SNeCT: Scalable Network Constrained Tucker Decomposition for Multi-Platform Data Profiling

Dongjin Choi and Lee Sael

Abstract—How do we integratively profile large-scale multi-platform genomic data that are high dimensional and sparse? Furthermore, how can we incorporate prior knowledge, such as the association between genes, in the analysis systematically to find better latent relationships? To solve this problem, we propose a **S**calable **Ne**twork **C**onstrained **T**ucker decomposition method (SNeCT). SNeCT adopts parallel stochastic gradient descent approach on the proposed parallelizable network constrained optimization function. SNeCT decomposition is applied to a tensor constructed from a large scale multi-platform multi-cohort cancer data, PanCan12, constrained on a network built from PathwayCommons database. The decomposed factor matrices are applied to stratify cancers, to search for top-k similar patients given a new patient, and to illustrate how the matrices can be used to identify significant genomic patterns in each patient. In the stratification test, combined twelve-cohort data is clustered to form thirteen subclasses. The similarity of the top-k patient to the query was high for 23 clinical features, including estrogen/progesterone receptor statuses of BRCA patients with average precision value ranges from 0.72 to 0.86 and from 0.68 to 0.86, respectively. We also illustrate how the factor matrices can be used for identifying significant patterns for each patient.
Resources are available at: https://github.com/leesael/GIFT

Index Terms—G.1.3.i Sparse, structured, and very large systems, G.1.6.a Constrained optimization, H.2.8.a Bioinformatics (genome or protein) databases

1 Introduction

Two of the major problems in cancer data analysis is stratification and clinical prediction. Stratification helps the researchers in understanding and exploring the genomic characteristics in relation to their current phenotypes and thus to recognize opportunities for clinical improvement on stratified groups of patients. There have been various works regarding stratification. In the perspective of personalized medicine, clinical predictions of an individual patient are needed and can be done by searching the integrated profile of a patient to existing records [1, 2]. However, not many works are done for patient profiling. Integrative profiling of multi-platform cancer data helps in better stratification and clinical predictions as analysis of multi-platform data, such as copy number variation (CNV), somatic mutation, gene expression, DNA methylation, and microRNA (miRNA) data, can provide more holistic view of a patient's biological status compared to using data from one or few platforms.

Related works in integrative cancer analysis

Need for integrative data analysis in cancer studies has been recognized. However, due to increased sized of data and a limited number of uniform data analysis framework, integrative analysis of multi-platform cancer data is still a challenging task. Existing methods are often limited in interpretability and scalability and often runs in a selected subset of data and features.

Previous integrative methods that run only on a small number of genes includes a work by [3] that has shown the effect of DNA (deoxyribonucleic acid) methylation and CNV (copy number variation) in gene expression of several known oncogenes for glioblastoma and ovarian cancer; PARADIGM method by [4] that has adopted graph inference approach on augmented pathway structure containing nodes for CNV, gene expression, protein expression and active protein information; and a work by [5] that has introduced an integrative statistical framework based on a sparse regression of gene expression values based on CNV, miRNA (Micro ribonucleic acid), and methylation.

Some of the integrative methods that have utilized only on small number of samples include a multiple-kernel based method by [6, 7, 8] that has combined kernels generated from individual platform data in a weighted linear fashion for stratification and predictions of ovarian cancer; a method by [9] that has applied multivariate Cox Lasso model and median time-to-event prediction algorithm on dataset integrated from the CNV, methylation, miRNA, and gene expression data; and iCluster method by [10] that has transformed the multi-platform data to latent space and then clustered the data on latent variable.

Also, many methods have ensembled the results of separately analyzed data for each platform. An ensemble approach is not a direct approach to integrative analysis problem. Methods in this category includes work by [11] that has evaluated the predictive power of patient survival and clinical outcome using clinical data in combination with one of CNV, methylation, mRNA (messenger ribonucleic acid), miRNA or protein expression data and work by [12]

- *D. Choi was with the Department of Computer Science and Engineering, Seoul National University, Seoul, Republic of Korea. He is currently with the Computational Science and Engineering, Georgia Institute of Technology, USA.*
- *L. Sael is with the Department of Computer Science and Engineering, Seoul National University, Seoul, Republic of Korea. E-mail: saellee@snu.ac.kr (corresponding author)*

Manuscript received XXX 00, 20XX; revised XXXX 00, 20XX.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2906205, IEEE/ACM Transactions on Computational Biology and Bioinformatics

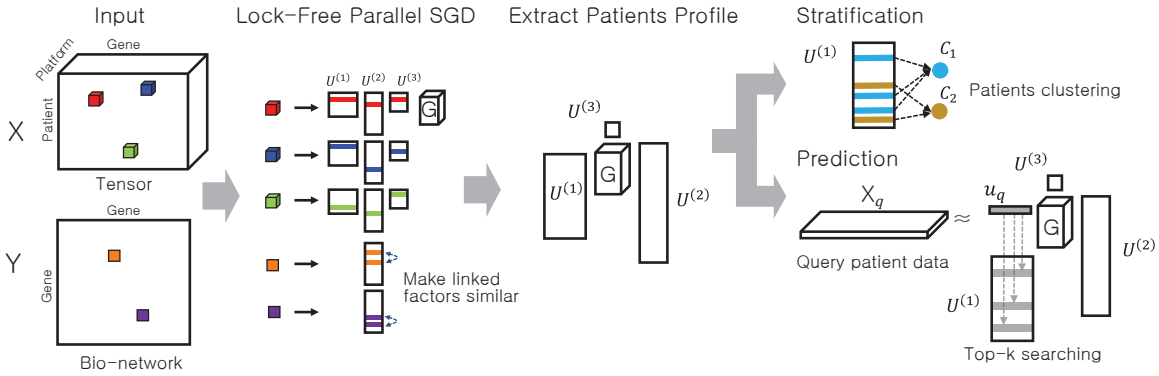

Fig. 1. Overview of the tensor decomposition of SNeCT and validation processes.

that have first clustered on individual data platforms, and then used the results of single-platform clusters as input to a second-level cluster analysis to form a cluster-of-cluster assignment (COCA).

Although work by [12] has utilized an indirect approach for integrative analysis, their work is significant in the aspect of multi-cancer analysis. The significance of multi-cancer analysis is that the analysis across multiple cancer types enables us to get a glimpse of the extent to which genomic signatures are shared across the different cancers. Also, the biological understanding of similarity and dissimilarity among the different cancer types can enable efficient management of diseases as well as treatment transfers between different cancer types of similar genomic signatures. The multi-platform multi-cancer data that was used by [12], i.e. the PanCan12 dataset, is still widely explored by many researchers.

Related works in tensor analysis

Tensors, i.e., multi-dimensional arrays, are a natural representation of multi-platform genomic data [13]. The core of tensor analysis is tensor decomposition, which can be considered as a higher-order singular value decomposition (HOSVD). Tensor analysis has been widely applied on various fields such as on network traffic [14], knowledge bases [15, 16], hyperlinks and anchor texts in the Web graphs [17], sensor streams [18, 19], and DBLP conference-author-keyword relations [20], and electronic heath record analysis [21] to name a few. Also, a few tensor analysis has been done on genome-wide experimental data, including analysis of gene expression analysis [22, 23] and analysis of bio-networks [24, 25]. However, tensor analysis has not yet gained popularity in genome data analysis compared to its low dimensional counterparts including singular-value decomposition (SVD) and non-negative matrix factorizations (NMF) due to computational challenges.

We identify two major challenges of tensor analysis in genome-wide multi-platform multi-cancer data analysis applications: data scalability and non-uniqueness in results. A scalable method is needed since there is an intermediate data explosion problem in the decomposition process even when the input tensor fits into the memory. There are several works addressing the scalability in a tensor analysis including our previous works [26, 27, 28, 29]. The non-uniqueness of results comes as characteristics of Tucker decomposition [1], which most genome based tensor analysis is based on.

Contributions

In this paper, we propose a tensor-based multi-platform data analysis method that enables stratification and clinical prediction of patients across multiple cancer types. The **S**calable **Ne**twork **C**onstrained **T**ucker decomposition method (SNeCT) (Figure 1) address the two tensor analysis challenges by utilizing parallel stochastic gradient descent (SGD) for scalably updating the factors and reducing the effect of non-uniqueness by imposing known bio-network as prior knowledge constraint. We also show that SNeCT can efficiently stratify cancer subtypes and predict clinical outcomes.

The contributions of this paper are listed in the following.

- Propose and develop a novel **S**calable **Ne**twork **C**onstrained **T**ucker decomposition algorithm (SNeCT).
- Perform stratification on multi-platform multi-cancer data and show similarities and differences between cancer types.
- Perform clinical prediction utilizing multi-platform genomic profiles.
- Provide a demonstration of individualized interpretation utilizing factor matrices.

2 Data Description

To test the effectiveness of SNeCT, we utilized the largest publicly available multi-platform genomic data of cancer patients, TCGA-Pancancer [31].

