC", "MYD88", "NCR1", "PRF1", "RETNLB", "SOCS1",
                  "TICAM1", "TNF") #, "IL.12", "IRG6")

Facs_lab <- c("CD4", "Treg", "Div_Treg", "Treg17", "Th1",
                    "Div_Th1", "Th17", "Div_Th17", "CD8", "Act_CD8",
                    "Div_Act_CD8", "IFNy_CD4", "IFNy_CD8","Treg_prop",
                    "IL17A_CD4")

Facs_wild <- c( "Treg", "CD4", "Treg17", "Th1", "Th17", "CD8",
                     "Act_CD8", "IFNy_CD4", "IL17A_CD4", "IFNy_CD8")
```

data imputation

Genes

```r
field <- hm %>%
  dplyr::filter(origin == "Field")

field <- unique(field)
genes_mouse_field <- field %>%
  dplyr::select(c(Mouse_ID, "IFNy", "CXCR3", "IL.6", "IL.13",# "IL.10",
                  "IL1RN","CASP1", "CXCL9", "IDO1", "IRGM1", "MPO",
                  "MUC2", "MUC5AC", "MYD88", "NCR1", "PRF1", "RETNLB", "SOCS1",
                  "TICAM1", "TNF"))

genes <- genes_mouse_field  %>%
  dplyr::select(-Mouse_ID)
#remove rows with only nas
genes <- genes[,colSums(is.na(genes))<nrow(genes)]
#remove colums with only nas
genes <- genes[rowSums(is.na(genes)) != ncol(genes), ]
genes_mouse_field <- genes_mouse_field[row.names(genes), ]
##select same rows in the first table
field <- field[row.names(genes), ]


###############lab
#select the genes and lab muce
lab <- hm %>%
  dplyr::filter(origin == "Lab", Position == "mLN") #selecting for mln to avoid
# duplicates
lab <- unique(lab)
gene_lab_mouse <- lab %>%
  dplyr::select(c(Mouse_ID, "IFNy", "CXCR3", "IL.6", "IL.13",# "IL.10",
                  "IL1RN","CASP1", "CXCL9", "IDO1", "IRGM1", "MPO",
                  "MUC2", "MUC5AC", "MYD88", "NCR1", "PRF1", "RETNLB", "SOCS1",
                  "TICAM1", "TNF"))

gene_lab_mouse <- unique(gene_lab_mouse)

genes_lab <- gene_lab_mouse[, -1]

#remove rows with only nas
genes_lab <- genes_lab[,colSums(is.na(genes_lab))<nrow(genes_lab)]

#remove colums with only nas
genes_lab <- genes_lab[rowSums(is.na(genes_lab)) != ncol(genes_lab), ]

genes_lab <- unique(genes_lab)

#select same rows in the first table
gene_lab_mouse <- gene_lab_mouse[row.names(genes_lab), ]

##select same rows in the first table
```

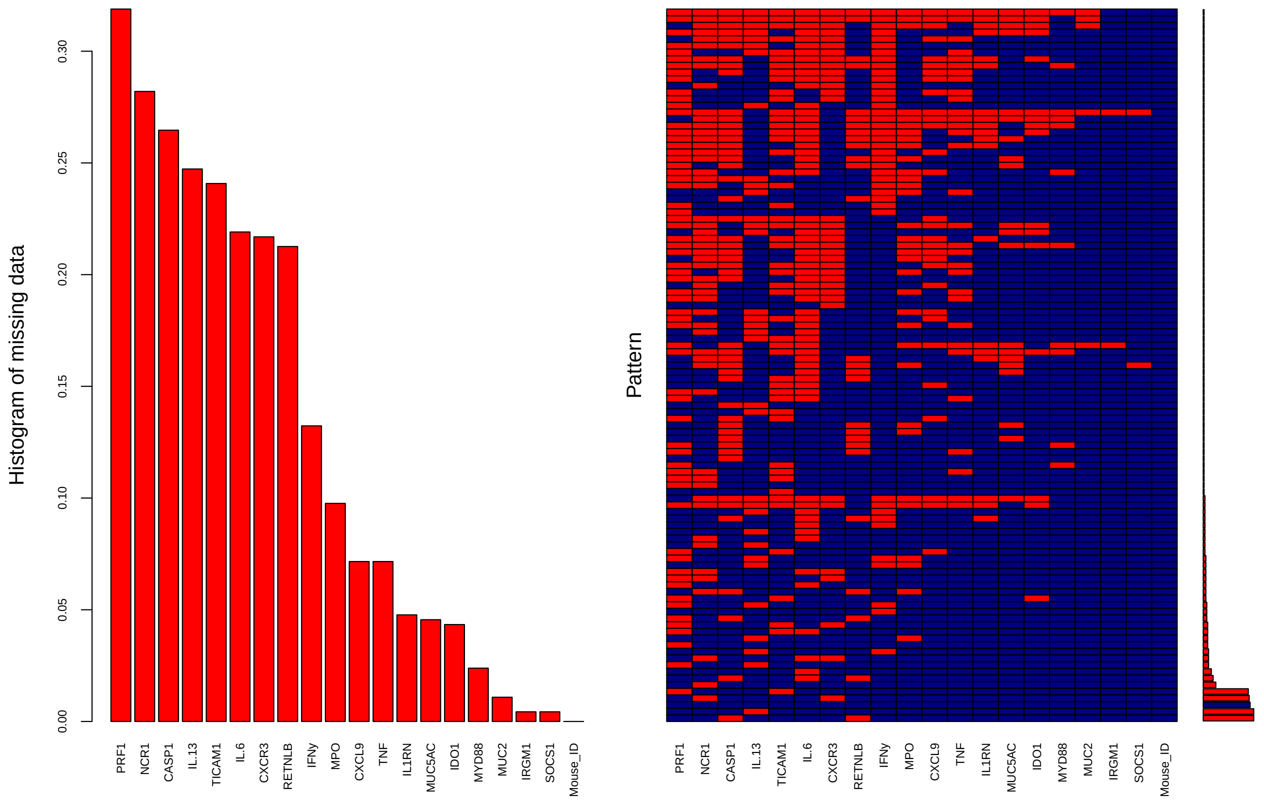
```
lab <- lab[row.names(genes_lab), ]

hm_genes <- rbind(gene_lab_mouse, genes_mouse_field)

hm_selection_g <- rbind(lab, field)

genes <- hm_genes %>%
  left_join(hm_selection_g %>%
              dplyr::select(c(Mouse_ID, origin)),
            by = "Mouse_ID")


genes <- genes %>%
  dplyr::select(-Mouse_ID)

genes$origin <- as.factor(genes$origin)

#dplyr::select(-Mouse_ID)
# looking at patterns of nas)
#pattern_na <-as.data.frame(md.pattern(field_genes))
sapply(hm_genes, function(x) sum(is.na(x)))
```

```
## Mouse_ID     IFNy    CXCR3     IL.6    IL.13    IL1RN    CASP1    CXCL9
##        0       61      100      101      114       22      122       33
##     IDO1    IRGM1      MPO     MUC2   MUC5AC    MYD88     NCR1     PRF1
##       20        2       45        5       21       11      130      147
##   RETNLB    SOCS1   TICAM1      TNF
##       98        2      111       33

## Mouse_ID     IFNy    CXCR3     IL.6    IL.13    IL.10    IL1RN    CASP1
##        0       62      110      111      124      230       31      131
##    CXCL9     IDO1    IRGM1      MPO     MUC2   MUC5AC    MYD88     NCR1
##       42       29       11       54       14       30       20      139
##     PRF1   RETNLB    SOCS1   TICAM1      TNF
##      158      108       11      121       42
```

```
# Discarding the origin
#genes <- genes %>% dplyr::select(-origin)


