BIOINFORMATICS

Vol. 00 no. 00 2015
Pages 1–6

Supplementary Information: Network regularized sparse non-negative TRI matrix factorization for PATHway identification (NTriPath)

Sunho Park[1], Seung-Jun Kim[2], Donghyeon Yu[3], Samuel Pena-Llopis[4,5], Jianjiong Gao[6], Jin Suk Park[1], Beibei Chen[1], Jessie Norris[1], Xinlei Wang[7], Min Chen[8], Minsoo Kim[1], Jeongsik Yong[9], Zabi Wardak[5,10], Kevin Choe[5,10], Michael Story[5,10], Timothy Starr[11,12], Jae-Ho Cheong[13,14], Tae Hyun Hwang[1,5]

[1]Department of Clinical Sciences, [4]Internal Medicine, [10]Radiation Oncology, and [5]Simmons Comprehensive Cancer Center, University of Texas Southwestern Medical Center, Dallas, Texas, USA.
[2]Department of Computer Science and Electrical Engineering, University of Maryland at Baltimore County, Baltimore, Maryland, USA.
[3]Department of Statistics, Keimyung University, Daegu, South Korea
[6]Center for Molecular Biology, Memorial Sloan Kettering Cancer Center, New York, USA.
[7]Department of Statistical Science, Southern Methodist University, Dallas, Texas, USA.
[8]Department of Mathematical Sciences, University of Texas at Dallas, Dallas, Texas, USA
[9]Department of Biochemistry, Molecular Biology and Biophysics, [9] Obstetrics, Gynecology & Women's Health, [11]Genetics, Cell Biology , [12] Masonic Cancer Center, University of Minnesota Twin Cities, Minneapolis, Minnesota, USA
[13]Department of Surgery, [14]Open NBI Convergence Technology Research Laboratory, Yonsei University College of Medicine, Seoul, South Korea

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

Note: The method introduced here is more general than that of the main paper in the sense that the factor matrix $\boldsymbol{U}$, along with the other matrices $\boldsymbol{V}$ and $\boldsymbol{S}$, is treated as a parameter to be estimated, instead of being fixed to an indicator matrix constructed from the patient's cancer type.

1 MOTIVATION

1.1 Nonnegative tri-matrix factorization (NMTF) for mutation data

The NMTF aims to approximate a data matrix $\boldsymbol{X}$ by the product of three factor matrices, such that $\boldsymbol{X} \approx \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$, where $\boldsymbol{U} \in \mathbb{R}_+^{n\times k_1}$, $\boldsymbol{S} \in \mathbb{R}_+^{k_1\times k_2}$ and $\boldsymbol{V} \in \mathbb{R}_+^{m\times k_2}$. Thus the objective (loss) function to estimate the factor matrices can be defined as

$$\min_{\boldsymbol{U}\geq 0, \boldsymbol{S}\geq 0, \boldsymbol{V}\geq 0} \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top\|_F^2, \tag{1}$$

where $\|\boldsymbol{M}\|_F$ is the Frobenius norm of a matrix $\boldsymbol{M}$, i.e., $\|\boldsymbol{M}\|_F = \sqrt{\sum_{i,j}[\boldsymbol{M}]_{ij}^2}$ [1].

[1] $[\boldsymbol{M}]_{ij}$ and M_{ij} both refer to the (i,j)th element in the matrix $\boldsymbol{M}$.

The NMTF can be applied to simultaneously co-cluster different types of entities, such as the clustering of documents and words (Ding *et al.*, 2006). In our case, it is used to co-cluster patients and genes at the same time, where the factor matrix $\boldsymbol{U}$ can be interpreted as the cluster indicator matrix for patients, the factor matrix $\boldsymbol{V}$ becomes the cluster indicator matrix for genes, and the factor matrix $\boldsymbol{S}$ represents the association between patient clusters and gene clusters. However, the solutions obtained from the NMTF formulation Eq. (1) do not necessarily yield biologically meaningful results.

1.2 Weighted loss function

Note that the data matrix $\boldsymbol{X}$ in the main paper, constructed from the mutation data, is significantly sparse: only about 1% of it is nonzero entries. From the perspective that mutated genes are more informative than normal ones, the Frobenius norm in (1) might not be appropriate to evaluate the goodness of the decomposition models since it is dominated by the errors on zero entries when the data matrix $\boldsymbol{X}$ is sparse. Let us define a weight matrix $\boldsymbol{W} \in \mathbb{R}^{n\times m}$, where W_{ij} is 1 if $X_{ij} > 0$, and otherwise is set to a nonnegative constant. Then the weighted version of the loss (1), which only

concerns errors on the non-zero entries in $\boldsymbol{X}$, is given by

$$\|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top\|_W^2 \triangleq \sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}\left(X_{ij} - [\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top]_{ij}\right)^2. \quad (2)$$

We use this weighted version of the norm, Eq. (2), for the loss function instead of the Frobenius norm in Eq. (1).

1.3 Incorporating prior knowledge on the structure of the factor matrices

We here consider three different types of prior knowledge: the cancer type of each patient, the pathway database and the gene-gene interaction network. With the cancer type information for each patient, we define $\boldsymbol{U}_0 \in \mathbb{R}^{n\times k_1}$ ($k_1 = \#$cancer types), each row of which is a length k_1 row vector containing a single 1 for the cancer type of the patient and 0 elsewhere. With the pathway database, we construct $\boldsymbol{V}_0 \in \mathbb{R}^{m\times k_2}$ ($k_2 = \#$pathways), each column of which is a length of n vector where 1 in the jth element indicates that the jth gene is a member of the corresponding pathway and 0 indicates that it is not a member. Thus we define that the constraints enforcing the factor matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are closed to $\boldsymbol{U}_0$ and $\boldsymbol{V}_0$, respectively:

$$\|\boldsymbol{U} - \boldsymbol{U}_0\|_F^2 < t_{U_0}, \quad \|\boldsymbol{V} - \boldsymbol{V}_0\|_F^2 < t_{V_0}, \quad (3)$$

where $t_{U_0}, t_{V_0} \geq 0$. These constraints enable us to obtain biologically meaningful results from the decomposition results of the tri-factorization models: 1) each row of $\boldsymbol{U}$ can be interpreted as a soft membership of the corresponding patient within the different cancers; 2) we can identify new member genes in the pathways by checking the newly added nonzero entries in $\boldsymbol{V}$; and 3) the factor $\boldsymbol{S}$ reveals how different cancer types are associated with the pathways.

