Briefings in Bioinformatics, 2017, 1–13

doi: 10.1093/bib/bbx119
Paper

Principal component analysis of binary genomics data

Yipeng Song, Johan A. Westerhuis, Nanne Aben, Magali Michaut, Lodewyk F. A. Wessels and Age K. Smilde

Corresponding author: Yipeng Song, Swammerdam Institute for Life Sciences, University of Amsterdam, Amsterdam, GE 1090, The Netherlands. Tel.: +31 (0)20 525 7638; E-mail: y.song2@uva.nl

Abstract

Motivation: Genome-wide measurements of genetic and epigenetic alterations are generating more and more high-dimensional binary data. The special mathematical characteristics of binary data make the direct use of the classical principal component analysis (PCA) model to explore low-dimensional structures less obvious. Although there are several PCA alternatives for binary data in the psychometric, data analysis and machine learning literature, they are not well known to the bioinformatics community. **Results:** In this article, we introduce the motivation and rationale of some parametric and nonparametric versions of PCA specifically geared for binary data. Using both realistic simulations of binary data as well as mutation, CNA and methylation data of the Genomic Determinants of Sensitivity in Cancer 1000 (GDSC1000), the methods were explored for their performance with respect to finding the correct number of components, overfit, finding back the correct low-dimensional structure, variable importance, etc. The results show that if a low-dimensional structure exists in the data, that most of the methods can find it. When assuming a probabilistic generating process is underlying the data, we recommend to use the parametric logistic PCA model, while when such an assumption is not valid and the data are considered as given, the nonparametric Gifi model is recommended.
Availability: The codes to reproduce the results in this article are available at the homepage of the Biosystems Data Analysis group (www.bdagroup.nl).

Key words: binary data; dimension reduction; logistic PCA; nonlinear PCA; optimal scaling; PCA

Introduction

Binary measurements have only two possible outcomes, such as presence and absence, or true and false, which are usually labeled as '1' and '0'. In many research problems, objects are characterized by multiple binary features, each depicting a different aspect of the object. In biological research, several examples of binary data sets can be found. Genome-wide measurements of genetic and epigenetic alterations are generating more and more high-dimensional binary data [1, 2]. One example is the high-throughput measurements of point mutation. Here, a feature is labeled as '1' when it is classified as mutated in a sample, '0' when it is not. Another often observed binary data set is the copy number aberrations (CNAs), which are gains

Yipeng Song is a PhD candidate in Biosystems Data Analysis group at the University of Amsterdam. His research is focused on the development of data fusion methods for continuous and binary biological data sets.
Johan A. Westerhuis is an assistant professor of Biosystems Data Analysis group at the University of Amsterdam. His current research focuses on the development of methods for the analysis of complex data obtained from metabolomics studies.
Nanne Aben is a PhD student in the Computational Cancer Biology group, Division of Molecular Carcinogenesis, Netherlands Cancer Institute, Amsterdam. His research focuses on the identification of biomarkers of drug (combination) response using multiple molecular data types.
Magali Michaut is a staff scientist in the Computational Cancer Biology group, Division of Molecular Carcinogenesis, Netherlands Cancer Institute, Amsterdam. Her research focuses on data integration and includes molecular subtyping, driver gene prioritization, drug/treatment response prediction and drug synergy biomarker identification.
Lodewyk F. A. Wessels is a group leader of the Computational Cancer Biology group, Division of Molecular Carcinogenesis, Netherlands Cancer Institute, Amsterdam. His research is focused on understanding the characteristics and vulnerabilities of cancer cells, how these are regulated and how they affect treatment response.
Age K. Smilde is a professor of Biosystems Data Analysis at the University of Amsterdam. His Biosystems Data Analysis group has a wide expertise in analyzing complex functional genomics data, and they work closely together with biologists.
Submitted: 22 July 2017; **Received (in revised form):** 19 August 2017

and losses of segments in chromosomal regions. Segments are labeled as '1' when aberration is present in a sample, otherwise '0' [3]. DNA methylation data can also be discretized as binary features, where '1' indicates a high level of methylation and '0' means a low level [2].

Compared with commonly used numerical data, binary data have some special mathematical characteristics, which should be taken into account during the data analysis. In binary measurements, '0' and '1' are abstract representations of two exclusive categories rather than numerical values 0 and 1. These two categories can also be encoded to any other two different labels, like '-1' and '1' or '-' and '+', without changing the meaning. Because '1' and '0' are only an abstract representation of two categories, they cannot be taken interpreted as numerical data. Furthermore, the measurement error of binary data is discrete in nature. Binary measurement error occurs when a category is assigned to the wrong label, such as when a mutated gene is misclassified as wild type. Therefore, the by default used Gaussian error assumption for continuous data in many statistical models is inappropriate for binary data analysis. Another aspect of binary data is that there can be an order in the two categories. For example, presence is often considered more important than absence. Finally, binary data can be generated from a discrete measurement process, but also a continuous measurement process [4].

PCA is one of the most popular methods in dimension reduction with numerous applications in biology, chemistry and many other disciplines [5]. PCA can map data points, which are in a high-dimensional space, to a low-dimensional space with minimum loss of variation. The derived low-dimensional features, which provide a parsimonious representation of the original high-dimensional data, can be used in data visualization or for further statistical analysis.

Classical linear PCA methods are appropriate for numerical data. The direct use of linear PCA on binary data does not take into account the distinct mathematical characteristics of binary data. Although there are several parametric and nonparametric PCA alternatives for binary data in the psychometric, data analysis and machine learning literature, they are not well known to the bioinformatics community. In this article, we are going to introduce, compare and evaluate some of these different approaches.

First, the theory of the different approaches is introduced together with their model properties and how the different models are assessed. Then, we will introduce three binary genomics data sets on which the models will be applied. Besides the real data, realistic simulations of binary data are used to uncover some of the properties of the different models.

Theory

There exist two separate directions in extending PCA for binary data: parametric and nonparametric. Parametric approaches are represented by logistic PCA methods, originating from the machine learning literature. In these methods, PCA is extended to binary data from a probabilistic perspective in a similar way, as linear regression is extended to logistic linear regression [6–8]. Also, a sparse extension [9] is available and has been applied into a genome-wide association study [10]. Nonparametric methods, originating from the psychometric and data analysis communities, include optimal scaling [11], multiple correspondence analysis [12] and many others [13]. In this direction, PCA is extended to binary data from a geometric perspective without assumptions of probability distribution.

Table 1. Notations used in the article

I	Number of samples ($i = 1 \cdots I$)
J	Number of variables ($j = 1 \cdots J$)
K	Number of components in simulation ($k = 1 \cdots K$)
R	Number of components in modeling ($r = 1 \cdots R$)
1_I	Column vector of ones ($I \times 1$)
μ	Column offset term ($J \times 1$)
I_R	Identity matrix ($R \times R$)
X	Binary data matrix ($I \times J$)
P	Probability matrix ($I \times J$)
Θ	Log-odds matrix ($I \times J$)
A	Score matrix ($I \times R$)
B	Loading matrix ($J \times R$)

The details for the motivation and rationale of above approaches for binary data will be explained later in this section. We will start by introducing classical PCA.

Notation

Notations used in this article are shown in Table 1.

Classical PCA

Classical PCA can be expressed as a projection-based approach following Pearson [14], finding the low-dimensional space that best represents a cloud of high-dimensional points. Suppose that the low-dimensional space is spanned by the columns of a rank R orthogonal loading matrix $B(J \times R)$, $R \ll \min(I, J)$. The orthogonal projection of a high-dimensional point x on this low-dimensional space is $BB^T x$. We find B by minimizing the Euclidean distance between x and its low-dimensional projection:

$$\begin{aligned} \min_{\mu,\mathbf{B}} \quad & \sum_{i}^{I} (\mathrm{x_i} - \mu - BB^T(\mathrm{x_i} - \mu))^2 \\ \text{subject to} \quad & B^T B = \mathrm{I_R} \end{aligned} \tag{1}$$

$\mathrm{x_i}$, $i = 1 \cdots I$, the ith row of matrix X, is the J-dimensional measurement of the ith object; μ is the column offset. The exact position of the ith data point in this low-dimensional space is represented by its R-dimensional score vector $\hat{\mathrm{a}}_\mathrm{i}$, $\hat{\mathrm{a}}_\mathrm{i} = \hat{B}^T(\mathrm{x_i} - \hat{\mu})$, where $\hat{B}$ and $\hat{\mu}$ are the estimated values of Equation (1). In matrix form, we have $\hat{A} = (\mathrm{X} - 1_I\hat{\mu}^T)\hat{B}$, $\hat{\mathrm{a}}_\mathrm{i}$ is the ith row of $\hat{A}$; 1_I is an I-dimensional vector of ones; the estimated offset $\hat{\mu}$ contains the column means of X, and $\mathrm{X} - 1_I\hat{\mu}^T$ is the column-centered X.

