Bioinformatics, 36(6), 2020, 1855–1863
doi: 10.1093/bioinformatics/btz793
Advance Access Publication Date: 18 October 2019
Original Paper

Data and text mining

Inferring subgroup-specific driver genes from heterogeneous cancer samples via subspace learning with subgroup indication

Jianing Xi [1,2], Xiguo Yuan[3], Minghui Wang[4], Ao Li[4], Xuelong Li[2,5,*] and Qinghua Huang[1,2,*]

[1]School of Mechanical Engineering and [2]Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, 710072, China, [3]School of Computer Science and Technology, Xidian University, Xi'an 710071, China, [4]School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China and [5]School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on June 17, 2019; revised on September 23, 2019; editorial decision on October 13, 2019; accepted on October 16, 2019

Abstract

Motivation: Detecting driver genes from gene mutation data is a fundamental task for tumorigenesis research. Due to the fact that cancer is a heterogeneous disease with various subgroups, subgroup-specific driver genes are the key factors in the development of precision medicine for heterogeneous cancer. However, the existing driver gene detection methods are not designed to identify subgroup specificities of their detected driver genes, and therefore cannot indicate which group of patients is associated with the detected driver genes, which is difficult to provide specifically clinical guidance for individual patients.

Results: By incorporating the subspace learning framework, we propose a novel bioinformatics method called DriverSub, which can efficiently predict subgroup-specific driver genes in the situation where the subgroup annotations are not available. When evaluated by simulation datasets with known ground truth and compared with existing methods, DriverSub yields the best prediction of driver genes and the inference of their related subgroups. When we apply DriverSub on the mutation data of real heterogeneous cancers, we can observe that the predicted results of DriverSub are highly enriched for experimentally validated known driver genes. Moreover, the subgroups inferred by DriverSub are significantly associated with the annotated molecular subgroups, indicating its capability of predicting subgroup-specific driver genes.

Availability and implementation: The source code is publicly available at https://github.com/JianingXi/DriverSub.

Contact: xuelong_li@nwpu.edu.cn or qhhuang@nwpu.edu.cn

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Cancer is a life-threatening disease which involves millions of patients and causes hundreds of thousands of deaths (Siegel *et al.*, 2019). A primary reason of cancer progression is that DNA mutations on driver genes can lead to the abnormalities of key functions involved in tumor cells (Vogelstein *et al.*, 2013). Consequently, the comprehensive understanding of driver genes is crucial for the development of cancer diagnosis and treatment (Bailey *et al.*, 2018). Recently, the next generation sequencing technique has provided the unprecedented opportunity to detect the mutated genes in cancer samples (Meyerson *et al.*, 2010; Yu *et al.*, 2014), which accumulates a large volume of gene mutation data (Hudson *et al.*, 2010; Tomczak *et al.*, 2015). Unfortunately, even we are able to detect all the mutated genes of a tumor sample, there are still a number of mutated genes that do not confer an advantage to tumor progression, where the related mutations are denoted as passenger mutations (De and Ganesan, 2017). Due to the passenger mutations, even if a gene is mutated in a tumor sample, we still cannot directly distinguish whether this mutation is contributing to tumorigenesis or not (Vogelstein *et al.*, 2013). Therefore, inferring driver genes from the mutation data of tumor samples is a fundamental task in cancer research (Bailey *et al.*, 2018; Vogelstein *et al.*, 2013).

A common strategy to discover driver genes from the mutation data is to incorporate the hypothesis that driver genes are mutated recurrently among a large cohort of tumor samples (Bailey *et al.*, 2018;

Tokheim *et al.*, 2016). Based on this hypothesis, most of the previous published approaches focus on establishing computational methods to detect recurrently mutated genes as driver gene candidates (Tokheim *et al.*, 2016). For example, MutSigCV is proposed to predict recurrently mutated driver genes according to the statistical significance of their mutation frequencies among all samples (Lawrence *et al.*, 2013). Furthermore, OncodriveCLUST detects driver genes which display significant biases toward mutation clustering within their related sequences (Tamborero *et al.*, 2013). Based on the computational approaches aforementioned, numerous driver gene candidates with high frequencies have been discovered, many of which have been validated to play important roles in the progression of tumor cells according to the follow-up biological experiments (Lawrence *et al.*, 2013; Tamborero *et al.*, 2013).

However, an amount of previous studies have reported that cancer is a heterogeneous disease, i.e. there are various subgroups for a certain type of cancer, and the mutated driver genes are distinct between different subgroups of the cancer (Alizadeh *et al.*, 2015; Vogelstein *et al.*, 2013). If a driver gene is mutated in the samples within one subgroup rather than in all samples, this gene is denoted as a subgroup-specific driver gene (Cyll *et al.*, 2017). When compared with recurrently mutated driver genes, subgroup-specific driver genes are the essential clues to unveil the distinction between different subgroups of heterogeneous cancers (Alizadeh *et al.*, 2015; Cyll *et al.*, 2017). Moreover, subgroup-specific driver genes are also regarded as the guidance of precise diagnosis and treatment on cancer patients (Dagogojack and Shaw, 2017). Nonetheless, when a certain subgroup includes only a small fraction of the investigated samples, the driver genes specific to this subgroup will be infrequently mutated among all samples (Cyll *et al.*, 2017). In this regard, to detect driver genes with relatively low mutation frequencies, MutSigCV also introduce corrections for variation by gene-specific background mutation rates, and can detect genes that are only mutant in 3–5% of samples (Lawrence *et al.*, 2013). In addition to mutation frequency, there are also a lot of methods consider multiple types of features, such as the amino acid change and flanking sequence composition (Carter *et al.*, 2009; Tan *et al.*, 2012), which do not rely on mutation frequency and can detect rare mutations. Still, the aforementioned methods are not designed to identify subgroup specificities of their detected driver genes, and therefore cannot indicate which group of patients are associated with the detected driver genes, which is difficult to provide specifically clinical guidance for individual patients (Cyll *et al.*, 2017). Consequently, there is an urgent need for the computational methods which are capable of predicting subgroup-specific driver genes (Cyll *et al.*, 2017).

Recently, there have been some attempts to detect subgroup-specific driver genes from heterogeneous cancer data (Pereira *et al.*, 2016). For the special case that the subgroups of samples are well-annotated, an intuitive approach is to divide the cancer samples into different subgroups according to the annotations, and detect the recurrently mutated genes from these annotated subgroups, respectively (Pereira *et al.*, 2016). With the subgroups annotations of samples, a maximum weight submatrix optimization based method is then proposed to simultaneously detect common and subgroup-specific driver genes (Zhang and Zhang, 2017). Nonetheless, all these methods are suitable to the ideal case where the information of subgroups are known (Pereira *et al.*, 2016; Zhang and Zhang, 2017). When the subgroup annotations of the cancer samples are not available, the aforementioned methods cannot predict subgroup-specific driver genes anymore (Cyll *et al.*, 2017). Since the information of subgroups are unavailable on most occasions (Vogelstein *et al.*, 2013), inferring subgroup-specific driver genes from cancer samples without subgroup annotations is an imperative task for the development of precision therapy for heterogeneous cancers (Cyll *et al.*, 2017).

In this article, we present a novel bioinformatics method called DriverSub, which can efficiently address the unavailability of subgroup annotation problem. Due the difficulty of acquiring a large amount of experimentally validated driver genes, we incorporate the unsupervised learning strategy which does not require any known driver genes. Furthermore, to simultaneously infer driver genes and their related subgroup specificities, we establish a transformation which can convert the mutation data of genes into representation vectors of genes (Yang *et al.*, 2019) through the subspace learning framework (Wang *et al.*, 2016; Zheng *et al.*, 2019). For the vectors of the investigated genes, the distances between the output vectors and the origin of the subspace can be used to discriminate driver genes, and the coordinate values in different dimensions of the vectors can indicate the subgroups specificities of the related genes. When we evaluate the performance of DriverSub through simulation datasets where the ground truth is known, DriverSub outperforms the previous methods on the prediction of subgroup-specific driver genes (Lawrence *et al.*, 2013; Tamborero *et al.*, 2013). For the inference of subgroup specificities of driver genes, when we further compare DriverSub with existing subspace learning methods (Hyvärinen, 2013; Jolliffe and Cadima, 2016), DriverSub also yields the best prediction of the subgroup indices of driver genes. Moreover, when DriverSub is applied on two real cancer datasets (Cancer Genome Atlas Network and Others, 2012; Cancer Genome Atlas Research Network and Others, 2014), the predicted genes are highly enriched for experimentally validated known driver genes (Sondka *et al.*, 2018). Moreover, the subgroups inferred by DriverSub are significantly associated with the annotated molecular subgroups, indicating the efficiency of DriverSub on inferring subgroup-specific driver genes.

