Benchmark for PLN

```
source("../functions/benchmark.R")
source("../functions/data_processing.R")
source("../functions/cooc_classes.R")
library(dplyr)
library(ggplot2)

set.seed(42)
```

Simulations: what does mu change?

The algo's performance with little aggregation was very poor: can we explain that by too sparse counts?

Difficult setting

Choose parameters:

```
# Run 6 times
n <- 14 # n species (median of above)
nrow <- 630 + 230 # median + nsamples to compensate lack of positive counts
mval <- log(.04) # mean value tested to get approx. mean .2 / col
nrep <- 20
crit <- "StARS"
mean <- rep(mval, n)

# # Uncomment to run Erdos-Renyi simulation
# # Erdos Renyi
# ncor <- 14
# g <- sample_gnm(n = n, m = ncor)

# Barabasi
m <- 2
gr <- sample_pa(n = n, m = m, directed=FALSE) # Network
ncor <- length(E(gr))

Omega <- generate_precmat(n = n, m = 1, u = .1, v = .3,
                          gr = gr) # Their correlation matrix

tri <- Omega[upper.tri(Omega, diag = FALSE)]
```

See what does the data look like:

```
g <- generate_obs(nrow = nrow, mean = mean, Omega = Omega)
head(g$obs)
```

Simulation

```
# # Uncomment to run simulations
# df.difficult <- repeat_simul(n = n, nrep = nrep, nrow = nrow,
#                               mean = mean, Omega= Omega, crit = crit)
#
# write.csv(df.difficult, "simul/difficult_setting_log0.04_25TP_Erdos.csv", row.names = FALSE)
```

Second simulation: easy setting

Choose parameters:

```
n <- 14 # n species (median of above)
nrow.easy <- 142 # median
mval <- log(.2) # mean value tested to get approx. mean .2 / col
nrep <- 20
crit <- "StARS"
mean.easy <- rep(mval, n)

# # Uncomment to run simulations with different parameters than the difficult setting
# # Erdos Renyi
# ncor <- 20
# gr <- sample_gnm(n = n, m = ncor)

# Barabasi
# m <- 1
# gr <- sample_pa(n = n, m = m, directed=FALSE) # Network
# ncor <- length(E(gr))

Omega <- generate_precmat(n = n, m = 1, u = .1, v = .3,
                          gr = gr) # Their correlation matrix

tri <- Omega[upper.tri(Omega, diag = FALSE)]
```

The data look like:

```
g <- generate_obs(nrow = nrow.easy, mean = mean.easy, Omega = Omega)
head(g$obs)
```

Simulation

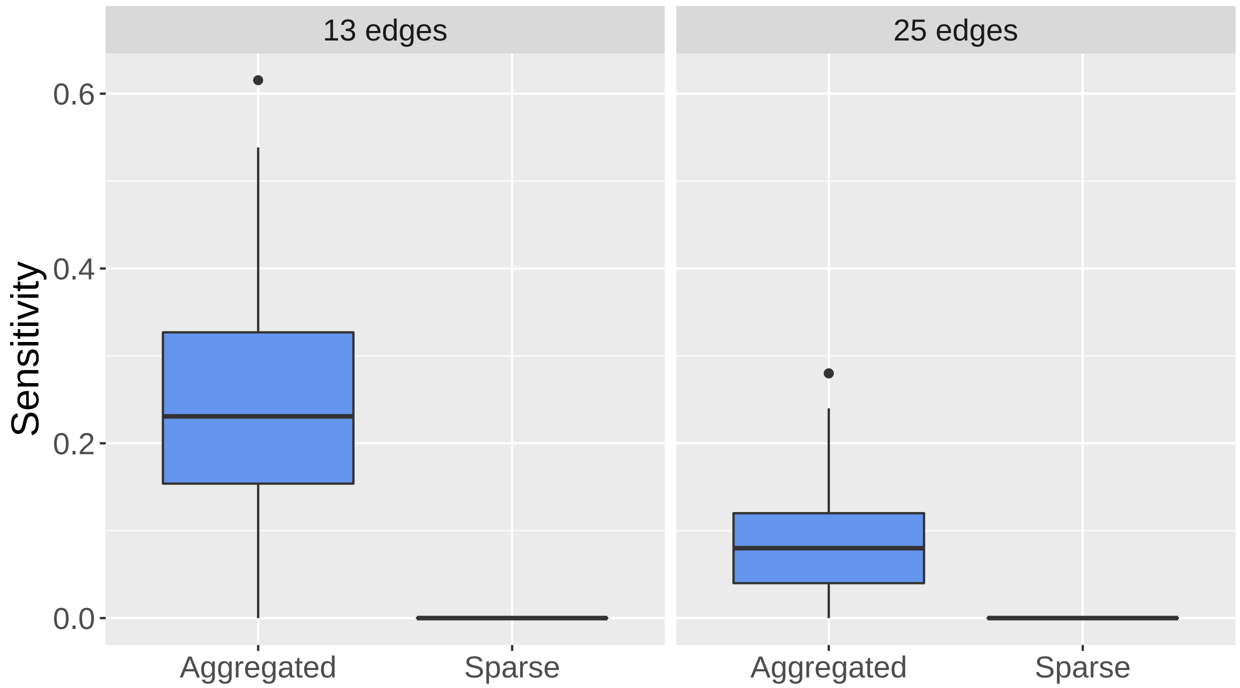
```
# # Uncomment to run simulations
# df.easy <- repeat_simul(n = n, nrep = nrep, mean = mean.easy,
#                         nrow = nrow.easy,
#                         Omega= Omega, crit = crit)
#
# write.csv(df.easy, "simul/easy_setting_log0.2_25TP_Erdos.csv", row.names = FALSE)
# df.easy
```