Another approach to explain PCA is the reconstruction-based approach [15]. A high-dimensional data point $\mathrm{x_i}$ is approximated by a linear function of the latent low-dimensional score $\mathrm{a_i}$ with orthogonal coefficients B, $\mathrm{x_i} \approx \mu + B\mathrm{a_i}$, $B^T B = \mathrm{I_R}$, μ is the offset term. Now, μ, $\mathrm{a_i}$ and B can be found simultaneously by minimizing the Euclidean distance between $\mathrm{x_i}$ and its low-dimensional linear approximation $\mu + B\mathrm{a_i}$:

$$\begin{aligned} \min_{\mu,\mathbf{a_i},\mathbf{B}} \quad & \sum_{i}^{I} (\mathrm{x_i} - \mu - B\mathrm{a_i})^2 \\ \text{subject to} \quad & B^T B = \mathrm{I_R} \end{aligned} \tag{2}$$

It is well known that the above two approaches for classical PCA are equivalent and the global optimal solution can be obtained from the R truncated singular value decomposition (SVD) of centered X [16]. The solution $\hat{\mu}$ contains the column means of X; $\hat{A}$ is the product of first R left singular vectors and the

diagonal matrix of first R singular values; $\widehat{B}$ contains the first R right singular vectors.

Above, the classical PCA was derived from a geometrical perspective. Bishop *et al.* [17] have derived another explanation for PCA from a probabilistic perspective, called probabilistic PCA. A high-dimensional point x_i can be regarded as a noisy observation of the true data point θ_i, which lies in a low-dimensional space. The model can be expressed as $x_i = \theta_i + \varepsilon_i$, and $\theta_i = \mu + B\,a_i$, μ is the offset term as before; B contains the coefficients; a_i represents the low-dimensional score vector. The noise term ε_i is assumed to follow a multivariate normal distribution with 0 mean and constant variance σ^2, $\varepsilon_i \sim N(0, \sigma^2 I_J)$. Thus, the conditional distribution of x_i is a normal distribution with mean θ_i and constant variance, $x_i|\mu, a_i, B \sim N(\mu + Ba_i, \sigma^2 I_J)$. μ, a_i and B can be obtained by maximum likelihood estimation.

$$\begin{aligned}\max_{\mu,\mathbf{a_i},\mathbf{B}} \quad & \sum_i^I \log(p(x_i|\mu, a_i, B)) \\ & = \sum_i^I \log(N(x_i|\mu + Ba_i, \sigma^2 I_J)) \\ & = \sum_i^I \sum_j^J \log(N(x_{ij}|\mu_j + a_i^T b_j, \sigma^2)) \\ \text{subject to} \quad & B^T B = I_R\end{aligned} \tag{3}$$

The above maximum likelihood estimation is equivalent to the least squares minimization in classical PCA from the perspective of frequentist statistics [6]. One important implication is that all the elements in the observed matrix X are conditionally independent of each other given the offset μ, the score matrix A and the loading matrix B, which is the key point for the further extension to binary data.

Logistic PCA and logistic SVD

The probabilistic interpretation of PCA under multivariate normal distribution for the observed data provides a framework for the further generalization to other data types [17]. As the Gaussian assumption is only appropriate for continuous numerical data, it is necessary to replace the Gaussian assumption by the Bernoulli distribution for binary observations in a similar way as from linear regression to logistic linear regression [6, 8, 18]. The ijth element in observed matrix X, x_{ij}, is a realization of the Bernoulli distribution with parameter p_{ij}, which is the ijth element in the probability matrix P. Specifically, the probability that x_{ij} equals '1' is p_{ij}. Similar to probabilistic PCA, all the elements in the observed matrix X are conditionally independent of each other given the parameter matrix P. The log likelihood for observation X given the probability matrix P is as follows:

$$l(P) = \sum_i^I \sum_j^J x_{ij} \log(p_{ij}) + (1 - x_{ij}) \log(1 - p_{ij}). \tag{4}$$

The log-odds of p_{ij} is θ_{ij}, where $\theta_{ij} = \log\left(\frac{p_{ij}}{1-p_{ij}}\right)$. It is the natural parameter of the Bernoulli distribution expressed in the exponential family form. Thus, $p_{ij} = \phi(\theta_{ij}) = (1 + e^{-\theta_{ij}})^{-1}$ and $\phi()$ is called the logistic function. The log likelihood for observation X, given log-odds Θ is represented as:

$$l(\Theta) = \sum_i^I \sum_j^J x_{ij} \log(\phi(\theta_{ij})) + (1 - x_{ij}) \log(1 - \phi(\theta_{ij})). \tag{5}$$

A low-dimensional structure can be assumed to exist in the log-odds Θ, $\Theta = AB^T + 1_I \mu^T$. Here, A is the object score matrix for the log-odds Θ; B is the loading matrix; μ is the offset. In the same way as in the probabilistic PCA, the solution of logistic PCA can be obtained by maximizing the conditional log likelihood $l(\Theta)$ in Equation (5).

There are mainly two approaches to fit the model [Equation (5)], logistic SVD [18] and logistic PCA [8]. The main difference between these two approaches is whether A and B are estimated simultaneously or sequentially. In the logistic SVD model, the score matrix A and loading matrix B are estimated simultaneously by alternating minimization [6, 19] or by a majorization–minimization (MM) algorithm [18].

On the other hand, logistic PCA only estimates B directly. After B is estimated, A is obtained by a projection-based approach in the same manner as classical PCA in Equation (1) [8]. Score matrix A is the low-dimensional representation of the log-odds $\widetilde{\Theta}$ of the saturated model in the subspace spanned by B. In matrix form, $A = (\widetilde{\Theta} - 1_I \mu^T)B$, μ is the offset term. Then, the log-odds Θ in Equation (5) can be represented as $\Theta = (\widetilde{\Theta} - 1_I \mu^T)BB^T + 1_I \mu^T$. The estimation of parameters $\widehat{\mu}$ and $\widehat{B}$ can be obtained by maximizing the conditional log likelihood $l(\widetilde{\Theta})$ in Equation (5). Then, the solution for the score matrix is $\widehat{A} = (\widetilde{\Theta} - 1_I \mu^T)\widehat{B}$.

Compared with logistic SVD, logistic PCA has fewer parameters to estimate, and thus is less prone to overfitting. In addition, the estimation of the scores of new samples in logistic SVD involves an optimization problem, while for logistic PCA, it is a simple matrix multiplication of new data and the loading matrix $\widehat{B}$.

In the saturated model used in the logistic PCA, there is a separate parameter for every individual observation. For binary data, the parameter (probability of success) of the observation '1' is 1, and the parameter of the observation '0' is 0. Thus, the ijth element in $\widetilde{\Theta}$ from the saturated model is $\tilde{\theta}_{ij} = \log\left(\frac{x_{ij}}{1-x_{ij}}\right)$. It is negative infinity when $x_{ij} = 0$; positive infinity when $x_{ij} = 1$. To project $\widetilde{\Theta}$ onto the low-dimensional space spanned by B, one needs a finite $\widetilde{\Theta}$. In logistic PCA, positive and negative infinities in $\widetilde{\Theta}$ are approximated by large numbers m and $-m$. When m is too large, the elements in the estimated probability matrix $\widehat{P}$ for generating the binary observation X are close to 1 or 0; when m is close to 0, the elements in $\widehat{P}$ are close to 0.5. In the original paper of logistic PCA, cross-validation is used to select m [8]. In this article, we select $m = 2.94$, which corresponds to a probability of success 0.95. This can be interpreted as using probabilities 0.95 and 0.05 to approximate the probabilities 1 and 0 in the saturated model.

Theory of nonlinear PCA with optimal scaling

Another generalization of PCA to binary data is nonlinear PCA with optimal scaling (the Gifi method). This method was primarily developed for categorical data, of which binary data are a special case [11, 20]. The basic idea is to quantify the binary variables to numerical values by minimizing some loss functions. The quantified variables are then used in a linear PCA model. The jth column of X, X_{*j}, is encoded into an $I \times 2$ indicator matrix G_j. G_j has two columns, '1' and '0'. If the ith object belongs to column '1', the corresponding element of G_j is 1, or otherwise, it is 0. A is the $I \times R$ object score matrix, which is the representation of X in a low-dimensional Euclidean space; Q_j is a $2 \times R$ quantification matrix, which quantifies this jth binary variable to a numerical value. For binary data, the rank of the quantification matrix Q_j is constrained to 1. This is the PCA solution in the Gifi method. Q_j can be expressed as $Q_j = z_j w_j^T$, where z_j is a

Table 2. Orthogonality properties of the scores and loadings of the four methods

	PCA	Gifi	Logistic SVD	Logistic PCA
Score matrix *A*	D	O	D	D
Loading matrix *B*	O	D	O	O

Note: O: the columns of this matrix are orthonormal vectors, $B^TB = I_R$; D: the columns of this matrix are orthogonal vectors, $B^TB = D_R$, D_R is a $R \times R$ diagonal matrix.

two-dimensional column vector with binary quantifications, and W_j is the vector of weights for *R* principal components. The loss function is expressed as:

$$\min_{\mathbf{A},\mathbf{z}_j,\mathbf{w}_j} \sum_{j=1}^{J} (\mathrm{A} - \mathrm{G}_j \mathrm{z}_j \mathrm{w}_j)^2. \quad (6)$$

To avoid trivial solutions, the score matrix *A* is forced to be centered and orthogonal, $1_I^T A = 0$, $A^TA = I_R$. The loss function is optimized by alternating least squares algorithms. For binary data, nonlinear PCA with optimal scaling is equivalent to multiple correspondence analysis and to PCA on standardized variables [13].