2 Materials and methods

2.1 Subspace learning and output vector

Since the biological experimental annotations of driver genes require extensive costs, we adopt the unsupervised learning strategy to overcome the lack of annotation problem. Here we use the subspace learning framework to obtain the vectorized representations of unannotated genes (Wang *et al.*, 2016; Zheng *et al.*, 2019), which can transform the input high-dimensional mutation data of genes into a low-dimensional subspace of gene representations (Yang *et al.*, 2019). In this study, the input mutation data are formed as a binary matrix (Hofree *et al.*, 2013), which consists of the vectors of mutations of the investigated genes $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i, \ldots, \boldsymbol{x}_P]$ (here P is the total gene number). The binary input mutation vector of the ith gene is an N-dimension vector, where the number of dimensions of the input vectors N is the total amount of the investigated samples, and the binary value of the jth coefficient of the vector indicate whether there are any mutations occurring in the ith gene of the jth sample (Hofree *et al.*, 2013).

Meanwhile, the output matrix $\boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_i, \ldots, \boldsymbol{z}_P]$ is composed of the low-dimensional output subspace vectors ($\forall \boldsymbol{z}_i \leftarrow \mathbb{R}^K$), where the number of dimension K of the subspace vectors should be far less than the number of dimension N of the input mutation vectors, i.e. $K \ll N$ (Wang *et al.*, 2016; Zheng *et al.*, 2019). Furthermore, for conserving the mutation information of input genes, the transformation of subspace learning should be constructed to be approximately invertible (Wang *et al.*, 2016). Consequently, the output subspace vectors can be used to reconstruct the input mutation profiles due to the approximate invertibility of the transformation. In contrast to the input vectors of mutation profiles, the output vectors in the low-dimensional subspace are more appropriate for the computational analysis, which can also circumvent problem caused by the curse of dimensionality (Wang *et al.*, 2016; Zheng *et al.*, 2019).

2.2 Subgroup specificity indication by subspace dimensions

Although the output vector can represent the gene's mutation profile across the investigated samples, how to indicate the subgroup indices of the genes is still an essential issue for inferring subgroup-specific driver genes. In other word, the challenge is how to determine which subgroup the investigated gene belongs to (Cyll *et al.*, 2017). Since the dimensions of the subspace can be used to reveal the intrinsically latent features of the input data (Wang *et al.*, 2016; Zheng *et al.*, 2019), the dimensions of the output vectors may also have the potential of indicating the related subgroups of these genes. Nevertheless, directly using the existing subspace learning cannot guarantee whether the subspace dimensions of output vectors can indicate the subgroup specificities of the driver genes (Hyvärinen, 2013; Jolliffe and Cadima, 2016). Therefore, to ensure the

relationship between the subspace dimensions and the subgroup indices of genes, we propose a novel subgroup indicating subspace learning method called DriverSub, which is based on subspace learning with additional constraints and regularizations on the subspace dimension.

To ensure that the subspace dimension can efficiently indicate the subgroup indices of genes, we introduce two prerequisites for the subgroup indicating subspace learning method. The first prerequisite is that the coordinate values of output vectors can be calculated as a metric for driver gene evaluation (Tokheim *et al.*, 2016). The second prerequisite is that the coordinate values of output vectors in different dimensions can be used to indicate the subgroup indices of the corresponding genes (Cyll *et al.*, 2017). To achieve the first prerequisite, we introduce the nonnegative constraints on output vectors to ensure the coordinate values are either positive or zero, where a larger value represents a better chance of being driver gene (Cai *et al.*, 2011). For the second prerequisite, we utilize sparsity-inducing regularization to ensure that most of the output vectors are sparse vectors (Zhou and Tao, 2013). For output vectors z_i, although L0-norm corresponds to the sparsity, the L0-norm optimization problem is non-convex and therefore difficult to solve (Ramirez *et al.*, 2013). Accordingly, we use the convex L1 optimization problem as a good approximation to L0-norm problem of sparsity, as suggested by Candes and Tao (2005) and Ramirez *et al.* (2013). When the coordinate values are sparse, the output vectors trend to be close to the coordinate axes or the coordinate planes, where the dimensions of the located coordinate axes or planes can indicate the subgroup specificities of the genes. The objective function of the subgroup indicating subspace learning is illustrated below:

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{z}_i} \quad & \sum_{i=1}^{P} \|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{z}_i\|_2^2 + \lambda_Z \sum_{i=1}^{P} \|\boldsymbol{z}_i\|_1 \\ \text{s.t.} \quad & \boldsymbol{W} \geq 0 \ \text{ and } \ \boldsymbol{z}_i \geq 0, \forall i = 1, \ldots, P \end{aligned} \tag{1}$$

where the reconstruction function is assumed to be linear function $f_{\boldsymbol{W}}(\boldsymbol{z}_i) = \boldsymbol{W}\boldsymbol{z}_i$, and the related parameters are in the matrix $\boldsymbol{W}$ (Zheng *et al.*, 2019). Moreover, the tuning parameter λ_Z is responsible for the sparsity-inducing regularization, which is a positive value that can control the distance between the output vectors and the coordinate axes/planes (Zhou and Tao, 2013). Accordingly, the subspace dimensions of vectors can efficiently indicate the subgroup indices of the related genes.

In addition to the two ideas aforementioned, overfitting issue of the subspace learning procedure is also a noteworthy problem (Li *et al.*, 2015). Hence, we further introduce the Frobenius norm regularization into the subspace learning procedure, which can prevent extreme values in the parameters of the subspace transformation (Li *et al.*, 2015). The objective function with Frobenius norm regularization is

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{z}_i} & \sum_{i=1}^{P} \|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{z}_i\|_2^2 + \lambda_Z \sum_{i=1}^{P} \|\boldsymbol{z}_i\|_1 + \lambda_W \|\boldsymbol{W}\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{W} \geq 0 \ \text{ and } \ \boldsymbol{z}_i \geq 0, \forall i = 1, \ldots, P \end{aligned} \tag{2}$$

where the term $\|\boldsymbol{W}\|_F^2$ denotes the squared Frobenius norm of the matrix $\boldsymbol{W}$, and the tuning parameter λ_W is a positive value to control the tolerability of extreme values (Li *et al.*, 2015). Here the two parameters λ_Z and λ_W are set to 0.001 and 0.1 empirically. Accordingly, the output vectors learned from the subspace learning above can both preserve the information of the input mutation profiles of the genes, and indicate the subgroup indices of the genes through the subspace dimensions. The overview of our proposed DriverSub is illustrated in Figure 1.

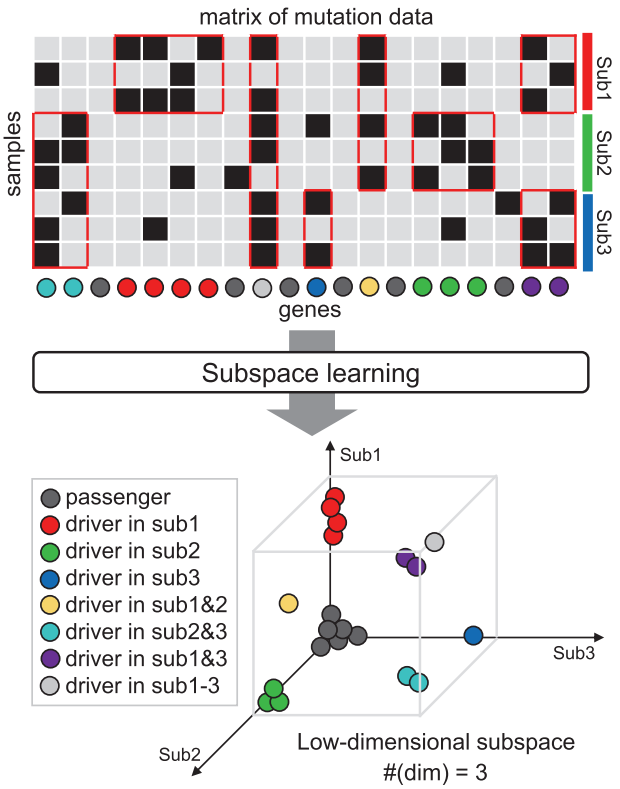

Fig. 1. The overview of our proposed DriverSub. To overcome the difficulty of acquiring a large amount of experimentally validated driver genes, DriverSub incorporate the unsupervised learning strategy which do not requires any known driver gene. To infer driver genes and their related subgroup specificities simultaneously under the situation of unavailability of subgroup annotations, DriverSub is established based on the subspace learning framework (Wang *et al.*, 2016; Zheng *et al.*, 2019) which can represent the investigated genes as output vectors through their mutation data. Accordingly, the driver genes can be discriminated through the distances between the output vectors and the origin. The subgroups specificities of the related genes can also be inferred according to the coordinate values in different dimensions of the vectors

