Computer Methods and Programs in Biomedicine 175 (2019) 223–231

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

MLSeq: Machine learning interface for RNA-sequencing data

Dincer Goksuluk[a,e], Gokmen Zararsiz[b,e,*], Selcuk Korkmaz[c,e], Vahap Eldem[d], Gozde Erturk Zararsiz[b], Erdener Ozcetin[f], Ahmet Ozturk[b,e], Ahmet Ergun Karaagaoglu[a]

[a] *Department of Biostatistics, School of Medicine, Hacettepe University, 06100, Ankara, Turkey*
[b] *Department of Biostatistics, School of Medicine, Erciyes University, 38030, Kayseri, Turkey*
[c] *Department of Biostatistics, School of Medicine, Trakya University, 22030, Edirne, Turkey*
[d] *Department of Biology, Faculty of Science, Istanbul University, 34452, Istanbul, Turkey*
[e] *Turcosa Analytics Solutions Ltd. Co., Erciyes Teknopark 5, 38030, Kayseri, Turkey*
[f] *Department of Industrial Engineering, Faculty of Engineering, Hitit University, 19030, Corum, Turkey*

ARTICLE INFO

Article history:
Received 23 December 2018
Revised 21 March 2019
Accepted 8 April 2019

Keywords:
RNA-Sequencing
Classification
Negative Binomial
Poisson
Linear discriminant analysis

ABSTRACT

Background and Objective: In the last decade, RNA-sequencing technology has become method-of-choice and prefered to microarray technology for gene expression based classification and differential expression analysis since it produces less noisy data. Although there are many algorithms proposed for microarray data, the number of available algorithms and programs are limited for classification of RNA-sequencing data. For this reason, we developed MLSeq, to bring not only frequently used classification algorithms but also novel approaches together and make them available to be used for classification of RNA sequencing data. This package is developed using R language environment and distributed through BIOCONDUCTOR network.

Methods: Classification of RNA-sequencing data is not straightforward since raw data should be preprocessed before downstream analysis. With MLSeq package, researchers can easily preprocess (normalization, filtering, transformation etc.) and classify raw RNA-sequencing data using two strategies: (i) to perform algorithms which are directly proposed for RNA-sequencing data structure or (ii) to transform RNA-sequencing data in order to bring it distributionally closer to microarray data structure, and perform algorithms which are developed for microarray data. Moreover, we proposed novel algorithms such as voom (an acronym for variance modelling at observational level) based nearest shrunken centroids (voomNSC), diagonal linear discriminant analysis (voomDLDA), etc. through MLSeq.

Materials: Three real RNA-sequencing datasets (i.e cervical cancer, lung cancer and aging datasets) were used to evalute model performances. Poisson linear discriminant analysis (PLDA) and negative binomial linear discriminant analysis (NBLDA) were selected as algorithms based on dicrete distributions, and voomNSC, nearest shrunken centroids (NSC) and support vector machines (SVM) were selected as algorithms based on continuous distributions for model comparisons. Each algorithm is compared using classification accuracies and sparsities on an independent test set.

Results: The algorithms which are based on discrete distributions performed better in cervical cancer and aging data with accuracies above 0.92. In lung cancer data, the most of algorithms performed similar with accuracies of 0.88 except that SVM achieved 0.94 of accuracy. Our voomNSC algorithm was the most sparse algorithm, and able to select 2.2% and 6.6% of all features for cervical cancer and lung cancer datasets respectively. However, in aging data, sparse classifiers were not able to select an optimal subset of all features.

Conclusion: MLSeq is comprehensive and easy-to-use interface for classification of gene expression data. It allows researchers perform both preprocessing and classification tasks through single platform. With this property, MLSeq can be considered as a pipeline for the classification of RNA-sequencing data.