Model properties

Offset

Including the column offset term μ in component models is equivalent to requiring the column mean of score matrix to be 0, i.e. $1_I^T A = 0$. In PCA and the Gifi method, the estimated $\widehat{\mu}$ equals the column mean of *X*. Therefore, including μ in the model has the same effect as column centering of *X*. In logistic PCA and logistic SVD, the *j*th element of μ, μ_j, can be interpreted as the log-odds of the marginal probability of the *j*th variable. When only the offset μ is included in the model, $\Theta = 1_I\mu^T$, the *j*th element of the solution, $\widehat{\mu}$, $\widehat{\mu}_j$, is the log-odds of the empirical marginal probability of the *j*th variable (the proportion of '1' in the *j*th column). When more components are included, $\Theta = 1_I\mu^T + AB^T$, the solution $\widehat{\mu}$ is not unique. If an identical offset is required for comparing component models with a different number of components, one can fix the offset term to the log-odds of the empirical marginal probability during the maximum likelihood estimation.

Orthogonality

Similar to PCA, the orthogonality constraint $B^TB = I_R$ in logistic PCA and logistic SVD is inactive. If *B* is not orthogonal, it can be made orthogonal by subjecting AB^T to an SVD algorithm. *B* equals the right-hand singular vectors, and *A* equals the product of the left-hand singular vectors and the diagonal matrix of singular values. This extra step will not change the objective value. Table 2 gives the orthogonality properties of the scores and loadings of the four methods discussed above.

Nestedness

Linear PCA models are nested in the number of components, which means the first *R* principal components in the $R+1$ components model are exactly the same as the *R* components model. For the Gifi method, this property only holds for the binary data case but not in general. For logistic PCA and logistic SVD, this property does not hold.

Model assessment

Error metric

To make a fair comparison between linear PCA, the Gifi method, logistic PCA and logistic SVD, the training error is defined as the average misclassification rate in using the derived low-dimensional structure to fit the training set *X*. Each of the four methods provides an estimation of the offset term, score matrix and loading matrix, $\widehat{\mu}$, $\widehat{\mathrm{A}}$ and $\widehat{\mathrm{B}}$. For linear PCA and the Gifi method, we take $1_I\widehat{\mu}^T + \widehat{\mathrm{A}}\widehat{\mathrm{B}}^T$ as an approximation of the binary matrix *X*; for logistic SVD and logistic PCA, $\phi(1_I\widehat{\mu}^T + \widehat{\mathrm{A}}\widehat{\mathrm{B}}^T)$ is used as an approximation for the probability matrix *P*, of which the observed matrix *X* was generated. As both approximations are continuous, we need to select a threshold to discretize them to binary fitting.

In the discretization process, two misclassification errors exist. '0' can be misclassified as '1', which we call err0, and '1' can be misclassified as '0', which we call err1. N_{err0} is the number of err0 in this process, and N_{err1} is the number of err1; N_0 is the number of '0' in the observed binary matrix *X*, and N_1 is the number of '1' in *X*. A commonly used total error rate is given by $(N_{err0} + N_{err1})/(N_0 + N_1)$, which gives equal weights to these two errors. However, this can lead to undesirable results for unbalanced binary data, i.e. when the proportions of '1' and '0' are extreme. Usually, unbalanced binary data sets are common in real applications, where sometimes the proportion of '1' in the observed matrix *X* can be <5%. In such a case, err0 is more likely to occur than err1, and hence, it seems inappropriate to give them equal weights. In unbalanced cases, a balanced error rate $0.5 \times (N_{err0}/N_0 + N_{err1}/N_1)$ is more appropriate [21]. To decide whether the predicted quantitative value represents a '0' or a '1', a threshold value has to be selected. This threshold value can be selected by minimizing the balanced error rate in a training set after which it can be applied to a test set to prevent biased (too optimistic) results.

Cross-validation

The training error is an overly optimistic estimator of the generalization error. It can be intuitively understood as the average misclassification rate in predicting an independent test set. Thus, we use cross-validation to approximate the generalization error. In this article, we use the cross-validation algorithm named EM-Wold [22, 23]. In this approach, validation sets of elements of the matrix *X* are selected in a diagonal style rather than a row-wise style. The left out part is considered as missing. In this way, the prediction of the left out part is independent of the left out part itself. It is possible to use this approach, as all the component models in this article can handle missing data. A 7-fold cross-validation procedure was used for all calculations in this article. In each of these folds, a component model is developed taking the missing data into account. The model is then used to make a prediction of the missing elements. This is repeated until all elements of *X* have been predicted in this way. The threshold of converting the continuous predictions to binary predictions in cross-validation was the same as the one used in computing the training error.

Data sets

Real data sets

The data we used are from the Genomic Determinants of Sensitivity in Cancer 1000 (GDSC1000) [2]. To facilitate the interpretability of the results, only three cancer types are included in

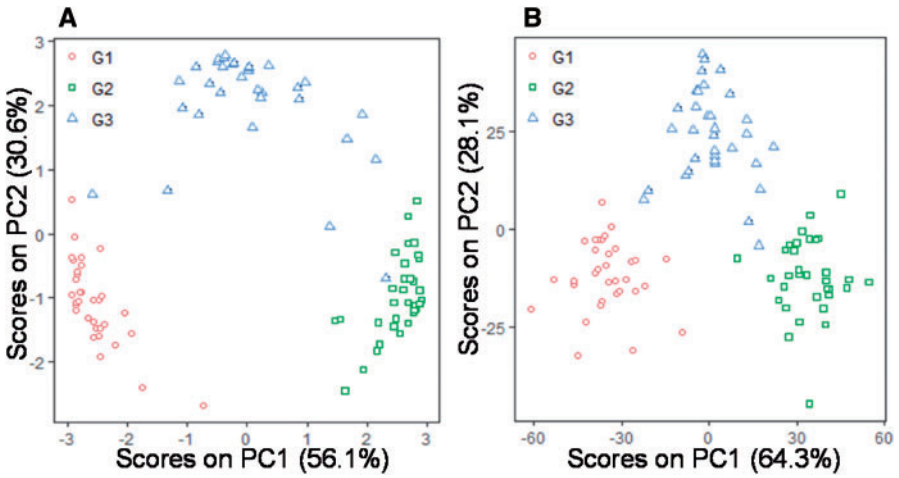

Figure 1. Score plot of the first two principal components (PCs) derived from linear PCA on the probability matrix P (**A**) and log-odds matrix Θ (**B**) used in the binary data simulation. G1, G2 and G3 are three simulated groups in the samples.

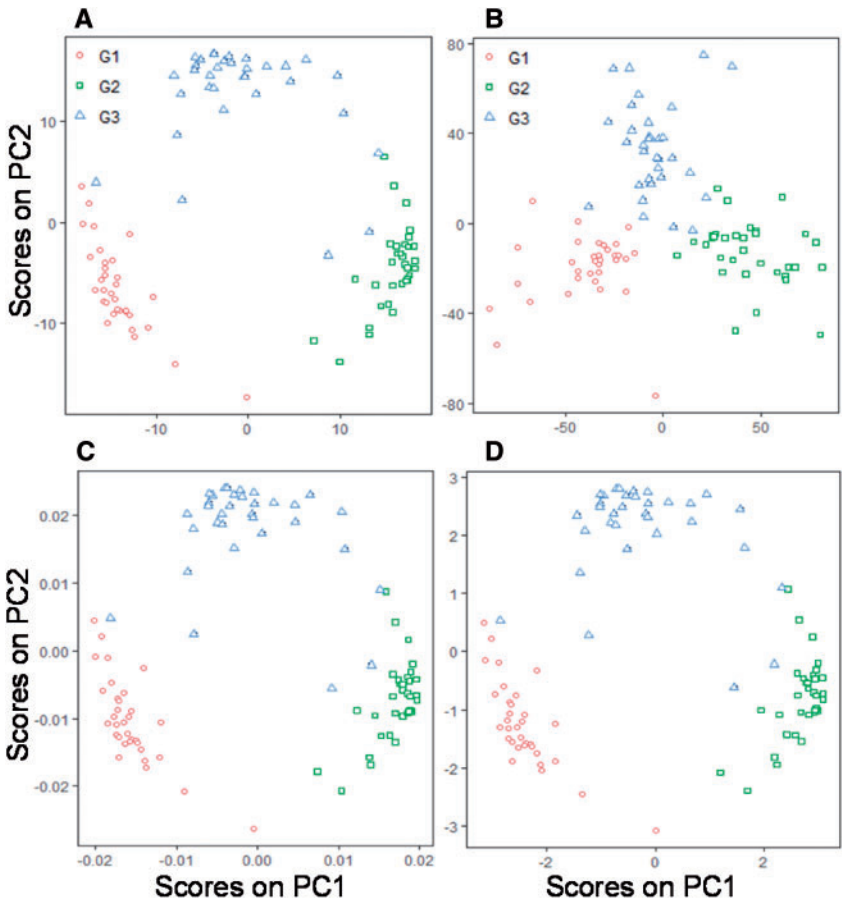

Figure 2. Score plot of first two PCs produced by the four different approaches. (**A**) logistic PCA; (**B**) logistic SVD; (**C**) the Gifi method; (**D**) PCA. G1, G2 and G3 are three simulated groups in the samples.