For the factor matrix $\boldsymbol{V}$, we additionally consider network regularization constraints, which are based on the assumption that any genes connected in the network are more likely to be placed in the same pathway. We use in the paper the human gene-gene interaction network (Zhang *et al.*, 2011). Denoting the adjacency matrix of the gene interaction network by $\boldsymbol{A} \in \mathbb{R}^{m\times m}$, where A_{ij} is 1 if the genes i and j are interacting with each other, otherwise 0, the network regularization constraint can be defined by

$$\sum_{k=1}^{k_2}\sum_{i=1}^{m}\sum_{j=1}^{m} A_{ij}(V_{ki} - V_{kj})^2 \leq t_{V_L} \quad (4)$$

where $t_{V_L} \geq 0$. Let us denote the Laplacian matrix by $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, where $\boldsymbol{D}$ is a diagonal matrix and its diagonal element is a row sum of $\boldsymbol{A}$, i.e., $D_{ii} = \sum_{j=1}^{m} A_{ij}$. Then the constraint Eq. (4) can be rewritten as

$$\mathrm{tr}\{\boldsymbol{V}^\top\boldsymbol{L}\boldsymbol{V}\} \leq t_{V_L}. \quad (5)$$

2 NETWORK REGULARIZED SPARSE NON-NEGATIVE TRI MATRIX FACTORIZATION FOR PATHWAY IDENTIFICATION (NTRIPATH)

Combining all the ideas in the previous section and the routines to avoid *inadmissible zeros* (Chi and Kolda, 2012), we propose a Network regularized sparse non-negative TRI matrix factorization for PATHway identification (NTriPath). In addition, we provide the convergence analysis of the proposed method, based on the technique that is used to prove the convergence of nonnegative matrix factorization algorithms (Lee and Seung, 2001; Ding *et al.*, 2006; Blondel *et al.*, 2008).

2.1 Objective function

Combining the objective function Eq. (2), the constraints obtained from the available prior knowledge (Eq. (3) and Eq. (5)), and the sparsity control constraints of the factor matrices, the optimization problem can be formulated as

$$\min_{\boldsymbol{U}\geq 0, \boldsymbol{S}\geq 0, \boldsymbol{V}\geq 0} f(\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}), \quad (6)$$

where

$$\begin{aligned} & f(\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}) \\ &= \frac{1}{2}\Big(\|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top\|_W^2 + \lambda_U\|\boldsymbol{U}\|_1^2 + \lambda_S\|\boldsymbol{S}\|_1^2 + \lambda_V\|\boldsymbol{V}\|_1^2 + \\ &+ \lambda_{U_0}\|\boldsymbol{U} - \boldsymbol{U}_0\|_F^2 + \lambda_{V_0}\|\boldsymbol{V} - \boldsymbol{V}_0\|_F^2 + \lambda_{V_L}\mathrm{tr}\{\boldsymbol{V}^\top\boldsymbol{L}\boldsymbol{V}\}\Big), \end{aligned} \quad (7)$$

where $\|\boldsymbol{M}\|_1 = \sum_{i,j}|M_{ij}|$ and $\{\lambda.\} \geq 0$ are user-specific regularization coefficients.

2.2 Update rule

To find the optimal solution of Eq. (6), we use the multiplicative update rule (Lee and Seung, 2001) which is computationally effective because the multiplicative factor can be easily calculated based on the gradient information. For example, consider the gradient of Eq. (6) with respect to $\boldsymbol{V}$, which can be decomposed into two positive terms:

$$\frac{\partial f}{\partial \boldsymbol{V}} = -\boldsymbol{\gamma}^N + \boldsymbol{\gamma}^P, \quad (8)$$

where

$$\boldsymbol{\gamma}^N = (\boldsymbol{W}\circ\boldsymbol{X})^\top\boldsymbol{U}\boldsymbol{S} + \lambda_{V_L}\boldsymbol{A}\boldsymbol{V} + \lambda_{V_0}\boldsymbol{V}_0, \quad (9)$$

$$\boldsymbol{\gamma}^P = \widehat{\boldsymbol{X}}^\top\boldsymbol{U}\boldsymbol{S} + \lambda_{G_0}\boldsymbol{V} + \lambda_{V_L}\boldsymbol{D}\boldsymbol{V} + \lambda_V\|\boldsymbol{V}\|_1\boldsymbol{E}, \quad (10)$$

where the notation $\circ$ stands for the Hadamard product (element-wise multiplication), $\boldsymbol{E} \in \mathbb{R}^{m\times k_2}$ is a matrix of ones and $\widehat{\boldsymbol{X}} = \boldsymbol{W}\circ(\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top)$. Then the multiplicative update rule for the factor matrix $\boldsymbol{V}$ is of the form

$$\boldsymbol{V} \leftarrow \boldsymbol{V}\circ\frac{[\boldsymbol{\gamma}^N]}{[\boldsymbol{\gamma}^P]}, \quad (11)$$

where $\frac{[\cdot]}{[\cdot]}$ is an element-wise division operator. The multiplicative rule Eq. (11) not only preserves the nonnegativity of parameters but also guarantees $\partial f/\partial \boldsymbol{V} = 0$ when the algorithm converges. Similarly, the update rules for all factor matrices, including $\boldsymbol{V}$, can

be written in element-wise form:

$$U_{ij} \leftarrow U_{ij} \frac{\Big[(\boldsymbol{W}\circ\boldsymbol{X})\boldsymbol{V}\boldsymbol{S}^\top + \lambda_{U_0}\boldsymbol{U}_0\Big]_{ij}}{\Big[\widehat{\boldsymbol{X}}\boldsymbol{V}\boldsymbol{S}^\top + \lambda_{U_0}\boldsymbol{U}\Big]_{ij} + \lambda_U\|\boldsymbol{U}\|_1}, \quad (12)$$

$$S_{ij} \leftarrow S_{ij} \frac{\Big[\boldsymbol{U}^\top(\boldsymbol{W}\circ\boldsymbol{X})\boldsymbol{V}\Big]_{ij}}{\Big[\boldsymbol{U}^\top\widehat{\boldsymbol{X}}\boldsymbol{V}\Big]_{ij} + \lambda_S\|\boldsymbol{S}\|_1}, \quad (13)$$

$$V_{ij} \leftarrow V_{ij} \frac{\Big[(\boldsymbol{W}\circ\boldsymbol{X})^\top\boldsymbol{U}\boldsymbol{S} + \lambda_{V_L}\boldsymbol{A}\boldsymbol{V} + \lambda_{V_0}\boldsymbol{V}_0\Big]_{ij}}{\Big[\widehat{\boldsymbol{X}}^\top\boldsymbol{U}\boldsymbol{S} + \lambda_{G_0}\boldsymbol{V} + \lambda_{V_L}\boldsymbol{D}\boldsymbol{V}\Big]_{ij} + \lambda_V\|\boldsymbol{V}\|_1}. \quad (14)$$