1. Introduction

Ongoing advancements in large-scale sequencing technologies have enabled researchers to decipher cellular transcriptome composition and dynamics at the single nucleotide level [1]. Measur-

[*] Corresponding author at: Erciyes University, Faculty of Medicine, Department of Biostatistics, 38030, Kayseri, Turkey.
E-mail address: gokmenzararsiz@hotmail.com (G. Zararsiz).

https://doi.org/10.1016/j.cmpb.2019.04.007
0169-2607/

ing gene expression levels via RNA-sequencing could not only uncover the mechanism of cellular processes, it can also serve as molecular fingerprints that are unique characteristics of different cell types. In a complex organism such as a human, each of the cell types has unique transcriptome profile and substantial perturbation in gene expression networks may lead to cancer initiation and progression [2]. In this context, monitoring transcriptome dynamics using high-dimensional RNA-sequencing data can aid in the elucidation of the genetic basis of diseases as well as cancer classification and progression [3]. Early efforts have achieved significant progresses in clustering and classification of continuous expression data; however, oligonucleotide probe-based hybridization technologies, i.e. microarrays, failed to detect the expression of fusion transcripts and genes at high dynamic ranges [4]. Recent studies also showed that microarray data might demonstrate greater systematic bias in low-intensity genes than discrete RNA-sequencing data [5]. Due to superior performance, RNA-sequencing currently appears to be method of choice for classification in gene expression studies. In the near future, the signature generated from training RNA-sequencing based gene-expression data using appropriate classification algorithms might be considered as a routine procedure at clinical decision-making points from diagnosis to therapies [6].

Both microarrays and RNA-sequencing provide $p \times n$ dimensional gene-expression data, where p denotes the number of features (e.g. genes, exons, etc.) and n denotes the number of observations (e.g. tissue samples). Microarray data contain the continuous log-intensities obtained from microarray experiments. Numerous algorithms have been proposed for classification based on the microarray technology. Unlike microarray technology, RNA-sequencing data contain the genome-mapped discrete read counts. Moreover, RNA-sequencing data are reported to be overdispersed, where the variance of the counts exceeds the mean. Since the distributional properties of both technologies are totally different, microarray-based classifiers are not directly applicable to the RNA-sequencing data. Hence, we should preprocess data for downstream analyses. There are two strategies to be used for clustering and classification of RNA-sequencing data. First option is to transform the RNA-sequencing data to bring it distributionally closer to microarrays and apply microarray-based algorithms. Variance stabilizing transformation (vst) [7], regularized logarithmic transformation (rlog) [8], logarithm of counts per million reads (log-cpm) [9] and variance modelling at observational level (voom) [10] are some of popular transformation techniques. Note that voom transformation exports transformed gene-expression matrix along with a precision weight matrices in the same dimension. Transformed values are obtained on a log-cpm scale and precision weights are estimated for each sample on each feature. Although it is possible to use voom transformed values in the classification, it might give better results when precision weight are introduced into classifiers along with the transformed values. Zararsiz et al. [11] presented voom-based diagonal discriminant classifiers and the sparse voom-based nearest shrunken centroids classifier by considering precision weight in the model. In other study, Zararsiz et al. [12] demonstrated the application of microarray-based classifiers and compared the performances of several classifiers in a comprehensive simulation study. Second option is to use classifiers based on discrete probability distributions such as negative binomial and Poisson distributions. Witten [13] considered modelling mapped read counts with Poisson distribution and proposed sparse Poisson linear discriminant analysis (PLDA) classifier. However, PLDA performs poorly when counts are overdispersed. The authors suggested a power transformation to deal with the overdispersion problem when the amount of overdispersion is up to moderate level. Dong et al. [14] extended this approach to negative binomial linear discriminant analysis (NBLDA) classifier. NBLDA is more complex model comparing to PLDA; however, it generally performs better when counts are highly overdispersed as expected.