2.1 Tensor construction with the PanCan12

Initially, TCGA-Pancancer data freeze version 4 [31], created by The Cancer Genome Atlas (TCGA) Research Network [32], was downloaded from the Sage Bionetworks repository, Synapse [33]. The PanCan12 contains level 3 processed multi-platform data with mapped clinical information of patients group into cohorts of twelve cancer

1. CANDECOMP/PARAFAC tensor decomposition provides uniqueness, however, they perform poorly when dimensions are highly uneven [30]

type: bladder urothelial carcinoma (BLCA), breast adenocarcinoma (BRCA), colon and rectal carcinoma (COAD, READ), glioblastoma multiforme (GBM), head and neck squamous cell carcinoma (HNSC), kidney renal clear cell carcinoma (KIRC), acute myeloid leukaemia (LAML), lung adenocarcinoma (LUAD), lung squamous cell carcinoma (LUSC), ovarian serous carcinoma (OV), and uterine corpus endometrial carcinoma (UCEC). Table 1 lists the Synapse IDs of the downloaded data for each platform used.

After the download, miRNA probes were mapped to target genes using miRWalk2.0 dataset [34] and methylation probes of methylation platform were mapped to corresponding gene symbols using the Illumina Infinium HumanMethylation450K BeadChip information. Average values were taken if multiple probes map to a gene for both miRNA and methylation. For CNV, values were mapped to genes in the sequence range. No changes were made to gene expression values of the downloaded data. Each mutation information was mapped to genes and summarized in binary format to describe the existence of a mutation in each gene. For each data types, when there is a gene that has no mapping, the value of the gene is considered missing. After mapping the platform values to genes, subjects (patients) and genes that had less than two evidence were removed from the dataset.

Resulting data for each platform were further min-max normalized so that each entry is scale bounded between 0 and 1. Min-max normalization was chosen to scale bound while preserving the data distribution and on the presumption that downloaded level 3 data are already well preprocessed and normalized without extreme outliers.

A cell of resulting 3-mode tensor, $\mathcal{X}$, contains a floating point value indexed on $[patient, gene, platform]$, as shown in Figure 1. The first mode spans over 4,555 patients; the second mode spans over 14,351 genes; and the third mode spans over five different platforms.

TABLE 1
TCGA PAN Cancer (PanCan12) freeze 4.7 [31] and Synapse repository.

Platform	Input Data	# of Genes	# of Samples
P1. miRNA (mir)	syn2491366	14,345	4198
P2. Methylation (met)	syn2486658	1,383	4919
P3. Somatic CNV (cnv)	syn1710678	876	3260
P4. mRNA (gex)	syn1715755	14,178	3599
P5. Somatic SNV (mut)	syn1729383	14,351	4933

2.2 Network constraint formation with pathway data

Edges between genes in a gene-gene network were used for constraining the factor matrices towards existing knowledge of gene associations. The initial bio-network of human gene associations was retrieved from well-received PathwayCommons v.8 [2] [35, 36]. The initial bio-network was then used to construct adjacency matrix of the gene network for the list of gene considered in the tensor construction. The resulting adjacency matrix, $\mathbf{Y}$, contains 665,429 number of association information of 14,351 genes.

2. The choice of gene-gene network is not restricted to PathwayCommons. A simple construction of appropriate adjacency matrix based on the chosen network and replacement of the input matrix $\mathbf{Y}$ suffice in changing the network in SNeCT.

3 METHODS

In this section, descriptions of the basic notations and operations used in tensor analyses are provided followed by the derivation of the SNeCT algorithm.

3.1 Tensor basics

Table 2 shows the definitions of symbols used in this paper

TABLE 2
Table of symbols.

Symbol	Definition
$\mathcal{X}$	a tensor (boldface Euler script)
x_{ijk}	(ijk)-th entry of $\mathcal{X}$
$\mathbf{A}$	a matrix (uppercase, bold letter)
$\mathbf{a}_i$	the i-th row vector of $\mathbf{A}$ (lowercase, bold letter)
a_{ij}	(ij)-th entry of $\mathbf{A}$
$\times_n$	n-mode matrix product
$\|\bullet\|$	Frobenius norm
$*$	Hadamard product
$\circ$	Outer product
$\oslash$	Element-wise division
$\Omega_{\mathcal{X}}$	index set of $\mathcal{X}$
$\Omega_{\mathcal{X}}^{n,i}$	subset of $\Omega_{\mathcal{X}}$ having i as the n-th index
I_n	length of n-th dimension of input tensor $\mathcal{X}$
J_n	length of n-th dimension of core tensor $\mathcal{G}$

3.1.1 Tensors

A tensor is a generalization of a multi-dimensional array denoted by a boldface Euler script, e.g. $\mathcal{X}$. An N-mode tensor is denoted as $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, and $(i_1 i_2 \cdots i_N)$-th elements of $\mathcal{X}$ is denoted as $x_{i_1 i_2 \ldots i_N}$. A matrix is denoted by an uppercase bold letter, e.g. $\mathbf{A}$. The i-th row vector of $\mathbf{A}$ is denoted by $\mathbf{a}_i$ in lowercase bold letter, and the (ij)-th entry of $\mathbf{A}$ is denoted by a_{ij}. All tensor and matrix indices are positive integers greater than or equal to 1. The mode-n matrix product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $\mathbf{A} \in \mathbb{R}^{I_n \times K}$ is denoted by $\mathcal{X} \times_n \mathbf{A}$ and has the size of $I_1 \times \cdots I_{n-1} \times K \times I_{n+1} \cdots \times I_N$. The element-wise definition is as follows:

$$(\mathcal{X} \times_n \mathbf{A})_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} a_{j i_n}. \quad (1)$$

Please view [30] for more detailed explanations about tensor operations. Descriptions based on 3-mode tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ are given for the following sections since our dataset, PanCan12 tensor, is a 3-mode tensor.

3.1.2 Higher order singular value decomposition

After construction of tensors, they can be decomposed in several ways for analysis and profiling. We focus on higher order singular value decomposition (HOSVD). HOSVD, also known as Tucker decomposition, is the generalization of singular value decomposition (SVD), which works on matrices. HOSVD decomposes a tensor into a core tensor and orthogonal factor matrices corresponding to modes. Specifically,

given a 3-mode data tensor $\mathcal{X}$, HOSVD decomposes $\mathcal{X}$ as follows:

$$\mathcal{X} \approx \tilde{\mathcal{X}} = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)}, \tag{2}$$

where $\mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$ is a core tensor, and $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times J_n}$ denotes the factor matrices for the n-th dimensions, respectively. HOSVD finds the factors by minimizing the following objective function:

$$f = \frac{1}{2}\|\mathcal{X} - \tilde{\mathcal{X}}\|^2 + \frac{\lambda}{2} R(\mathcal{G}, \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}), \tag{3}$$

where $R(\mathcal{G}, \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)})$ is the L_2 regularization term. Performance comparison of HOSVD and our method is provided in the Analysis Section.

3.2 SNeCT decomposition

3.2.1 Addressing missing values

In experimental data, values can be missing for several reasons. They can be missing because particular variable was not measured, because measured but filtered due to inconsistence, or because they are not important. In a typical tensor decomposition, positions of missing values are filled with 0s. However, this degrades the accuracy significantly when there are a large number of such cases. That is, considering the cell corresponding to missing value fill with value 0, typical decomposition algorithms try to learn the factor vectors such that when reconstructed will result in 0, which is not true. Thus, it is important to be able to distinguish a missing value from the value 0.

In SNeCT, missing data is addressed by ignoring the missing data in the optimization. That is, factor values are learned using only the non-missing values by optimizing on reducing reconstruction error for the values we know in confidence (observed values). Rewriting the objective function in Equation (3) such that reconstruction error is calculated only at the indices of observed entries in input tensor $\mathcal{X}$ specified by $\Omega_{\mathcal{X}}$ is as follows.

$$f_r = \frac{1}{2} \sum_{(i_1 i_2 i_3) \in \Omega_{\mathcal{X}}} (x_{i_1 i_2 i_3} - \tilde{x}_{i_1 i_2 i_3})^2 + \frac{\lambda_r}{2}(\|\mathcal{G}\|^2 + \sum_{n=1}^{3} \|\mathbf{U}^{(n)}\|^2), \tag{4}$$

where $\tilde{x}_{i_1 i_2 i_3} = \mathcal{G} \times_1 \mathbf{u}^{(1)}_{i_1} \times_2 \mathbf{u}^{(2)}_{i_2} \times_3 \mathbf{u}^{(3)}_{i_3}$. Note that optimization function with reconstruction error calculated only at the observed entries has constantly been proven to be more accurate than otherwise in our previous studies [28, 37, 38] and will not be further discussed in this paper.

