Structured factorization for single-cell gene expression data: simulations

Antonio Canale, Luisa Galtarossa, Davide Risso, Lorenzo Schiavon, Giovanni Toto

Last modified: June 07, 2023

This vignette contains the code developed and used for computing the aggregated evaluation metrics of the simulation study.

```
library(dplyr)
```

Introduction

To generate the matrix of gene expression, we consider the zero mean data generating process defined as

$$y_{ij} = \lfloor \exp(z_{ij}) \rfloor, \quad z_{ij} = \sum_h C_{hij} + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim N(0, \sigma^2).$$

To illustrate the validity and generality of `COSIN`, we consider three scenarios in which η and λ have different sparsity structures. In particular,

1. in scenario 1 we consider two rank-one contributions which induce two groups of genes:

$$\begin{gathered}\eta_{\cdot 1}, \eta_{\cdot 2} \sim N_n(0, I_n), \quad \lambda_{\cdot 1} \sim N_p(0, I_p), \\ \lambda_{j2} = 0, \quad j > p/2, \quad \lambda_{m2} = 1, \quad m \leq p/2;\end{gathered}$$

2. in scenario 2 we consider two rank-one contributions which induce two groups of cells:

$$\begin{gathered}\lambda_{\cdot 1}, \lambda_{\cdot 2} \sim N_p(0, I_p), \quad \eta_{\cdot 1} \sim N_n(0, I_n), \\ \eta_{i2} \sim N(0, 0.05^2), \quad i > n/2, \quad \eta_{l2} = 1, \quad l \leq n/2;\end{gathered}$$

3. in scenario 3 we consider three rank-one contributions which induce both two groups of cell and two groups of genes:

$$\begin{gathered}\eta_{\cdot 1}, \eta_{\cdot 3} \sim N_n(0, I_n), \quad \lambda_{\cdot 1}, \lambda_{\cdot 2} \sim N_p(0, I_p), \\ \eta_{i2} \sim N(0, 0.05^2), \quad i > n/2, \quad \eta_{l2} = 1, \quad l \leq n/2, \\ \lambda_{j3} \sim N_p(0, 0.05^2), \quad j > p/2, \quad \lambda_{m3} = 1, \quad m \leq p/2.\end{gathered}$$

In the next sections we show how to load matrices, compute the aggregated evaluation metrics and explore the results. We report here only the steps related to the third scenario since it is the only one reported in the paper. The steps for the other two scenarios are the same.

Load tables

We load the tables containing the evaluating metrics related to COSIN:

```
# # assigning names to columns (scenario 1)
# colnames_s1 <- c("n", "p", "sigma", "s", "r", "RMSE_1", "RMSE_2", "RMSE_tot",
#                  "F1_1", "F1_2", "F1_tot", "MAD_Y", "MAE_Y", "RMSE_el",
#                  "kstar", "sd_1", "sd_2", "sd_tot", "meta")
# # assigning names to columns (scenario 2)
# colnames_s2 <- c("n", "p", "sigma", "s", "r", "RMSE_1", "RMSE_2", "RMSE_tot",
#                  "MAD_Y", "MAE_Y", "RMSE_el", "kstar", "sd_1", "sd_2", "sd_tot", "meta")
metrics_s3 <- rbind(readRDS("simulations/metrics/table_metrics_meta_NA_p100_s3.RDS"),
                    readRDS("simulations/metrics/table_metrics_nometa_NA_p100_s3.RDS"),
                    readRDS("simulations/metrics/table_metrics_meta_NA_p1000_s3.RDS"),
                    readRDS("simulations/metrics/table_metrics_nometa_NA_p1000_s3.RDS"))
metrics_s3_woNA <- rbind(readRDS("simulations/metrics/table_metrics_meta_woNA_p100_s3.RDS"),
                         readRDS("simulations/metrics/table_metrics_nometa_woNA_p100_s3.RDS"),
                         readRDS("simulations/metrics/table_metrics_meta_woNA_p1000_s3.RDS"),
                         readRDS("simulations/metrics/table_metrics_nometa_woNA_p1000_s3.RDS"))
# assigning names to columns
colnames_s3 <- c("n", "p", "sigma", "s", "r", "RMSE_1", "RMSE_2",
                 "RMSE_3", "RMSE_tot", "MAD_Y", "MAE_Y", "RMSE_el",
                 "kstar", "sd_1", "sd_2", "sd_3", "sd_tot", "meta")
colnames(metrics_s3) <- colnames(metrics_s3_woNA) <- colnames_s3
```