In this paper, we present **MLSeq**, a comprehensive and user-friendly R package, for classification of RNA-sequencing data. **MLSeq** consists both microarray-based and discrete-based classifiers along with the preprocessing approaches. These approaches include both normalization techniques, i.e. deseq median ratio and trimmed mean of M values normalization methods, and the transformation techniques, i.e. vst, rlog, log-cpm and voom. Researchers can import their raw RNA-sequencing count data, preprocess and build one of the numerous classification models, optimize the model parameters, evaluate the model performances, compare different classification models and predict the test cases based on the build models. We detailed mathematical background of discrete classifiers in material and methods section, fitted each model to real RNA-sequencing data sets and reported findings in results section throughout this paper.

2. Material and methods

2.1. Notations

Let $\mathbf{X}$ denote $p \times n$ dimensional RNA-sequencing gene-expression data matrix, with p genes (or genomic features, exons, etc.) and n samples. This matrix contains the non-negative and integer-valued counts which are the number of mapped sequence reads to each gene. A similar $p \times n$ dimensional gene-expression data matrix is obtained from microarray technology. However, it consists of continuous values which are the log-intensities obtained from microarray spots. Let x_{ij} be the elements of RNA-sequencing count matrix for i-th gene $(i = 1, 2, \ldots, p)$ and j-th sample $(j = 1, 2, \ldots, n)$, $x_i = (X_{i1}, X_{i2}, \ldots, X_{in})$ be the i-th row and $x_j = (X_{1j}, X_{2j}, \ldots, X_{pj})^T$ be the j-th column of $\mathbf{X}$ matrix. $X_{i.}$ is the total read counts mapped to i-th gene, i.e *gene total.* $X_{.j}$ is the *library size* (total read counts) for sample j and $X_{.}$ is the total library size. Let $\mathbf{y}$ be a vector of length n denoting the class labels of each sample and K is the number of classes $(y_j \in k = 1, 2, \ldots, K)$. Finally, we suppose that $x^* = (X_1^*, X_2^*, \ldots, X_p^*)$ is a vector of a new test observation whose class label y^* will be predicted. Hence, we train a classifier between $\mathbf{X}$ and $\mathbf{y}$, and classify the class label y^* of a test data x^* based on the trained model.

2.2. Normalization

Although RNA-sequencing provides less biased gene-expression data compared to microarrays, there are still systematic variations affecting the results of downstream analysis. Systematic variations contain both experimental artifacts and biases. These variations may arise from both between-sample variations including library size (sequencing depth) and the presence of majority fragments, and within-sample variations including gene length and sequence composition (GC content). Normalization is defined as the determination and correction of these systematic variations. Hence, it is crucial to normalize the raw RNA-sequencing data before fitting statistical models [15].

MLSeq provides two effective normalization approaches including deseq median ratio [7] and trimmed mean of M values (TMM) [16]. Both approaches assume that most of the genes are not differentially expressed between classes. TMM trims the data with genes having extreme log-fold-changes (M_{ij}^r, as default 30%) and extreme absolute intensities (A_i, as default 5%). Next, it normalizes the data based on the weighted log-fold-changes of a reference sample (r). These weights are computed from the inverse asymptotic variances by delta method. By default, reference sample is selected based on the closeness to the mean upper-quartile. TMM normalization is applied as follows:

$$\log_2(\mathrm{TMM}_j^r) = \frac{\sum_{i=1}^{p'} \psi_{ij}^r M_{ij}^r}{\sum_{i=1}^{p'} \psi_{ij}^r}, \tag{1}$$

where $M_{ij}^r = \frac{\log_2(x_{ij}/X_{.j})}{\log_2(x_{ir}/X_{.r})}$ and $\psi_{ij}^r = \frac{X_{.j}-x_{ij}}{X_{.j}x_{ij}} + \frac{X_{.r}-x_{ir}}{X_{.r}x_{ir}}$, $x_{ij}, x_{ir} > 0$. In the formula, $X_{.r}$ corresponds to the library size of the reference sample, and p' corresponds to genes which are not trimmed and used in the calculation. Likewise, deseq median ratio method generates a pseudo-reference sample from the geometric mean across the samples. Size factors ($\widehat{s}_j$) are estimated as