the data analysis: BRCA (breast invasive carcinoma, 48 human cell lines), LUAD (lung adenocarcinoma, 62 human cell lines) and SKCM (skin cutaneous melanoma, 50 human cell lines). Each cell line is a sample in the data analysis. For these samples, three different binary data sets are available: mutation, CNA and methylation data. For the mutation data, there are 198 mutation variables. Each variable is a likely cancer driver or suppressor gene. A gene is labeled as '1' when it is classified as mutated in a sample and as '0' when classified as wild type. The mutation data are sparse (Supplementary Figure S1A): roughly 2% of the data matrix is labeled as '1'. The CNA data have 410 observed CNA variables. Each variable is a copy number region in a chromosome. It is labeled as '1' for a specific sample when it is identified as aberrated and it is labeled as '0' otherwise.

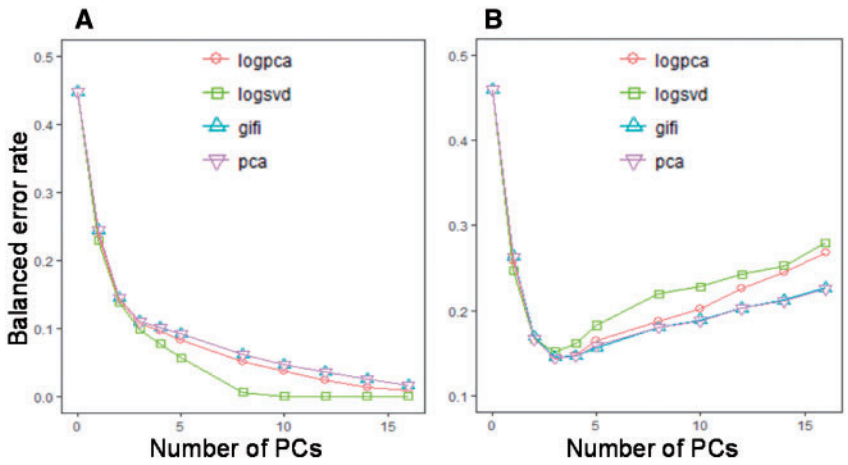

Figure 3. The balanced training error (**A**) and cross-validation error (**B**) for the balanced simulated data set produced by four different approaches with different number of components. (**A**) Training error; (**B**) cross-validation error. logpca: logistic PCA; logsvd: logistic SVD; gifi: the Gifi method; pca: linear PCA.

The CNA data set is also sparse (Supplementary Figure S1B): roughly 7% of the data matrix is labeled as '1'. For the methylation data, there are 38 methylation variables. Each variable is a CpG island located in gene promoter area. In each variable, '1' indicates a high level of methylation and '0' indicates a low level. The methylation data set is relatively balanced compared with other data sets (Supplementary Figure S1C): roughly 27% of the data matrix is labeled as '1'.

Simulated binary data sets

Binary data matrices with an underlying low-dimensional structure can be simulated either from a latent variable model or the noise corruption of a prestructured binary data set. We use both of these two approaches to study the properties of different binary PCA methods.

Simulated binary data based on a latent variable model

Data sets with different levels of sparsity and with low-dimensional structures were simulated according to the logistic SVD model. The offset term μ is used to control the sparsity, and the log-odds Θ are defined to have a low-dimensional structure. The observed binary matrix X is generated from the corresponding Bernoulli distributions.

Each element in the $J \times K$ loading matrix B is sampled from the standard normal distribution. The Gram–Schmidt algorithm is used to force $B^TB = I_K$. K is set to 3. The simulated $I \times K$ score matrix A has three group structures in the samples. I samples are divided into three groups of equal size. The three group means are set manually to force sufficient difference between the groups. For example, the first two group means are set to $a_1^* = [2, -1, 3]^T$ and $a_2^* = [-1, 3, -2]^T$. The third group mean is $a_3^* = [0, 0, 0]^T - a_1^* - a_2^*$. The scores in first group are sampled from the multivariate normal distribution $N(a_1^*, I_K)$, the scores in second group from $N(a_2^*, I_K)$ and the scores in the third group from $N(a_3^*, I_K)$. In this way, scores between groups are sufficiently different, and scores within the same group are similar.

When the elements in AB^T are close to 0, the corresponding probabilities are close to 0.5. In this case, the binary observations are almost a random guess. When their absolute values are large, the corresponding probabilities are close to 1 or 0, and the binary observations are almost deterministic. The scale of AB^T should be in a reasonable interval, not too large and not too small. A constant C is multiplied to AB^T to control the scale for generating proper probabilities. In addition, the offset term μ is included to control the level of sparsity in the simulated binary data set. $\Theta = CAB^T + 1_I\mu^T$. Then Θ is transformed to the probability matrix P by the logistic function $\phi()$, and x_{ij} in X is a realization of Bernoulli distribution with parameter p_{ij}, which is the ijth element of probability matrix P.

Simulated binary data based on noise corruption of prestructured binary data

Another approach of simulating binary data is by the noise corruption of a prestructured data set. Compared with the latent variable model, this approach provides an intuitive understanding of the low-dimensional structure in the observed binary data. Prestructured binary data set X_{true} has structured and unstructured parts. The structured part is simulated as follows. K versions of I-dimensional binary vectors are simulated first, and each element is sampled from the Bernoulli distribution with probability P, which is the level of sparsity in the binary data simulation. Each of these K binary vectors is replicated 10 times. In the unstructured part of X_{true}, all the elements are randomly sampled from Bernoulli distribution with probability P. The observed binary data X is a noise-corrupted version of the prestructured binary data set X_{true}. If the noise level is set to 0.1, all the elements in the binary data X_{true} have a probability of 0.1 to be bit-flipped. The observed binary matrix X has K groups of 10 highly correlated variables, and the other variables are not correlated. The K groups are taken as the low-dimensional structure. The above simulation process is illustrated in the Supplementary Figure S4.

Results

All the computations are done in R [24]. The linear PCA model is fitted using the SVD method after centering the data [25]. The Gifi method is fitted using the alternating least squares approach by Homals package [11]. The logistic SVD and logistic PCA models are fitted using an MM algorithm with offset term [8, 18]. The computational time of the four approaches with respect to different samples I and variables J are compared and summarized in Supplementary Table S1. The Gifi method,