We load the tables containing the evaluating metrics related to GLM-PCA:

```
# # assigning names to columns (scenario 1)
# colnames_glmpca_s1 <- c("n", "p", "sigma", "r", "alpha_a", "MAE_a",
#                         "MAE", "RMSE_1", "RMSE_2", "RMSE_tot",
#                         "rmse_etaLambda", "meta")
# # assigning names to columns (scenario 2)
# colnames(metrics_glmpca_s2) <- c("n", "p", "sigma", "r", "alpha_a", "MAE_a",
#                                  "MAE", "RMSE_1", "RMSE_2", "RMSE_tot",
#                                  "rmse_etaLambda", "meta")
metrics_glmpca_s3 <- readRDS("simulations/glmpca/table_metrics_nometa_glmpca_s3.RDS")
metrics_glmpca_s3_meta <- readRDS("simulations/glmpca/table_metrics_meta_glmpca_s3.RDS")
colnames_glmpca_s3 <- c("n", "p", "sigma", "r", "alpha_a", "MAE_a",
                        "MAE", "RMSE_1", "RMSE_2", "RMSE_3", "RMSE_tot",
                        "rmse_etaLambda", "meta")
colnames(metrics_glmpca_s3) <- colnames(metrics_glmpca_s3_meta) <- colnames_glmpca_s3
```

Aggregate metrics

First, we aggregate the evaluation metrics related to COSIN applied to data sets with and without NAs:

```
# with NAs
metrics_s3 <- as.data.frame(metrics_s3)
metrics_s3$s <- NULL
metrics_s3$r <- NULL
# aggregating metrics
metrics_s3 <- metrics_s3 %>% group_by(n, p, meta, sigma) %>%
  summarise_all(.funs = c("median" = median, "IQR" = IQR))
# ordering columns
metrics_s3 <- metrics_s3[, c(1:4, rbind(5:16, 5:16+12))]
# changing colnames
colnames(metrics_s3) <- colnames(metrics_s3) %>%
  stringr::str_replace_all("_median", " median") %>%
  stringr::str_replace("_IQR", " IQR")

# without NAs
metrics_s3_woNA <- as.data.frame(metrics_s3_woNA)
metrics_s3_woNA$s <- NULL
metrics_s3_woNA$r <- NULL
# filtering columns
metrics_s3_woNA <- metrics_s3_woNA[, c(1:3,4:6,16)]
# aggregating metrics
metrics_s3_woNA <- metrics_s3_woNA %>% group_by(n, p, meta, sigma) %>%
  summarise_all(.funs = c("median" = median, "IQR" = IQR))
```

Then, we aggregate the evaluation metrics related to `GLM-PCA` with meta-covariate applied to data sets without NAs:

```
metrics_glmpca_s3_meta <- as.data.frame(metrics_glmpca_s3_meta)
metrics_glmpca_s3_meta$r <- NULL
# aggregating metrics
metrics_glmpca_s3_meta <- metrics_glmpca_s3_meta %>% group_by(n, p, meta, sigma) %>%
  summarise_all(.funs = c("median" = median, "IQR" = IQR))
```

Finally, we aggregate the evaluation metrics related to `GLM-PCA` without meta-covariate applied to both kinds of data set:

```
metrics_glmpca_s3 <- as.data.frame(metrics_glmpca_s3)
metrics_glmpca_s3$r <- NULL
# aggregating metrics
metrics_glmpca_s3 <- metrics_glmpca_s3 %>% group_by(n, p, meta, sigma) %>%
  summarise_all(.funs = c("median" = median, "IQR" = IQR))
```