$$\widehat{s}_j = \frac{m_j}{\sum_{j=1}^n m_j}, \quad m_j = \mathrm{median}_i \left\{ \frac{x_{ij}}{(\prod_{j=1}^n x_{ij})^{1/n}} \right\}, \tag{2}$$

where m_j's are median of ratios of each sample to the pseudo-reference sample. More details on these normalization approaches can be found in related papers [7,15,16].

2.3. Discrete classification models

To model the RNA-sequencing data, a vast of literature has focused on the differential expression (DE) problem. Earlier studies considered the Poisson distribution [17–19]:

$$x_{ij}|y_j = k \sim \mathrm{Poisson}(\mu_{ij}), \tag{3}$$

for RNA-sequencing data where $\mu_{ij} = X_{i.}s_j$. Nagalakshmi et al. [20] reported that the RNA-sequencing data is overdispersed, i.e variance of counts exceeds its mean in presence of biological replicates. Since Poisson distribution assumes mean and variance of a random variable is equal, Poisson based models may not be a proper choice for modelling RNA-sequencing data. Recent studies considered negative Binomial (NB) distribution to take overdispersion effect into account as in Eq. (4),

$$x_{ij}|y_j = k \sim \mathrm{NB}(\mu_{ij}, \phi_i), \tag{4}$$

where ϕ_i refers to the dispersion parameter for i-th gene [7–9,21]. Here, NB distribution is parametrized using mean μ_{ij} and dispersion parameter ϕ_i satisfying that $\mathrm{Var}(x_{ij}) = \mu_{ij} + \mu_{ij}^2\phi_i$. Moreover, negative binomial and Poisson models are extended with the e_{ik} term in several studies [7,16,17,22,23]. e_{ik} term allows i^{th} gene being differentially expressed among K classes. Hence, e_{ik} is called as differential expression parameter. Extended Poisson model is

$$x_{ij}|y_j = k \sim \mathrm{Poisson}(\mu_{ij}e_{ik}), \tag{5}$$

where $e_{ik} = (\sum X_{iC_k} + \beta)/(\sum \mu_{iC_k} + \beta)$ and C_k is the vector of indices of the observations in class k. Note that we put Gamma(β, β) prior on e_{ik} which yields $e_{ik} \neq 1$. Extended negative binomial model is

$$x_{ij}|y_j = k \sim \mathrm{NB}(\mu_{ij}e_{ik}, \phi_i). \tag{6}$$

2.3.1. Poisson linear discriminant analysis

Witten [13] proposed PLDA and sparse PLDA for the classification of RNA-sequencing data. Sparse version of PLDA is closely related to the nearest shrunken centroids (NSC) classifier [24] in terms of shrinkage strategy. However, the samples in class k are assumed to follow a Poisson distribution as in Eq. (5) rather than normal distribution in NSC. Suppose we wish to estimate class label of a test observation x^* using the trained model. The posterior probability of a test sample belonging to class k can be estimated using Bayes' rule under independence of genes,

$$P(y^* = k|x^*) \propto f_k(x^*)\pi_k, \tag{7}$$

where f_k is the probability density function of the sample in class k, and π_k is the prior probability for class k. We assume that f_k is parametrized as in Poisson model 5. Introducing estimated train set parameters into Eq. (7) and taking the logarithm of the posterior probabilities, we obtain the discrimination function as,

$$\delta_k^{\mathrm{PLDA}}(x^*) = \log P(y^* = k|x^*) = \sum_{i=1}^{p} x_i^* \log \widehat{e}_{ik} - \widehat{s}^* \sum_{i=1}^{p} \widehat{e}_{ik}\widehat{X}_{i.} + \log \widehat{\pi}_k. \tag{8}$$