3.2.2 Objective function with network constraint

Network constraint can be enforced by introducing adjacency matrix of genes Y of network G in the objective function. Y informs the association between genes that composes the input tensor. That is, for our input tensor with three modes [patient, gene, platform], Y corresponds to the adjacency matrix of genes that composes the second mode of input tensor $\mathcal{X}$ and factor matrix $\mathbf{U^{(2)}}$. Based on the assumption that associated genes should have similar values, we aimed to constrain the factor values of gene k_1 and gene k_2 to be similar if they are associated, i.e., $y_{k_1,k_2} = 1$. To include the similarity constraint to HOSVD, we modified the optimization function of tucker decomposition to include the adjacency matrix Y in the regularization term. Specifically, we add the network regularization term $\lambda_g f_g$ to the objective function of Equation (4) as follows:

$$\begin{aligned} \lambda_g f_g &= \frac{\lambda_g}{2} \sum_{l=1}^{J_2} \Big[\sum_{(k_1 k_2) \in \Omega_G} y_{k_1 k_2} (u^{(2)}_{k_1 l} - u^{(2)}_{k_2 l})^2 \Big] \\ &= \frac{\lambda_g}{2} \sum_{(k_1 k_2) \in \Omega_G} y_{k_1 k_2} \|\mathbf{u}^{(2)}_{k_1} - \mathbf{u}^{(2)}_{k_2}\|^2 \end{aligned} \tag{5}$$

where Ω_G is the observed edge set of G and λ_g is a hyper-parameter controlling the strength of the regularization. Minimizing f_g guides the algorithm such that the factors of associated genes become similar, i.e., $\mathbf{u}^{(2)}_{k_1}$ and $\mathbf{u}^{(2)}_{k_2}$ have similar values when there is an edge between gene k_1 and gene k_2 in the network G. The f_g can be adapted to work for both symmetric and asymmetric gene networks unlike the previous works [39, 40].

3.2.3 Efficient parallelizable update rules

We present a multi-core algorithm to minimize the objective function $f_{opt} = \frac{1}{2} f_r + \frac{1}{2}\lambda_g f_g$ and factorize the given tensor $\mathcal{X}$ into HOSVD form. SNeCT adopts parallel stochastic gradient descent (SGD) optimization technique [41] and thus is highly memory-efficient and scalable to large datasets and multiple cores.

We reformulated f_r so that it is SGD-amenable as follows:

$$\begin{aligned} f_r = \frac{1}{2} \sum_{(i_1 i_2 i_3) \in \Omega_{\mathcal{X}}} \Big[&\big(x_{i_1 i_2 i_3} - \tilde{x}_{i_1 i_2 i_3}\big)^2 \\ &+ \frac{\lambda_r}{|\Omega_{\mathcal{X}}|}\|\mathcal{G}\|^2 + \lambda_r \sum_{n=1}^{3} \frac{\|\mathbf{u}^{(n)}_{i_n}\|^2}{|\Omega^{n,i_n}_{\mathcal{X}}|} \Big], \end{aligned} \tag{6}$$

where, $\Omega^{n,i_n}_{\mathcal{X}}$ is the subset of $\Omega_{\mathcal{X}}$ with i_n as the n-th index.

Equation (5) was used without modification since f_g is already in a SGD-amenable form.

With the SGD-amenable objective function f_{opt}, we solved for gradients to derive the Parallelizable update rule. That is, the gradients of $f_{opt} = \frac{1}{2} f_r + \frac{1}{2}\lambda_g f_g$ with respect to factors for a given data point $x_{\alpha=(i_1 i_2 i_3)}$ is calculated as follows:

$$\begin{aligned} \left.\frac{\partial f_{opt}}{\partial \mathbf{u}^{(1)}_{i_1}}\right|_{\alpha} &= -\big(x_\alpha - \tilde{x}_\alpha\big)\big[\mathcal{G} \times_2 \mathbf{u}^{(2)}_{(i_2)} \times_3 \mathbf{u}^{(3)}_{(i_3)}\big] + \frac{\lambda}{|\Omega^{1,i_1}_{\mathcal{X}}|}\mathbf{u}^{(1)}_{i_1}, \\ \left.\frac{\partial f_{opt}}{\partial \mathcal{G}}\right|_{\alpha} &= -\big(x_\alpha - \tilde{x}_\alpha\big) \times_1 \mathbf{u}^{(1)\mathsf{T}}_{i_1} \times_2 \mathbf{u}^{(2)\mathsf{T}}_{i_2} \times_3 \mathbf{u}^{(3)\mathsf{T}}_{i_3} + \frac{\lambda}{|\Omega_{\mathcal{X}}|}\mathcal{G}. \end{aligned} \tag{7}$$

$\left.\frac{\partial f_{opt}}{\partial \mathbf{u}^{(2)}_{i_2}}\right|_{\alpha}$ and $\left.\frac{\partial f_{opt}}{\partial \mathbf{u}^{(3)}_{i_3}}\right|_{\alpha}$ are calculated symmetrically as the above equations.

Gradients of f_{opt} with respects to $y_{\beta=(k_1 k_2)}$ are calculated as follows:

$$\left.\frac{\partial f_{opt}}{\partial \mathbf{u}^{(2)}_{k_1}}\right|_{\beta} = \lambda_g y_\beta(\mathbf{u}^{(2)}_{k_1} - \mathbf{u}^{(2)}_{k_2}), \quad \left.\frac{\partial f_{opt}}{\partial \mathbf{u}^{(2)}_{k_2}}\right|_{\beta} = \lambda_g y_\beta(\mathbf{u}^{(2)}_{k_2} - \mathbf{u}^{(2)}_{k_1}) \tag{8}$$

The above equations are naturally generalizable to mode-N tensors.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2906205, IEEE/ACM Transactions on Computational Biology and Bioinformatics

Algorithm 1 SNeCT

Require: Input data: tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, network matrix $\mathbf{Y} \in \mathbb{R}^{I_c \times K}$, number of parallel cores P, and network-constrained mode c
Hyperparameters: core size $(J_1, J_2, \cdots, J_N)$, learning rate η, and regularization factors λ and λ_g
Ensure: Core tensor $\mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$, and factor matrices $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \cdots, \mathbf{U}^{(N)}$
1: Initialize $\mathcal{G}$, $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times J_n}$ for $n = 1, 2, \cdots, N$ randomly
2: **repeat**
3: **for** $\forall x_{i_1 i_2 \cdots i_N} \in \mathcal{X}$, $\forall y_{k_1 k_2} \in \mathbf{Y}$ in random order **do in parallel**
4: **if** $x_{i_1 i_2 \cdots i_N} \in \mathcal{X}$ is selected **then**
5: $\mathcal{D} \leftarrow \mathcal{G} * (\mathbf{u}^{(1)}_{i_1} \circ \mathbf{u}^{(2)}_{i_2} \circ \cdots \circ \mathbf{u}^{(N)}_{i_N})$ ▷ Cache intermediate data tensor
6: $\tilde{x}_{i_1 i_2 \cdots i_N} \leftarrow \sum_{j_1} \sum_{j_2} \cdots \sum_{j_N} d_{j_1 j_2 \cdots j_N}$ ▷ Calculate $\tilde{x}_{i_1 i_2 \cdots i_N}$ by summing all element of $\mathcal{D}$
7: **for** $n = 1, 2, \cdots, N$ **do**
8: $\mathbf{u}^{(n)}_{i_n} \leftarrow \mathbf{u}^{(n)}_{i_n} - \eta\big((\tilde{x}_{i_1 i_2 \cdots i_N} - x_{i_1 i_2 \cdots i_N}) \cdot Collapse(\mathcal{D}, n) + \frac{\lambda}{|\Omega^{n, i_n}_{\mathcal{X}}|} \mathbf{u}^{(n)}_{i_n}\big)$ ▷ Update corresponding factor rows
9: **end for**
10: $\mathcal{G} \leftarrow \mathcal{G} - \eta P\big((\tilde{x}_{i_1 i_2 \cdots i_N} - x_{i_1 i_2 \cdots i_N}) \cdot \mathcal{D} \oslash \mathcal{G} + \frac{\lambda}{|\Omega_{\mathcal{X}}|} \mathcal{G}\big)$ ▷ Update core tensor (**executed by only one core**)
11: **end if**
12: **if** $y_{k_1 k_2} \in \mathbf{Y}$ is picked **then**
13: $\mathbf{u}^{(c)}_{k_1} \leftarrow \mathbf{u}^{(c)}_{k_1} - \eta \lambda_g y_{k_1 k_2} (\mathbf{u}^{(c)}_{k_1} - \mathbf{u}^{(c)}_{k_2})$ ▷ Update network-constrained factors
14: $\mathbf{u}^{(c)}_{k_2} \leftarrow \mathbf{u}^{(c)}_{k_2} - \eta \lambda_g y_{k_1 k_2} (\mathbf{u}^{(c)}_{k_2} - \mathbf{u}^{(c)}_{k_1})$
15: **end if**
16: **end for**
17: **until** convergence conditions are satisfied
18: $\mathbf{Q}^{(n)}, \mathbf{R}^{(n)} \leftarrow$ QR decomposition of $\mathbf{U}^{(n)}$ ▷ Apply QR decomposition to enforce orthogonality
19: **for** $n = 1, 2, \cdots, N$ **do**
20: $\mathbf{U}^{(n)} \leftarrow \mathbf{Q}^{(n)}$, $\mathcal{G} \leftarrow \mathcal{G} \times_n \mathbf{R}^{(n)}$,
21: **end for**
22: **return** $\mathcal{G}$, $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \cdots, \mathbf{U}^{(N)}$