2.3 Convergence analysis

We will prove that (*convergence*) the algorithm converges under the update rules Eq. (12)-(14) and that (*correctness*) at convergence its solution satisfies Karush Kuhn Tucker (KKT) optimality conditions, i.e., the algorithm converges to a local minima.

To prove the convergence, we will show that alternatively updating $\boldsymbol{U}$, $\boldsymbol{S}$ and $\boldsymbol{V}$ will monotonically decrease the objective function Eq. (6):

$$\begin{aligned} f(\boldsymbol{U}^0,\boldsymbol{S}^0,\boldsymbol{V}^0) \;&\geq\; f(\boldsymbol{U}^1,\boldsymbol{S}^0,\boldsymbol{V}^0) \geq f(\boldsymbol{U}^1,\boldsymbol{S}^1,\boldsymbol{V}^0) \\ &\geq f(\boldsymbol{U}^1,\boldsymbol{S}^1,\boldsymbol{V}^1) \geq f(\boldsymbol{U}^2,\boldsymbol{S}^1,\boldsymbol{V}^1) \geq ..., \end{aligned} \quad (15)$$

where $\boldsymbol{U}^t$ is the solution of $\boldsymbol{U}$ at iteration t. We here only prove for the case of updating $\boldsymbol{V}$ given $\boldsymbol{U}$ and $\boldsymbol{S}$ since the other cases, updating $\boldsymbol{U}$ or $\boldsymbol{S}$ given the other factors, can be proved in a similar way. To do this, we will make use of the auxiliary function (Lee and Seung, 2001): the function $Z(\boldsymbol{V},\overline{\boldsymbol{V}})$ is called an auxiliary function for the function f (when $\boldsymbol{U}$ and $\boldsymbol{S}$ are fixed) if, for any $\boldsymbol{V}$ and $\overline{\boldsymbol{V}}$, it satisfies

$$Z_V(\boldsymbol{V},\overline{\boldsymbol{V}}) \geq f(\boldsymbol{V}), \quad Z_V(\overline{\boldsymbol{V}},\overline{\boldsymbol{V}}) = f(\overline{\boldsymbol{V}}). \quad (16)$$

Note that, $\overline{\boldsymbol{V}}$ corresponds to the current solution of $\boldsymbol{V}$ in our proof. We also need the following lemma to construct the auxiliary function for our problem.

LEMMA 1. *For a symmetric nonnegative matrix $\boldsymbol{B}$ and a positive vector $\boldsymbol{b}$, the matrix, diag$\Big(\frac{[\boldsymbol{B}\boldsymbol{b}]}{[\boldsymbol{b}]}\Big) - \boldsymbol{B}$, is positive semi-definite (Blondel* et al.*, 2008).*

Proof. Note that the operator diag$(\boldsymbol{b})$ creates a square matrix with the elements of $\boldsymbol{b}$ on the diagonal. See (Blondel *et al.*, 2008) for the proof.

THEOREM 1. *(Convergence) The objective function Eq. (6) is monotonically decreasing under the update rules (12)-(14).*

Proof. We will show that updating $\boldsymbol{V}$ given $\boldsymbol{U}$ and $\boldsymbol{S}$ with the update rule (14) monotonically decreases the objective function Eq. (6). Define $\boldsymbol{v} = \text{vec}(\boldsymbol{V}^\top)$ and $\boldsymbol{v}_c = \text{vec}(\boldsymbol{V})$, where vec is a vectorization operator which converts a matrix into a column vector. In addition, let $\boldsymbol{P} \in \mathbb{R}^{mk_2\times mk_2}$ denote the permutation matrix such that $\boldsymbol{v}_c = \boldsymbol{P}\boldsymbol{v}$ (also $\boldsymbol{v} = \boldsymbol{P}^\top\boldsymbol{v}_c$ since $\boldsymbol{P}\boldsymbol{P}^\top = \boldsymbol{I}_{mk_2}$, where $\boldsymbol{I}_{mk_2}$ is an $mk_2 \times mk_2$ identity matrix). Then the objective function Eq. (6) can be rewritten in terms of $\boldsymbol{v}$ with the fixed $\boldsymbol{U}$ and $\boldsymbol{S}$:

$$f(\boldsymbol{v}) = c - \boldsymbol{v}^\top\text{vec}\Big((\boldsymbol{U}\boldsymbol{S})^\top(\boldsymbol{W}\circ\boldsymbol{X}) + \lambda_{V_0}\boldsymbol{V}_0^\top\Big) + \frac{1}{2}\boldsymbol{v}^\top\boldsymbol{H}\boldsymbol{v}, \quad (17)$$

where c is a constant irrelevant to $\boldsymbol{V}$, and $\boldsymbol{H}$ is the Hessian matrix:

$$\begin{aligned} \boldsymbol{H} \;&=\; \text{blockdiag}(\sum_{i=1}^m W_{i1}\boldsymbol{z}_i\boldsymbol{z}_i^\top, ..., \sum_{i=1}^m W_{in}\boldsymbol{z}_i\boldsymbol{z}_i^\top) \\ &+\; \lambda_{V_L}\boldsymbol{P}^\top(\boldsymbol{I}_{k_2}\otimes\boldsymbol{L})\boldsymbol{P} + \lambda_{V_0}\boldsymbol{I}_{mk_2} + \lambda_V\boldsymbol{1}\boldsymbol{1}^\top, \end{aligned} \quad (18)$$