Eq. (8) is linear in x^* and includes i-th feature (or gene) in the model unless $\widehat{e}_{ik} = 1$ for all classes. However, this is the non-sparse form of PLDA classifier since $e_{ik} \neq 1$. Sparse PLDA (sPLDA) shrinks the $\widehat{e}_{ik}$ towards 1 using soft-thresholding method. Shrinked $\widehat{e}_{ik}$ values are estimated by

$$\widehat{e}_{ik} = \begin{cases} \frac{a}{b} - \frac{\lambda}{b}, & \text{if } \sqrt{b}\left(\frac{a}{b} - 1\right) > \lambda \\ \frac{a}{b} + \frac{\lambda}{b}, & \text{if } \sqrt{b}(1 - \frac{a}{b}) > \lambda \ , \\ 1, & \text{if } \sqrt{b}\left|1 - \frac{a}{b}\right| < \lambda \end{cases} \tag{9}$$

where $a = \sum X_{iC_k} + \beta$ and $b = \sum \bar{x}_{iC_k} + \beta$. λ is the tuning parameter generally optimized by the cross-validation method. As λ increases, the number of removed features also increases. When counts are overdispersed up to moderate level, Witten [13] suggested to perform power transformation (i.e $x_{ij}^{\alpha} \leftarrow x_{ij}$, $\alpha \in (0, 1]$) on raw counts in order to overcome the overdispersion problem. More details on PLDA and sPLDA classifiers can be found in Witten [13].

2.3.2. Negative Binomial linear discriminant analysis

Dong et al. [14] presented negative-binomial linear discriminant analysis (NBLDA), an extension of PLDA, to classify RNA-sequencing data. NBLDA considers negative binomial distribution given with Eq. (6). This model takes overdispersion effect into account by considering an extra dispersion parameter ϕ_i. By Bayes' rule, discriminant function of NBLDA is obtained as follows:

$$\begin{aligned}\delta_k^{\mathrm{NBLDA}}(x^*) &= \sum_{i=1}^{p} x_i^* (\log \widehat{e}_{ik} - \log(1 + \widehat{X}_{i.}\widehat{s}^*\widehat{e}_{ik}\widehat{\phi}_i)) \\ &\quad - \sum_{i=1}^{p} \widehat{\phi}_i^{-1} \log(1 + \widehat{X}_{i.}\widehat{s}^*\widehat{e}_{ik}\widehat{\phi}_i) + \log \widehat{\pi}_k.\end{aligned} \tag{10}$$

It is clear from the Eq. (10) that dispersion parameter ϕ_i has an important role in the model. In contrast to Poisson model, a feature having non-zero dispersion ($\phi_i \neq 0$) will be included in the model even if it is not differentially expressed among classes, i.e $e_{ik} = 1$ for all $k = 1, 2, \ldots, K$. Furthermore, the effect of an insignificant feature on the discriminant score will be directly proportional to the magnitude of its dispersion parameter. We modified NBLDA model by shrinking overdipersion estimates towards zero if it is below pre-defined threshold level such that

$$\tilde{\phi}_i = \begin{cases} 0, & \text{if } \phi_i \leq \varepsilon \\ \phi_i, & \text{otherwise} \end{cases}, \tag{11}$$

where ε is threshold. Witten [13] considered $\phi_i = 0.1$ as moderate overdispersion. Hence, one may define $\varepsilon = 0.1$ to remove features having overdispersion below moderate level and $e_{ik} = 1$. The NBLDA model 10 becomes sparse when shrunken dispersion estimates $\tilde{\phi}_i$ are used within discriminant function of NBLDA model. PLDA model is more sparse than NBLDA as a result of overdispersion parameter. As $\varepsilon \to \infty$, sparse NBLDA (sNBLDA) converges to sPLDA. Sparse version of NBLDA is available through R packages NBLDA [25] and MLSeq [26].