Calculations of the $\tilde{x}$ and gradients of the tensor components include redundant computations. We have previously observed that introduction of intermediate data $\mathcal{D}$ reduces redundancy in computation [42]. $\mathcal{D} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$ for a tensor entry x_{i_1, i_2, i_3} is defined as follows:

$$\mathcal{D} \leftarrow \mathcal{G} * (\mathbf{u}^{(1)}_{i_1} \circ \mathbf{u}^{(2)}_{i_2} \circ \cdots \circ \mathbf{u}^{(N)}_{i_N}). \tag{9}$$

Utilizing the intermediate data $\mathcal{D}$, $\tilde{x}_{i_1 i_2 i_3}$ can be calculated by sum of all elements of $\mathcal{D}$ and gradients for the tensor components can be calculated as follows [42]:

$$\mathbf{u}^{(n)}_{i_n} \leftarrow \mathbf{u}^{(n)}_{i_n} - \eta\big((\tilde{x}_{i_1 i_2 \cdots i_N} - x_{i_1 i_2 \cdots i_N}) \cdot Collapse(\mathcal{D}, n) + \frac{\lambda}{|\Omega^{n, i_n}_{\mathcal{X}}|} \mathbf{u}^{(n)}_{i_n}\big), \tag{10}$$

for $n = 1, 2, 3$ and where $Collapse(\mathcal{D}, n)$ outputs a vector with length of i_n which contains the sum of k-th slice of D over n-th mode as its k-th element.

$$\mathcal{G} \leftarrow \mathcal{G} - \eta P\big((\tilde{x}_{i_1 i_2 \cdots i_N} - x_{i_1 i_2 \cdots i_N}) \cdot \mathcal{D} \oslash \mathcal{G} + \frac{\lambda}{|\Omega_{\mathcal{X}}|} \mathcal{G}\big) \tag{11}$$

where $\oslash$ is element-wise division. The reformulation of the gradients makes the computation $O(N)$ time faster [42].

3.2.4 Parallel learning model

SNeCT optimizes the objective function f_{opt} by lock-free parallel SGD update similar to HOGWILD! [43] using the reformulated gradients and intermediate data tensor. A possible complication with the lock-free form is when there are frequent memory conflicts, e.g., when a factor row or core tensor is accessed by multiple parallel update attempts. However, a lock-free form of parallel SGD is known to converge when memory conflict is rare [43]. Our previous study showed that allocation of one designated core for core tensor update computation while allowing other factor values, i.e. $\mathbf{u}^{(n)}_{i_n}$, in a lock-free form minimizes the memory conflict in Tucker decompositions allowing for near linear convergence [42]. SNeCT also assigns a single core for core tensor update thus SNeCT guarantees near-linear convergence to a local optimum.

Algorithm 1 shows detailed procedures of decomposition of a general N-mode tensor $\mathcal{X}$ and network constraint $\mathbf{Y}$ which represents the similarity of c-th mode entities. In the beginning, SNeCT initializes $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$, $\cdots$, $\mathbf{U}^{(N)}$, and $\mathcal{G}$ randomly (line 1 of Algorithm 1). The outer loop (lines 2-17) repeats until the factors converge. In the inner loop (lines 3-16), SNeCT conducts parallel updates of factor rows corresponding to each data point $x_{i_1 i_2 i_3}$ or $y_{j_1 j_2}$ in random order. SNeCT reduces the time cost efficiently by caching intermediate data tensor $\mathcal{D}$ (line 5). $\mathcal{D}$ is used to compute the approximation of tensor cell (line 6), update the factor rows (line 8), and update the core tensor (line 10). In line 10, element-wise division operator $\oslash$ is used to efficiently calculate core tensor gradient. Also, frequent conflicts are removed by updating core tensor using only one core (line 10). Lines 12 and 13 updates factor rows for network constrained mode, $U^{(c)}$ in the tensor. The last step of SNeCT (lines 18-21) is to QR decomposes all factor matrices $\mathbf{U}$ and updates the result to ensures factor matrixes are column-wise orthogonal.

3.3 Model selection

The hyper-parameters involved in SNeCT are the core size $[J_1, J_2, J_3]$, initial learning rate η_0, reconstruction error

weight λ_r, and graph constraint weight λ_g. We selected hyper-parameters that resulted in the best training and testing reconstruction error via the random grid search method on a subset of patient samples of size 142. The subset of data, not the whole PanCan12 data, was used to reduce the computational burden in model selection.

We carefully selected the validation subset such that the selected represents the original PanCan12 tensor distribution. To do this, we first decomposed the PanCan12 tensor with small core size [$J_1 = 20$, $J_2 = 20$, $J_3 = 5$] and used the rows of U_1 factor matrix to cluster the samples via hierarchical clustering and selected a reasonable cutoff point resulting in 142 clusters. Then one random patient sample was selected per cluster to generate the validation tensor of size $142 \times 14,351 \times 5$.

The training set and the testing set was randomly selected among cells of subset tensor with the ratio of 9:1, respectively. Training error and testing error was calculated for each hyper-parameter combination. The hyper-parameter set that generated the lowest testing error was selected. The selected hyper-parameters are as follows: the core size [$J_1 = 78$, $J_2 = 48$, $J_3 = 5$], the initial learning rate $\eta_0 = 0.02$, and the regularization factors $\lambda_r = 0.1$, and $\lambda_g = 0.1$.

3.4 SNeCT performance evaluation

Evaluation of SNeCT has three parts: comparing SNeCT with existing Tucker decomposition methods, evaluating the quality of multi-platform patient profile on stratification and clinical prediction, applying factors to explaining genomic landscape of individual patient. The cancer stratification test included the analysis of the effectiveness of network constraint and clinical prediction contains the top-k search using generated patient profile. Details of patient profiles, stratification, and top-k search follow.

3.4.1 Multi-platform patient profile

The multi-platform patient profile is simply the rows of patient factor matrix. That is, in our decomposed PanCan12 tensor as shown in 1, a patient profile is a row in $U^{(1)}$ that corresponds to the patient. No additional normalization of is necessary since factor matrix $U^{(1)}$ is already column-wise orthonormal.

3.4.2 Stratification

Stratification of cancer patients was done by k-means clustering using the Euclidean distance on patient profiles. Several distance measures were tested, including Euclidean, cosine and Mahalanobis distance, however, we have found no significant differences in the results.

Cluster size was selected based on the gap statistics introduced by [44]. Gap statistics formalizes on the widely applied "elbow" method for selecting cluster sizes. In gap statistics, a number of clusters is selected based on the one that maximizes the gap statistics or based on when the increase of the gap statistic begins to slow down [44].

3.4.3 Top-k search

When a new query patient q arrives with data $\mathcal{X}_q$ which is a tensor representing the patient's genomic data, q, we need to first generate the multi-platform profile of the query patient. This was done by finding the profile for the patient using the pre-calculated factor matrices and core tensor. The following equation was solved with SNeCT algorithm to find the patient profile.

$$\mathbf{u}_q = \arg\min_{\mathbf{u}} \|\mathcal{X}_q - \mathcal{G} \times_1 \mathbf{u} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)}\|. \quad (12)$$

After generation of new profile, $\mathbf{u}_q$, it was used to seek top-k similar patients by calculating the distance between the query profile and patient profiles encoded in the patient factor matrix $\mathbf{U}^{(1)}$. It takes 550 ms on average to search for a query against the training set.

4 ANALYSES

The performance of SNeCT was evaluated in various perspectives listed in the following:

- Evaluated the computational complexity and scalability in comparison with existing Tucker tensor decomposition methods.
- Performed cancer stratification using the multi-platform patient profiles generated with SNeCT. In the stratification test, we compared the effectiveness of network constraint on survival analysis. We compared stratification results of SNeCT with Similarity Network Fusion (SNF) [8]. We also made an empirical examination of the quality of the stratified clusters to validate the quality of patient profiles.
- Performed similar patient searches for clinical predictions with patient profiles. We performed top-k searches and measured precision of clinical feature values of the searched results.
- Showed how the factors are effective in extracting a coarse-grained but a holistic view of a patient's genomic landscape.