where $\boldsymbol{z}_i \in \mathbb{R}^{k_2}$ is the ith row vector of the matrix $(\boldsymbol{U}\boldsymbol{S})$, i.e., $\boldsymbol{z}_i = ([\boldsymbol{U}\boldsymbol{S}]_{i,:})^\top$, blockdiag constructs a block diagonal matrix from given block matrices and $\boldsymbol{1}$ is a mk_2-vector consisting of all 1s. It is convenient to rewrite the Hessian matrix $\boldsymbol{H}$ as a sum of the term involving the Laplacian matrix and the other terms, i.e., $\boldsymbol{H} = \boldsymbol{H}_{\backslash L} + \lambda_{V_L}\boldsymbol{P}^\top(\boldsymbol{I}_{k_2}\otimes\boldsymbol{L})\boldsymbol{P}$. To define the auxiliary function for the function f, we first consider its Taylor expansion at $\overline{\boldsymbol{v}}$ ($\triangleq \text{vec}(\overline{\boldsymbol{V}}^\top)$):

$$\begin{aligned} f(\boldsymbol{v}) \;&=\; f(\overline{\boldsymbol{v}}) \\ &+\; (\boldsymbol{v}-\overline{\boldsymbol{v}})^\top \bigtriangledown f(\overline{\boldsymbol{v}}) + \frac{1}{2}(\boldsymbol{v}-\overline{\boldsymbol{v}})^\top\boldsymbol{H}(\boldsymbol{v}-\overline{\boldsymbol{v}}). \end{aligned} \quad (19)$$

Then we define the following function $Z_V(\boldsymbol{v},\overline{\boldsymbol{v}})$ in vectorial form and show that it is the auxiliary function for the function f:

$$\begin{aligned} Z_V(\boldsymbol{v},\overline{\boldsymbol{v}}) \;&=\; f(\overline{\boldsymbol{v}}) + (\boldsymbol{v}-\overline{\boldsymbol{v}})^\top \bigtriangledown f(\overline{\boldsymbol{v}}) \\ &+\; \frac{1}{2}(\boldsymbol{v}-\overline{\boldsymbol{v}})^\top\boldsymbol{\Gamma}(\overline{\boldsymbol{v}})(\boldsymbol{v}-\overline{\boldsymbol{v}}), \end{aligned} \quad (20)$$

where the diagonal matrix $\boldsymbol{\Gamma}(\overline{\boldsymbol{v}})$ is set to

$$\boldsymbol{\Gamma}(\overline{\boldsymbol{v}}) = \text{diag}\Big(\frac{\big[\big(\boldsymbol{H}_{\backslash L} + \lambda_{V_L}\boldsymbol{P}^\top(\boldsymbol{I}_{k_2}\otimes\boldsymbol{D})\boldsymbol{P}\big)\overline{\boldsymbol{v}}\big]}{\overline{\boldsymbol{v}}}\Big). \quad (21)$$

It is obvious that $Z_V(\overline{\boldsymbol{v}},\overline{\boldsymbol{v}}) = f(\overline{\boldsymbol{v}})$, and one can show that $Z_V(\boldsymbol{v},\overline{\boldsymbol{v}}) \geq f(\boldsymbol{v})$ for any positive vector $\boldsymbol{v}$ using Lemma 1 (to show $\boldsymbol{H}_{\backslash L}$ is positive semi-definite) and the fact that

$$\begin{aligned} &\boldsymbol{v}^\top\Big(\text{diag}\Big(\frac{\big[\boldsymbol{P}^\top(\boldsymbol{I}_{k_2}\otimes\boldsymbol{D})\boldsymbol{P}\overline{\boldsymbol{v}}\big]}{\overline{\boldsymbol{v}}}\Big) - \boldsymbol{P}^\top(\boldsymbol{I}_{k_2}\otimes\boldsymbol{L})\boldsymbol{P}\Big)\boldsymbol{v} \\ &\quad= \boldsymbol{v}_c^\top(\boldsymbol{I}_{k_2}\otimes\boldsymbol{A})\boldsymbol{v}_c \geq 0. \end{aligned} \quad (22)$$

For simplicity, we define $\widetilde{\boldsymbol{H}} = \boldsymbol{H}_{\backslash L} + \lambda_{V_L}\boldsymbol{P}^\top(\boldsymbol{I}_{k_2}\otimes\boldsymbol{D})\boldsymbol{P}$.

Now, we need to find a global minimum, $\widehat{\boldsymbol{v}}$, of $Z_V(\boldsymbol{v},\overline{\boldsymbol{v}})$:

$$\widehat{\boldsymbol{v}} = \arg\min_{\boldsymbol{v}} Z_V(\boldsymbol{v},\overline{\boldsymbol{v}}). \quad (23)$$

From the definition of the auxiliary function in Eq. (16), it is obvious

$$f(\overline{\boldsymbol{v}}) = Z_V(\overline{\boldsymbol{v}},\overline{\boldsymbol{v}}) \geq Z_V(\widehat{\boldsymbol{v}},\overline{\boldsymbol{v}}) \geq f(\widehat{\boldsymbol{v}}). \quad (24)$$

Thus we can confirm that the solution $\widehat{\boldsymbol{v}}$ monotonically decreases the function f. On the other hand, the minimum solution is found

by solving $\frac{\partial Z_V(\boldsymbol{v},\overline{\boldsymbol{v}})}{\partial \boldsymbol{v}} = 0$:

$$\widehat{\boldsymbol{v}} = \overline{\boldsymbol{v}} - \boldsymbol{\Gamma}_V^{-1}(\overline{\boldsymbol{v}}) \bigtriangledown f(\overline{\boldsymbol{v}}) \quad (25)$$

$$= \overline{\boldsymbol{v}} - diag\Big(\frac{[\overline{\boldsymbol{v}}]}{[\widetilde{\boldsymbol{H}}\overline{\boldsymbol{v}}]}\Big) \bigtriangledown f(\overline{\boldsymbol{v}}) \quad (26)$$

$$= \overline{\boldsymbol{v}} \circ \frac{[\widetilde{\boldsymbol{H}}\overline{\boldsymbol{v}} - \bigtriangledown f(\overline{\boldsymbol{v}})]}{[\widetilde{\boldsymbol{H}}\overline{\boldsymbol{v}}]}. \quad (27)$$