2.3 Subspace learning algorithm via alternative optimization

To obtain both the output vectors and the parameters of the transformation for the subspace learning (Zheng *et al.*, 2019), we adopt an alternative optimization strategy to solve the learning task (Cai *et al.*, 2011). Note that the objective function in Equation (2) of subspace learning is equivalent to the following formulation:

$$\begin{aligned} & \min_{\boldsymbol{W},\boldsymbol{Z}} \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{Z}\|_F^2 + \lambda_Z \|\boldsymbol{Z}\|_1 + \lambda_W \|\boldsymbol{W}\|_F^2, \\ & \text{s.t.}\ \boldsymbol{W} \geq 0 \ \text{ and } \ \boldsymbol{Z} \geq 0. \end{aligned} \tag{3}$$

To solve the optimization problem of the subspace learning with subgroup index indication, we firstly introduce two matrices of Lagrange multipliers $\boldsymbol{\Phi} = [\phi_{jk}]$ and $\boldsymbol{\Psi} = [\psi_{ki}]$ on the two constraints of matrices $\boldsymbol{W} \geq \boldsymbol{0}$ and $\boldsymbol{Z} \geq \boldsymbol{0}$ (Cai *et al.*, 2011). Here the elements ϕ_{jk} and ψ_{ki} of the two matrices are the Lagrange multipliers for the element-wise constraints $w_{jk} \geq 0$ and $z_{ki} \geq 0$, respectively. The summation of all the element-wise Lagrange multipliers can be written in matrix format $\sum_{j,k} \phi_{jk} w_{jk} = \mathrm{Tr}\{\boldsymbol{\Phi}\boldsymbol{W}^{\mathrm{T}}\}$ and $\sum_{k,i} \psi_{ki} z_{ki} = \mathrm{Tr}\{\boldsymbol{\Psi}\boldsymbol{Z}^{\mathrm{T}}\}$. Furthermore, since the matrix $\boldsymbol{Z}$ is restricted to be nonnegative, the term of L1 norm penalty $\|\boldsymbol{Z}\|_1$ can be written as the summation of the elements of matrix $\boldsymbol{Z}$, which is also equivalent to the term $\mathrm{Tr}\{\boldsymbol{J}\boldsymbol{Z}^{\mathrm{T}}\}$ (here $\boldsymbol{J}$ is a K by P matrix of ones). Based on the property of matrix that $\|\boldsymbol{A}\|_F^2 = \mathrm{Tr}\{\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\}$, the Lagrange function can be further written as the formulation below:

$$\begin{aligned} \mathcal{L} = & \mathrm{Tr}\{(\boldsymbol{X} - \boldsymbol{W}\boldsymbol{Z})(\boldsymbol{X} - \boldsymbol{W}\boldsymbol{Z})^{\mathrm{T}}\} + \lambda_Z \mathrm{Tr}\{\boldsymbol{J}\boldsymbol{Z}^{\mathrm{T}}\} \\ & + \lambda_W \mathrm{Tr}\{\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\} + \mathrm{Tr}\{\boldsymbol{\Phi}\boldsymbol{W}^{\mathrm{T}}\} + \mathrm{Tr}\{\boldsymbol{\Psi}\boldsymbol{Z}^{\mathrm{T}}\}. \end{aligned} \tag{4}$$

Since there are two variables to be solved, i.e. matrix $\boldsymbol{W}$ for the parameters of the reconstruction function, and matrix $\boldsymbol{Z}$ for the

output vectors, we conduct the alternative solving strategy by optimizing one variable when the other variable is fixed (Cai *et al.*, 2011). Specifically, when the values of a matrix are fixed, we can solve the updating rules of the other matrix by its partial derivative of $\mathcal{L}$. The partial derivatives of the Lagrange function ℓ with respect to matrix $\boldsymbol{W}$ and $\boldsymbol{V}$ are

$$\begin{aligned}\partial\mathcal{L}/\partial\boldsymbol{Z} &= -2\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X} + 2\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{Z} + \lambda_Z\boldsymbol{J} + \boldsymbol{\Psi},\\ \partial\mathcal{L}/\partial\boldsymbol{W} &= -2\boldsymbol{W}\boldsymbol{Z}^{\mathrm{T}} + 2\boldsymbol{W}\boldsymbol{Z}\boldsymbol{Z}^{\mathrm{T}} + 2\lambda_W\boldsymbol{W} + \boldsymbol{\Phi}.\end{aligned} \tag{5}$$

Accordingly, we derive the Karush-Kuhn-Tucker (KKT) conditions with respect to the Lagrange function $\mathcal{L}$ (Cai *et al.*, 2011), i.e. $\partial\mathcal{L}/\partial\boldsymbol{Z} = \boldsymbol{0}$, $\partial\mathcal{L}/\partial\boldsymbol{W} = \boldsymbol{0}$, $\boldsymbol{Z}\circ\boldsymbol{\Psi} = \boldsymbol{0}$ and $\boldsymbol{W}\circ\boldsymbol{\Phi} = \boldsymbol{0}$, where the symbol $\circ$ denotes to the element-wise product of two matrices, and the matrix $\boldsymbol{0}$ is all-zero matrix. According to the KKT conditions, we can obtain the following equations for both $\boldsymbol{Z}$ and $\boldsymbol{W}$:

$$\begin{aligned}\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{Z} + \frac{\lambda_Z}{2}\boldsymbol{J}\right)\circ\boldsymbol{Z} &= (\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X})\circ\boldsymbol{Z},\\ (\boldsymbol{W}\boldsymbol{Z}\boldsymbol{Z}^{\mathrm{T}} + \lambda_W\boldsymbol{W})\circ\boldsymbol{W} &= (\boldsymbol{X}\boldsymbol{Z}^{\mathrm{T}})\circ\boldsymbol{W}.\end{aligned} \tag{6}$$

Based on the two equations in Equation (6), we can yield the alternatively updating algorithms of the two matrices $\boldsymbol{Z}$ and $\boldsymbol{W}$:

$$\begin{aligned}\boldsymbol{Z} &\leftarrow \boldsymbol{Z}\circ(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X})./\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{Z} + \frac{\lambda_Z}{2}\boldsymbol{J}\right),\\ \boldsymbol{W} &\leftarrow \boldsymbol{W}\circ(\boldsymbol{X}\boldsymbol{Z}^{\mathrm{T}})./(\boldsymbol{W}\boldsymbol{Z}\boldsymbol{Z}^{\mathrm{T}} + \lambda_W\boldsymbol{W}).\end{aligned} \tag{7}$$

where the symbol ./ denotes the element-wise division. By alternatively using the two formulas, the objective functions in Equation (7) are guaranteed to be convergent under the two updating rules (Cai *et al.*, 2011). The output vectors and the learned transformation will be obtained after the convergence of this algorithm. In addition, compared with L1-norm which can control the distance between the output vector and the coordinate axes and plains, L2-norm has more advantages in controlling the distance between the output vector and the origin. Consequently, we also derive the alternatively updating roles for L2-norm regularization on z_i (Supplementary Material), and implement two options for users to choose L1- or L2-norm in the source code. In summary, by adopting the alternative optimization strategy, DriverSub can simultaneously obtain the output vectors and the transformation's parameters of the subspace learning.