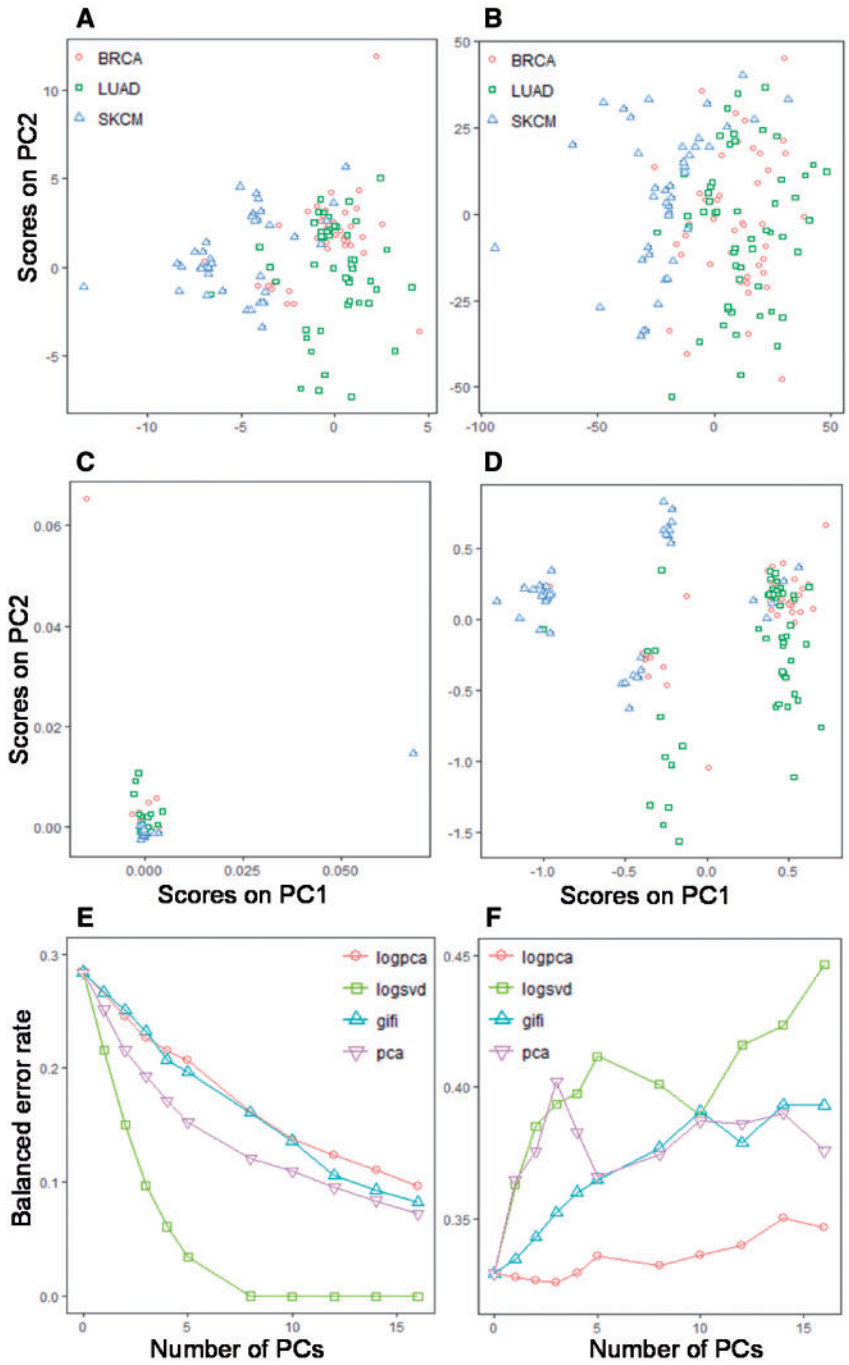

Figure 4. Score plot of the first two PCs, training and cross-validation error plot of the four different approaches for the mutation data. (**A**) Score plot of logistic PCA; (**B**) Score plot of logistic SVD; (**C**) Score plot of the Gifi method; (**D**) Score plot of PCA; (**E**) Training error plot; (**F**) Cross-validation error plot. The legend of the training error and cross-validation error plot is the same as Figure 3.

logistic SVD and logistic PCA are fast enough for real-life applications.

Balanced simulation

The goal of this simulation is to evaluate the abilities of the four approaches in finding back the embedded low-dimensional structures in the sample space and variable space. The simulation process is performed as described in 'Simulated binary data based on a latent variable model' section. The offset term μ is set to 0 to simulate balanced binary data. The parameters are set to $I = 99$, $J = 50$, $K = 3$, $C = 10$. The simulated balanced binary data are shown in Supplementary Figure S2. First, a classical PCA on the simulated probability matrix P and the log-odds Θ were performed. Figure 1 shows the score plots of these two PCA analyses. The difference between the score plots of linear PCA on P (Figure 1A) and on log-odds Θ (Figure 1B) is obvious. The scores of the linear PCA model on P lie in the margin of the figure, while for Θ, they lie more in the center of the figure. This difference is related to the nonlinear logistic function $\phi()$, which

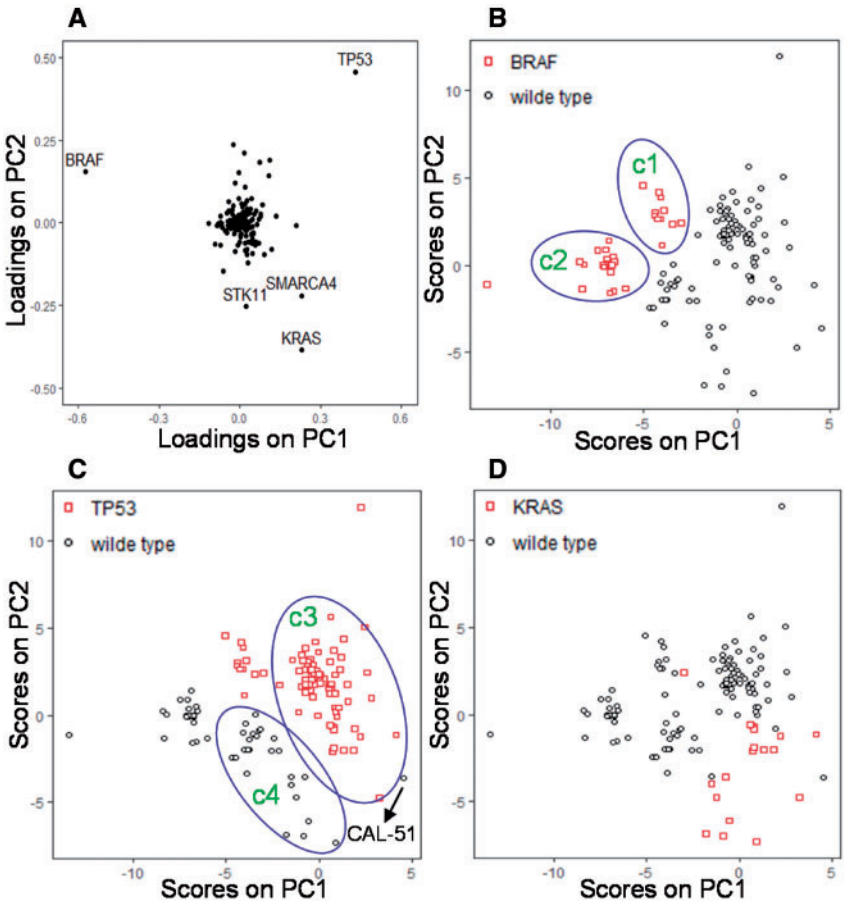

Figure 5. Loading plot (**A**) and score plots of the first two PCs derived from logistic PCA model on mutation data. The score plots (**B–D**) are labeled according to the mutation patterns. (**B**) *BRAF* mutation-labeled score plot; (**C**) *TP53* mutation-labeled score plot; (**D**) *KRAS* mutation-labeled score plot. Red square: mutated; black dot: wild type. c1, c2, c3 and c4 are the plausible four clusters in the samples on mutation data.

transforms Θ to P. Furthermore, PCA on the log-odds matrix describes more variation in the first two PCs.

Logistic PCA, logistic SVD, Gifi and linear PCA are used to model the binary matrix X. Two principal components are used. Offset terms are included in the model. The score plots produced by these different approaches are shown in Figure 2. The similarity between Figure 1 and Figure 2 indicates that the logistic SVD model approximates the underlying log-odds Θ from the binary observation X, while the other approaches approximate the probability matrix P.

Another observation is that the score plots derived from logistic PCA (Figure 2A), Gifi (Figure 2C) and linear PCA (Figure 2D) are similar except for some scale differences. The similarity between the Gifi and linear PCA for balanced binary data set is understandable. For binary data, the Gifi method is equivalent to PCA on standardized binary variables. As the proportion of '1' and '0' of each binary variable is similar in a balanced simulated data set, the column mean and SD of each binary variable are close to 0.5. Thus, the standardization of each binary variable will change 0 and 1 binary data to -1 and 1 data. Therefore, except for the difference in scale, Gifi and linear PCA are almost the same for balanced binary data. For logistic PCA, the score matrix A is a low-dimensional representation of the log-odds $\tilde{}$ from the saturated model, $A = (\tilde{\Theta} - 1_I \mu^T)B$, and the $\tilde{}$ is estimated by $2m(X - 1)$. This is equivalent to changing 0 and 1 to $-m$ and m. Thus, the true difference between linear PCA and logistic PCA is how to find the low dimension spanned by loading matrix B. Logistic PCA finds it by minimizing the logistic loss, and linear PCA finds it by minimizing the least squares loss.

The training error and CV error for different models are shown in Figure 3. We add the zero-component model in which only the offset term μ is included as the baseline for evaluating the different methods with different numbers of components. The estimated offset $\hat{\mu}$ in the zero-component model is the column mean of X for PCA and the Gifi method, while it is the logit transform of the column mean of X for logistic PCA and logistic SVD. All approaches successfully find the three components truly underlying the data. It can also be observed that logistic SVD is more eager to overfit the data. It shows a lower balanced error rate, but a higher cross-validation error rate for more than three components compared with the other methods.