Contribution RMSE computed on data sets without NAs

Median of contribution RMSE in 50 replicates, with varying (n, p, σ). Interquartile range is also reported.

```
# COSIN
round(metrics_s3_woNA, 2)
```

```
## # A tibble: 12 x 10
## # Groups:   n, p, meta [6]
##        n     p  meta sigma RMSE_1_median RMSE_~1 RMSE_~2 RMSE_~3 RMSE_~4 RMSE_~5
##    <dbl> <dbl> <dbl> <dbl>         <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1    50   100     0   0.1          0.58    0.56    0.37    0.29    0.28    0.18
##  2    50   100     0   1            0.6     0.58    0.43    0.26    0.27    0.21
##  3    50   100     1   0.1          0.63    0.58    0.39    0.25    0.28    0.26
##  4    50   100     1   1            0.62    0.57    0.39    0.3     0.26    0.14
##  5   200   100     0   0.1          0.6     0.51    0.24    0.33    0.33    0.11
##  6   200   100     0   1            0.46    0.45    0.26    0.37    0.34    0.03
##  7   200   100     1   0.1          0.54    0.45    0.25    0.35    0.33    0.19
##  8   200   100     1   1            0.48    0.46    0.26    0.41    0.32    0.05
##  9   200  1000     0   0.1          0.86    0.47    0.63    0.15    0.1     0.06
## 10   200  1000     0   1            0.62    0.53    0.18    0.19    0.09    0.34
## 11   200  1000     1   0.1          0.87    0.47    0.63    0.12    0.08    0.06
## 12   200  1000     1   1            0.62    0.55    0.53    0.12    0.07    0.04
## # ... with abbreviated variable names 1: RMSE_2_median, 2: RMSE_3_median,
## #   3: RMSE_1_IQR, 4: RMSE_2_IQR, 5: RMSE_3_IQR
```

```
# GLM-PCA with meta-covariates
round(metrics_glmpca_s3_meta[,c(1:4,8:10,16:18)], 2)
```

```
## # A tibble: 6 x 10
## # Groups:   n, p, meta [3]
##       n     p  meta sigma RMSE_1_median RMSE_2~1 RMSE_~2 RMSE_~3 RMSE_~4 RMSE_~5
##   <dbl> <dbl> <dbl> <dbl>         <dbl>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1    50   100     1   0.1          1.85     1.76    0.84    2.13    1.25    0.18
## 2    50   100     1   1            0.63     0.79    0.76    0.17    0.16    0.13
## 3   200   100     1   0.1          1.56     1.44    0.74    0.58    0.78    0.03
## 4   200   100     1   1            0.48     0.66    0.74    0.12    0.11    0.07
## 5   200  1000     1   0.1          0.99     0.86    0.72    0.23    0.42    0.05
## 6   200  1000     1   1            0.81     0.68    0.73    0.61    0.12    0.05
## # ... with abbreviated variable names 1: RMSE_2_median, 2: RMSE_3_median,
## #   3: RMSE_1_IQR, 4: RMSE_2_IQR, 5: RMSE_3_IQR
```

```
# GLM-PCA without meta-covariates
round(metrics_glmpca_s3[,c(1:4,8:10,16:18)], 2)
```

```
## # A tibble: 6 x 10
## # Groups:   n, p, meta [3]
##       n     p  meta sigma RMSE_1_median RMSE_2~1 RMSE_~2 RMSE_~3 RMSE_~4 RMSE_~5
##   <dbl> <dbl> <dbl> <dbl>         <dbl>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1    50   100     0   0.1          2.67     2.64    0.87    1.82    1.3     0.15
## 2    50   100     0   1            1.15     1.22    0.77    0.44    0.18    0.21
## 3   200   100     0   0.1          1.65     1.85    0.75    0.43    0.41    0.06
## 4   200   100     0   1            0.85     1.09    0.72    0.43    0.17    0.09
## 5   200  1000     0   0.1          2.02     2.26    0.76    1.29    0.64    0.06
## 6   200  1000     0   1            0.6      1.07    0.73    0.31    0.1     0.11
## # ... with abbreviated variable names 1: RMSE_2_median, 2: RMSE_3_median,
## #   3: RMSE_1_IQR, 4: RMSE_2_IQR, 5: RMSE_3_IQR
```

Out-of-sample MAE computed on data sets with NAs

Median of out-of-sample MAE in 50 replicates, with varying (n, p, σ). Interquartile range is also reported.

```
# COSIN
round(metrics_s3[,c(1:4,15:16)], 4)
```

```
## # A tibble: 12 x 6
## # Groups:   n, p, meta [6]
##        n     p  meta sigma ‘MAE_Y median‘ ‘MAE_Y IQR‘
##    <dbl> <dbl> <dbl> <dbl>          <dbl>       <dbl>
##  1    50   100     0   0.1          0.746       0.524
##  2    50   100     0   1            4.61        2.83
##  3    50   100     1   0.1          0.754       0.508
##  4    50   100     1   1            4.56        2.84
##  5   200   100     0   0.1          0.505       0.275
##  6   200   100     0   1            4.55        2.62
##  7   200   100     1   0.1          0.507       0.278
##  8   200   100     1   1            4.54        2.60
##  9   200  1000     0   0.1          0.420       0.112
## 10   200  1000     0   1            4.79        1.49
## 11   200  1000     1   0.1          0.417       0.120
## 12   200  1000     1   1            4.79        1.50
```

```
# GLM-PCA without meta-covariates
round(metrics_glmpca_s3[,c(1:4,6,14)], 4)
```

```
## # A tibble: 6 x 6
## # Groups:   n, p, meta [3]
##       n     p  meta sigma MAE_a_median MAE_a_IQR
##   <dbl> <dbl> <dbl> <dbl>        <dbl>     <dbl>
## 1    50   100     0   0.1          1.47      1.43
## 2    50   100     0   1            6.14      2.86
## 3   200   100     0   0.1          1.35      1.13
## 4   200   100     0   1            7.33      4.19
## 5   200  1000     0   0.1          1.43      0.778
## 6   200  1000     0   1            6.93      2.46
```