2.4 Driver gene inference by normalized distance

After the output vectors are obtained through the subspace learning algorithm, the next step is to infer driver genes and their subgroup indices from the learned output vectors. Since a larger coordinate value in the nonnegative output vector indicates a stronger trend toward driver gene, we use the coordinate values to calculate the distances between the origin and the output vector. Here the distance is adopted as the evaluation metric for driver gene. Since the scales of vectors are different among the dimensions of the subspace, to overcome the imbalance among the scales of the dimensions, we adopt the normalization procedure where the coordinate values of output vectors are divided by their related standard deviations (Cai *et al.*, 2011). Subsequently, we can use the normalized vectors to calculate their Euclidean distances from the origin. Finally, the investigated genes can be sorted according to their distance scores, and the top ranked genes are inferred as the candidates of driver genes.

Our proposed DriverSub can also infer the related subgroup samples of the driver genes. For a predicted driver gene, we can use the coordinate values in the dimensions of the output vector to indicate its related subgroups. Here the parameter matrix $\boldsymbol{W}$ can reflect the relationship between subgroup samples and subspace dimensions, where the value of the non-negative element w_{jk} can be used as the indicating score of the jth sample to the kth subgroup. For the jth sample, if the score w_{jk^*} is the maximum scores from w_{j1} to w_{jK}, then the jth sample is assigned to the k^*th subgroup. Furthermore, we can acquire the corresponding samples of each dimension of the subspace via the parameters of the learned transformation. For the output vector of a predicted gene, if the coordinate value of the kth dimension is larger than zero, then the investigated gene is considered to be related to the kth subgroup. Finally, by retrieving the subgroup samples from the subspace dimensions according to their coordinate values in these dimensions, we can acquire the subgroup samples of the driver genes.

3 Results

3.1 Simulation dataset evaluations

3.1.1 Overall performance

In this section, we conduct a series of experiments on the evaluation of the performance of subgroup-specific driver gene prediction. To overcome the unavailability of ground truth caused by the lack of known driver genes, we conducted simulation data analysis where the ground truth of driver genes is available (Hofree *et al.*, 2013). The details of the generation of simulation data is provided in Supplementary Material. In this study, we compare our proposed DriverSub with two existing methods MutSigCV (Lawrence *et al.*, 2013) and OncodriveCLUST (Tamborero *et al.*, 2013), which are widely used in driver gene prediction. In the comparison analysis, all the parameters used in these competing methods are set to the default values (we also try different parameters of the two existing methods, and the results are shown in Supplementary Material and Supplementary Figs S1 and S2; Lawrence *et al.*, 2013; Tamborero *et al.*, 2013). For DriverSub, the predicted driver genes are prioritized according to their distances between the output vectors and the origin. For MutSigCV and OncodriveCLUST, the driver gene candidates are prioritized according to their false discovery rate controlled q-values (Lawrence *et al.*, 2013; Tamborero *et al.*, 2013). Accordingly, the investigated genes are be prioritized by the three competing methods, where the top ranked gene candidates have the potential to be driver genes (Tokheim *et al.*, 2016).

When we analyze the top ranked genes of these competing methods with various rank thresholds, we can calculate a series of precisions as the fractions of predicted genes which are ground truth driver genes, and recalls as the fractions of ground truth driver genes that are successfully predicted (Jing *et al.*, 2019). By comparing these precisions and recalls, we can evaluate the prediction results of these methods. When we compare the results of top 100–500 genes with a step of 100, we can draw these precision recall curves of the competing methods, where the x- and y-axis are recalls and precisions, respectively (Chen *et al.*, 2018). If the location of a curve is closer to the top and right, then the related values of precision and recall are also higher (Chen *et al.*, 2018). Thus, the precision recall curves can be used to evaluate the prediction performances of the competing methods.

As shown in Figure 2, the curve of our DriverSub is the closest to the top and right among the curves of the three methods, indicating that our method yields better prediction results than those of the other two competing methods. Generally, for the simulation data with different numbers of subgroups, the precision recall curves of the three competing methods show similar trends. Taking the case of simulated data with four subgroups as an example, the precisions of the top 300 genes for MutSigCV and OncodriveCLUST are 0.94 and 0.73, respectively, while the precision for our proposed DriverSub is 0.99. At the same time, the recalls of the top 300 genes for DriverSub is 0.61, which is also larger than those for MutSigCV and OncodriveCLUST, respectively.

When we further evaluate the top 400 or 500 genes predicted by the competing methods, compared with the results for OncodriveCLUST, DriverSub and MutSigCV achieve better prediction performances among all the simulation data with different subgroups. When applied on simulation data with four subgroups, DriverSub and MutSigCV yield the precisions ranging in a narrow interval from 0.75 to 0.76, indicating that the two methods achieve comparable performances in the prediction tasks. Nevertheless, the performances of DriverSub still show a slight advantage where precision of DriverSub is about 0.01 larger than that of MutSigCV. In summary, our proposed DriverSub can successfully predict driver

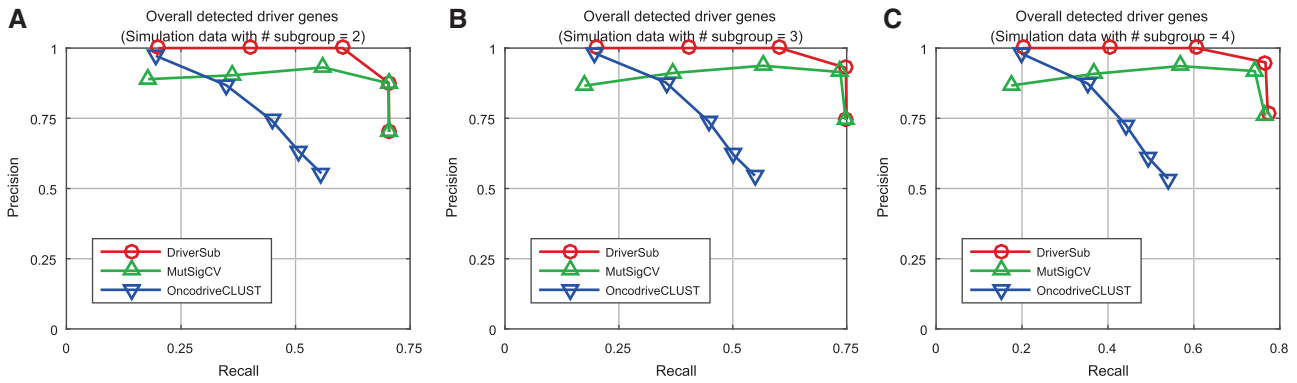

Fig. 2. Precision recall curve for the predicted driver genes by DriverSub, MutSigCV and OncodriveCLUST, on simulation datasets with (**A**) two subgroups, (**B**) three subgroups and (**C**) four subgroups. The precision recall curves are plotted according to the top ranked 100–500 predicted genes with a step of 100

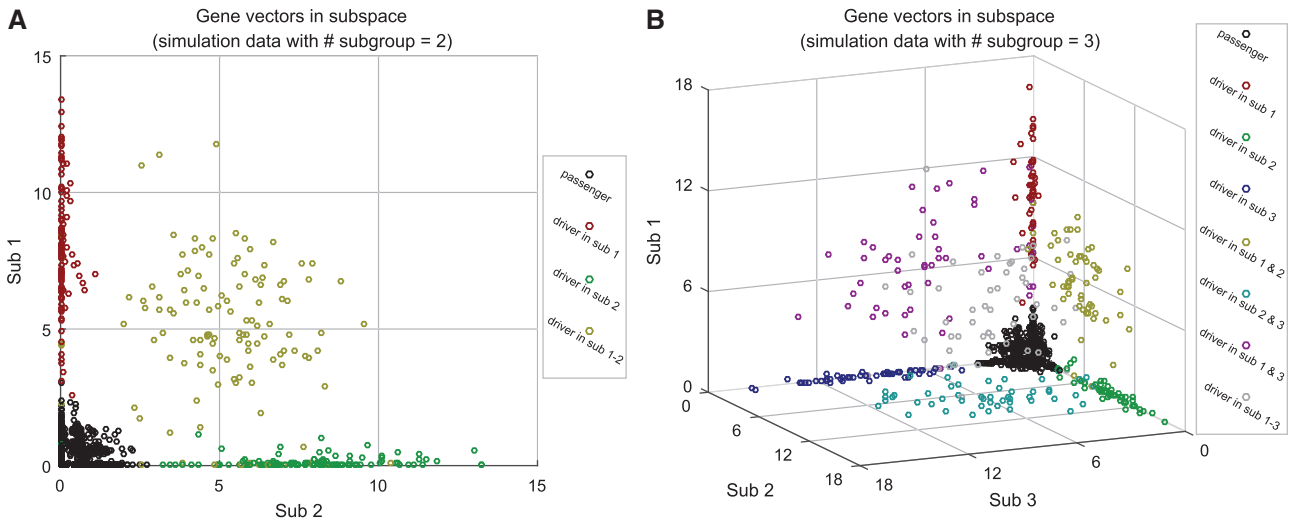

Fig. 3. The visualization of subspace output vectors yielded by DriverSub on simulation datasets with (**A**) two subgroups and (**B**) three subgroups. The black circlets represent ground truth genes with passenger mutations, and the colored circlets indicate ground truth driver genes in different numbers of subgroups, as shown in the legends

genes, and achieve comparable or better performances than those of the existing methods.