Real data

The binary mutation, CNA and methylation data sets are analyzed using the four different approaches. The score plots and error plots from different approaches on the real mutation data set are shown in Figure 4. The cross-validation results of PCA, Gifi and logistic SVD in Figure 4F do not support the assumption that a low-dimensional structure exists in the mutation data. For the cross-validation result of logistic PCA (Figure 4F), the minimum cross-validation error was achieved using three components. However, this minimum was only slightly lower than the zero-component model. Although the

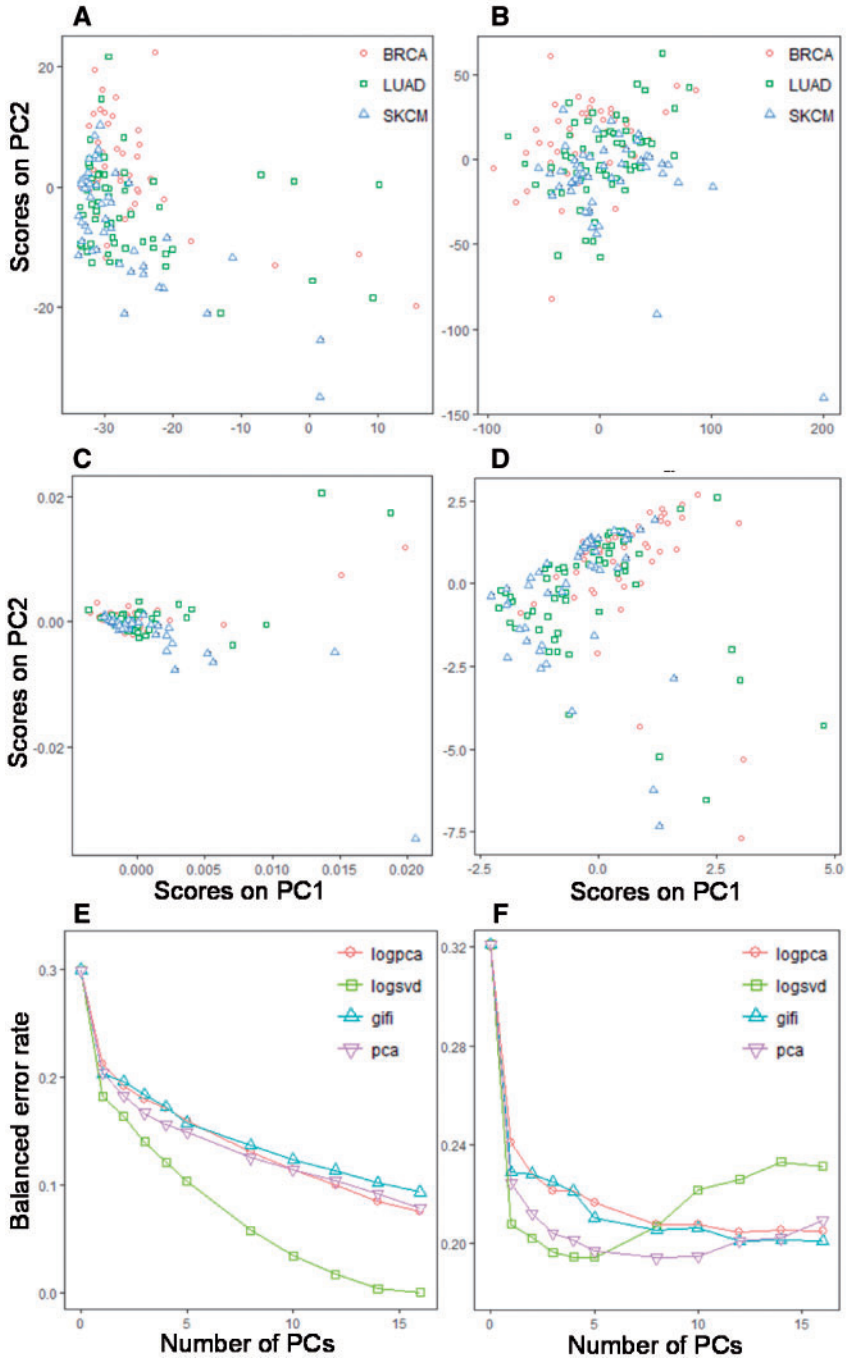

Figure 6. Score plot of the first two PCs, training and cross-validation error plot of the four different approaches for the CNA data. (**A**) Score plot of logistic PCA; (**B**) score plot of logistic SVD; (**C**) score plot of the Gifi method; (**D**) score plot of PCA; (**E**) training error plot; (**F**) cross-validation error plot. The legends for the score plot and training error plot are the same as Figure 4.

cross-validation result of logistic PCA is ambiguous, we can observe four clusters in the score plot.

To explore the clusters in more detail, Figure 5 shows the loading plot and score plots with different mutation status of the logistic PCA model. With the corresponding loading values (Figure 5A), we determined that these clusters were largely defined by *TP53*, *BRAF* and *KRAS* mutation status. Interestingly, these genes also have the highest mutational load, suggesting that variables with a higher aberration frequency contain more information. Cluster 1 (c1 in Figure 5B) is *BRAF* mutant and *TP53* mutant type; while Cluster 2 (c2 in Figure 5B) is *BRAF* mutant and *TP53* wild type. Cluster 3 (c3 in Figure 5C) mostly consists of *BRAF* wild and *TP53* mutant cell lines, a configuration that often occurs in all three analyzed cancer types. Cluster 4 (c4 in Figure 5C) contains *BRAF* wild and *TP53* wild-type cell lines, which again is a configuration that occurs across cancer types. Finally, we observed subclusters of LUAD cell lines toward the bottom of Clusters 3 and 4, which consist of *KRAS*-mutant cell lines (Figure 5D). As *BRAF* and *KRAS* mutations both activate the mitogen-activated protein kinase (MAPK) pathway in a similar

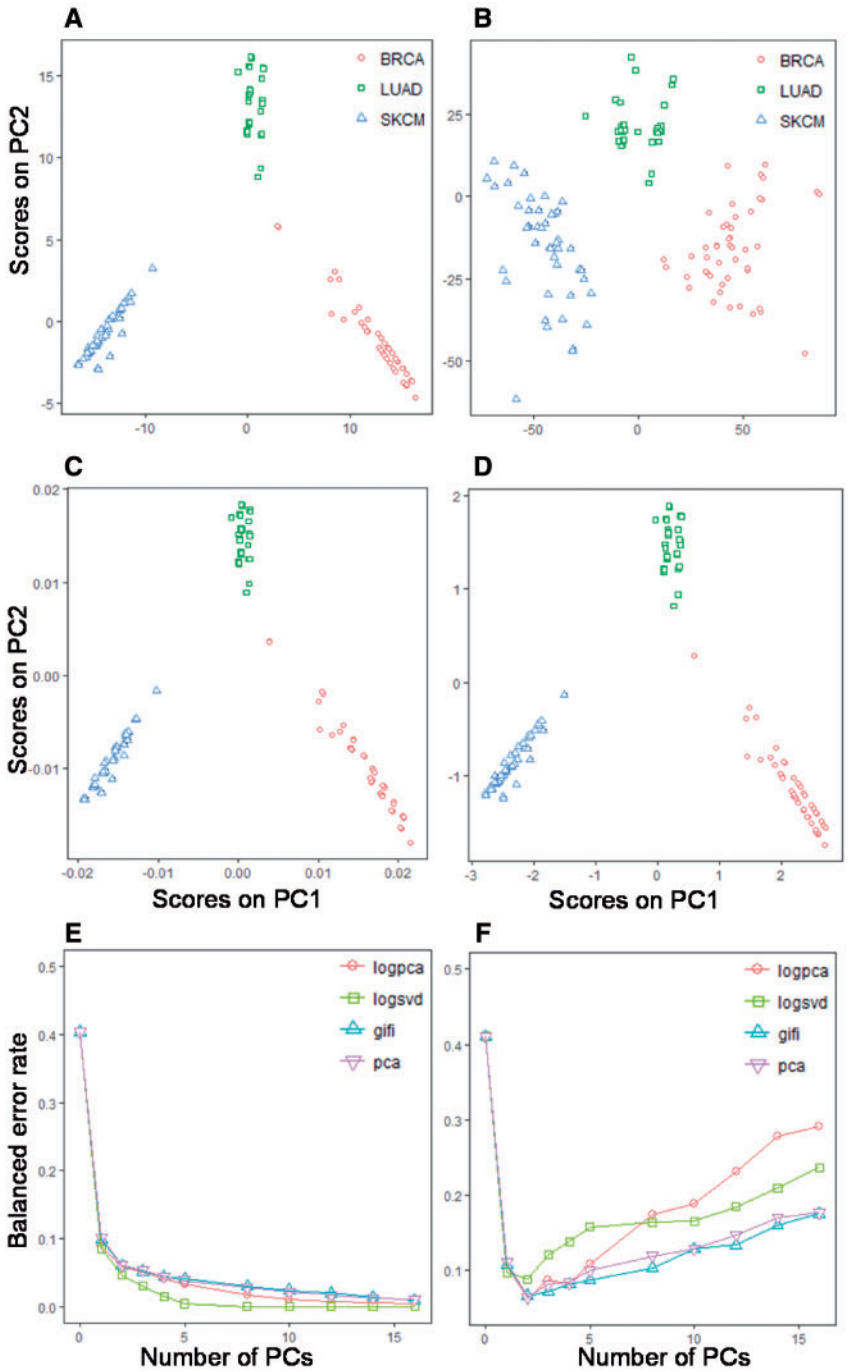

Figure 7. Score plot of the first two PCs, training and cross-validation error plot of the four different approaches for the methylation data. (**A**) Score plot of logistic PCA; (**B**) score plot of logistic SVD; (**C**) score plot of the Gifi method; (**D**) score plot of PCA; (**E**) training error plot; (**F**) cross-validation error plot. The legends for the score plot and training error plot are the same as Figure 4.

fashion, double mutants are redundant and hence rarely observed. Our results are in line with this mutual exclusivity pattern: with the exception of a single *BRAF*/*KRAS* double mutant, we find *BRAF* mutations only in Clusters 1 and 2 and *KRAS* mutations only in Clusters 3 and 4. One notable exception of the above characterization of the clusters is CAL-51 (labeled in Figure 5C). Given its *TP53* wild-type status, CAL-51 would be expected in Cluster 4, but it resides in the bottom left of Cluster 3. This shift left is likely because of mutations in both *SMARCA4* and *PIK3CA*, which have the third and fourth most negative loading values on PC1.