#had to remove as they were disturbing the imputation: Worms_presence, MC.Eimeria.FEC,  Heligmosomoides
#vis_miss(field)
# The frequency distribution of the missing cases per variable can be obtained
# as:
init <- mice(genes, maxit = 0)
#we want to impute only the specific variables
meth <- init$method
```

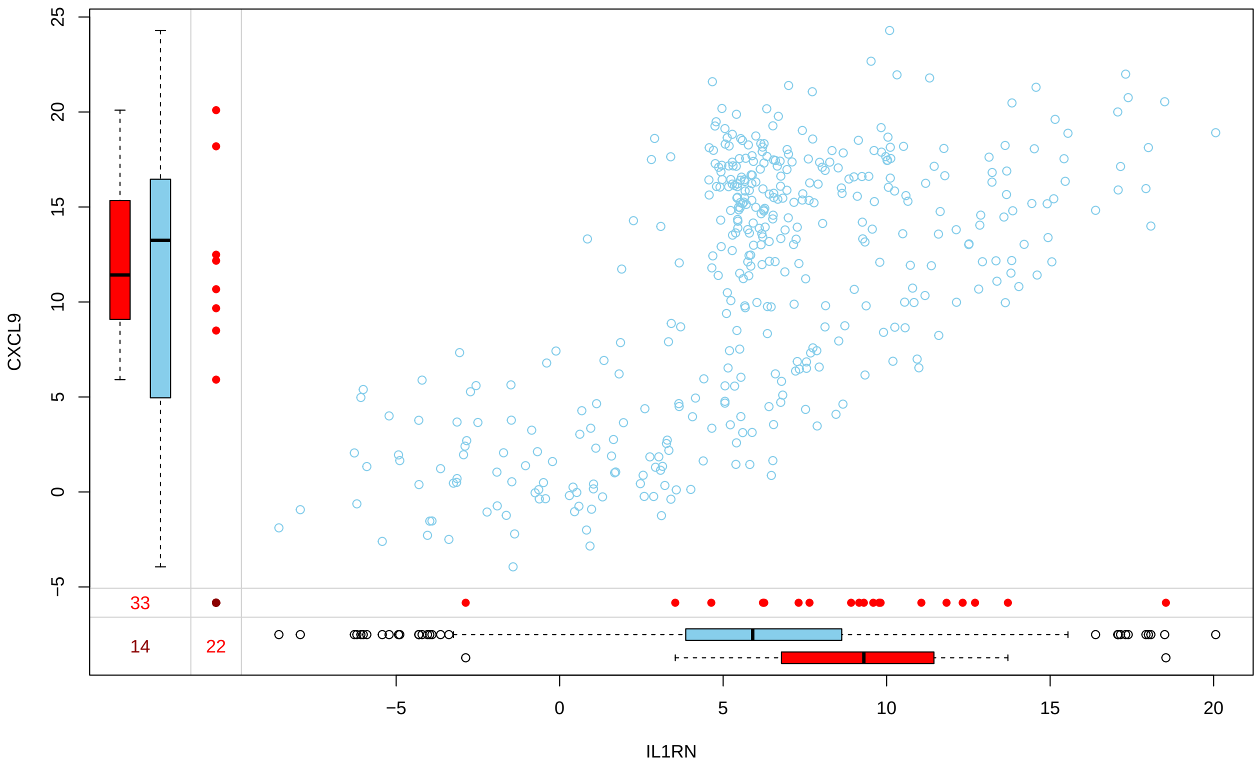
```
aggr_plot <- aggr(hm_genes, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE, labels=names(hm_genes)
```

```
## Warning in plot.aggr(res, ...): not enough vertical space to display frequencies
## (too many combinations)
```



```
##
##  Variables sorted by number of missings:
##  Variable       Count
##      PRF1 0.318872017
##      NCR1 0.281995662
##     CASP1 0.264642082
##     IL.13 0.247288503
##    TICAM1 0.240780911
##      IL.6 0.219088937
##     CXCR3 0.216919740
##    RETNLB 0.212581345
##      IFNy 0.132321041
##       MPO 0.097613883
##     CXCL9 0.071583514
##       TNF 0.071583514
##     IL1RN 0.047722343
##    MUC5AC 0.045553145
##      IDO1 0.043383948
##     MYD88 0.023861171
##      MUC2 0.010845987
##     IRGM1 0.004338395
##     SOCS1 0.004338395
##  Mouse_ID 0.000000000
```

```
marginplot(hm_genes[c(6,8)])
```



```r
# removing il 10
#genes <- genes %>%
 # dplyr::select(-IL.10)
# removed already at previous step (because of large missing numbers)
# m=5 refers to the number of imputed datasets. Five is the default value.
igf <- mice(genes, m = 5, seed = 500) # method = meth,
```

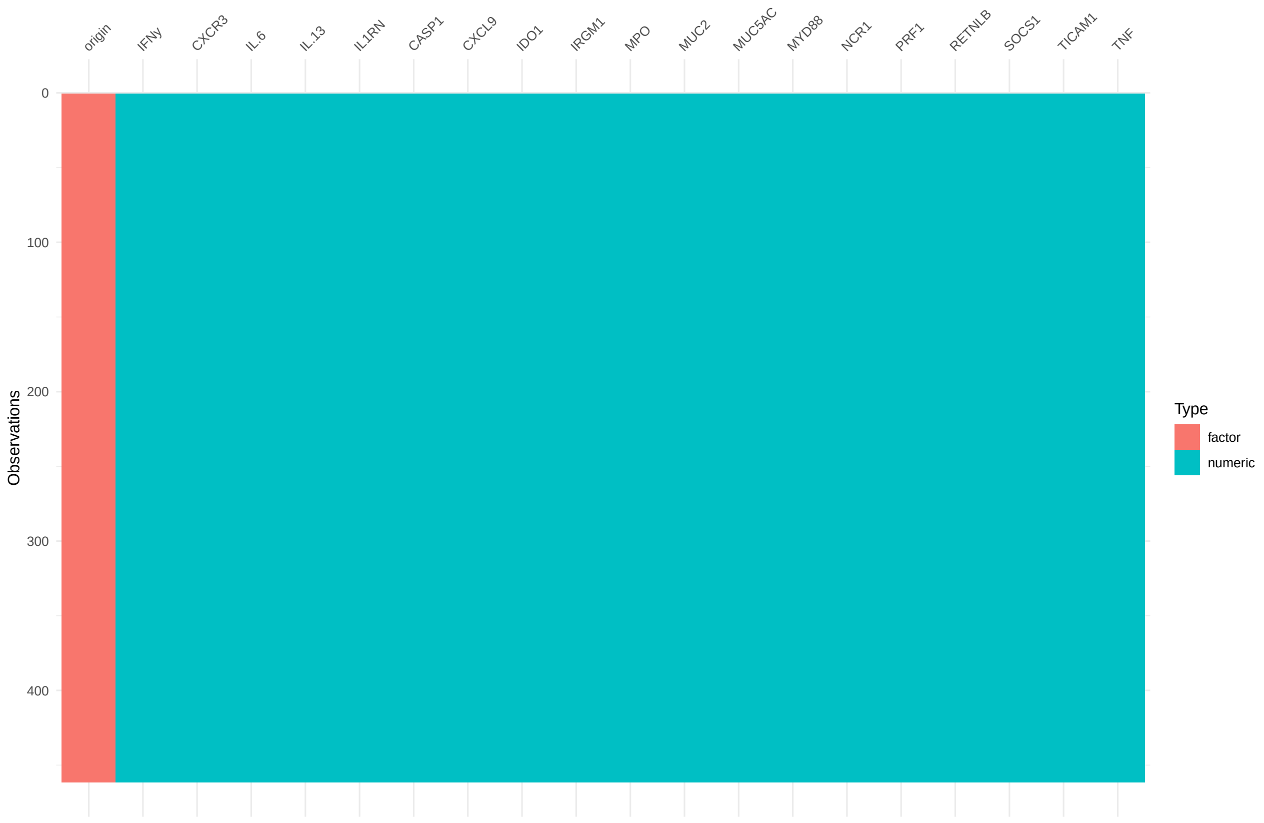
```
##
##  iter imp variable
##   1   1  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   1   2  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   1   3  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   1   4  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   1   5  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   2   1  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   2   2  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   2   3  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   2   4  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   2   5  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   3   1  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   3   2  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   3   3  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   3   4  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   3   5  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   4   1  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   4   2  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   4   3  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   4   4  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   4   5  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   5   1  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   5   2  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   5   3  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   5   4  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
##   5   5  IFNy  CXCR3  IL.6  IL.13  IL1RN  CASP1  CXCL9  IDO1  IRGM1  MPO  MUC2  MUC5AC  MYD88  NCR1
```