3.1.2 Subgroup-specific driver gene prediction performance

For our proposed DriverSub, a distinct advantage beyond the predictive capability of driver genes is that DriverSub can further estimate the subgroup specificities of the predicted genes. For the output vectors learned by DriverSub, their coordinate values in different dimensions can be used to indicate the related subgroups of these investigated genes. Specifically, a larger value in the kth dimension of the vector indicates a larger tendency of the investigated gene to belong to the kth subgroup. When there are more than one coefficients of the output vector representing relatively larger values, the investigated gene is considered as driver gene mutated in multiple subgroups, and the subgroups are indicated by the dimensions of these coefficients. Moreover, if all the coefficients of the output vector are large values, then the investigated gene is regarded as a driver gene recurrently mutated among all samples. Consequently, in addition to the prediction of driver genes, DriverSub is also capable of inferring the related subgroups of the investigated driver genes.

To demonstrate the advantage of our approach in predicting subgroup-specific driver genes, we further evaluate the inferred subgroup indices of the investigated genes. Since our proposed DriverSub is based on subspace learning, we also compare DriverSub with two widely used subspace learning methods, principle component analysis (Jolliffe and Cadima, 2016) and independent component analysis (Hyvärinen, 2013). Based on the simulation datasets with different subgroups, the comparison results of three subspace learning methods are evaluated through precision recall curves (Supplementary Fig. S3). Through the precision recall curves of the three methods, we can see that the curves for our DriverSub are closest to the top and right among all cases, indicating the efficiency of our approach in inferring the related subgroups of driver genes. For example, when applied on simulation data with two subgroups, the precisions of the driver genes in only one subgroup predicted by our approach is averagely 27.5% higher than those of the other competing methods, and the metrices are also higher for driver genes in three and four subgroups, respectively. Moreover, we also test the performances of DriverSub with regularizations of L1- and L2-norm, and the comparison results demonstrate that the prediction results with the two types of regularizations are comparable to each other (Supplementary Figs S4 and S5).

Furthermore, to intuitively interpret the prediction results of driver genes and their subgroup specificities for our proposed DriverSub, we also draw the coordinate graphs of the output vectors under the cases of two and three subgroups (Fig. 3), where the circlets are the learned output vectors in the subspace. In Figure 3A, the black circlets represent the ground truth genes with passenger mutations. The red circlets and green circlets denote the ground truth driver genes which are mutated in the first subgroup (Sub 1) and in the second subgroup (Sub 2), respectively. The yellow circlets indicate the ground truth driver genes mutated in both the first and the

second subgroups. As shown in Figure 3A, nearly most black circlets are surrounding the origin of the coordinate plane, indicating our approach can successfully filter out the genes with passenger mutations by their distances from the origin. Also, most red circlets and green circlets are beneath the y-axis (Sub 1) and x-axis (Sub 2) respectively, and most yellow circlets are located in the first quadrant of the subspace. These phenomena demonstrate that the driver genes in either Subs 1 or 2 can be efficiently distinguished through the output vectors yielded by our approach.

For the situation where there are three subgroups in the simulation data (Fig. 3B), we further use blue circlets to represent driver genes mutated in the third subgroup (Sub 3). Also, the cyan circlets indicate the driver genes in both Subs 2 and 3, whereas the violet circlets denote the driver genes in both Subs 1 and 3. Moreover, the light gray circlets represent the driver genes mutated in all the three subgroups (Subs 1–3). Similar to the case with two subgroups, the black circlets of passenger genes in Figure 3B are also close to the origin of the coordinate space. Meanwhile, the red, green and blue circlets are beneath the z-axis (Sub 1), x-axis (Sub 2) and y-axis (Sub 3), respectively. Furthermore, the yellow, cyan and violet circlets are located at the xz-plane (Subs 1 and 2), the xy-plane (Subs 2 and 3), and the yz-plane (Subs 1 and 3). Moreover, most of the light gray circlets are distributed in the first octant of the coordinate space. Based on the results above, we can conclude that our approach can successfully infer driver genes and their related subgroups according to the output vectors learned through the unsupervised learning paradigm.

For the cases where the subgroup number is larger than 3, we also extend the number of subgroup to a greater number (here up to 7) in the subgroup analysis. As shown in Supplementary Figure S6, we can observe a parabolic trend of the performance of DriverSub, where it shows an increase followed by a decrease in precision as the subgroup number increases. From a biological stand point, a possible explanation of the phenomenon is that, when the number of subgroups becomes larger, the fractions of subgroup-specific drivers in the cohort also decrease rapidly. Consequently, even though the proposed DriverSub is able to represent the subgroup specificities of driver mutations, the mutation frequencies of subgroup-specific drivers are likely to be indistinguishable from the passenger mutations in this extreme case. Moreover, since the two parameters λ_Z and λ_W are fixed in the aforementioned analysis of DriverSub, we also conduct an experiment to see how the prediction results will change when these parameters are perturbed (Supplementary Fig. S7), where the sensitivity analysis shows that DriverSub is robust to perturbations of these parameters.

3.2 Real cancer data applications

3.2.1 Enrichment analysis for known driver genes

To demonstrate the applicability of our proposed DriverSub on real cancer data, we apply our method on datasets of two types of cancers, breast cancer (Cancer Genome Atlas Network and Others, 2012) and bladder cancer (Cancer Genome Atlas Research Network and Others, 2014). The two datasets contain somatic mutations of 507 breast cancer samples and 130 bladder cancers samples, respectively, which are curated by The Cancer Genome Atlas (TCGA) program (Tomczak *et al.*, 2015) and are downloaded from cBioPortal database (Gao *et al.*, 2013). For the two datasets, the subspace dimension K of DriverSub is set to 4 experimentally (details in Supplementary Material). When analyzing the prediction results on real cancer data, we use a list of experimentally validated known driver genes which are curated by Cancer Gene Census (CGC; Sondka *et al.*, 2018). Subsequently, we can use Fisher's exact test to assess whether the genes predicted by DriverSub are significantly enriched for known driver genes or not (Subramanian *et al.*, 2005). To demonstrate the advantage of our proposed DriverSub, we also apply the two existing methods MutSigCV and OncodriveCLUST on the real cancer datasets with default settings, and compare their enrichment results.

Due to the practical reason, the top ranked predicted genes are more likely to be chosen in the further experimental validation, we select the top ranked predicted genes for the enrichment analysis. As suggested by previous study (Hou *et al.*, 2018), we first provide the full lists of the top 500 driver gene candidates predicted by DriverSub, MutSigCV and OncodriveCLUST on breast cancer and bladder cancer in Supplementary Tables S1–S6, respectively. Here, we apply Fisher's exact test on the predicted results of the three competing methods, where the corresponding P-values can be used to assess whether the results are enriched for known driver genes. For the results of DriverSub, MutSigCV and OncodriveCLUST, their P-values for breast cancer data are 1.34e-07, 9.12e-01 and 2.88e-07, respectively, where DriverSub yields the most significant P-values. For bladder cancer data, the P-values of MutSigCV and OncodriveCLUST are 5.68e-02 and 2.70e-05, and our proposed DriverSub yields a P-value of 8.59e-18. Generally, most of the P-values of the three competing methods are <0.05, indicating that the results of these methods are significantly enriched for known driver genes. Furthermore, the P-values yielded by our proposed DriverSub are much smaller than those of the other two competing methods, demonstrating the superior capability of DriverSub in driver gene prediction.