The score plots and error plots from different approaches on CNA data are shown in Figure 6. There is some evidence from the cross-validation results from all the models in Figure 6F for a five-dimensional structure in the data. However, in the score plots of Figure 6, the samples with different cancer types are not well separated, and there is no clear evidence of natural clusters. Therefore, we do not zoom in further on this data type.

The score plots and error plots from different approaches on methylation data are shown in Figure 7. The three cancer types are well separated in all the score plots. The similar and specific structure in the score plots of logistic PCA, the Gifi method and linear PCA may be related to the unique structure of the methylation data (Supplementary Figure S1C). Different cancer types have different methylation patterns, represented by unique sets of features. In addition, there is some evidence from the cross-validation results from all the models in Figure 7F for a two-dimensional structure in the data.

We use the score plot derived from logistic PCA model on methylation data as an example to interpret the result. The first two principal components from the logistic PCA applied to the methylation data show three clusters, which perfectly represent the three cancer types (Figure 7A). The corresponding loading values also roughly fall into three cancer-type-specific clusters (Figure 8), as most variables are exclusively nonzero in a single cancer type. Notable exceptions are *GSTM1* and *ARL17A*, which are nonzero in two cancer types, and hence, each reside between two clusters, and variables *GSTT1* and *DUSP22*, which are nonzero in all three cancer types and hence reside toward the center of the plot.

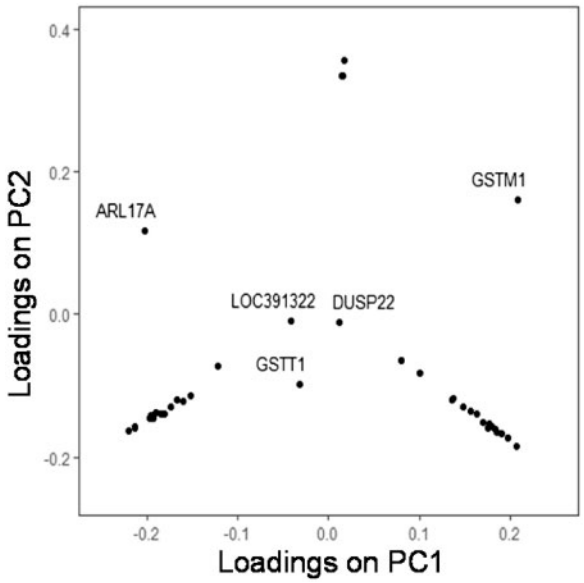

Figure 8. Loading plot of logistic PCA model on methylation data. The gene names corresponding to the methylation variables, which are interpreted in the article, are labeled on the plot.

Unbalanced simulation

The goal of the unbalanced simulation is to evaluate the effect of sparsity of simulated data on the ability of the four approaches in finding back the underlying low-dimensional structures in variable space. As the offset μ in logistic SVD model can be interpreted as the log-odds of marginal probabilities, we can use the log-odds of the empirical marginal probabilities from the real data sets with different sparsity levels as the offset in the simulation. The simulation process is performed as was described in 'Simulated binary data based on a latent variable model' section. The offset term μ is set to log-odds of column means of real data to simulate unbalanced binary data. The parameters I and J are set to the size of corresponding real data. The constant C is selected as 20, K is set to 3. The simulated data are shown in Supplementary Figure S3. We evaluate the effect of sparsity on the different models' abilities of finding back the simulated low-dimensional structure. The CV error plots of different models are shown in Figure 9. All the approaches are successful in finding back three significant PCs.

Feature selection

For the assessment of feature importance, the binary data are simulated according to the description in 'Simulated binary data based on noise corruption of prestructured binary data' section. I is 198; J is 100; K is set to 3. The sparsity level is set to 0.2, and the noise level is 0.1. The simulated data are shown in Supplementary Figure S4. There are noisy corrupted structures in the first 30 variables. For feature selection purposes, we estimate the importance of each feature in the model. This is performed as follows, $\frac{1}{3}\left(b_{j1}^2 + b_{j2}^2 + b_{j3}^2\right)$, where b_{j2} is the loading in the second PCs for jth variable. The process is repeated 15 times, and the mean and SD of the average squared loading for the 100 variables are shown in Figure 10. It can be observed that highly correlated binary variables have large loadings, and the variance of the loadings derived from logistic SVD is much higher than other approaches. This indicates that the logistic SVD model cannot make stable estimation of loadings.

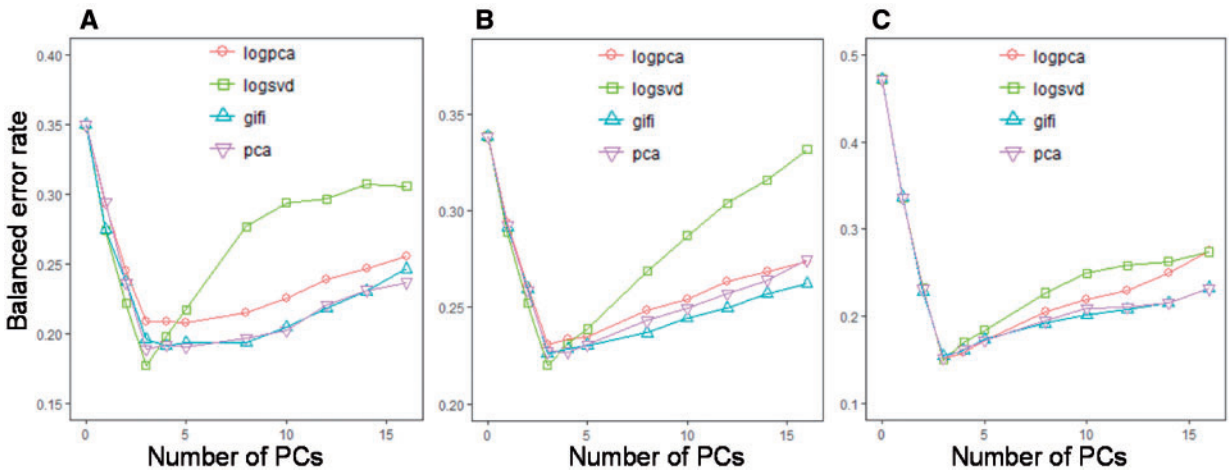

Figure 9. Cross-validation error plot for simulated unbalanced data sets with different levels of sparsity. (**A**) Sparsity similar as mutation data; (**B**) sparsity similar as CNA data; (**C**) sparsity similar as methylation data.

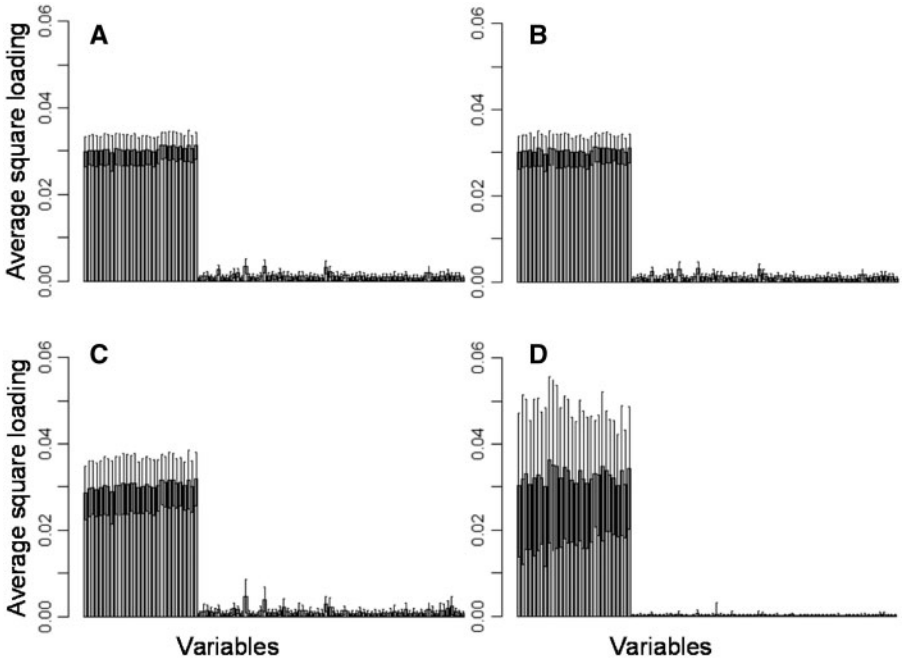

Figure 10. Barplot with 1 SD error bar of the mean square loadings of linear PCA model (**A**), the Gifi method (**B**), logistic PCA (**C**) and logistic SVD (**D**).