Boxplot of the out-of-sample MAE computed on data sets with NAs

Boxplot of the out-of-sample MAE of the competing models under different values of (n, p) with $\sigma^2 = 1$:

```
metrics <- rbind(readRDS("simulations/metrics/table_metrics_meta_NA_p100_s3.RDS"),
                 readRDS("simulations/metrics/table_metrics_nometa_NA_p100_s3.RDS"),
                 readRDS("simulations/metrics/table_metrics_meta_NA_p1000_s3.RDS"),
                 readRDS("simulations/metrics/table_metrics_nometa_NA_p1000_s3.RDS"))
colnames(metrics) <- c("n", "p", "sigma", "s", "r", "RMSE_1", "RMSE_2",
                       "RMSE_3", "RMSE_tot", "MAD_Y", "MAE_Y", "RMSE_el",
                       "kstar", "sd_1", "sd_2", "sd_3", "sd_tot", "meta")
metrics <- as.data.frame(metrics)
metrics$n_p_sigma_s <- paste(metrics$n, metrics$p, metrics$sigma, metrics$s, sep="_")
metrics <- metrics[, c("meta", "n_p_sigma_s", "r", "MAE_Y")]

metrics_glmpca <- readRDS("simulations/metrics/table_metrics_nometa_glmpca_s3.RDS")
colnames(metrics_glmpca) <- c("n", "p", "sigma", "r", "alpha_a", "MAE_a",
                              "MAE", "RMSE_1", "RMSE_2", "RMSE_3", "RMSE_tot",
                              "rmse_etaLambda", "meta")
metrics_glmpca <- as.data.frame(metrics_glmpca)
metrics_glmpca$n_p_sigma_s <- paste(metrics_glmpca$n, metrics_glmpca$p, metrics_glmpca$sigma, 3, sep="_
metrics_glmpca <- metrics_glmpca[, c("meta", "n_p_sigma_s", "r", "MAE_a")]
```