4.1 Tensor decomposition performance comparison

We compared SNeCT with two Tucker tensor decomposition methods: a method by [39] and alternating least squares implementation in MATLAB tensor toolbox version 2.6 [45]. The method by [39] is a network constraint tensor decomposition approach and, to the best of our knowledge, it is the only network contained decomposition method existing in the literature. Also, a naive application of Tensor toolbox without the network constraining term (naive Tucker) provides the bases for a computationally optimized tensor decomposition.

4.1.1 Scalability comparison

We evaluated data scalability of SNeCT decomposition and comparing them to [39] and naive Tucker decomposition [45]. In the scalability test, running time per iteration was calculated because time complexity of [39] is too high, i.e., the method did not converge to a feasible optimum after 3 days on the original data. (Detailed comparison of the complexity of SNeCT and method by [39] follows in later section.) Also, we created sampled datasets by randomly selecting part of patients and platforms with a certain ratio to verify the efficiency of SNeCT on 'big data' scenario.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2906205, IEEE/ACM Transactions on Computational Biology and Bioinformatics

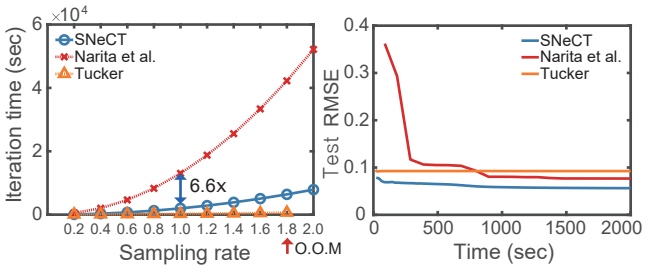

Fig. 2. Running time per iteration (A) and convergence (B) of SNeCT, [39] and naive Tucker decomposition.

Figure 2A shows the running time of three methods per iteration. The naive Tucker decomposition was the fastest among three methods since the decomposition is computationally optimized via batch matrix operations. However, the naive Tucker decomposition suffered from high memory requirement (*M-bottleneck problem* [46]) and failed to run after sampling rate of 2.0 showing O.O.M (out of memory) error (red arrow in Figure 2A). Compared to method by [39], running time per iteration of SNeCT (1974s) was 6.6× faster than that of [39] (13036s) on PanCan12 tensor. This shows that our proposed SNeCT is a data scalable and efficient network constrained Tucker decomposition method.

4.1.2 Reconstruction error comparison

Not only the running time but also the accuracy and convergence property of decomposition are critical parts of the performance of our method. The training and the testing RMSEs of SNeCT for the whole PanCan12 tensor are provided in the Figure 3.

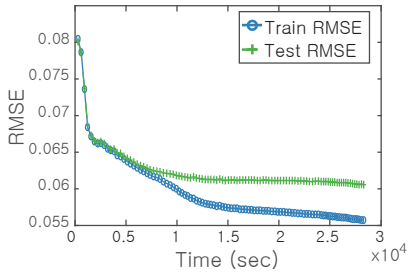

Fig. 3. Test and training RMSE of SNeCT with graph constraint of $\lambda = 1$ over total running time on PanCan12 tensor.

For accuracy comparison, we computed the testing set reconstruction RMSE the three methods on a reduced dataset. Reduced dataset used due to high time complexity of method by [39] (Table 3). The reduced set consists of 10% of randomly chosen patients and 10% of randomly chosen genes. We divided the input tensor entry set into the training set and testing set with a ratio of 9:1. Figure 2(B) shows the convergence property of three methods. A lower test RMSE for the same time means faster convergence and higher accuracy. As shown in the figure, SNeCT converged to a better point within a much fewer second while method by [39] and the naive Tucker decomposition failed to converge to feasible optimum points within the observed time range.

4.1.3 Computational complexity analysis

Table 3 summarizes the analytical complexity comparison between SNeCT and [39]. SNeCT outperforms the existing method in terms of both time complexity and memory usage. SNeCT achieves low time complexity by direct parallelization of SGD updates while caching intermediate data tensor and low memory usage by avoiding batch matrix calculation used by [39]. On the other hand, [39] follows a gradient descent approach which is hardly parallelizable and takes high memory requirement due to the batch calculation of large matrices for gradients.

TABLE 3
Comparison of time complexity (per iteration) and memory usage of SNeCT with existing network-regularized HOSVD algorithm of [39]. SNeCT shows lower time complexity and memory usage. For simplicity, we assume that data tensor $\mathcal{X}$ has an order of N, all modes are of size I, of rank J, and one mode has network constraint. P is the number of parallel cores.

	Time complexity (per iter.)	**Memory usage**
SNeCT	$\mathcal{O}(\lvert\Omega_{\mathcal{X}}\rvert J^N N/P + \lvert\Omega_{\mathbf{Y}}\rvert J/P)$	$\mathcal{O}(J^N P)$
[39]	$\mathcal{O}(\lvert\Omega_{\mathcal{X}}\rvert J^N N^2 + \lvert\Omega_{\mathbf{Y}}\rvert J)$	$\mathcal{O}(J^{N-1} I)$

4.2 Stratification

We performed stratification test on the PanCan12 data to show that the proposed SNeCT method reduces the high dimensional multi-platform cancer genomic data well, thus generating patient profiles that can be used for clinical predictions.

4.2.1 Effects of multi-platform multi-cancer analysis compared with a competing method

TABLE 4
Log-rank values of multi-platform data analysis. SNeCT-1 is SNeCT on individual cancer types. SNeCT-12 is SNeCT on all 12 cancer types. Single-platform data contains the average log-rank values of single-platform data analysis result detailed in Supplemental Material Table II.

		Multi-platform			Single-platform	
Ctype	Size	SNF	SNeCT-1	SNeCT-12	SNF	SNeCT-1
BLCA	126	1.5	2.8	**21.0**	3.8	4.8
BRCA	889	2.7	4.8	**12.6**	5.3	5.5
COAD	419	7.0	1.6	**9.2**	2.4	4.0
GBM	267	7.2	0.9	5.4	**8.0**	5.1
HNSC	310	**12.4**	5.4	5.9	8.3	4.4
KIRC	498	4.6	2.6	2.8	**6.5**	**6.5**
LAML	197	2.4	**5.2**	0.9	3.3	2.2
LUAD	357	**13.6**	4.7	1.7	4.2	5.5
LUSC	340	0.1	2.9	**7.7**	2.9	3.9
OV	485	2.3	2.4	4.7	**8.9**	2.7
READ	163	3.7	6.7	**7.2**	3.6	3.2
UCEC	499	9.4	1.3	**16.8**	8.1	8.1
Sum	4550	67.0	41.4	**95.8**	65.4	56.0

Before going into details of cluster analysis performed using multi-platform multi-cancer PanCan12 data, we answer two questions on whether multi-platform analysis and multi-cancer analysis improves stratification performance evaluated by log-rank values obtained from survival analysis of each cluster. Survival analysis was acquired using

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2906205, IEEE/ACM Transactions on Computational Biology and Bioinformatics

the Cox proportional hazards regression model in the R survival package (Fig 5). We used right-censored survival data for patients: days-to-death for deceased patients and days-to-last contact for living patients as right-censored data. We also compared SNeCT with SNF (Similarity network fusion) [8], a well known multi-platform stratification method. Since SNF is not built for large scale analysis, we first divided the PanCan12 dataset by their tissue of origin (ctype) and then performed clustering on each of the divided datasets. Since no correct number of subgroups of each cancer types are known, we unbiasedly chose cluster size of five and performed survival analysis on each cluster.

Does the integrative analysis of multi-platform data improve stratification? Table 4 show the log-rank values of multi-platform data analysis for SNF and SNeCT compared with the best log-rank values of single-platform data analysis across SNF and SNeCT results. Details of the single-platform analysis is provided in the Supplemental Material Table II. We can see that for 9 out of 12 cancer types, multi-platform analysis improved the log-rank values compared to single-platform analysis.

Does analysis across multiple cancer types (multi-cancer) improve stratification? Comparing SNeCT-1 and SNeCT-12, we can see that across cancer analysis improved the log-rank values of for 10 out of 12 cancer types. Also, multi-cancer analysis (SNeCT-12) had higher log-rank values for 7 out of 12 cases compared to SNF.