```
summary(igf)
```

```
## Class: mids
## Number of multiple imputations:  5
## Imputation methods:
##   IFNy  CXCR3   IL.6  IL.13  IL1RN  CASP1  CXCL9   IDO1  IRGM1    MPO   MUC2
##  "pmm"  "pmm"  "pmm"  "pmm"  "pmm"  "pmm"  "pmm"  "pmm"  "pmm"  "pmm"  "pmm"
## MUC5AC  MYD88   NCR1   PRF1 RETNLB  SOCS1 TICAM1    TNF origin
##  "pmm"  "pmm"  "pmm"  "pmm"  "pmm"  "pmm"  "pmm"  "pmm"     ""
## PredictorMatrix:
##       IFNy CXCR3 IL.6 IL.13 IL1RN CASP1 CXCL9 IDO1 IRGM1 MPO MUC2 MUC5AC MYD88
## IFNy     0     1    1     1     1     1     1    1     1   1    1      1     1
## CXCR3    1     0    1     1     1     1     1    1     1   1    1      1     1
## IL.6     1     1    0     1     1     1     1    1     1   1    1      1     1
## IL.13    1     1    1     0     1     1     1    1     1   1    1      1     1
## IL1RN    1     1    1     1     0     1     1    1     1   1    1      1     1
## CASP1    1     1    1     1     1     0     1    1     1   1    1      1     1
##       NCR1 PRF1 RETNLB SOCS1 TICAM1 TNF origin
## IFNy     1    1      1     1      1   1      1
## CXCR3    1    1      1     1      1   1      1
## IL.6     1    1      1     1      1   1      1
## IL.13    1    1      1     1      1   1      1
## IL1RN    1    1      1     1      1   1      1
## CASP1    1    1      1     1      1   1      1
```

```
# to check each column with imputed data
## igf$imp$IFNy
#Now we can get back the completed dataset using the complete()
complete_genes <- complete(igf, 1)

#sapply(complete_field, function(x) sum(is.na(x)))
#visualize missingness
vis_dat(complete_genes)
```

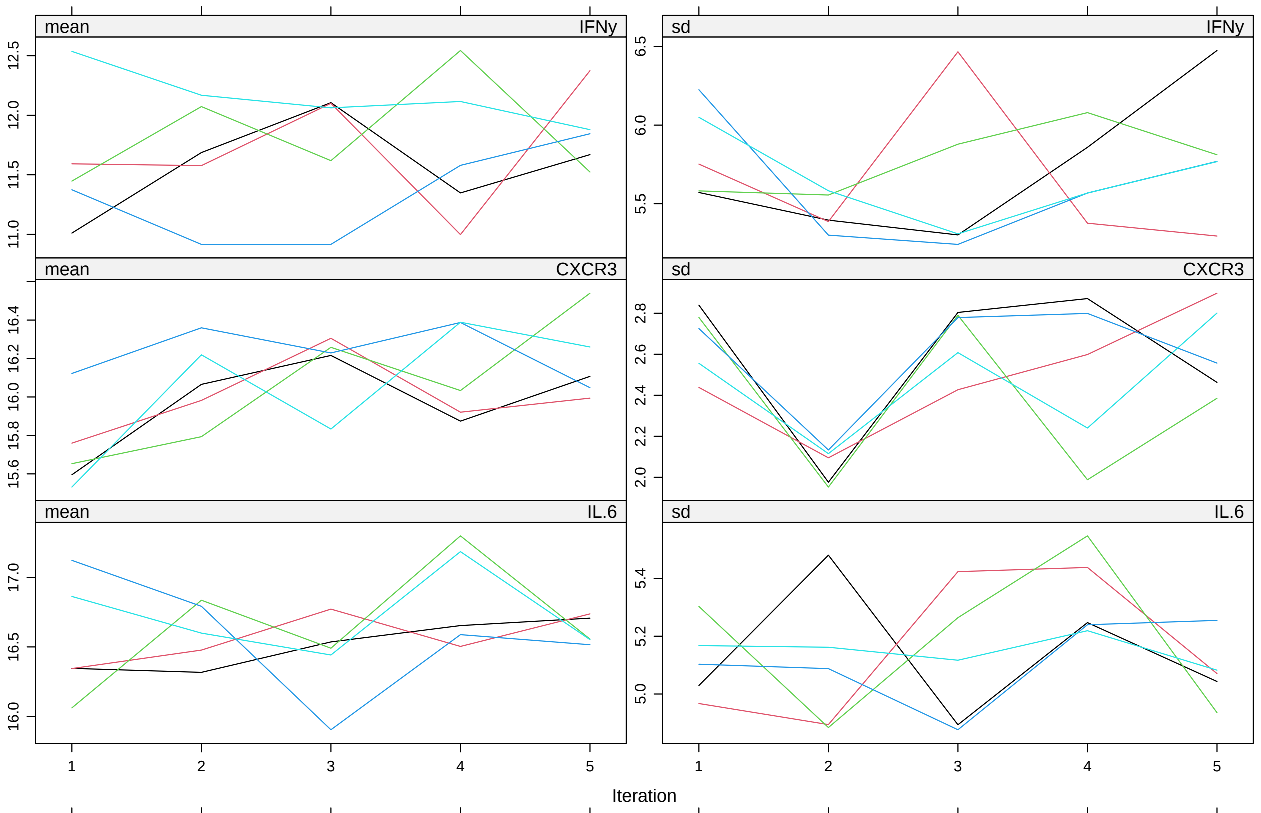
```
## Warning: `gather_()` was deprecated in tidyr 1.2.0.
## i Please use `gather()` instead.
## i The deprecated feature was likely used in the visdat package.
##   Please report the issue at <https://github.com/ropensci/visdat/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