Since top 500 is much too large a gene set to start from for any kind of experimental prioritization, we further down sample from here and demonstrate the enrichment P-values of the investigated methods. As shown in Supplementary Table S3, when the number of the top ranked genes becomes smaller, the P-values of our method are also the smallest among those of these methods. Taking the results for top 200 genes as an example, the P-values of DriverSub, MutSigCV and OncodriveCLUST are 1.46e-06, 8.35e-02 and 1.23e-02 for breast cancer data. For bladder cancer data, the P-values of MutSigCV and OncodriveCLUST 6.69e-06 and 4.16e-04, and DriverSub yields a P-value of 2.96e-07 in comparison. Consequently, when we only focus on a small number of the top ranked predicted genes, the superior capability of our method still holds up for the prediction of known driver genes.

3.2.2 Venn diagram analysis

To further analyze the results of the three methods on the two real cancer datasets, we evaluate the predicted genes by investigating whether they are also shared by the results of other methods. Hence, we draw the Venn diagram of the predicted genes for the three competing methods (Fig. 4), which can illustrate the shared genes of these methods. For the prediction results on breast cancer data (Fig. 4A), there are four predicted genes shared by all the three methods, including known driver genes PIK3CA and TP53 (Sondka *et al.*, 2018). According to previous studies, PIK3CA plays oncogenic roles in breast cancer (Mukohara, 2015), and TP53 is also a driver gene that encodes a tumor suppressor protein (Yin *et al.*, 2002). For the results on bladder cancer data (Fig. 4B), there are six driver genes shared by the three methods, and three of them are known driver

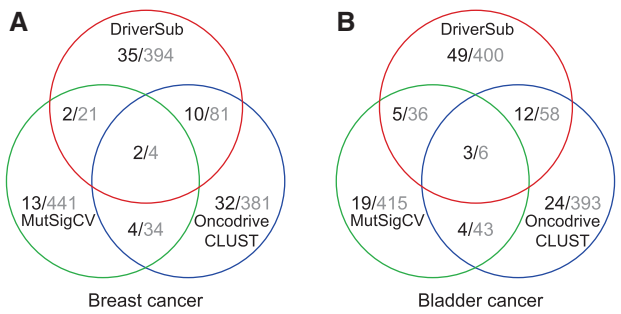

Fig. 4. Venn diagram of intersections between the predicted genes for DriverSub, MutSigCV and OncodriveCLUST on real datasets of (**A**) breast cancer samples and (**B**) bladder cancer samples. The red, green and blue circles represent the predicted results of DriverSub, MutSigCV and OncodriveCLUST, respectively. The gray numbers after the slash indicate the numbers of included genes, and the black numbers before the slash indicate the numbers of driver genes which are validated by experiments (Sondka *et al.*, 2018). (Color version of this figure is available at *Bioinformatics* online.)

gene KDM6A, PIK3CA and TP53 (Sondka *et al.*, 2018). Generally, most of the shared genes have been reported by CGC known driver genes lists.

Furthermore, for the results on breast cancer data, there are a total of 81 genes predicted by both DriverSub and OncodriveCLUST rather than MutSigCV (Fig. 4A). Specifically, 10 of them are known driver genes, such as MTOR, MAP2K4, MAP3K1, NCOA2 and SPEN. Meanwhile, there are also 21 genes shared by the results of DriverSub and MutSigCV rather than OncodriveCLUST, which include known driver genes such as AFF4 and AKAP9 (Sondka *et al.*, 2018). When we investigate the results on bladder cancer data, we can also observe similar phenomenon that there are many predicted genes shared by the results of both DriverSub and the other methods (Fig. 4B). For the prediction results on bladder cancer data, there are 58 genes shared by DriverSub and OncodriveCLUST rather than MutSigCV (Fig. 4B), including 12 known driver genes such as CLTC, DDX3X, EP300 and SPEN (Sondka *et al.*, 2018). As for the genes predicted by both DriverSub and MutSigCV rather than OncodriveCLUST, there are totally 36 shared genes including known driver genes as BRCA2, NOTCH1, FGFR3, CREBBP and PTPN13 (Sondka *et al.*, 2018). In conclusion, these shared genes indicate the consistency between the predicted genes for DriverSub and those for the other existing methods.

Despite the genes predicted by both DriverSub and the existing methods, there are also some predicted genes that are unique to our proposed DriverSub. For the results on breast cancer data, there are 394 driver gene candidates predicted by exclusively DriverSub (Fig. 4A), which include 35 known driver genes. For examples, ERBB2, BRCA1, BRCA2 and CDH1 are exclusively predicted by our method, and these genes have been reported to play crucial roles in breast cancer (Cancer Genome Atlas Network and Others, 2012). Here the fraction of known driver genes in the results exclusively yielded by DriverSub is comparable or larger than the fractions of the other two competing methods. For the results on bladder cancer data, there are totally 400 genes unique to our proposed DriverSub (Fig. 4B), including 49 known driver genes such as ERBB3, LRP1B, NOTCH2 and TSC1 (Sondka *et al.*, 2018). When we investigate the fraction of known driver genes in the results unique to these competing methods, the fractions for MutSigCV and OncodriveCLUST are 4.6 and 6.1%, respectively. In comparison, DriverSub yields a fraction of 12.3%, which is larger than those of the two other methods. Note the fact that a fair number of genes are identified as drivers uniquely by one of the methods, we also explore why there are stark differences in these inferences between methods (details in Supplementary Material). Consequently, DriverSub can successfully predict a bunch of driver genes that are missed by the existing methods.

3.2.3 Analysis of subgroup-specific driver genes

The major distinction between our proposed DriverSub and the existing methods is that DriverSub can infer subgroup-specific driver genes, and the existing methods mainly focus on the prediction of recurrently mutated driver genes. When compared with recurrently mutated driver genes, subgroup-specific driver genes are mutated in only a small fraction of samples, which lead to relatively low frequencies of mutations (Cyll *et al.*, 2017). Therefore, we further analyze the mutation frequencies of the driver genes predicted by DriverSub. For PIK3CA and TP53 which are predicted by both DriverSub and other existing methods on breast cancer data, the mutation frequencies of the two genes are 41.03 and 39.05%, respectively. In comparison, the mutation frequencies of driver genes exclusively predicted by our proposed DriverSub are much lower than those of the shared predictions (Supplementary Fig. S8A), where the frequencies are 1.78% for ERBB2, 4.14% for BRCA1, 4.93% for BRCA2 and 6.71% for CDH1. For the results on bladder cancer data, we can also observe similar phenomenon that the predicted genes unique to DriverSub show relatively lower mutation frequencies than those of the genes shared by both DriverSub and other existing methods (Supplementary Fig. S8B). Therefore, our proposed DriverSub can efficiently predict driver genes mutated in only a small fraction of samples.

We should note that predicting subgroup-specific driver genes is not equivalent to detecting rarely mutated genes. Admittedly, due to the fact that subgroup-specific driver genes are mutated in only a small fraction of samples, many of them show low mutation frequencies. Thus, by investigating the lowly frequently mutated genes, we can confirm whether this method satisfies the necessary conditions for prediction subgroup-specific driver genes. For a fair comparison, we also check the mutation frequencies of driver genes that are exclusively predicted by the other methods, demonstrating that not only our proposed DriverSub but also the other two methods, can successfully detect driver genes with relatively low frequencies (Supplementary Fig. S9), where the number of lowly frequently mutated driver genes for DriverSub is generally comparable with those for MutSigCV and OncodriveCLUST. Consequently, compared with the capability of identifying rare mutated genes, the main contribution of DriverSub to the detection of subgroup-specific driver genes is the ability to infer the subgroup indication of the tested genes.