4.2.2 Multi-platform multi-cancer cluster assignment

TABLE 5
12 pathological disease types assigned to clusters of profiles factorized via SNeCT with graph constraint of $\lambda = 1.0$.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	Total
BLCA	16	32	2	19	0	22	3	0	0	0	32	0	0	126
BRCA	17	3	600	172	1	70	0	0	0	0	26	0	0	889
COAD	4	0	2	2	0	91	317	0	0	0	1	2	0	419
GBM	4	1	1	2	3	7	0	0	248	0	1	0	0	267
HNSC	0	242	1	6	0	1	0	0	0	0	60	0	0	310
KIRC	14	1	1	0	471	4	0	0	1	0	6	0	0	498
LAML	0	0	0	0	0	9	0	0	0	188	0	0	0	197
LUAD	302	2	2	7	1	12	0	0	0	0	29	0	0	357
LUSC	26	32	0	29	0	7	0	0	0	0	246	0	0	340
OV	0	0	1	3	0	1	1	348	0	0	0	0	131	485
READ	1	1	0	5	0	9	145	0	0	0	1	1	0	163
UCEC	3	1	3	117	1	348	1	0	0	0	10	13	2	499
Total	387	315	613	362	477	581	467	348	249	188	412	17	134	4550

We evaluated the gap statistics to select the cluster size for k-means clustering of patient profiles. The rate of increase in the gap statistics slows down after a cluster size of 10. Details of the gap statistics result are provided in the Supplemental Material Fig. 1. For the convenience of comparison, we stratified patient profiles into 13 clusters similar to the cluster-of-cluster assignments (COCA) analysis by Hoadley et al. [12]. Table 5 shows the clustering result mapping the tissue-of-origin to each cluster. A strong correlation between the tissue-of-origin to each cluster is observed which is similar to the COCA result. The weighted average of Jaccard similarity between the best matching COCA clusters and SNeCT clusters is 0.75 (details provided in Supplemental Material Table I). We also plotted the PCA (principle component analysis) scatter plot of the patient profile generated by SNeCT using the multi-platform multi-cancer PanCan12 dataset (Fig. 4).

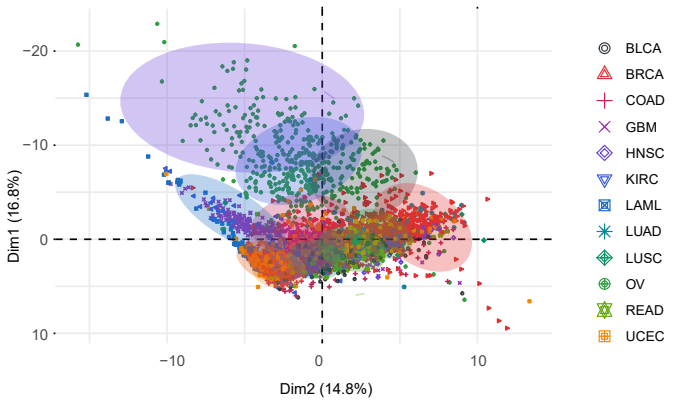

Fig. 4. PCA scatter plot of patient profile generated by SNeCT with graph constraint of $\lambda = 1.0$. Ellipses denote thirteen cluster locations. Points are patient samples color-coded by the type of cancer.

4.2.3 Effect of network constraint on survival analysis

We imposed three different levels of network regularization to determine the effect of network constraints on decomposition results: λ_g value of 0 (not constrained), 0.1, and 1. Resulting patient profiles were each clustered using k-mean with $k = 13$ and survival analysis performed on each clusters. Again the survival analysis for the thirteen clusters was acquired using the Cox proportional hazards regression model again using right-censored survival data for patients. Survival analysis and the log-rank statistics values in Figure 5 shows that, although there is little difference between two weight values, graph constraint produces better clustering results.

Cluster result of the patient factor matrix, $\mathbf{U}^{(1)}$, was highly correlated with the tissue-of-origin as was observed by Hoadley et al. [12]. Six clusters, C1-LUAD-enriched, C3-BRCA/Luminal, C5-KIRC, C8-OV-1, C9-GBM, C10-LAML, and C13-OV-2 each dominantly composed of cancer samples from a single tissue-of-origin. C1-LUAD-enriched cluster included 302 out of 357 LUAD patients with relatively good prognosis, that is, neoplasm cancer status is tumor free for 186 out of all 217 tumor free cases with a precision of 0.73 (recall of 0.85). C3-BRCA/Luminal cluster grouped 600 BRCA cases with 13 other cancer types. The BRCA patients in C3 had positive estrogen and progesterone receptor status with a precision of 0.95 and 0.85, respectively, and HER2 status was mixed tending to have more negative status with a precision of 0.73. Furthermore, C3 contained 8 out of 9 metastatic cases and contained 34 out of 43 cases with known other malignancy histological type. It tells us that C3 grouped patients with BRCA Luminal A and Luminal B molecular subtypes of breast cancer. Four other clusters that form somewhat mutually exclusive collectively exhaustive groups are C5-KIRC that contained 471 patients classified as KIRC , C9-GBM that contained 248 cases of GBM patients , C10-LAML that contained 188 cases of LAML , and C12-UCEC-small, a small cluster, that contained 13 UCEC cases with very high survival ratio as shown in Supplemental Material Fig. 2.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2906205, IEEE/ACM Transactions on Computational Biology and Bioinformatics

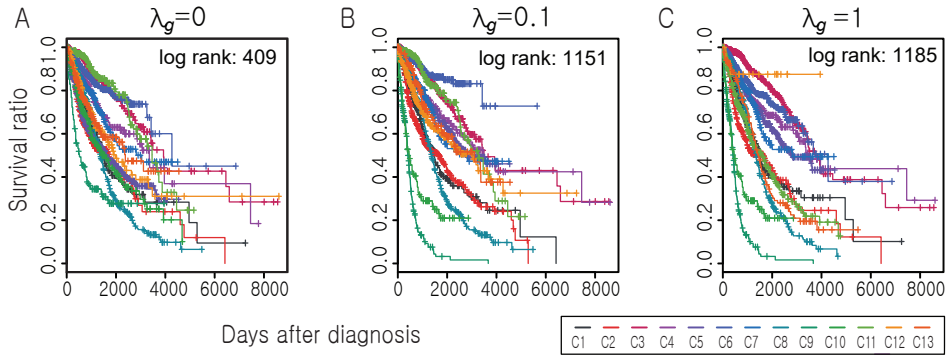

Fig. 5. Predicted survival curves for clustered patients. X-axis is survival time (days after diagnosis) and Y-axis is survival rate.

TABLE 6
Top-k search precisions and a recall (Top-R).

Cohort	Clinical Features	Top1	Top5	Top10	TopR
BRCA	estrogen receptor status	0.72	0.85	0.86	0.81
	progesterone receptor status	0.86	0.71	0.71	0.68
	her2/neu IHC receptor status	0.53	0.51	0.49	0.55
	neoplasm cancer status	0.84	0.80	0.81	0.77
COAD	braf gene analysis result	1.00	0.80	0.70	0.92
	colon polyps present	0.47	0.56	0.52	0.59
	1st relatives with cancer	0.84	0.78	0.75	0.84
	venous invasion	0.61	0.60	0.63	0.72
GBM	histological type	0.96	0.94	0.94	0.78
	icd-o-3 histology	1.00	1.00	1.00	0.77
	neoplasm cancer status	0.85	0.82	0.83	0.77
HNSC	hpv status by p16 testing	0.78	0.78	0.77	0.73
KIRC	histological type	1.00	0.99	0.99	0.73
	icd-o-3 histology	1.00	0.99	0.99	0.73
	number packs/year smoked	0.50	0.30	0.20	1.00
LAML	calgb cytogenetics risk cat.	0.85	0.84	0.81	0.65
OV	neoplasm histologic grade	0.79	0.75	0.76	0.77
READ	braf gene analysis result	1.00	1.00	1.00	1.00
	1st relatives with cancer	0.93	0.77	0.78	0.86
UCEC	menopause status	0.71	0.76	0.76	0.77
	neoplasm cancer status	0.83	0.75	0.77	0.77

4.3 Clinical predictions

One of significant utility of SNeCT is in the generation of multi-omics patient profiles for genome base similar patients search. Clinical information of similar patients in existing database to a new patient can be used for clinical predictions of the new patient.

4.3.1 *Clinical similarity of top-k search*

Clinical prediction performance of SNeCT was evaluated using 10% of the PanCan12 data as the test set. That is, we generated the factor matrices from 90% of the data and used 10% of the data as test set or new queries. Then we search each query patient (testing set) to the patients in the database (training set) using the patient factor matrix $\mathbf{U}^{(1)}$ and selected top-k similar patients to the query.

Clinical similarities of each patient to the top-k patients was measured by determining whether the top-k patients had the same clinical labels as the query patients. Seventeen clinical features, fifteen of which are listed in the second column of Table 6, was tested and evaluated when available. Numerical values of clinical terms were converted to categorical labels prior to the analysis. More specifically, "age at initial pathogenic diagnosis" where converted to two labels, i.e, older than 50 and less or equal to 50; "began age smoking in years" converted to four labels, i.e., less than 10, 10 to 20, 20 to 30, and more than 30; and "number of pack years smoked" were converted to five labels, i.e, less than 13 a year, 13 to 25 a year, 25 to 53 a year, 53 to 105 a year, and more than 105 a year.

We calculated the average precision over test cases for each of the seventeen clinical features on top-1, top-5, and top-10 similar patients. Only precision values are calculated for top-1, 5, and 10 as the number of retrievals are fixed. To evaluate the recall, we calculated R-precision (top-R) [3]. R-precision computes the precision over R number of retrievals where R value computes the number of samples with the same clinical values as the query in the database and varies from query to query.