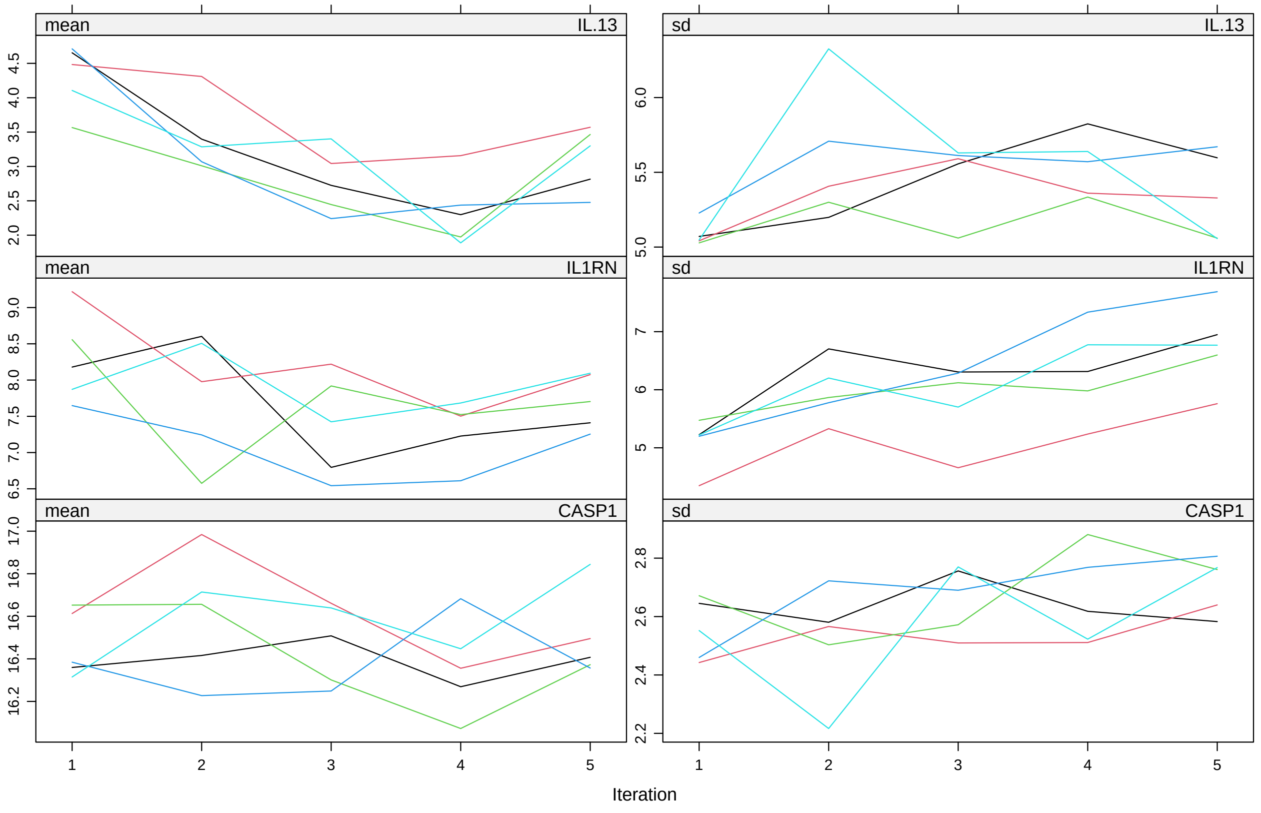


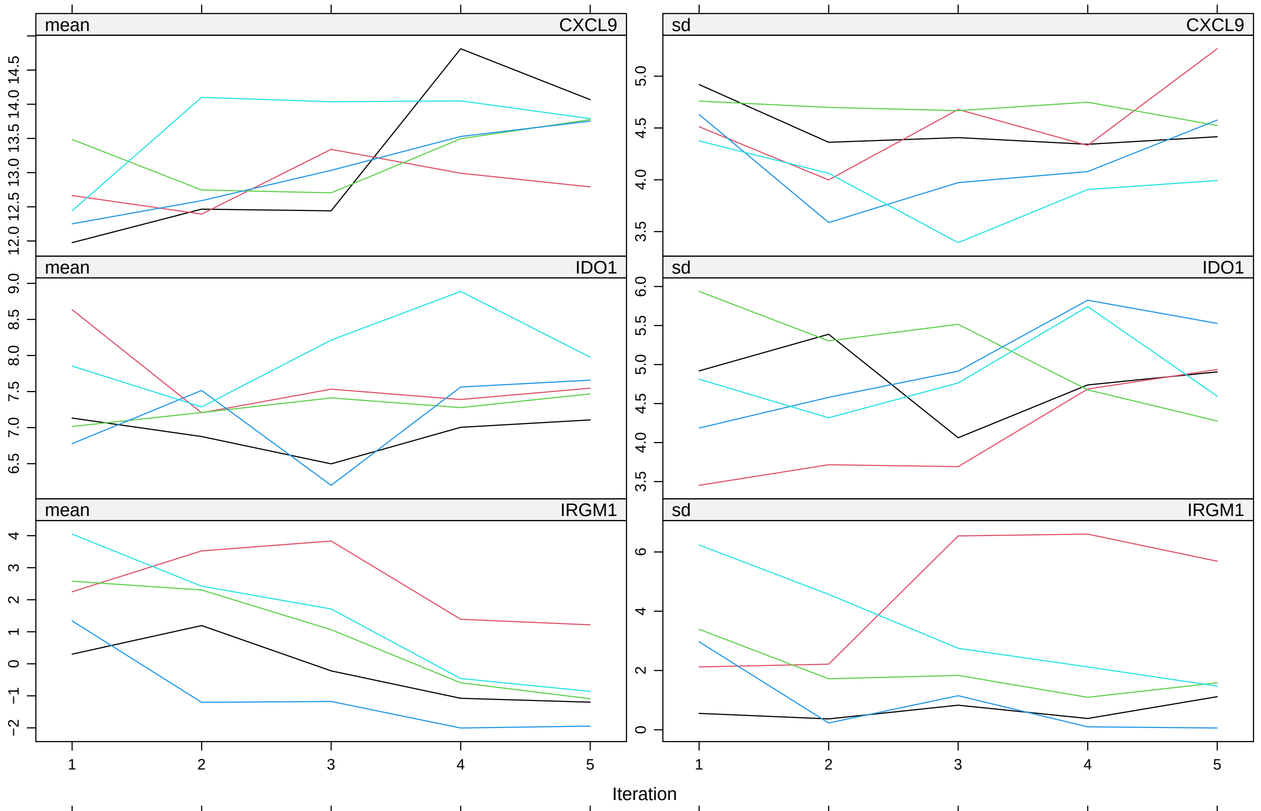
```
#remove the non imputed genes from our data set
hm_selection_g <- hm_selection_g %>%
  dplyr::select(-c("IFNy", "CXCR3", "IL.6", "IL.13", #"IL.10",
                "IL1RN","CASP1", "CXCL9", "IDO1", "IRGM1", "MPO",
                "MUC2", "MUC5AC", "MYD88", "NCR1", "PRF1", "RETNLB", "SOCS1",
                "TICAM1", "TNF", "origin"))
# add the new imputed genes to the data
hm_selection_g <- cbind(hm_selection_g, complete_genes)
```

