# Supplementary Material

# Sparse Multivariate Factor Analysis Regression Models and Its Applications to Integrative Genomics Analysis

**Yan Zhou**

Merck & Co., PA 19454, USA

**email:* yan.zhou1@merck.com

**and**

**Pei Wang**

Icahn School of Medicine at Mount Sinai, New York, NY 10026, USA

**email:* pei.wang@mssm.edu

**and**

**Xianlong Wang**

Fred Hutchinson Cancer Research Center, Seattle, WA 98109, USA

**email:* xianlong.wang@gmail.com

**and**

**Ji Zhu, Peter X.-K. Song**

University of Michigan, MI 48109, USA

**email:* jizhu@umich.edu, pxsong@umich.edu

## 1. Some Empirical Results on Computational Complexity

Fitting an mFARM model involves two separate operations. One is related to a factor analysis of residuals, which is implemented by the EM algorithm; and the other is the operation of blockwise coordinate descent algorithm to search for sparse group lasso solution in the estimation of the association map matrix $\mathbf{\Theta}$. In our computation, given the number of latent factors $K$ and a pair of tuning parameters $(\lambda_1, \lambda_2)$, when the association parameter matrix $\mathbf{\Theta}$ is fixed, the computational complexity is $O(NQK)$ per iteration in the estimation of the factor loadings $B$ and uniqueness $\mathbf{\Psi} = \sigma^2\mathbf{I}$. When covariance matrix $\mathbf{\Sigma}$ is given, the computational cost of solving the association matrix $\mathbf{\Theta}$ by the sparse group lasso via the popular blockwise coordinate descent algorithm is $O(NPQ)$. Here $N$ is the sample size, $P$ is the number of markers, $Q$ is the number of genes, and $K$ is the number of latent factors.

To demonstrate the actual run-time in model fitting, here we present a simulation experiment focusing on computation time. Using the Simulation II setup outlined in the paper, we report the computation time in various scenarios in the following Table 1 in terms of average running time in seconds over 50 simulations to solve smFARM under the selected tuning parameters $(\lambda_1, \lambda_2)$. All calculations were carried out on a computer with an Intel Xeon 2.30 GHz processor.

[Table 1 about here.]

With no surprise the computational cost increases along the increase in the number of latent factors, $K$. This is because the more complicated the factor model is the heavier computational burden the EM algorithm encounters to estimate loading coefficients. In practice, instead of trying a wide range of $K$ values, one may narrow down such range by identifying the top $K$ eigenvalues of sample covariance matrix of $Y$. At this moment this strategy is learned from our empirical experience, which needs further theoretical investigation.

We also provide our computing code available online at the following webpage: `http://www.umich.edu/~songlab/software.html`.

## 2. Demonstration of Robustness via Simulation Experiment

To test the robustness of our proposed methodology for data that do not follow the mFARM model, we considered a simulation experiment in this section by following steps (suggested by a reviewer) of data generation under the setup of Simulation II given in the paper. The summary results from this simulation study over 50 rounds of simulations are listed in Table 2. These steps of data generation are given as follows.

a. Create a random sparse matrix with correlation for the coefficient matrix $\mathbf{\Theta}$, as described in the paper.

b. Simulate CNAs (i.e. $\mathbf{x}_i$) and then regress these CNAs to get gene expression responses $\mathbf{y}_i$, namely $\mathbf{y}_i = \mathbf{\Theta}\mathbf{x}_i + \mathbf{u}_i,\ i = 1, \dots, N$.

c. Simulate correlations in the residual errors of gene expressions $\mathbf{y}_i$ by a non-diagonal covariance matrix $\mathbf{B}\mathbf{B}^T + \mathbf{\Psi}$, namely $\mathbf{u}_i$ is independently drawn from $\mathrm{MVN}_Q(\mathbf{0}, \mathbf{B}\mathbf{B}^T + \mathbf{\Psi})$.

Fitting the above simulated data by the proposed smFARM model and the existing remMap method in which errors are assumed to be iid from $\mathrm{MVN}_Q(\mathbf{0}, \mathbf{I})$, we compare their performances in terms of MCC, sensitivity, and total false (TF), as shown in Table 2.

[Table 2 about here.]

From Table 2, it is easy to see that the findings from the previous simulations reported in the paper are in agreement with the results in the above table. Both remMap and smFARM$_{K=0}$ perform equally well in terms of TF, sensitivity and MCC. Comparing our smFARM with $K > 0$, which accounts for the latent factors, to the remMap or smFARM$_{K=0}$, which ignores latent factors, smFARM$_{K=2}$ is clearly more effective to identify true signals than remMap or smFARM$_{K=0}$.

Table 1: An average running time over 50 simulations in seconds to solve smFARM under the selected tuning parameters $(\lambda_1, \lambda_2)$.

| Method | K=0 | K=1 | K=2 | K=3 |
|---|---|---|---|---|
| smFARM | 3.34(1.64) | 360.92(89.16) | 384.91(80.16) | 404.81(99.29) |

Table 2: Impact of different number of latent factors $K$ on regulator selection and group selection.

| | | | Regulator Selection | | | Group Selection | | | Average Running |
|---|---|---|---|---|---|---|---|---|---|
| SNR | $K_{\text{true}}$ | Method | TF | Sen | MCC | TF | Sen | MCC | Time (Seconds) |
| **Simulation II Setup** | | | | | | | | | |
| 1:3:5 | 2 | smFARM$_{K=2}$ | 64.10(15.90) | 0.75(0.03) | 0.82(0.04) | 11.88(1.66) | 0.63(0.05) | 0.76(0.04) | 327.52(71.53) |
| | | smFARM$_{K=0}$ | 67.46(15.94) | 0.75(0.03) | 0.82(0.04) | 12.18(1.48) | 0.63(0.05) | 0.76(0.03) | 2.84(3.93) |
| | | remMap | 45.35(9.20) | 0.83(0.04) | 0.88(0.03) | 10.20(2.43) | 0.69(0.06) | 0.80(0.05) | 1.57(0.28) |

**Note:**For each Total False (TF), Sensitivity (Sen), or Matthews correlation coefficient (MCC) measurement, we report mean values together with their standard errors on 50 replicates. smFARM$_{K=K_0}$ represents fitting the smFARM on a given number of latent factors $K_0$.