We can rewrite the above update equation in matrix form exactly the same as our update rule (14):

$$\widehat{\boldsymbol{V}} = \overline{\boldsymbol{V}} \circ \frac{[(\boldsymbol{W} \circ \boldsymbol{X})^\top \boldsymbol{U}\boldsymbol{S} + \lambda_{V_L} \boldsymbol{A}\overline{\boldsymbol{V}} + \lambda_{V_0} \boldsymbol{V}_0]}{[\overline{\boldsymbol{X}}^\top \boldsymbol{U}\boldsymbol{S} + \lambda_{G_0}\overline{\boldsymbol{V}} + \lambda_{V_L} \boldsymbol{D}\overline{\boldsymbol{V}} + \lambda_V \|\overline{\boldsymbol{V}}\|_1 \boldsymbol{E}]} \quad (28)$$

where $\overline{\boldsymbol{X}} = \boldsymbol{W} \circ (\boldsymbol{U}\boldsymbol{S}\overline{\boldsymbol{V}}^\top)$. Thus we can conclude that the objective function (when $\boldsymbol{U}$ and $\boldsymbol{S}$ are given) is monotonically decreasing under the update rule (14).

THEOREM 2. *(Correctness) At convergence, the solution satisfies the KKT optimality conditions.*

Proof. It is straightforward based on the nature of the multiplicative rule (preserving the positivity of parameters) and the definitions of our update rules (see Eq. (11)).

2.4 Inadmissible zero avoidance

There is still a computational issue in implementing the proposed algorithm. The multiplicative rules might lead to the inadmissible zero problem (Chi and Kolda, 2012): an entry in the factor matrices is stuck at zero when it becomes zero although the zero value does not satisfy stationarity conditions. In practice, the inadmissible zero often appears due to the finite precision of machines. To solve this problem, one can examine the KKT conditions for the solution in each update and then replace the inadmissible zero entries with a small positive number κ as in (Chi and Kolda, 2012; Seung-Jun *et al.*, 2012). We here present only an avoidance method of the inadmissible zero for the factor $\boldsymbol{V}$ since its extension for the factor matrices is straightforward.

Note that, the KKT conditions for the factor matrix $\boldsymbol{V}$ to the problem (6) can be written in element-wise form:

$$V_{ij} \geq 0, \quad \gamma_{ij}^P - \gamma_{ij}^N \geq 0, \quad V_{ij}\left(\gamma_{ij}^P - \gamma_{ij}^N\right) = 0. \quad (29)$$

The KKT condition states that if $V_{ij} > 0$, the multiplicative factor should be equal to 1; otherwise it should be less than or equal to 1. Thus we just replace the zero entry whose corresponding multiplicative factor is greater than 1 with κ to prevent the inadmissible zero from occurring. The NTriPath algorithm, including the update rules with the inadmissible zero avoidance methods, is summarized in Table 1.

3 SIMULATION ANALYSIS

We performed experiments using synthetic datasets to evaluate the performance of NTriPath to discover cancer-type-specific pathways and new member genes in the pathways.

Algorithm 1 NTriPath

procedure NTRIPATH($\boldsymbol{X}$,λ_U,λ_{U_0},λ_S,λ_V,λ_{V_0},λ_{V_L})
  Set $\kappa > 0$, $\kappa_{\text{tol}} > 0$ and $\epsilon > 0$ to small values.
  Set $\boldsymbol{U} \leftarrow \min\{\boldsymbol{U}_0, \kappa\}$, $\boldsymbol{V} \leftarrow \min\{\boldsymbol{V}_0, \kappa\}$ and fill all entries of $\boldsymbol{S}$ with one. Initialize $\{\boldsymbol{\alpha}, \widetilde{\boldsymbol{U}}\} \leftarrow 0^{n \times k_1}$, $\{\boldsymbol{\beta}, \widetilde{\boldsymbol{S}}\} \leftarrow 0^{k_1 \times k_2}$ and $\{\boldsymbol{\gamma}, \widetilde{\boldsymbol{V}}\} \leftarrow 0^{m \times k_2}$.
  while not converged **do**
    $\widehat{\boldsymbol{X}} \leftarrow \boldsymbol{W} \circ (\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top)$
    $\alpha_{ij} = \dfrac{\left[(\boldsymbol{W}\circ\boldsymbol{X})\boldsymbol{V}\boldsymbol{S}^\top + \lambda_{U_0}\boldsymbol{U}_0\right]_{ij}}{\left[\widehat{\boldsymbol{X}}\boldsymbol{V}\boldsymbol{S}^\top + \lambda_{U_0}\boldsymbol{U}\right]_{ij} + \lambda_U\|\boldsymbol{U}\|_1 + \epsilon}$
    $\widetilde{U}_{ij} = \begin{cases} \kappa & \text{if } U_{ij} < \kappa_{\text{tol}} \text{ and } \alpha_{ij} > 1 \\ 0 & \text{otherwise} \end{cases}$
    $\boldsymbol{U} \leftarrow (\boldsymbol{U} + \widetilde{\boldsymbol{U}}) \circ \boldsymbol{\alpha}$.
    $\widehat{\boldsymbol{X}} \leftarrow \boldsymbol{W} \circ (\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top)$
    $\beta_{ij} = \dfrac{\left[\boldsymbol{U}^\top(\boldsymbol{W}\circ\boldsymbol{X})\boldsymbol{V}\right]_{ij}}{\left[\boldsymbol{U}^\top\widehat{\boldsymbol{X}}\boldsymbol{V}\right]_{ij} + \lambda_S\|\boldsymbol{S}\|_1 + \epsilon}$
    $\widetilde{S}_{ij} = \begin{cases} \kappa & \text{if } S_{ij} < \kappa_{\text{tol}} \text{ and } \beta_{ij} > 1 \\ 0 & \text{otherwise} \end{cases}$
    $\boldsymbol{S} \leftarrow (\boldsymbol{S} + \widetilde{\boldsymbol{S}}) \circ \boldsymbol{\beta}$.
    $\widehat{\boldsymbol{X}} \leftarrow \boldsymbol{W} \circ (\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top)$
    $\gamma_{ij} = \dfrac{\left[(\boldsymbol{W}\circ\boldsymbol{X})^\top\boldsymbol{U}\boldsymbol{S} + \lambda_{V_L}\boldsymbol{W}\boldsymbol{V} + \lambda_{V_0}\boldsymbol{V}_0\right]_{ij}}{\left[\widehat{\boldsymbol{X}}^\top\boldsymbol{U}\boldsymbol{S} + \lambda_{G_0}\boldsymbol{V} + \lambda_{V_L}\boldsymbol{D}\boldsymbol{V}\right]_{ij} + \lambda_V\|\boldsymbol{V}\|_1 + \epsilon}$
    $\widetilde{V}_{ij} = \begin{cases} \kappa & \text{if } V_{ij} < \kappa_{\text{tol}} \text{ and } \gamma_{ij} > 1 \\ 0 & \text{otherwise} \end{cases}$
    $\boldsymbol{V} \leftarrow (\boldsymbol{V} + \widetilde{\boldsymbol{V}}) \circ \boldsymbol{\gamma}$.
  end while
  return $\boldsymbol{U}$, $\boldsymbol{S}$, $\boldsymbol{V}$.
end procedure