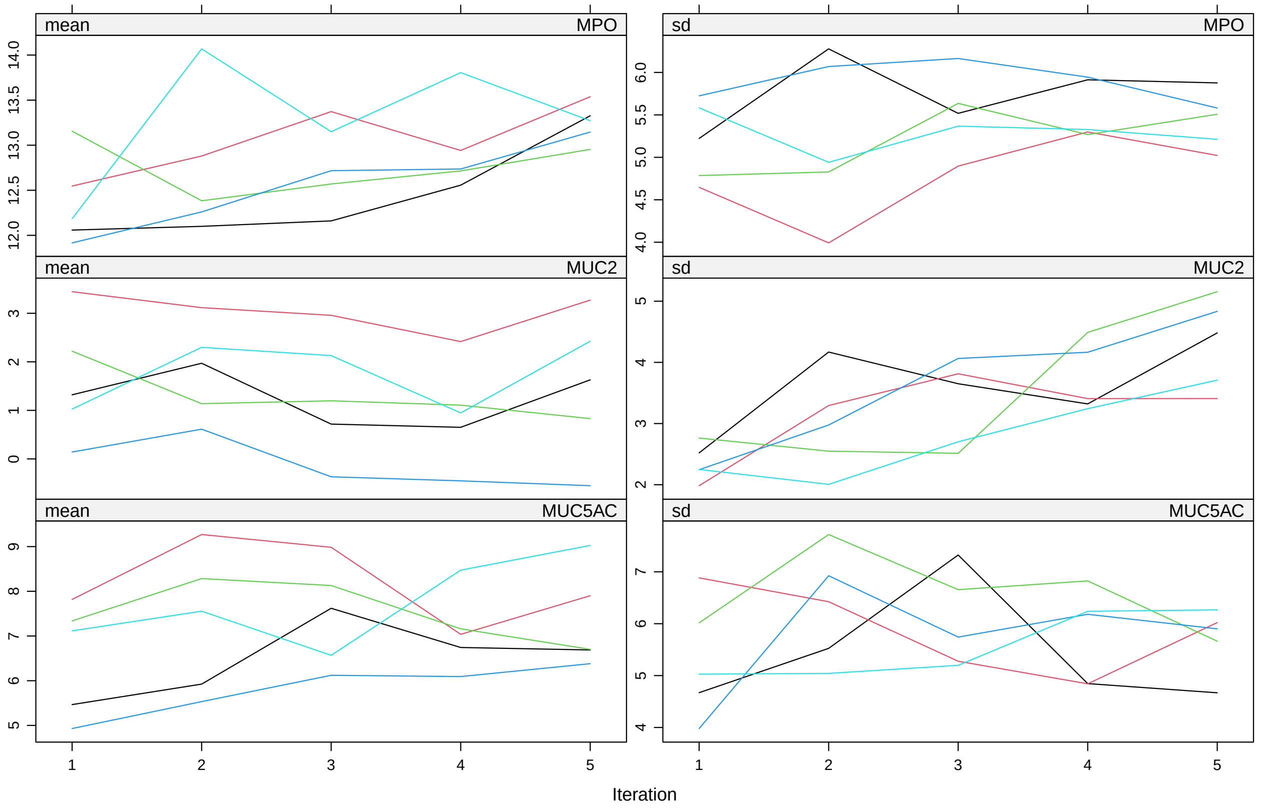
inspect the trace lines for convergence:

