Supplement 3 - modgo demonstration Application to a bigger dataset

By Francisco Ojeda, George Koliopanos, and Andreas Ziegler

Table of contents

1 Specification of High Performance Computer

We run all of the analysis below through an HPC with four nodes. Each node contains 2 X AMD EPYC 7742 64-Core Processor (128 cores each node) and 2TB of RAM. We suggest NOT to run this quarto report in a local computer.

2 Data set

For the evaluation of *modgo* in bigger datasets, we selected Golub dataset (Golub et al. 1999). It contains 6 categorical variables and 7129 gene expressions variables (quantitative). We remove two of the categorical variables, specifically, the sample number sample number and tissue.mf. In the section below, we download and then remove the samples that contains at least one NA value.

```
# Download data set
golub <-  as.data.frame(read.csv(url("https://www.openintro.org/data/csv/golub.csv")))

# Set up binary and categorical variables
binary_variables <- c("BM.PB","Gender")
categorical_variables <- c("Source","cancer")

# Remove samples with any NA
golub_prep <- golub %>% dplyr::select(-Samples) %>% dplyr::select(-'tissue.mf')
    golub_prep <- na.omit(golub_prep)
# Replace string values with factors
for (i in c(binary_variables,categorical_variables)){
  golub_prep[,i] <- as.numeric(factor(golub_prep[,i]))-1
}
```

3 Modgo run using multiple cores

In the following part, *modgo* is run on the HPC using multiple cores with the help or rslurm library. We create 500 simulated data sets. According to the number of cores selected, each *modgo* run creates a corresponding number of simulated data sets (500/number of cores).

```
# number of cores
n <- 250


# create data frame that contains the parameters for slurm_modgo
# function
data_example <- data.frame(x = 1:n,cores = rep(x = n,times = n))

path <- paste0("/cluster/storage/data/MODGO/modgo_original",n)
dir.create(path)
setwd(path)

submit <- TRUE
# Use multiple cores for running modgo using rslurm
sjob <- slurm_apply(slurm_modgo, params = data_example ,
                    jobname = 'slurm_modgo_original',
                    nodes = n, cpus_per_node = 1,
                    submit = submit)
```

After *modgo* has finished running on all cores, we need to collect all results and merge them.

```
# Merge the simulated data sets from the multiples runs
simulated_data_set <- mapply(c,(
  lapply(0:(n-1),
   function(k)
    readRDS(paste0(path,"/_rslurm_slurm_modgo_original/results_",k,".RDS"))[[1]]$Simulated

sim_data_set_path <- paste0(path,"/golub_500_sim_datasets.RDS")
saveRDS(simulated_data_set,sim_data_set_path)

# Calculate mean correlation matrix from the multiple runs
mean_correlation_sets <- Reduce("+",(
  lapply(0:(n-1),
  function(k) as.matrix(
  readRDS(
  paste0(path,"/_rslurm_slurm_modgo_original/results_",k,".RDS"))[[1]]$Correlations["Mean"

mean_corr_path <-  paste0(path,"/golub_mean_corrs.RDS")
saveRDS(mean_correlation_sets,mean_corr_path)
```

4 Result plots

We use two figures to illustrate *modgo*'s performance. Figure 1 displays distribution plots of the original, and one simulated data set for three variables:
1. AFFX.BioC.3_st = gene expression (quantitative variable)
2. Source = hospital name where the data were collected (categorical variable)
3. BM.PB = from where they collected cells (BM= bone marrow, PB = peripheral blood)
Figure 2 displays correlation plots for the original data set, the mean correlation of 500 simulated data sets, and the difference between the previous two.

Figure 1: Distribution plots of the original data set and one simulated data set. Panel A depicts the distribution of one gene's expression variable. Panel B provides the center's information. Panel C illustrates whether the samples were taken from bone marrow (BM) or peripheral blood (PB)

```
knitr::include_graphics("C:/Users/geokolcc/OneDrive - Hochgebirgsklinik Davos/Desktop/modg
```