3.1 Data preparation

We first generated the mutation matrix $\boldsymbol{X}$ containing 250 patient samples and 1000 genes (see Fig. 1). $\boldsymbol{U}$ represents five patient subgroups (e.g., A, B, C, D, and E) and $\boldsymbol{V}_0$ represents 10 pathways consisted of 100 genes per pathway. Each subgroup included between 1-7 altered pathways. We introduced different mutation rates (e.g., subgroup C has a higher mutation rate compared to other subgroups) to investigate whether different mutation rates for each subgroup would affect the performance of NTriPath to discover cancer-type-specific altered pathways. We generated the gene-gene interaction networks $\boldsymbol{A}$ and used it as prior knowledge. The member genes in the pathway are densely connected in the gene-gene interaction networks.

3.2 Experiments

We ran NTriPath using a synthetic dataset with $\lambda_V = 0.1$, $\lambda_{V_0} = 0.1$, $\lambda_{V_L} = 0.1$, $\lambda_S = 0.001$. We set W_{ij} for non zero entries to 1 and for zero entries to 0.1 in all the experiments. Fig. 2 shows the plot of the objective function over iterations for a typical run of the NTriPath on the synthetic data. We found that the algorithm converged well and almost reached the minimum value at iteration 20. Thus we set the maximum iteration to 20 in all the

experiments. The results of NTriPath are shown in Fig. 1. Learned $\boldsymbol{S}$ indicates that NTriPath could accurately identify subgroup-specific altered pathways. For example, there are four pathways associated with subgroup A (e.g., A1, A4, A5, and A10) and learned $\boldsymbol{S}$ indicated that NTriPath accurately identified subgroup A-specific altered pathways. $\widetilde{\boldsymbol{X}}$ which represents the reconstructed matrix $\boldsymbol{X}$ based on $\boldsymbol{USV}^\top$ indicates that NTriPath could reconstruct original matrix $\boldsymbol{X}$. We performed additional experiments using the synthetic dataset to access NTriPath for discovering new member genes in the pathways. We introduced a set of mutated genes into subgroups A and C along with 10th pathway into the mutation matrix $\boldsymbol{X}$ (See the red box in the mutation matrix $\boldsymbol{X}$ in Fig. 3). In addition, the newly added mutated genes interact with the member genes in the 10th pathway through the gene-gene interaction networks (See the red box in the gene-gene interaction networks in Fig. 3). The initial pathway information $\boldsymbol{V}_0$ does not include those genes as member genes in the 10th pathway (See the red box in the mutation matrix $\boldsymbol{V}_0$ in Fig. 3). The purpose of this experiment is to investigate whether NTriPath can correctly identify new member genes. We ran NTriPath with all the same parameters used in the previous experiments. Results indicated that NTriPath could accurately identify the new member genes in the 10th pathway (See the blue box in the pathway information $\boldsymbol{V}$ in Fig. 3).

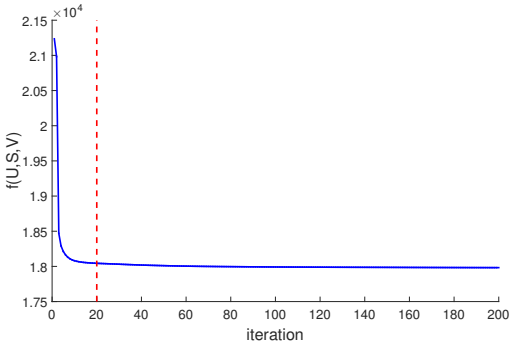

Fig. 2. Plot of objective function over iterations for a typical run of NTriPath on the synthetic data described in Fig. 1. One can see that the method almost reached the minimum value at iteration 20.

Finally, we ran experiments to evaluate the performance of how accurately NTriPath identified subgroup-specific pathway associations in genome-wide scale settings. We generated synthetic datasets containing 250 patients with five subgroups (e.g., 50 patients per subgroup) and 12,000 genes with 120 pathways. Each pathway consisted of 100 genes. Each subgroup included between 5-15 altered pathways. In addition, we generated the gene-gene interaction networks, where member genes in the same pathway were densely connected. We considered different sparsity levels of the mutation data matrix for each patient subgroup using probability P. For example, $P = 0.10$ indicates that X_{ij} is set to 1 if the random number sampled from a uniform distribution on the interval $[0, 1]$ is less than and equal to P. A higher value of P indicates a higher mutation rate, and a lower value of P indicates a lower mutation rate in the mutation matrix X. We ran NTriPath on these synthetic datasets with different values of P and repeated 50 times for each P. We used all the same parameter setting in the previous experiments. To evaluate the accuracy of the subgroup-pathway association $\boldsymbol{S}$, we used ROCArea (Joachims, 2005), which is a widely used performance measure for ranking-based classification algorithms, such as ranking-SVM (Joachims, 2002). In the main paper, we identified cancer-type-specific altered pathways by ranking the ith row elements in the $\boldsymbol{S}$ matrix for the ith cancer type. We used the same strategy to identify the altered pathways for each subgroup. Thus to make the predictions correct, the elements in the matrix $\boldsymbol{S}$ corresponding to the altered pathways for each subgroup should be higher (i.e., more highly ranked) than those of the other pathways. Denote pos_i by a set of the altered pathways for the ith subgroup and neg_i by the remain pathways. Then the ROCArea for the ith subgroup is defined based on the number of the incorrectly predicted pairs between positive labels (pathways in pos_i) and negative labels (pathways in neg_i) (Joachims, 2005):

$$\text{ROCArea}_i = 1 - \frac{\#SwappedPairs_i}{\#pos_i \cdot \#neg_i} \tag{30}$$

where

$$SwappedPairs_i = \{(j,l)|(j \in pos_i, l \in neg_i)\ and\ (S_{ij} < S_{il})\}, \tag{31}$$

where $\#pos_i$ is the number of elements in the set pos_i. The closer the ROCArea value is to 1, the more accurate the method is. All the results are summarized in Table 1, where the ROCArea values were averaged among 50 trials, and the mean and standard deviation values were reported. Although the ROCArea of the method slightly decreased as the sparsity level of the data matrix, $\boldsymbol{X}$, increases (or as P decreases), the results reconfirm that the NtriPath could accurately identify subgroup-pathway associations in most cases.