When we further analyze the related subgroups of the known driver genes predicted by DriverSub, we can observe that the coordinate values of output vectors $\boldsymbol{Z}$ of the driver genes vary distinctly among the different subgroup (Fig. 5A and B). This result indicates the necessity of inferring subgroup-specific driver genes for heterogeneous cancer samples. Taking the result on breast cancer data as an example (Fig. 5A), there is only one large coefficient in the output vector of ERBB2, indicating that ERBB2 gene is frequently mutated in only one subgroup, rather than in all the subgroups. For the MTOR gene, the related output vector includes two relatively large scores, which represents that MTOR is mutated in two subgroups of the investigated samples. Likewise, according to the results yielded by DriverSub on bladder cancer data (Fig. 5B), driver genes such as GATA1, CBL and HOXC11 are mutated in only one subgroup rather than in all subgroups. Genes such as AKAP9, APC, CDKN2A and SF3B1 are mutated in two subgroups of the samples. In comparison, the recurrently mutated driver gene TP53 demonstrates large coefficients in all the dimensions of its related output vector, which indicates that TP53 is highly mutated in all subgroup of the bladder cancer samples. Based on these phenomena aforementioned,

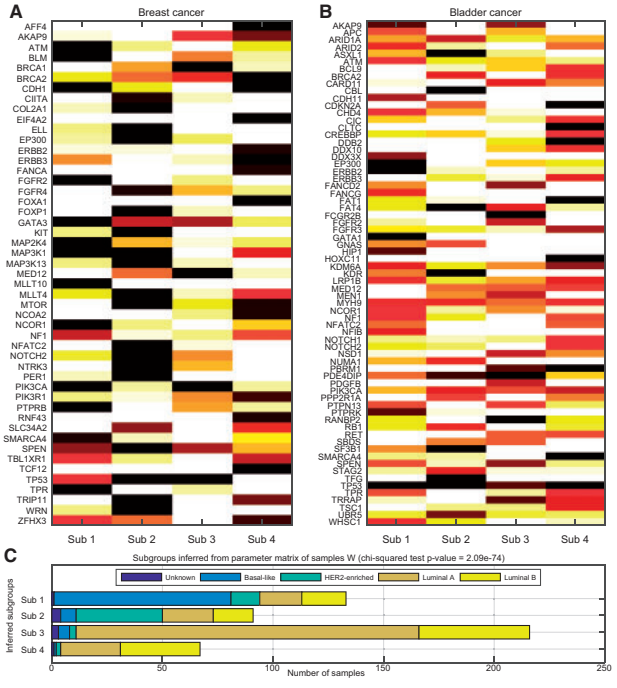

Fig. 5. The heatmaps of output vectors for experimentally validated driver genes, yielded by DriverSub on real datasets of (**A**) breast cancer and (**B**) bladder cancer. (**C**) Composition of inferred subgroups in terms of PAM50 subgroups of breast cancer, where the subgroups are inferred from parameter matrix of samples $\boldsymbol{W}$. In the heatmaps, darker color indicates higher coordinate values of output vectors $\boldsymbol{Z}$. The values in output vectors demonstrate that a bunch of driver genes are mutated in only a fraction of subgroups rather than in all subgroups. (Color version of this figure is available at *Bioinformatics* online.)

we can conclude that our proposed DriverSub can efficiently infer the subgroup specificities of the predicted driver genes.

We further use DriverSub to infer subgroups of the TCGA samples of breast cancer, of which the annotated molecular subgroups are available. For the application on DNA mutation data, the output parameter matrix $\boldsymbol{W}$ learned by DriverSub can provide information of the relationship between inferred subgroups and individual samples. By simply retrieving the maximum elements of the parameter matrix of samples $\boldsymbol{W}$, we can infer DNA mutation-based subgroups of the investigated samples. Through statistical analysis of chi-squared test, we can observe that the DNA mutation-based subgroups inferred by DriverSub are significantly associated with the annotated breast cancer subgroups (P-value = 5.59e-11, details in Supplementary Fig. S10). Nevertheless, we should note that the subgroup annotations of TCGA breast cancer is defined by PAM50 (Prediction Analysis of Microarray 50) RNA expressions (Cancer Genome Atlas Network and Others, 2012), and RNA expression based subgroups are distinct from DNA mutation-based subgroups to a certain degree (Hofree *et al.*, 2013). Since there are little studies on DNA mutation-based molecular subgroups for breast and bladder cancers (Hofree *et al.*, 2013), there is still a lack of the evidence of direct assessment of DNA mutation-based subgroup inferred by DriverSub. To circumvent this shortage, we further used the learned parameter matrix of samples $\boldsymbol{W}$ to train a map between the two types of subgroups (details in Supplementary Material), which is used to assess whether the inferred subgroups can be mapped back to the annotated molecular subgroups. The results demonstrate that the subgroups inferred through the map from parameter matrix of DriverSub are highly significantly associated with the annotated PAM50 subgroups (shown in Fig. 5C), indicating that the output of DriverSub is highly informative for inferring the associated subgroups of driver genes.

3.2.4 Predictions by integrating copy number variations

Since there are a number of cancer types such as breast cancer, are predominantly copy number driven, we further add copy number variation (CNV) data into the analysis. When compared with MutSigCV and OncodriveCLUST that are not compatible with the format of CNV data as their inputs, integrating CNV data into our proposed DriverSub is rather straight forward. Accordingly, we also apply DriverSub on data integrated of both mutations and CNVs (details in Supplementary Material), and the predicted driver genes are demonstrated in Supplementary Tables S7 and S8 for breast cancer and bladder cancer, respectively. Taking breast cancer as an example, when we evaluate whether the results are enriched for known driver genes, DriverSub's result on both mutations and CNVs yields a P-value of 8.36e-12 by Fisher's exact test, which is more significant than that of the results on only mutations. For bladder cancer, the corresponding P-value of the result on both mutations and CNVs is 6.09e-15, indicating the significance of enrichment for known drivers. Specifically, some well-known drivers such as ERBB2 and BRCA1/2, are more frequently altered by CNVs rather than mutations in breast cancers. For example, gene ERBB2 is altered in 45.2% breast cancer samples in this case, which may also be responsible for its higher predicted rank than that in the case of only mutations (rank = 42 versus rank = 438). Moreover, the analysis of subgroup-specific driver genes on the data of both mutations and CNVs also shows that DriverSub can predict driver genes with distinct patterns across the investigated samples (Supplementary Fig. S11) and inferring subgroups closely associated with the recorded PAM50 defined subgroups of breast cancer (Supplementary Fig. S12), indicating its efficiency in subgroup-specific driver inference.

4 Discussion

Inferring subgroup-specific driver genes is crucial for the understanding of cancer heterogeneity and the development of precision medicine. When the subgroup annotations of cancer samples are unavailable, the existing methods are not capable of detecting subgroup-specific driver genes from heterogeneous caner samples. Thus, it is an imperative task to infer subgroup-specific driver genes in the situation where the annotations of subgroups are unknown. To predict subgroup-specific driver genes from heterogeneous cancer data, we propose a subspace learning-based method called DriverSub to simultaneously predict driver genes and their subgroup specificities from mutation data of genes. Through the comparison with existing methods on simulation datasets, the results of DriverSub demonstrate not only better predictions of driver genes, but also more accurate indications of the subgroup specificities of genes. The applications of our method on real heterogeneous cancer data also illustrate that the results of DriverSub are highly enriched for experimentally validated known driver genes. When assessed by known molecular subgroups of breast cancers, the subgroups inferred by DriverSub display significant association with the annotated subgroups. In summary, DriverSub show effective capability of predicting subgroup-specific driver genes.

There are three potential reasons which might be considered to be responsible for the remarkable performance of DriverSub. The first aspect is that subspace learning framework can transform the high-dimensional discrete mutation data of genes into the low-dimensional vectors with continuous values, where the vectors with continuous values are more suitable for numerical analysis than the discrete data of mutations (Yang *et al.*, 2019). Also, compared with high-dimensional mutation data, the transformed low-dimensional vectors can circumvent the problem caused by the curse of dimensionality (Wang *et al.*, 2016). The second aspect is that subspace learning can adopt the dimensions in the subspace to indicate the subgroup specificities of genes. By regarding each dimension in the subspace as subgroups of cancer, the coordinate values in different dimension are adopted to indicate the related subgroups of driver genes. The third aspect is that the distances between output vectors and the origin of the subspace coordinate can demonstrate the potential of the investigated genes to be driver genes, which can be used as a metric to predict driver genes.