Graphes and testing

Read the data and plot:

```
easy_erdos13 <- read.csv("simul/easy_setting_log0.2_13TP_Erdos.csv")
easy_erdos25 <- read.csv("simul/easy_setting_log0.2_25TP_Erdos.csv")
easy_barabasi13 <- read.csv("simul/easy_setting_log0.2_13TP_Barabasi.csv")
easy_barabasi25 <- read.csv("simul/easy_setting_log0.2_25TP_Barabasi.csv")

difficult_erdos13 <- read.csv("simul/difficult_setting_log0.04_13TP_Erdos.csv")
difficult_erdos25 <- read.csv("simul/difficult_setting_log0.04_25TP_Erdos.csv")
difficult_barabasi13 <- read.csv("simul/difficult_setting_log0.04_13TP_Barabasi.csv")
difficult_barabasi25 <- read.csv("simul/difficult_setting_log0.04_25TP_Barabasi.csv")

df <- rbind(easy_erdos13, easy_barabasi13,
      difficult_erdos13, difficult_barabasi13,
      easy_erdos25, easy_barabasi25,
      difficult_erdos25, difficult_barabasi25)

Setting <- c(rep("Aggregated", 2*20),
             rep("Sparse", 2*20),
             rep("Aggregated", 2*20),
             rep("Sparse", 2*20))

Model <- rep(c(rep("Erdős-Rényi", 20), rep("Barabási-Albert", 20)),4)

edges <- c(rep("13 edges", 4*20), rep("25 edges", 4*20))

df <- cbind(df, Setting, Model, edges)

df <- df %>% mutate(sensitivity = TP/(TP+FN))
```

```
ggplot(df) + geom_boxplot(aes(y = sensitivity, x = Setting), fill = "cornflowerblue") +
                                facet_grid(~ edges) + theme(axis.title.x = element_blank(),
                                                            text = element_text(size = 18)) +
  ylab("Sensitivity")
```



Wilcoxon tests:

```
df13 <- df %>% filter(edges == "13 edges")
df13 <- df13 %>% dplyr::select(sensitivity, Setting)

df13.e <- df13 %>% filter(Setting == "Aggregated")
df13.d <- df13 %>% filter(Setting == "Sparse")
wilcox.test(df13.d$sensitivity, df13.e$sensitivity, alternative = "less")
```

```
## Warning in wilcox.test.default(df13.d$sensitivity, df13.e$sensitivity,
## alternative = "less"): cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  df13.d$sensitivity and df13.e$sensitivity
## W = 40, p-value = 1.184e-15
## alternative hypothesis: true location shift is less than 0
```

```
df25 <- df %>% filter(edges == "25 edges")
df25 <- df25 %>% dplyr::select(sensitivity, Setting)

df25.e <- df25 %>% filter(Setting == "Aggregated")
df25.d <- df25 %>% filter(Setting == "Sparse")
wilcox.test(df25.d$sensitivity, df25.e$sensitivity, alternative = "less")
```

```
## Warning in wilcox.test.default(df25.d$sensitivity, df25.e$sensitivity,
## alternative = "less"): cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  df25.d$sensitivity and df25.e$sensitivity
## W = 160, p-value = 1.648e-12
## alternative hypothesis: true location shift is less than 0
```