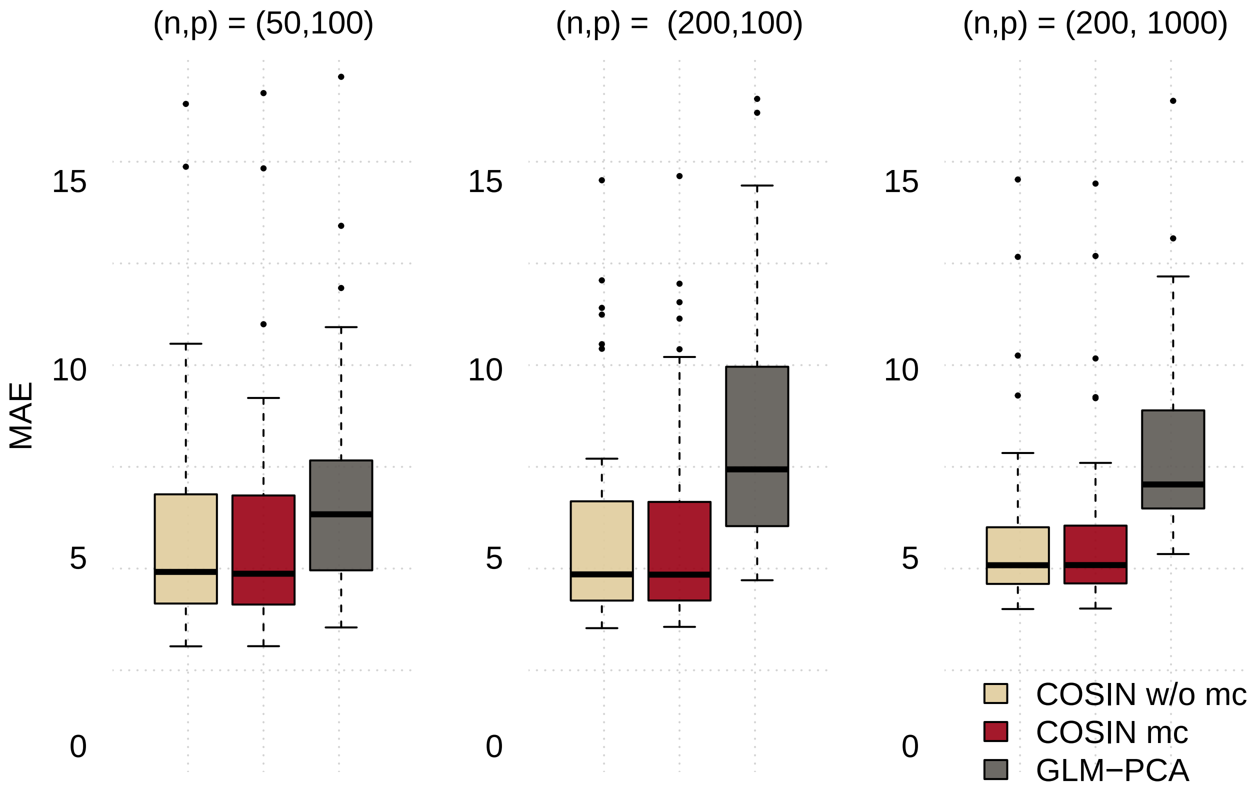
```r
scenario_list <- c("50_100_1_3", "200_100_1_3", "200_1000_1_3")

datatoplot_list <- list()
for(scenario in scenario_list) {
  datatoplot_list[[scenario]] <- cbind(metrics[metrics$meta==0 & metrics$n_p_sigma_s==scenario,"MAE_Y"]
                                       metrics[metrics$meta==1 & metrics$n_p_sigma_s==scenario,"MAE_Y"]
                                       metrics_glmpca[metrics_glmpca$n_p_sigma_s==scenario,"MAE_a"])
}

scenario_titles <- list("50_100_1_3" = "(n,p) = (50,100)",
                        "200_100_1_3" = "(n,p) =  (200,100)",
                        "200_1000_1_3" = "(n,p) = (200, 1000)")

unipdred <- scales::alpha("#9B0014", 0.9)
mybeige <- scales::alpha("#e1cd9d", 0.9)
mygrey <- scales::alpha("#5e5a55", 0.9)

par(mfrow=c(1,3), mar=c(3,4.5,3,0), xpd = FALSE)
for(scenario in scenario_list) {
  # 50x3 data set in which each column contains MAE of the 50 simulations computed by a model
  datatoplot <- as.data.frame(datatoplot_list[[scenario]])
  # plot settings
  if(scenario == "50_100_1_3"){
    plot(1, 1, pch = "", xlim = c(0.2,3.8), ylim = c(0, 17.5),
         axes = FALSE, xlab = "", ylab="MAE", cex.lab=1.5)
  }else{
    plot(1, 1, pch = "", xlim = c(0.2,3.8), ylim = c(0, 17.5),
         axes = FALSE, xlab = "", ylab="")
  }
  par(xpd = FALSE)
  grid(nx = 4, ny = 7)
  # boxplot
  par(xpd=T)
  boxplot(datatoplot,  col = c( mybeige, unipdred, mygrey),
          add = TRUE, axes = FALSE, pch = 20)
  #axis(1, at = c(1, 2, 3), labels = c("COSIN","COSIN2", "GLMPCA"), cex.axis = 1.5, tick = FALSE)
  axis(2, tick = FALSE, las = 1, cex.axis=1.5)
  #abline(h = 0, lwd = 2)
  title(main = scenario_titles[[scenario]], cex.main=1.5, font.main = 1)
  if(scenario =="200_1000_1_3"){
     par(xpd=T)
     legend("bottomright", legend = c("COSIN w/o mc","COSIN mc", "GLM-PCA"),
            fill = c(mybeige, unipdred, mygrey), cex=1.5, pt.cex = 2,
            bg = "white", horiz =F, inset = -0.05, bty="n")
  }
}
```


(n,p) = (50,100)
(n,p) = (200,100)
(n,p) = (200, 1000)
MAE
0
5
10
15
COSIN w/o mc
COSIN mc
GLM−PCA