Despite the distinguished performance achieved by DriverSub on subgroup-specific driver gene prediction, there are still many opportunities to improve the inferring of subgroup-specific driver genes from heterogeneous cancers. First, DriverSub only incorporate the information from mutation data, and the information beyond mutations such as copy number alternations, transcriptome and epigenome can also be integrated into a model of multi-omics (Peng *et al.*, 2019; Shi *et al.*, 2017). Second, as the cancer samples accumulated through time, we can expand the datasets to include more than thousands of samples in the future, which can offer a more comprehensive view of the subgroups in heterogeneous cancers (Liu and Zhang, 2015; Weinstein *et al.*, 2013). Finally, in addition to subgroup-specific driver genes, there are also some other factors such as personalized drug response that can also contribute to the development of precision medicine (Ding *et al.*, 2016; Yang *et al.*, 2019), which can also be investigated in future work.

Acknowledgements

We would like to thank Junping Li and Yifan Hao for their assistance in preparing some of figures.

Funding

This work was supported by the National Natural Science Foundation of China [61901322, 61971393, 61871361, 61571341 and 61571414], in part by the Natural Science Foundation of Shaanxi, China [Grant 2019JC-13], in part by the Natural Science Foundation of Guangdong, China [Grant 2017A030312006], and in part by Science and Technology Program of Guangzhou [Grant 201704020134].

Conflict of Interest: none declared.

References

Alizadeh,A.A. *et al.* (2015) Toward understanding and exploiting tumor heterogeneity. *Nat. Med.*, **21**, 846–853.

Bailey,M.H. *et al*. (2018) Comprehensive characterization of cancer driver genes and mutations. *Cell*, **173**, 371–385.
Cai,D. *et al*. (2011) Graph regularized nonnegative matrix factorization for data representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, **33**, 1548–1560.
Cancer Genome Atlas Network and Others. (2012) Comprehensive molecular portraits of human breast tumours. *Nature*, **490**, 61.
Cancer Genome Atlas Research Network and Others (2014) Comprehensive molecular characterization of urothelial bladder carcinoma. *Nature*, **507**, 315.
Candes,E.J. and Tao,T. (2005) Decoding by linear programming. *IEEE Trans. Inform. Theory*, **51**, 4203–4215.
Carter,H. *et al*. (2009) Cancer-specific high-throughput annotation of somatic mutations: computational prediction of driver missense mutations. *Cancer Res*., **69**, 6660–6667.
Chen,X. *et al*. (2018) Predicting miRNA–disease association based on inductive matrix completion. *Bioinformatics*, **34**, 4256–4265.
Cyll,K. *et al*. (2017) Tumour heterogeneity poses a significant challenge to cancer biomarker research. *Br. J. Cancer*, **117**, 367–375.
Dagogojack,I. and Shaw,A.T. (2017) Tumour heterogeneity and resistance to cancer therapies. *Nat. Rev. Clin. Oncol*., **15**, 81–94.
De,S. and Ganesan,S. (2017) Looking beyond drivers and passengers in cancer genome sequencing data. *Ann. Oncol*., **28**, 938–945.
Ding,Z. *et al*. (2016) Evaluating the molecule-based prediction of clinical drug responses in cancer. *Bioinformatics*, **32**, 2891–2895.
Gao,J. *et al*. (2013) Integrative analysis of complex cancer genomics and clinical profiles using the cbioportal. *Sci. Signal*, **6**, pl1–pl1.
Hofree,M. *et al*. (2013) Network-based stratification of tumor mutations. *Nat. Methods*, **10**, 1108.
Hou,Y. *et al*. (2018) MaxMIF: a new method for identifying cancer driver genes through effective data integration. *Adv. Sci*., **5**, 1800640.
Hudson,T.J. *et al*. (2010) International network of cancer genome projects. *Nature*, **464**, 993–998.
Hyvärinen,A. (2013) Independent component analysis: recent advances. *Philos. Trans. A Math. Phys. Eng. Sci*., **371**, 20110534.
Jing,F. *et al*. (2019) An integrative framework for combining sequence and epigenomic data to predict transcription factor binding sites using deep learning. *IEEE/ACM Trans. Comput. Biol. Bioinform*, doi: 10.1109/TCBB.2019.2901789.
Jolliffe,I.T. and Cadima,J. (2016) Principal component analysis: a review and recent developments. *Philos. Trans. A Math. Phys. Eng. Sci*., **374**, 20150202.
Lawrence,M.S. *et al*. (2013) Mutational heterogeneity in cancer and the search for new cancer-associated genes. *Nature*, **499**, 214–218.
Li,Z. *et al*. (2015) Robust structured subspace learning for data representation. *IEEE Trans. Pattern Anal. Mach. Intell*., **37**, 2085–2098.
Liu,Z. and Zhang,S. (2015) Tumor characterization and stratification by integrated molecular profiles reveals essential pan-cancer features. *BMC Genomics*, **16**, 503.
Meyerson,M. *et al*. (2010) Advances in understanding cancer genomes through second-generation sequencing. *Nat. Rev. Genet*., **11**, 685–696.
Mukohara,T. (2015) Pi3k mutations in breast cancer: prognostic and therapeutic implications. *Breast Cancer*, **7**, 111.
Peng,C. *et al*. (2019) Capsule network based modeling of multi-omics data for discovery of breast cancer-related genes. *IEEE/ACM Trans. Comput. Biol. Bioinform*.
Pereira,B. *et al*. (2016) The somatic mutation profiles of 2, 433 breast cancers refines their genomic and transcriptomic landscapes. *Nat. Commun*., **7**, 11479–11479.
Ramirez,C. *et al*. (2013) Why l1 is a good approximation to l0: a geometric explanation. *J. Uncertain Syst*., **7**, 203–207.
Shi,Q. *et al*. (2017) Pattern fusion analysis by adaptive alignment of multiple heterogeneous omics data. *Bioinformatics*, **33**, 2706–2714.
Siegel,R.L. *et al*. (2019) Cancer statistics, 2019. *CA Cancer J. Clin*., **69**, 7–34.
Sondka,Z. *et al*. (2018) The COSMIC Cancer Gene Census: describing genetic dysfunction across all human cancers. *Nat. Rev. Cancer*, **18**, 696–705.
Subramanian,A. *et al*. (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA*, **102**, 15545–15550.
Tamborero,D. *et al*. (2013) OncodriveCLUST: exploiting the positional clustering of somatic mutations to identify cancer genes. *Bioinformatics*, **29**, 2238–2244.
Tan,H. *et al*. (2012) A novel missense-mutation-related feature extraction scheme for 'driver' mutation identification. *Bioinformatics*, **28**, 2948–2955.
Tokheim,C. *et al*. (2016) Evaluating the evaluation of cancer driver genes. *Proc. Natl. Acad. Sci. USA*, **113**, 14330–14335.
Tomczak,K. *et al*. (2015) The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge. *Contemp. Oncol. (Pozn)*, **19**, A68.
Vogelstein,B. *et al*. (2013) Cancer genome landscapes. *Science*, **339**, 1546–1558.
Wang,K. *et al*. (2016) Joint feature selection and subspace learning for cross-modal retrieval. *IEEE Trans. Pattern Anal. Mach. Intell*., **38**, 2010–2023.
Weinstein,J.N. *et al*. (2013) The cancer genome atlas pan-cancer analysis project. *Nat. Genet*., **45**, 1113.
Yang,J. *et al*. (2019) A novel approach for drug response prediction in cancer cell lines via network representation learning. *Bioinformatics*, **35**, 1527–1535.
Yin,Y. *et al*. (2002) p53 stability and activity is regulated by mdm2-mediated induction of alternative p53 translation products. *Nat. Cell Biol*., **4**, 462.
Yu,Z. *et al*. (2014) CLImAT: accurate detection of copy number alteration and loss of heterozygosity in impure and aneuploid tumor samples using whole-genome sequencing data. *Bioinformatics*, **30**, 2576–2583.
Zhang,J. and Zhang,S. (2017) Discovery of cancer common and specific driver gene sets. *Nucleic Acids Res*., **45**, e86–86.
Zheng,R. *et al*. (2019) Sinnlrr: a robust subspace clustering method for cell type detection by nonnegative and low rank representation. *Bioinformatics*, **35**, 3642–3650.
Zhou,T. and Tao,D. (2013) Double shrinking sparse dimension reduction. *IEEE Trans. Image Process*., **22**, 244–257.