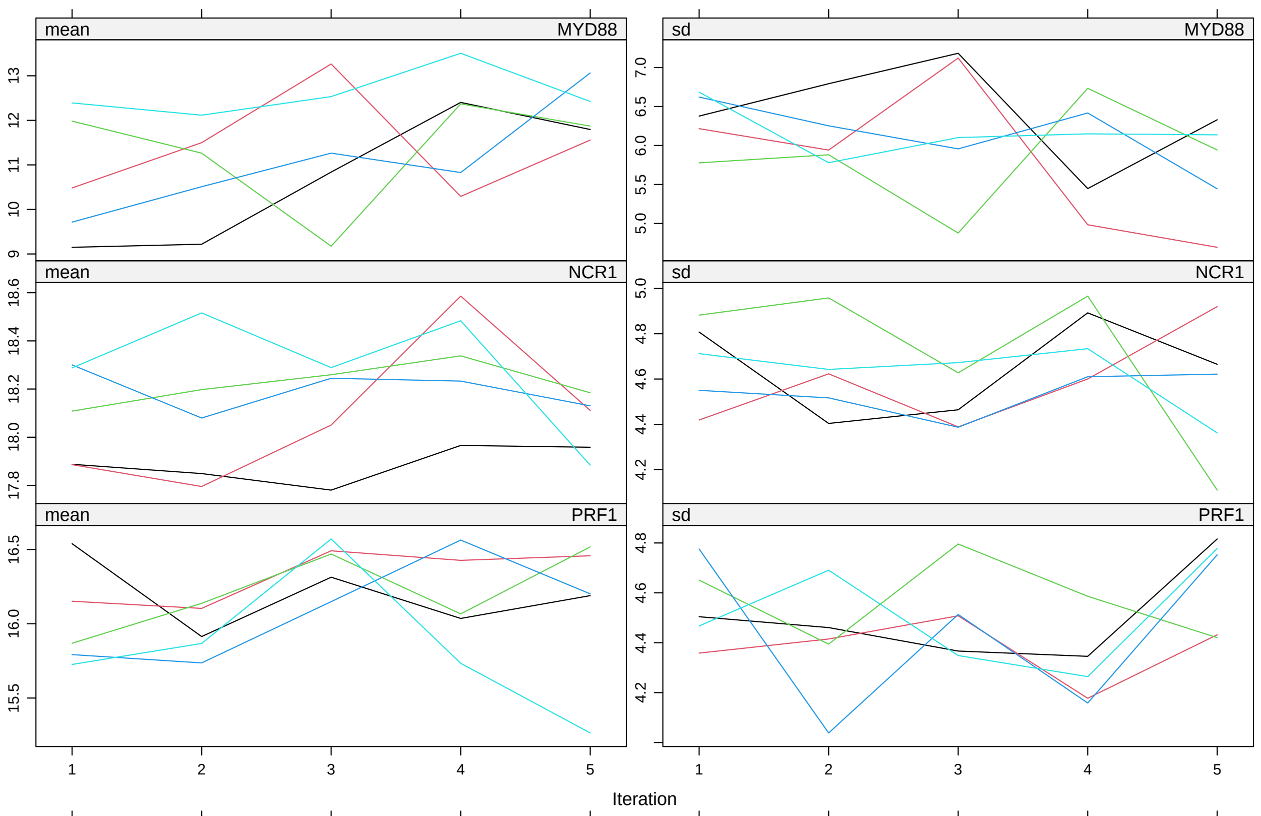
```
plot(igf)
```


mean
IFNy
sd
IFNy
mean
CXCR3
sd
CXCR3
mean
IL.6
sd
IL.6
Iteration

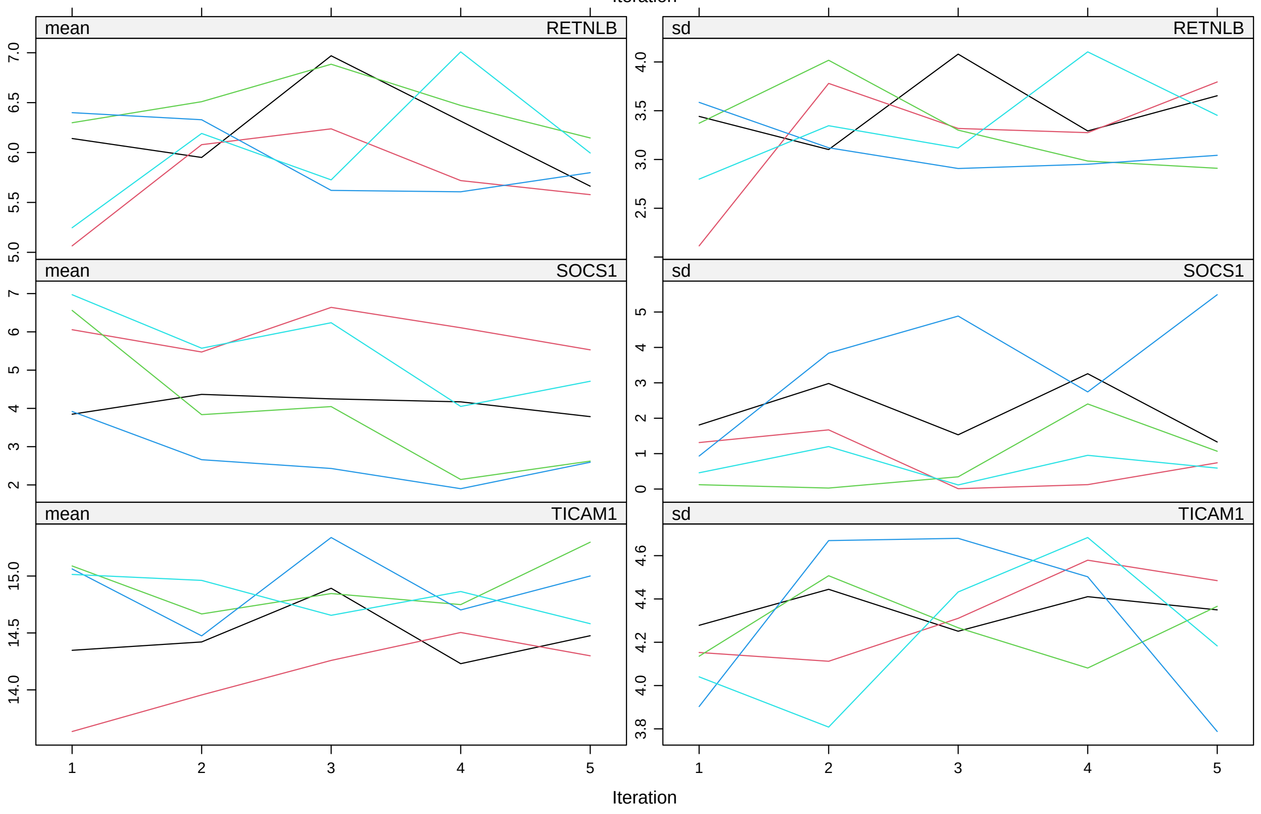

mean
IL.13
sd
IL.13
mean
IL1RN
sd
IL1RN
mean
CASP1
sd
CASP1
Iteration


mean CXCL9
sd CXCL9
mean IDO1
sd IDO1
mean IRGM1
sd IRGM1
Iteration

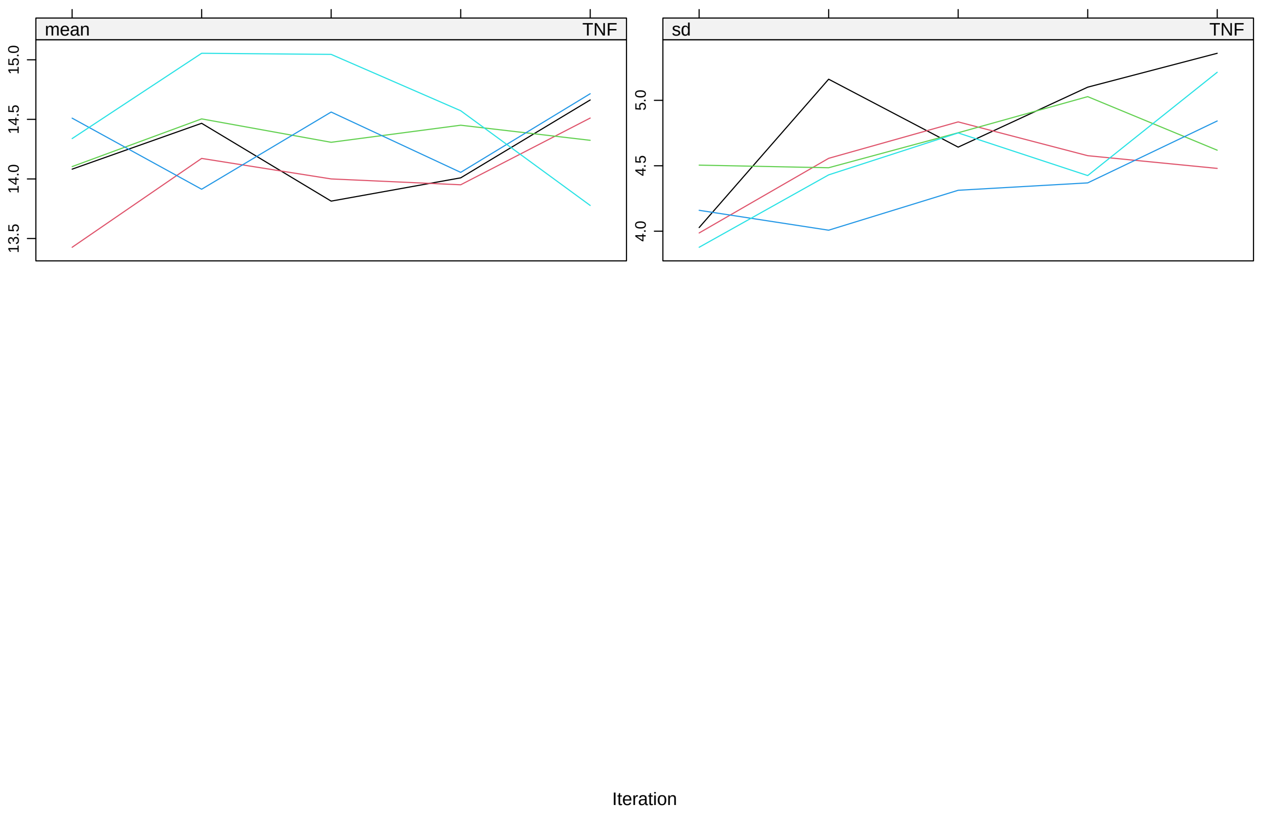

mean MPO
sd MPO
mean MUC2
sd MUC2
mean MUC5AC
sd MUC5AC
Iteration



mean
MYD88
sd
MYD88
mean
NCR1
sd
NCR1
mean
PRF1
sd
PRF1
Iteration

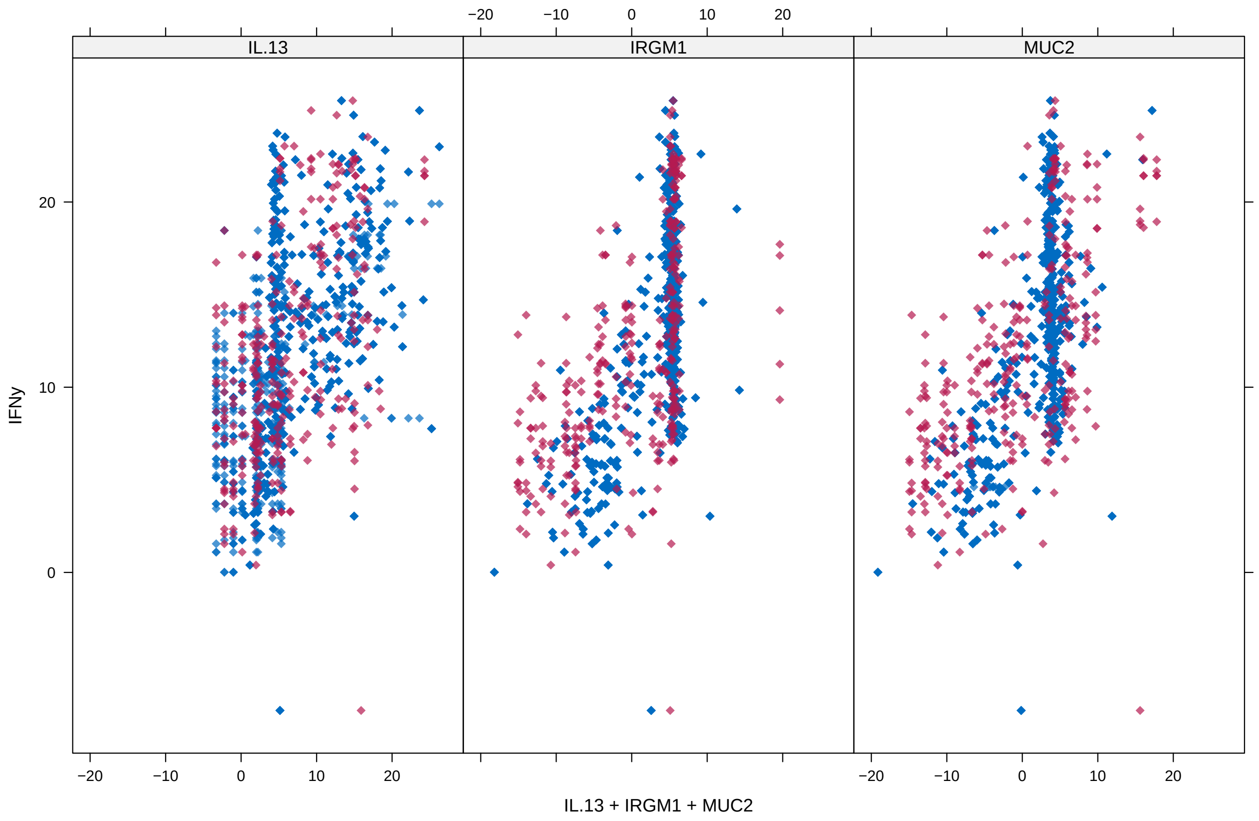


mean
RETNLB
sd
RETNLB
mean
SOCS1
sd
SOCS1
mean
TICAM1
sd
TICAM1
Iteration