2.4. Transformation and microarray-based classification models

Another approach for classification of RNA-sequencing data is to transform counts into continuous scale and bring it distributionally closer to microarrays, and fit transformed counts to microarray-based classifiers. Although transformation is not required for RNA-sequencing data, it allows for the use any machine learning algorithm which works for continuous distributions for clustering and/or classification of RNA-sequencing data. A number of early RNA-sequencing publications applied shifted-log transformation ($z_{ij} = \log_2 x_{ij} + 1$). An extension of this transformation is logarithm of counts per million reads (log-cpm) transformation which is available in edgeR [9] and limma [27] packages. Although both packages aim to calculate log-cpm values, there are slight differences between calculated log-cpm values. edgeR package allows changing pseudo counts which is added to observed counts while limma takes it constant at 0.5. The log-cpm values from limma package is simply calculated by the log of the division of the counts by the library sizes and multiplication by one million as follows:

$$z_{ij} = \log_2\left(\frac{x_{ij} + 0.5}{X_{.j} + 1} \times 10^6\right). \tag{12}$$

Logarithmic transformations provide less-skewed distribution, however, the variances are still unequal. Anders and Huber [7] proposed a variance-stabilizing transformation (vst) which can be used to obtain variance stabilized transformed values. Let X be a random variable with mean-variance relation ν and u is a vst transformation, a variance stabilized values of a random variable X is derived as

$$u(x) = \int^{x} \frac{1}{\sqrt{\nu(\mu)}} d\mu, \tag{13}$$

where $\mathrm{var}(X) = \nu(E(X)) = \mu + \phi\mu^2$. The vst transformed values, $u(X)$, now has stabilized variance. In addition to vst and log-cpm transformations, Love et al. [8] presented a regularized logarithmic (rlog) transformation. This method uses a shrinkage approach as used in DESeq2 paper [8]. Rlog transformed values are similar to vst or shifted-log transformed values for genes with higher counts while shrunken for genes with lower counts. Let β_{i0} be the baseline gene expressions for each sample, and β_{ij} be the shrunken log-fold-changes of the normalized counts. Rlog transformation is defined as below:

$$z_{ij} = \log_2(\mu_{ij}/s_j) = \beta_{i0} + \beta_{ij}. \tag{14}$$

In differential expression analysis, it is reported that vst transformation does not work effectively for data with unequal library sizes [8]. However, in our experiences, both vst and rlog methods perform very similarly in classification analysis for data with unequal library sizes. Although rlog transformation performs better when sample size is small, it is computationally intensive when sample size is large.

Finally, Law et al. [10] presented voom transformation to enable limma work with differential expression analysis of RNA-sequencing data. Unlike other transformation methods, voom takes **X** as input and returns **Z** and **W** matrices ($z_{ij} \in \mathbf{Z}, w_{ij} \in \mathbf{W}$), where **Z** corresponds to the log-cpm transformed values, and **W** corresponds to the precision weights calculated from the mean-variance relationship of the data at observational level. More details about voom method can be found in Law et al. [10]. After transforming the data, we can directly train a microarray-based classifier, between **Z** and **y**. However, if voom is the transformation method of choice, the classifier should consider both **Z** and **W** in order to predict **y**. We discuss the voom transformation based classification in latter sections.

2.4.1. Application of microarray-based classifiers to RNA-sequencing data

Transformed gene expression data can be modeled using microarray based classification algorithms. **MLSeq** is integrated with several machine learning packages such as caret [28] and e1071 [29] in R software. One may fit classifiers which are available in related packages through MLSeq library. These classifiers include popular machine-learning classifiers (e.g. support vector machines, k-nearest neighbours, etc.), discriminant classifiers (e.g. penalized linear discriminant analysis, flexible discriminant analysis, etc.), ensemble classifiers (e.g. random forests, boosted trees, etc.), decision tree classifiers (e.g. C5.0, CART, etc.), neural network classifiers (e.g. multilayer perceptron, model averaged neural networks, etc.) and so on. Researchers can select an appropriate classifier, train model using count data, optimize model parameters and compare model performances with other classifiers.