Discussion

In this article, four methods were discussed that all aim to explore binary data using low-dimensional scores and loadings. It was shown that each of the methods has different goals and therefore produces slightly different results. Linear PCA treats the binary data as numeric data and tries to use low rank score and loading matrices to fit the numerical data. For binary data, the quantification process in the Gifi method is simply a standardization of the binary variables. After quantification, the Gifi method tries to use low rank score and loading matrices to fit the quantified binary variables. Both logistic PCA and logistic SVD assume that the binary data follow a Bernoulli distribution, and try to find an optimal estimation of the log-odds matrix, which lies in the low-dimensional space. Logistic SVD tries to estimate the low-dimensional log-odds matrix directly, while logistic PCA estimates this matrix by the projection of the log-odds matrix from the saturated model on an approximated low-dimensional space.

For all the four approaches, it is not the level of sparsity, which is the problem, but how the sparsity is distributed over the variables. As shown in our experimental results, the low-dimensional structure of binary variables, which can be simulated from a latent variable model or by noise corruption of a prestructured binary data set, is the key issue for the results of data analysis, rather than the sparsity of the data set. When there is a low-dimensional structure in our simulation process, all the approaches can successfully find the correct number of components with different sparsity levels.

In both the analysis of the simulated data and of the real data, the performance of the linear PCA method, in the criteria of training error and cross-validation error, is similar to other specially designed algorithms for binary data. In addition, as the global optimum of the linear PCA model can always be achieved, the solution is stable. However, the linear PCA model on binary data obviously contradicts the mathematical characteristics of binary data and the assumptions of the linear PCA model itself. In addition, the fitted values, elements in the product of score and loading matrix, can only be regarded as an approximation to numerical 0 and 1, and are thus difficult to interpret.

The results of linear PCA and the Gifi method are similar, especially when the sparsity in each variable is approximately equal. Furthermore, there are signs of overfitting in the analysis of the CNA data by the Gifi model. However, compared with linear PCA, the interpretability of the Gifi method is better. The mathematical characteristics of binary data are taken into account from the geometrical perspective, and the solutions can be interpreted as an approximation of the optimally quantified binary variables.

On the other hand, logistic PCA and logistic SVD methods take into account the mathematical characteristics of binary data from the probabilistic perspective. Fitted values, elements in the product of the derived score and loading matrices, can be interpreted as the log-odds for generating the binary data, and the log-odds can again be transformed to probability. The problem for logistic SVD is that it is not robust with respect to the score and loading estimation, although it is able to select the correct number of components [18]. As both score matrix A and loading matrix B are free parameters to fit in the optimization, the estimation of AB^T will not hesitate to move to infinity to minimize the loss function. This represents itself in such a way that logistic SVD is prone to overfit (as can be seen from the cross-validation results) and the large variation in the loading estimation. The non-robustness problem is mitigated in the logistic PCA model. As only the loading matrix B is freely estimated in logistic PCA to find the optimal model, while the score matrix A is fixed, given the loadings, the logistic PCA model is less prone to overfitting. Thus, the estimation of the loadings of binary variables is more stable compared with logistic SVD. Furthermore, as the fitted values are the linear projection of the log-odds matrix of the saturated model, its scale is constrained by the scale of the approximate log-odds matrix of the saturated model, which can be specified in advance.

When assuming a probabilistic generating process is underlying the binary data, we recommend to use the parametric

logistic PCA model. When such an assumption is not valid and the data are considered as given, the nonparametric Gifi model is recommended.

Key Points

- Exploration of binary genomics data requires specialized versions of PCA.
- A balanced error rate has to be used when the binary data are sparse.
- Logistic SVD gives a good approximation of the data but provides loadings with large uncertainties. If binary sparse genomics data contain a low-dimensional structure, logistic SVD, logistic PCA, nonlinear PCA (Gifi) and linear PCA are all able to find it.

Supplementary Data

Supplementary data are available online at http://bib.oxfordjournals.org/.

Acknowledgements

Y.S. gratefully acknowledges the financial support from China Scholarship Council (NO.201504910809).

Funding

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC synergy grant agreement n° 319661 COMBATCANCER.

References

1. Cancer Genome Atlas Research Network. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature* 2008;**455**(7216):1061–8.
2. Iorio F, Knijnenburg TA, Vis DJ, *et al*. A landscape of pharmacogenomic interactions in cancer. *Cell* 2016;**166**(3):740–54.
3. Wu HT, Hajirasouliha I, Raphael BJ. Detecting independent and recurrent copy number aberrations using interval graphs. *Bioinformatics* 2014;**30**(12):i195–203.
4. Young FW, de Leeuw J, Takane Y. Quantifying qualitative data. In: E. D. Lantermann, H. Feger (eds). In: *Similarity and Choice. Papers in Honour of Clyde Coombs*. Berne: Hans Huber, 1980.
5. Jolliffe I. *Principal Component Analysis*. New York, USA: Springer, 2002.
6. Collins M, Dasgupta S, Schapire RE. A generalization of principal component analysis to the exponential family. In: *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2001.
7. Schein AI, Saul LK, Ungar LH. A generalized linear model for principal component analysis of binary data. In: Proceedings of the 9 th International Workshop on Artificial Intelligence and Statistics. Florida, USA: Society for Artificial Intelligence and Statistics, 2003.
8. Landgraf AJ. Generalized principal component analysis: dimensionality reduction through the projection of natural parameters. PhD thesis, The Ohio State University, 2015.
9. Lee S, Huang JZ, Hu J. Sparse logistic principal components analysis for binary data. *Ann Appl Stat* 2010;**4**(3):1579–601.
10. Lee S, Epstein MP, Duncan R, Lin X. Sparse principal component analysis for identifying ancestry-informative markers in genome-wide association studies. *Genet Epidemiol* 2012;**36**(4):293–302.
11. de Leeuw J, Mair P. Gifi methods for optimal scaling in R: the package homals. *J Stat Soft* 2009;**31**(4):1–21.
12. Mori Y, Kuroda M, Makino N. *Nonlinear Principal Component Analysis and its Applications*. New York, USA: Springer, 2016.
13. Kiers HAL. *Three-way Methods for the Analysis of Qualitative and Quantitative Two-way Data*. Leiden, Netherlands: DSWO Press, 1989.
14. Pearson K. On lines and planes of closest fit to systems of points in space. *Philos Mag* 1901;**2**(11):559–72.
15. Zou H, Hastie T, Tibshirani R. Sparse principal component analysis. *J Comput Graph Stat* 2006;**15**(2):265–86.
16. ten Berge JM. *Least Squares Optimization in Multivariate Analysis*. Leiden, Netherlands: DSWO Press, 1993.
17. Tipping ME, Bishop CM. Probabilistic principal component analysis. *J R Stat Soc Series B Stat Methodol* 1999;**61**(3):611–22.
18. de Leeuw J. Principal component analysis of binary data. Applications to roll-call analysis. Department of Statistics, The University of California, Los Angeles, USA: Elsevier Science, 2003.
19. Udell M, Horn C, Zadeh R, *et al*. Generalized low rank models. *Found Trends Mach Learn* 2016;**9**(1):1–118.
20. Gifi A. *Nonlinear Multivariate Analysis*. New York, NY: Wiley, 1990, This is a publication under a collective pseudonym.
21. Wei Q, Dunbrack RL, Jr. The role of balanced training and testing data sets for binary classifiers in bioinformatics. *PLoS One* 2013;**8**(7):e67863.
22. Wold S. Cross-validatory estimation of the number of components in factor and principal components models. *Technometrics* 1978;**20**(4):397–405.
23. Bro R, Kjeldahl K, Smilde AK, *et al*. Cross-validation of component models: a critical look at current methods. *Anal Bioanal Chem* 2008;**390**(5):1241–51.
24. R Development Core Team (2008). R: a language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.
25. Stacklies W, Redestig H, Scholz M, *et al*. pcaMethods–a bioconductor package providing PCA methods for incomplete data. *Bioinformatics* 2007;**23**(9):1164–7.