Let's compare the distributions of original and imputed data using a some useful plots.First of all we can use a scatterplot and plot Ozone against all the other variables. Let's first plot the variables for which we have few missing values.

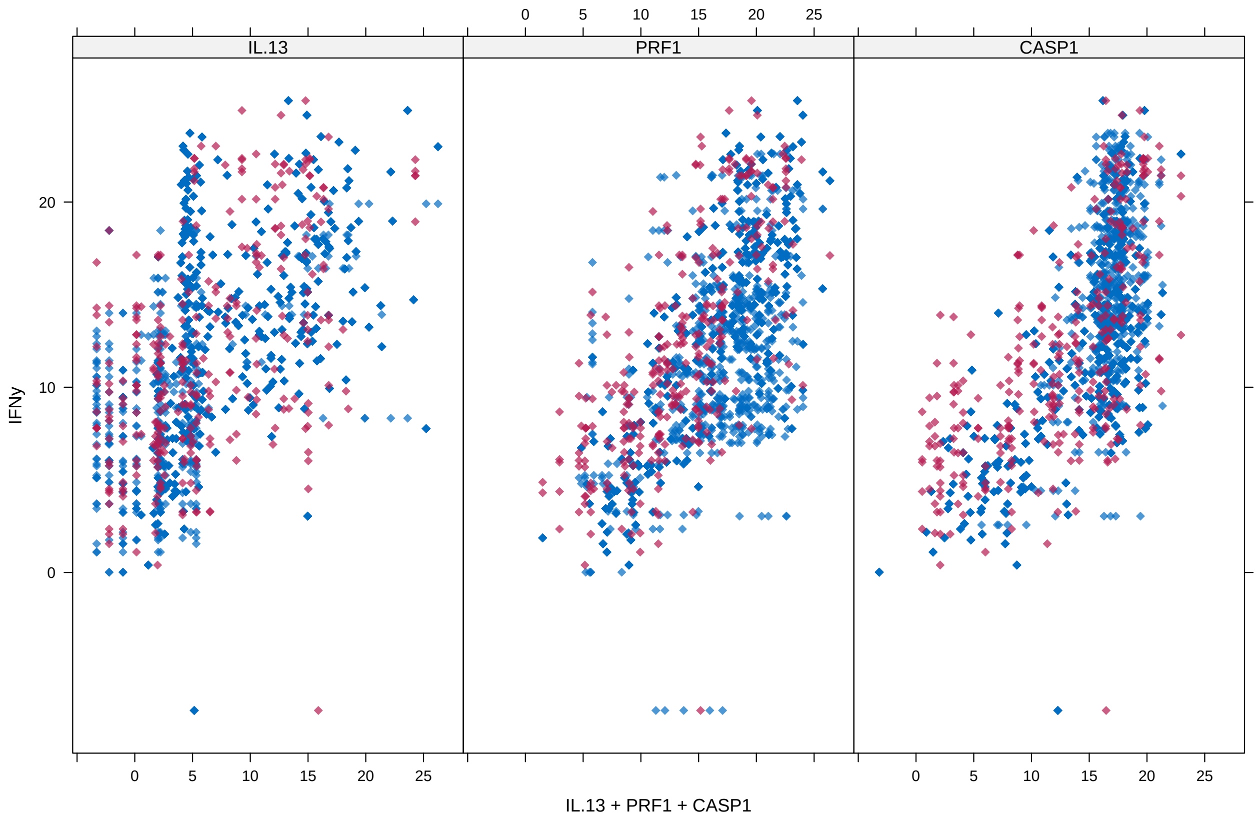
```
xyplot(igf, IFNy ~ IL.13 + IRGM1 + MUC2, pch=18,cex=1)
```



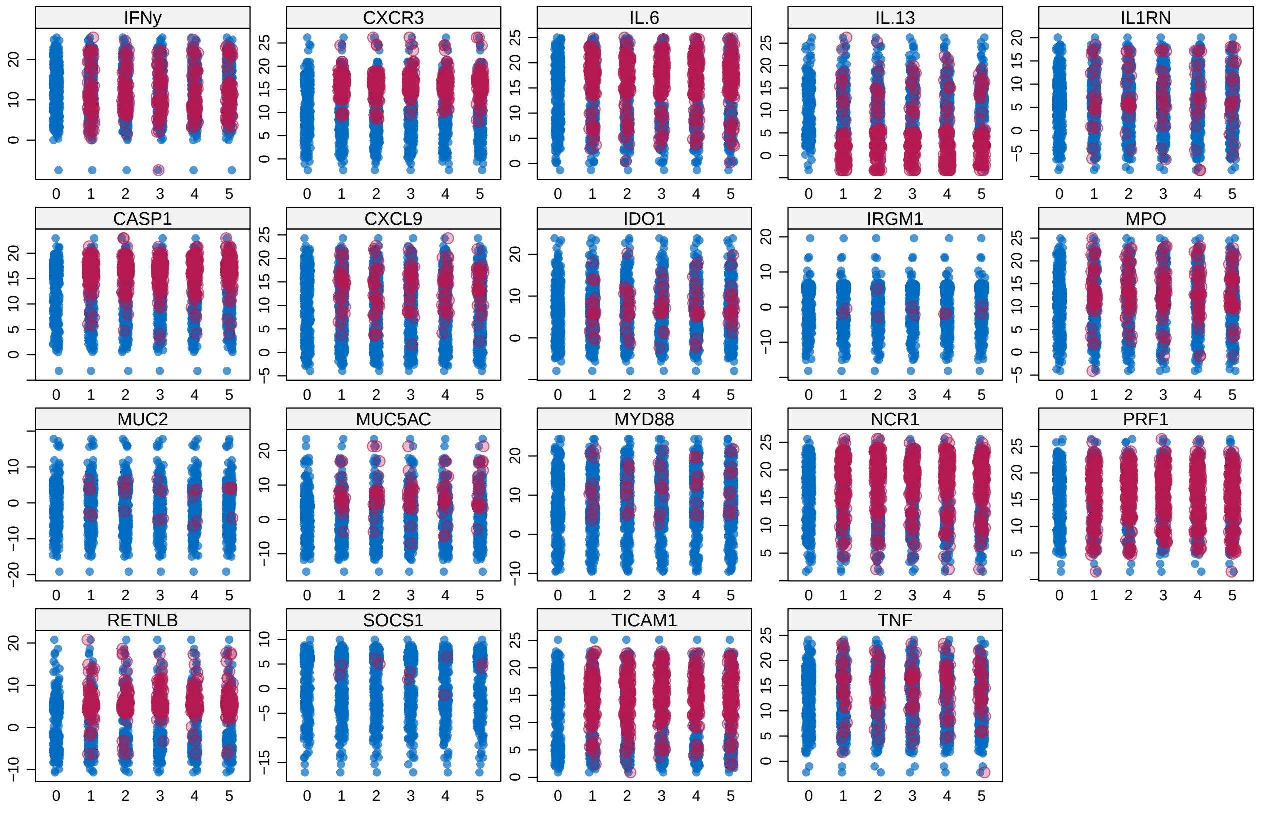
What we would like to see is that the shape of the magenta points (imputed) matches the shape of the blue ones (observed). The matching shape tells us that the imputed values are indeed "plausible values".