Table 1. Experiments with genome-wide scale synthetic datasets for subgroup-pathway association identification. The ROCArea values were averaged among 50 trials, and the mean values and their standard deviation values (in the parentheses) were reported. The first column shows the index for the subgroup and the number of the pre-defined pathways for the corresponding subgroup in the parentheses. For example, subgroup 1 has 10 altered pathways and subgroup 3 has 14 altered pathways.

	$P = 0.10$	$P = 0.07$	$P = 0.04$	$P = 0.01$
1 (10)	1.000(0.000)	0.987(0.013)	0.940(0.015)	0.890(0.041)
2 (10)	1.000(0.000)	0.986(0.014)	0.934(0.017)	0.882(0.045)
3 (14)	1.000(0.000)	0.981(0.014)	0.942(0.010)	0.874(0.040)
4 (9)	1.000(0.003)	0.975(0.020)	0.923(0.018)	0.874(0.044)
5 (11)	1.000(0.000)	0.996(0.008)	0.964(0.012)	0.919(0.027)

REFERENCES

Blondel,V.D., diep Ho,N., Dooren,P., Louvain,U..D. and Lematre,A.G. (2008) Weighted nonnegative matrix factorization and face feature extraction. In *In Image and Vision Computing* pp. 1–17.

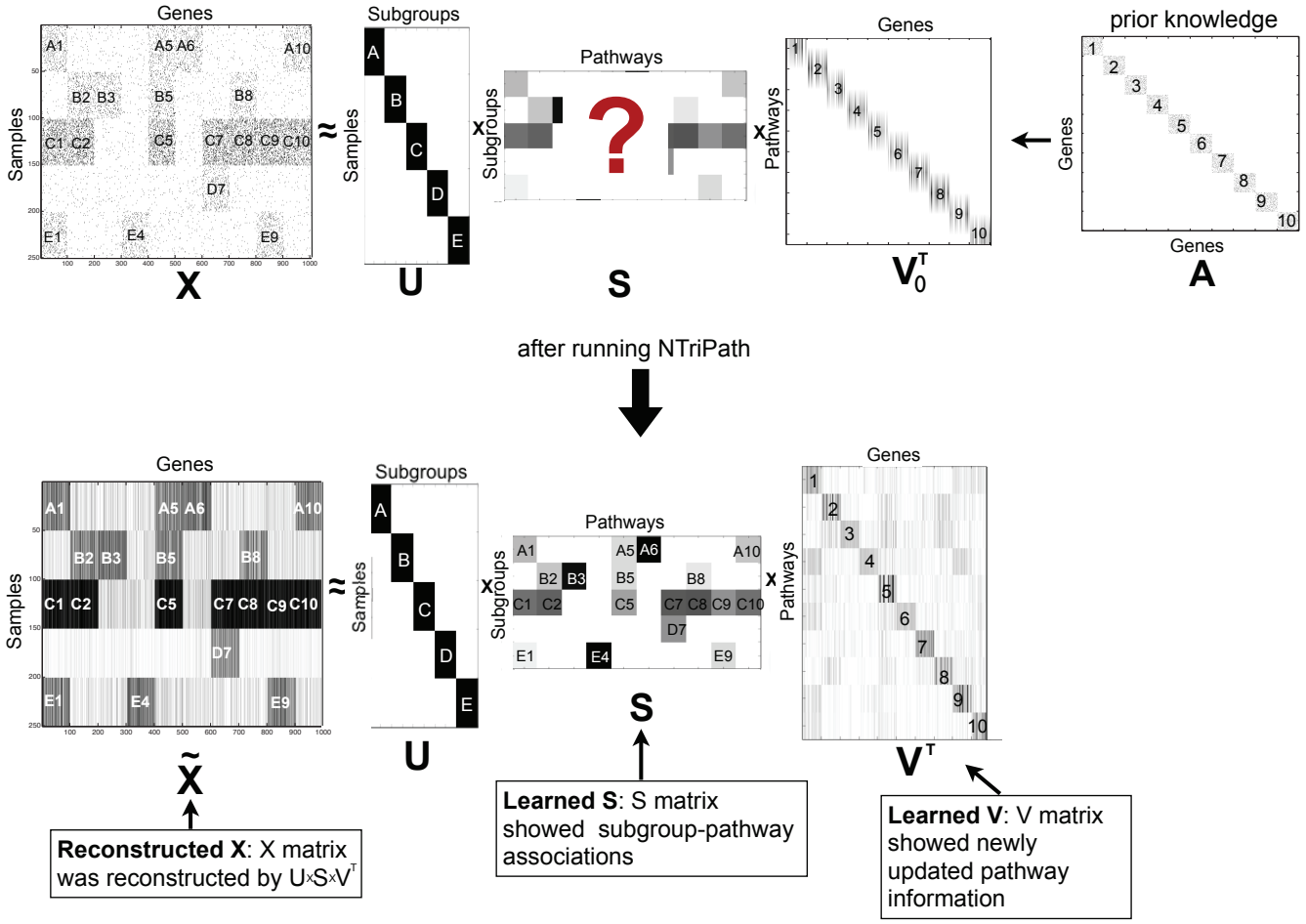

Fig. 1. Simulation on subgroup-specific altered pathway identification. $\boldsymbol{X}$ represents the patient mutation data. There are five subgroups and each subgroup has a different mutation rate. In particular, group C has a higher mutation rate compared to other groups. $\boldsymbol{U}$ represents patient subgroup information and $\boldsymbol{S}$ represent patient subgroup and pathway association. $\boldsymbol{V}_0$ and $\boldsymbol{A}$ represent initial pathway information and gene-gene interaction networks, respectively. NTriPath outputs two solutions $\boldsymbol{S}$ and $\boldsymbol{V}$. Learned $\boldsymbol{S}$ represents inferred subgroup and pathway associations by NTriPath. Learned $\boldsymbol{V}$ represents newly learned pathway information. The mutation matrix $\boldsymbol{X}$ can be reconstructed by $\boldsymbol{USV}^\top$ and $\widetilde{\boldsymbol{X}}$ represents the reconstructed mutation matrix $\boldsymbol{X}$.