Overall, "age at initial pathologic diagnosis" and "vital status" coincided well with all top-k retrievals with average precision over all the test data ranging from 0.76 to 0.81 and from 0.66 to 0.68, respectively. No significant features were found for LUAD, LUSC, and BLCA other than the two clinical features mentioned above. Other clinical features that are cohort-specific or have high average precision values are listed in Table 6. Looking at the precision values, we can see that the search successfully retrieved BRCA patients with similar estrogen and progesterone receptor status in most cases while less so in terms of her2/neu IHC receptor status. Also, most search results match that of the query for the braf gene analysis results in the COAD and READ test cases.

4.4 Holistic view of individual's genome: an example

One of the unique aspects of the tensor factorization result is on the capability to interpret each patient based on the learned latent factors. To illustrate how factor matrices can be used interpret individuals' genome landscape, we provide a brief example of a given patient i. For the patient i, SNeCT generates patient profile $\mathbf{u}_i^{(1)}$. If the patient is a new patient we can use the Eq. 12 to generate the profile. We then calculate the personalized subtype matrix as follows: $\mathcal{S} = \mathcal{G} \times_1 \mathbf{u}_i^{(1)}$ ($\in \mathbb{R}^{J_2 \times J_3}$). $\mathcal{S}$ provides a personalized weight information for subtypes for the gene and the platform

3. The precision at R equals the recall by definition.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2906205, IEEE/ACM Transactions on Computational Biology and Bioinformatics

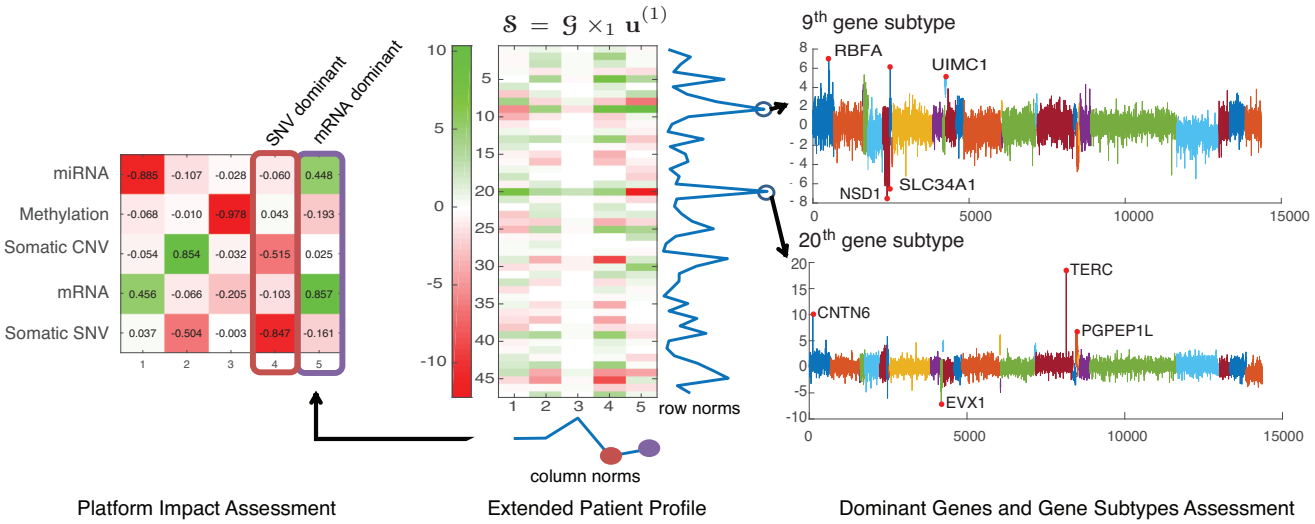

Fig. 6. A holistic view of genomic landscape of a patient. Visualization of personalized subtype matrix $\mathcal{S}$ matrix of patient ID TCGA-BS-A0UV in cohort UCEC assigned to cluster C12.

modes. For example, the center of Fig. 6 shows the heatmap of $\mathcal{S}$ for the sample patient TCGA-BS-A0UV. Each row of $\mathcal{S}$ represents a subtype for gene mode, thus norm of each row represents the influence of each subtype to the patient. Each column of $\mathcal{S} \times_3 \mathbf{U}^{(3)}$ represents the platform mode. The norm of each column shows the influence of each platform to the patient and right side of Fig. 6 shows the associated gene factor values. For example, sample patient TCGA-BS-A0UV is significantly associated with the 9^{th} and 20^{th} gene subtypes. In the 9^{th} subtype, genes RBFA, UIMC1, NSD1, and SLC34A1, were significant; and in 20^{th} subtype, genes CNTN6, EVX1, TERC, and PGPEP1L were significant. The patterns of the factor value are distinct as can be shown in the gene factor value distribution that is color-coded by the gene factor clusters. Further gene enrichment analysis can be made to the gene cluster of interest to identify which functional process distinguishes a patient. We have performed k-means clustering (k=30) using the gene factor matrix and performed KEGG enrichment analysis using ClusterProfiler [47] package in BioConductor. The enriched KEGG pathway terms for each group is provided in the Supplemental Material Table IV. With the analysis, we can determine which gene subtypes dominantly characterize the patient and which platform data were important in finding the dominant characteristics such that we can trace back to the significant genes and functions.

5 Discussion and Conclusions

In this paper, we have proposed a large-scale network constrained Tucker decomposition method (SNeCT) that is based on parallelizable stochastic gradient descent. With SNeCT, it is possible to systematically analyze high dimensional multi-platform genomic data constrained on prior knowledge of feature associations in a form of a network. It is a general purpose approach that can be applied in various combinations of multi-platform data. This is important as the availability and variety of multi-platform genomic data increases and the need for fast and intuitive methods becomes higher. However, existing methods either run in a small-scale analysis or combine multiple analysis methods thus requiring a large number of hyper-parameter tuning and expert knowledge.

The practicality of SNeCT was shown on the PAN-CAN12 dataset where the stratification result, based on k-means clustering of multi-platform patient profiles generated by SNeCT, showed a high correlation to subsets of clinical features and to the tissue of origin, which is similar to the observation made by Hoadley et al. [12]. Also, unlike existing methods, SNeCT can be applied to search for top-K similar patients given a new patient, which has various utilities such as in using the clinical information of top-k patients to perform diagnostics and prognostic predictions of a new patient. Furthermore, we showed how the combination of factor matrices can be used for individualized genomic interpretation of a patient.

There are considerations to make when using SNeCT, such as choosing normalization and gene mapping methods for construction of the input tensor, which values to be considered as missing and which to be '0', sizes of latent factors, appropriate network (removing negative associations), number of clusters in stratification studies, and value of K in the top-K search in the clinical predictions. All these considerations affect the result significantly. However, these are common problems in analysis and several solutions can be found in existing literature. Also, possible suggestion for extending SNeCT are modifying network constraint to works on signed graphs and generation of factor matrices that are sparse and interpretable [29, 38]. Even with the limitations and possible improvements, we conclude that SNeCT provides a powerful tool for integrative analysis of multi-platform to the bioinformatics community.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation

of Korea (2015R1C1A2A01055739, 2018R1A1A3A0407953, 2018R1A5A1060031).

REFERENCES

[1] S. Kim, L. Sael, and H. Yu, "A mutation profile for top-k patient search exploiting gene-ontology and orthogonal non-negative matrix factorization," *Bioinformatics*, vol. 31, no. 22, pp. 3653–3659, 2015.

[2] ——, "Identifying cancer subtypes based on somatic mutation profile," in *Proceedings of DTMBIO*. ACM, 2014, pp. 19–22.

[3] R. Louhimo and S. Hautaniemi, "Cnamet: an r package for integrating copy number, methylation and expression data," *Bioinformatics*, vol. 27, no. 6, pp. 887–888, 2011.

[4] C. J. Vaske, S. C. Benz, J. Z. Sanborn, D. Earl, C. Szeto, J. Zhu, D. Haussler, and J. M. Stuart, "Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using paradigm," *Bioinformatics*, vol. 26, no. 12, pp. i237–i245, 2010.

[5] K.-A. Sohn, D. Kim, J. Lim, and J. H. Kim, "Relative impact of multi-layered genomic data on gene expression phenotypes in serous ovarian tumors," *BMC systems biology*, vol. 7, no. 6, p. S9, 2013.

[6] J. Thomas and L. Sael, "Overview of integrative analysis methods for heterogeneous data," in *Proceedings of BigComp*. IEEE, 2015, pp. 266–270.

[7] ——, "Maximizing information through multiple kernel-based heterogeneous data integration and applications to ovarian cancer," in *Proceedings of EDB*. ACM, 2016, pp. 95–98.