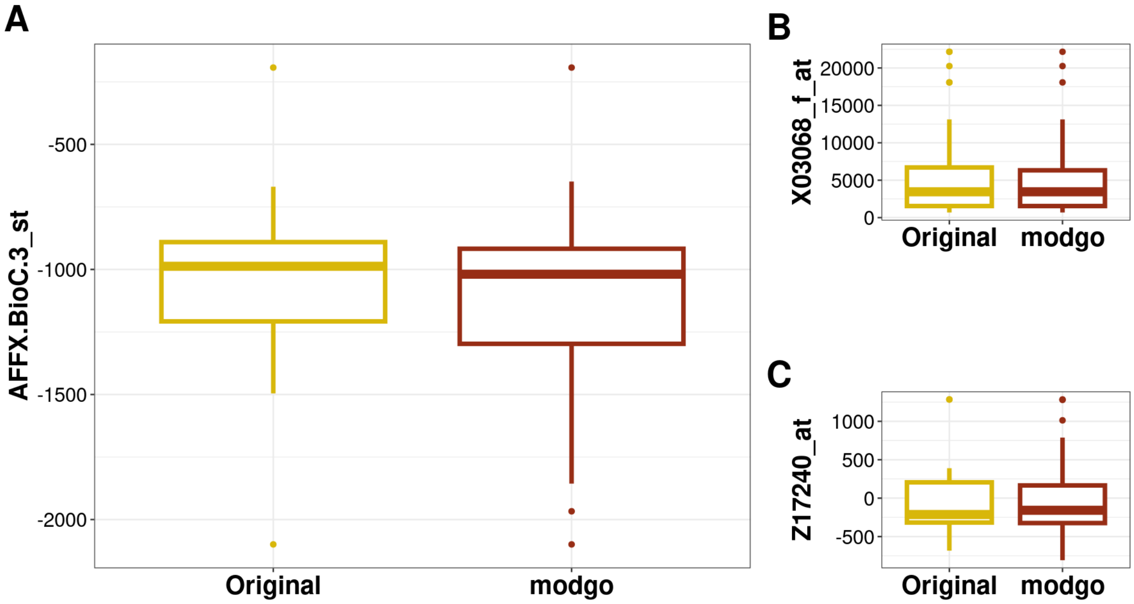
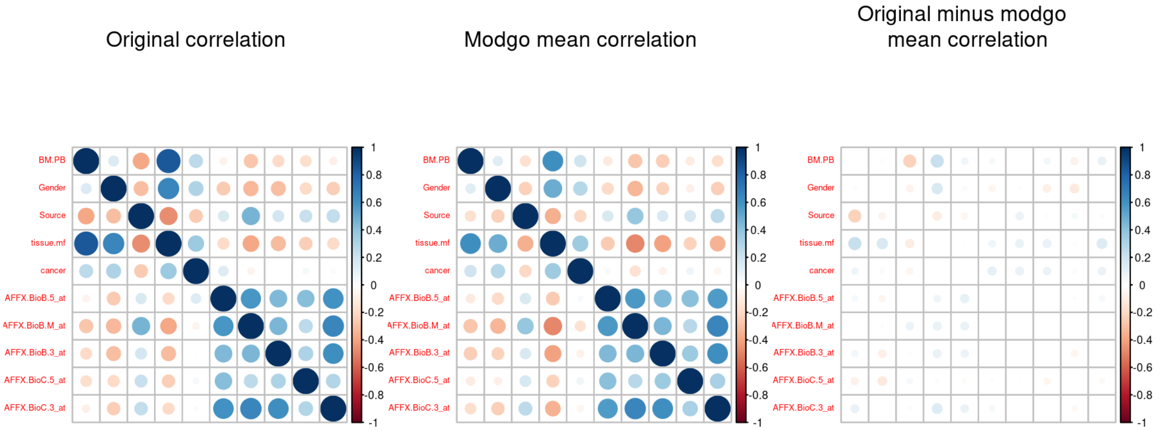


Figure 2: Correlation matrix plots for the Golub data. Displayed are the correlation matrices as heat maps for the original data (left), the mean correlation of modgo simulations (middle), and the difference between the original correlation and the mean correlation matrix of modgo simulations (right).

```
knitr::include_graphics("C:/Users/geokolcc/OneDrive - Hochgebirgsklinik Davos/Desktop/modg
```



Original correlation

Modgo mean correlation

Original minus modgo mean correlation

Reference

Golub et al. 1999. "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring." *Science*, 531–37. https://www.science.org/doi/10.1126/science.286.5439.531?url_ver=Z39.88-2003&rfr_id=ori:rid:crossref.org&rfr_dat=cr_pub%20%200pubmed.