Now let's plot the variables with many missing data points.

```
xyplot(igf,IFNy ~ IL.13 + PRF1 + CASP1, pch=18,cex=1)
```

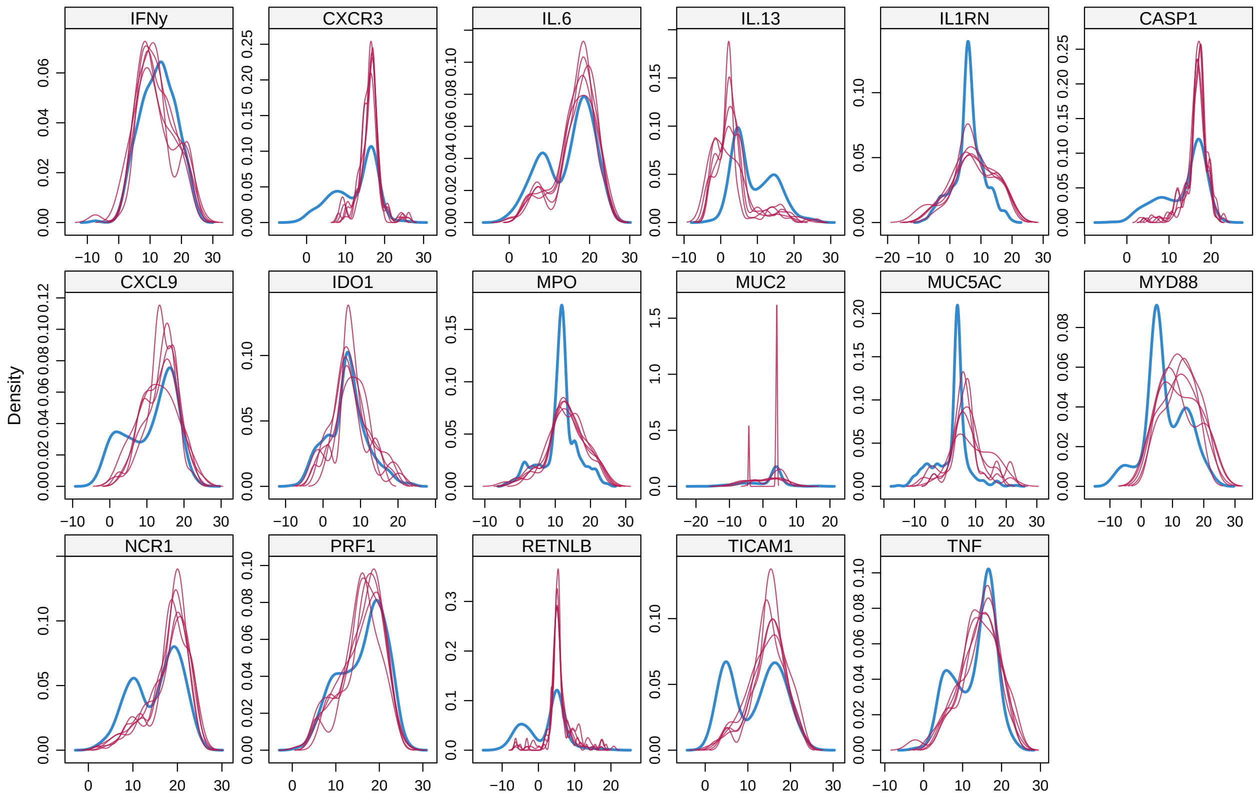


```
stripplot(igf, pch = c(20,21), cex = 1.2)
```



```r
#bwplot(igf)
```

```r
densityplot(igf)
```



The density of the imputed data for each imputed dataset is showed in magenta while the density of the observed data is showed in blue. Again, under our previous assumptions we expect the distributions to be similar.

Another useful visual take on the distributions can be obtained using the stripplot() function that shows the distributions of the variables as individual points

Facs

```r
###############lab
#select the facs and lab muce
lab <- hm %>%
  dplyr::filter(origin == "Lab", Position == "mLN") #selecting for mln to avoid

# duplicates
lab <- unique(lab)

facs_mouse <- lab %>%
  dplyr::select(c(Mouse_ID, all_of(Facs_lab))) #choosing the same with the wild

facs_mouse <- unique(facs_mouse)

facs_lab <- facs_mouse[, -1]
```

```r
#remove rows with only nas
facs_lab <- facs_lab[,colSums(is.na(facs_lab))<nrow(facs_lab)]
#remove colums with only nas
facs_lab <- facs_lab[rowSums(is.na(facs_lab)) != ncol(facs_lab), ]

#select same rows in the first table
facs_mouse_lab <- facs_mouse[row.names(facs_lab), ]





##########################Field
###########field
# somehow the field samples have the origin na,
# fix that
field <- hm %>%
  dplyr::filter(origin == "Field")

field <- unique(field)
facs_mouse <- field %>%
  dplyr::select(c(Mouse_ID, all_of(Facs_wild)))
facs_field <- facs_mouse[,-1]
#remove rows with only nas
facs_field <- facs_field[,colSums(is.na(facs_field))<nrow(facs_field)]
#remove colums with only nas
facs_field <- facs_field[rowSums(is.na(facs_field)) != ncol(facs_field), ]

#select same rows in the first table
facs_mouse_field <- facs_mouse[row.names(facs_field), ]

# full join the two tables
facs_data <- full_join(facs_mouse_lab, facs_mouse_field)
```

```
## Joining, by = c("Mouse_ID", "CD4", "Treg", "Treg17", "Th1", "Th17", "CD8",
## "Act_CD8", "IFNy_CD4", "IFNy_CD8", "IL17A_CD4")
```

```r
length(intersect(hm_selection_g$Mouse_ID, facs_data$Mouse_ID))
```

```
## [1] 99
```

```r
facs_data <- facs_data %>%
  left_join(hm)
```

```
## Joining, by = c("Mouse_ID", "CD4", "Treg", "Div_Treg", "Treg17", "Th1",
## "Div_Th1", "Th17", "Div_Th17", "CD8", "Act_CD8", "Div_Act_CD8", "IFNy_CD4",
## "IFNy_CD8", "Treg_prop", "IL17A_CD4")
```

We don't need to impute anything for the facs data as we have a complete data set

join the gene expression data with the facs data

```r
setdiff(facs_data$Mouse_ID, hm_selection_g$Mouse_ID)
```

```
## [1] "AA0772" "AA0790" "AA0799" "AA0807"
```

```
facs_data <- facs_data %>%
  dplyr::filter(Mouse_ID %in% setdiff(facs_data$Mouse_ID, hm_selection_g$Mouse_ID))


# We expect 477 mice in the new data frame
472 + 5
```

```
## [1] 477
```

```
#now combine the two data frames
hm_select <- rbind(hm_selection_g, facs_data)

hm_select <- unique(hm_select)
```

```
 ##save the imputed data
write.csv(hm_select, "output_data/2.imputed_MICE_data_set.csv", row.names = FALSE)
```