Chi,E.C. and Kolda,T.G. (2012) On tensors, sparsity, and nonnegative factorizations. *SIAM Journal on Matrix Analysis and Applications,* **33** (4), 1272–1299.

Ding,C., Li,T., Peng,W. and Park,H. (2006) Orthogonal nonnegative matrix tri-factorizations for clustering. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, Philadelphia, PA.

Joachims,T. (2002) Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* pp. 133–142.

Joachims,T. (2005) A support vector method for multivariate performance measures. In *Proceedings of the International Conference on Machine Learning (ICML)* pp. 377–384.

Lee,D.D. and Seung,H.S. (2001) Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, (Leen,T., Dietterich,T. and Tresp,V., eds),. pp. 556–562.

Seung-Jun,K., TaeHyun,H. and Giannakis,G.B. (2012) Sparse robust matrix tri-factorization with application to cancer genomics. In *Procedding of 3rd International workshop on Cognitive Information Processing* pp. 1–6.

Zhang,S., Li,Q., Liu,J. and Zhou,X.J. (2011) A novel computational framework for simultaneous integration of multiple types of genomic data to identify microrna-gene regulatory modules. *Bioinformatics,* **27** (13), i401–i409.

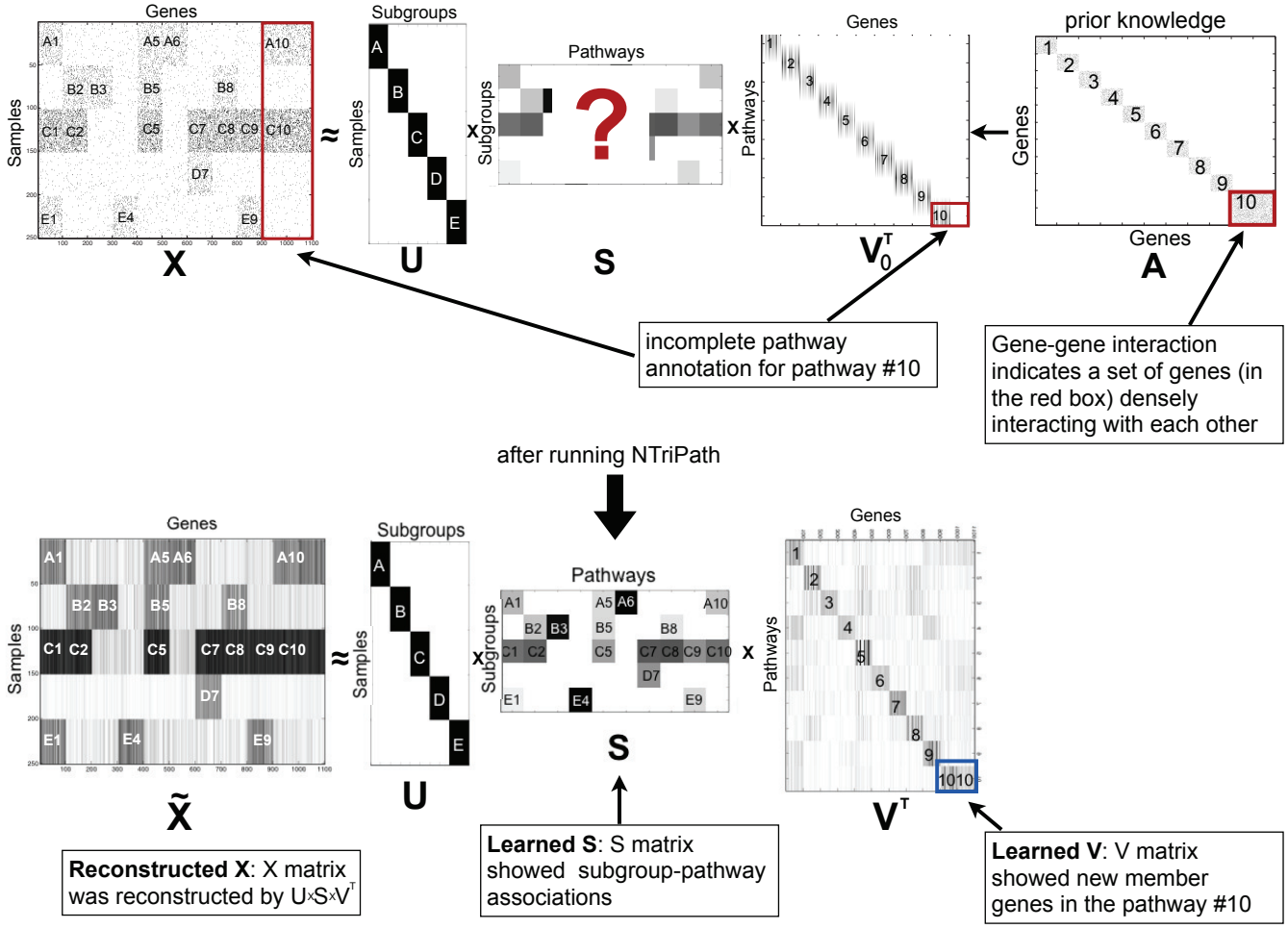

Fig. 3. Simulation on new pathway member gene identification. We introduce additional mutated genes into subgroup A and C along with member genes of 10th pathway. Newly added mutated genes are connected with member genes in the 10th pathway through the networks $\boldsymbol{A}$. The initial pathway information $\boldsymbol{V}_0$ does not include those genes as member genes in the 10th pathway. After running NTriPath, learned $\boldsymbol{V}$ indicates that NTriPath could accurately identify newly introduced mutated genes as new member genes of 10th pathway.