[8] B. Wang, A. M. Mezlini, F. Demir, M. Fiume, Z. Tu, M. Brudno, B. Haibe-Kains, and A. Goldenberg, "Similarity network fusion for aggregating data types on a genomic scale," *Nature Methods*, vol. 11, no. 3, pp. 333–337, 2014.

[9] P. K. Mankoo, R. Shen, N. Schultz, D. A. Levine, and C. Sander, "Time to recurrence and survival in serous ovarian tumors predicted from integrated genomic profiles," *PloS one*, vol. 6, no. 11, p. e24709, 2011.

[10] Q. Mo, S. Wang, V. E. Seshan, A. B. Olshen, N. Schultz, C. Sander, R. S. Powers, M. Ladanyi, and R. Shen, "Pattern discovery and cancer gene identification in integrated cancer genomic data," *Proceedings of the National Academy of Sciences*, vol. 110, no. 11, pp. 4245–4250, 2013.

[11] Y. Yuan, E. M. Van Allen, L. Omberg, N. Wagle, A. Amin-Mansour, A. Sokolov, L. A. Byers, Y. Xu, K. R. Hess, L. Diao *et al.*, "Assessing the clinical utility of cancer genomic and proteomic data across tumor types," *Nature biotechnology*, vol. 32, no. 7, p. 644, 2014.

[12] K. A. Hoadley, C. Yau, D. M. Wolf, A. D. Cherniack, D. Tamborero, S. Ng, M. D. Leiserson, B. Niu, M. D. McLellan, V. Uzunangelov *et al.*, "Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin," *Cell*, vol. 158, no. 4, pp. 929–944, 2014.

[13] L. Sael, I. Jeon, and U. Kang, "Scalable tensor mining," *Big Data Research*, vol. 2, no. 2, pp. 82–86, 2015.

[14] K. Maruhashi, F. Guo, and C. Faloutsos, "Multiaspectforensics: Pattern mining on large-scale heterogeneous networks with tensor analysis," in *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*. IEEE, 2011, pp. 203–210.

[15] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell, "Toward an architecture for never-ending language learning." in *AAAI*, vol. 5. Atlanta, 2010, p. 3.

[16] M. Nickel, V. Tresp, and H.-P. Kriegel, "Factorizing yago: scalable machine learning for linked data," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 271–280.

[17] T. Kolda and B. Bader, "The tophits model for higher-order web link analysis," in *Workshop on link analysis, counterterrorism and security*, vol. 7, 2006, pp. 26–29.

[18] J. Sun, S. Papadimitriou, and S. Y. Philip, "Window-based tensor analysis on high-dimensional and multi-aspect streams." in *ICDM*, vol. 6, 2006, pp. 1076–1080.

[19] F. Cong, Q.-H. Lin, L.-D. Kuang, X.-F. Gong, P. Astikainen, and T. Ristaniemi, "Tensor decomposition of EEG signals: A brief review," *Journal of Neuroscience Methods*, vol. 248, pp. 59–69, 2015.

[20] T. G. Kolda and J. Sun, "Scalable tensor decompositions for multi-aspect data mining," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 2008, pp. 363–372.

[21] J. Henderson, J. C. Ho, A. N. Kho, J. C. Denny, B. A. Malin, J. Sun, and J. Ghosh, "Granite: Diversified, sparse tensor factorization for electronic health record-based phenotyping," in *2017 IEEE International Conference on Healthcare Informatics (ICHI)*, Aug 2017, pp. 214–223.

[22] L. Omberg, G. H. Golub, and O. Alter, "A tensor higher-order singular value decomposition for integrative analysis of DNA microarray data from different studies." *PNAS*, vol. 104, no. 47, pp. 18 371–6, 2007.

[23] S. P. Ponnapalli, M. A. Saunders, C. F. van Loan, and O. Alter, "A higher-order generalized singular value decomposition for comparison of global mRNA expression from multiple organisms," *PLoS ONE*, vol. 6, no. 12, 2011.

[24] O. Alter and G. H. Golub, "Reconstructing the pathways of a cellular system from genome-scale signals by using matrix and tensor computations," *Proceedings of the National Academy of Sciences*, vol. 102, no. 49, pp. 17 559–17 564, 2005. [Online]. Available: http://www.pnas.org/cgi/doi/10.1073/pnas.0509033102

[25] X. Xiao, A. Moreno-Moral, M. Rotival, L. Bottolo, and E. Petretto, "Multi-tissue Analysis of Co-expression Networks by Higher-Order Generalized Singular Value Decomposition Identifies Functionally Coherent Transcriptional Modules," *PLoS Genetics*, vol. 10, no. 1, 2014.

[26] B. Jeon, I. Jeon, L. Sael, and U. Kang, "Scout: Scalable coupled matrix-tensor factorization-algorithm and discoveries," in *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*. IEEE, 2016, pp. 811–822.

[27] I. Jeon, E. E. Papalexakis, C. Faloutsos, L. Sael, and U. Kang, "Mining billion-scale tensors: algorithms and discoveries," *The VLDB Journal*, vol. 25, no. 4, pp. 519–544, 2016.

[28] K. Shin, L. Sael, and U. Kang, "Fully scalable methods

for distributed tensor factorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 100–113, 2017.

[29] J. Lee, D. Choi, and L. Sael, "CTD: Fast, accurate, and interpretable method for static and dynamic tensor decompositions." *PloS One*, vol. 13, no. 7, p. e0200579, 2018.

[30] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[31] *The TCGA-Pancancer 4.7*, Accessed 25 June 2017. [Online]. Available: https://www.synapse.org/#!Synapse:syn1758011

[32] K.-W. Chang, W.-t. Yih, and C. Meek, "Multi-relational latent semantic analysis," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1602–1612.

[33] L. Omberg, K. Ellrott, Y. Yuan, C. Kandoth, C. Wong, M. R. Kellen, S. H. Friend, J. Stuart, H. Liang, and A. A. Margolin, "Enabling transparent and collaborative computational analysis of 12 tumor types within the cancer genome atlas," *Nature genetics*, vol. 45, no. 10, p. 1121, 2013.

[34] H. Dweep and N. Gretz, "miRWalk2.0: a comprehensive atlas of microRNA-target interactions," *Nature Methods*, vol. 12, no. 8, pp. 697–697, 2015.

[35] E. G. Cerami, B. E. Gross, E. Demir, I. Rodchenkov, Ö. Babur, N. Anwar, N. Schultz, G. D. Bader, and C. Sander, "Pathway commons, a web resource for biological pathway data," *Nucleic acids research*, vol. 39, no. suppl_1, pp. D685–D690, 2010.

[36] *Pathway Commons v8 All*, Accessed 7 July 2017. [Online]. Available: http://www.pathwaycommons.org/archives/PC2/v8/

[37] S. Oh, N. Park, L. Sael, and U. Kang, "Scalable Tucker factorization for sparse tensors - algorithms and discoveries," in *ICDE*. Paris, France: IEEE Computer Society, 2018.

[38] J. Lee, S. Oh, and L. Sael, "GIFT: Guided and Interpretable Factorization for Tensors - An Application to Large-Scale Multi-platform Cancer Analysis," *Bioinformatics*, vol. bty490, 2018.

[39] A. Narita, K. Hayashi, R. Tomioka, and H. Kashima, "Tensor factorization using auxiliary information," *Data Mining and Knowledge Discovery*, vol. 25, no. 2, pp. 298–324, 2012.

[40] W.-J. Li and D. Y. Yeung, "Relation regularized matrix factorization," in *21ST INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE (IJCAI-09), PROCEEDINGS*, 2009, p. 1126.

[41] M. Zinkevich, M. Weimer, L. Li, and A. J. Smola, "Parallelized stochastic gradient descent," in *Advances in neural information processing systems*, 2010, pp. 2595–2603.

[42] D. Choi, J.-G. Jang, and U. Kang, "Fast, accurate, and scalable method for sparse coupled matrix-tensor factorization," *arXiv preprint arXiv:1708.08640*, 2017.

[43] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *Advances in neural information processing systems*, 2011, pp. 693–701.

[44] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[45] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

[46] J. Oh, K. Shin, E. E. Papalexakis, C. Faloutsos, and H. Yu, "S-hot: Scalable high-order tucker decomposition," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 761–770.

[47] G. Yu, L.-G. Wang, Y. Han, and Q.-Y. He, "clusterProfiler: an R Package for Comparing Biological Themes Among Gene Clusters," *OMICS: A Journal of Integrative Biology*, 2012.

Dongjin Choi is s a graduate student in the Computational Science and Engineering, Georgia Institute of Technology. He received his B.S. in Computer Science and Engineering from Seoul National University. His research interests include numerical linear algebra and machine learning.

Lee Sael is a BK Associate Professor in the Deptartment of Computer Science and Engineering of Seoul National University. She received her Ph.D. in Computer Science from Purdue University in 2010, and her B.S. in Computer Science from Korea University in 2005. She has published in numerous journals and proceedings in the areas of Bioinformatics, Data Mining, and Machine Learning.