2.4.2. Voom-Based classifiers

Zararsiz et al. [11] presented voomNSC classifier, which combines voom transformation [10] and NSC method [24,30] into a single and powerful classifier. This classifier extends voom method for RNA-sequencing based classification studies. The authors also presented the extensions of diagonal discriminant classifiers [31], i.e. voom-based diagonal linear discriminant analysis (voomDLDA) and voom based diagonal quadratic discriminant analysis (voomDQDA) classifiers. All three classifiers are able to work with high-dimensional ($n \ll p$) RNA-sequencing data. VoomDLDA and voomDQDA approaches are non-sparse and use all features in the model, while voomNSC is sparse and only use a small-subset of features. The authors demonstrated that voomNSC provides more accurate and sparse solutions compared to other classifiers.

The input data for voomDLDA and voomDQDA classifiers is either the raw or normalized sequencing read counts. If raw counts are provided, normalization can be performed within MLSeq. First, a voom transformation is applied, log-cpm value matrix (**Z**) and precision weights matrix (**W**) are obtained. Next, weighted statistics are calculated in order to build the classification rules. Let $\bar{z}_{w^{ik}} = \sum_{j=1}^{n} w_{ijk} z_{ijk} / \sum_{j=1}^{n} w_{ijk}$ be the class-specific weighted mean for k^{th} class, $\bar{z}_{w^{i}} = \sum_{j=1}^{n} w_{ij} z_{ij} / \sum_{j=1}^{n} w_{ij}$ be the overall weighted mean, $\widehat{\mathbf{\Sigma}}_{w^{ik}} = diag(s^2_{w^{1k}}, \ldots, s^2_{w^{pk}})$ be the diagonal weighted sample covariance matrices for k^{th} class and $\widehat{\mathbf{\Sigma}}_{w^{i}} = diag(s^2_{w^{1}}, \ldots, s^2_{w^{p}})$ be the weighted pooled covariance matrix. The genes are assumed to be independent each other. Hence, the off-diagonal elements of the covariance matrix are all set to be zero. The weighted variance of i^{th} gene in class k can be calculated as $s^2_{w^{ik}} = \frac{\sum_{j=1}^{n} w_{ijk}}{(\sum_{j=1}^{n} w_{ijk})^2 - \sum_{j=1}^{n} w_{ijk}^2} \sum_{j=1}^{n} w_{ijk}(z_{ijk} - \bar{z}_{ik})^2$ and the pooled variance can be calculated as $s^2_{w^{i}} = \frac{\sum_{j=1}^{n} w_{ij}}{(\sum_{j=1}^{n} w_{ij})^2 - \sum_{j=1}^{n} w_{ij}^2} \sum_{j=1}^{n} w_{ij}(z_{ij} - \bar{z}_{i})^2$. The discriminating function of voomDLDA classifier is as follows:

$$\delta_k^{\mathrm{DLDA}}(x^*) = -\sum_{i=1}^{p} \frac{(z_i^* - \bar{z}_{w^{ik}})^2}{s^2_{w^{i}}} + 2\log(\widehat{\pi}_k). \tag{15}$$

The difference between voomDLDA and voomDQDA methods is that voomDQDA separately calculates gene specific weighted variances ($s^2_{w^{ik}}$) for each class and uses estimated values in the denominator of Eq. (15). However, these models may produce very complex models when number of features are high because the number of covariances to be estimated increases with the number of features. To obtain a simpler and more interpretable model, voomNSC classifier may be preferred. VoomNSC primarily calculates the weighted difference scores for each gene in each class,

$$d_{w^{ik}} = \frac{\bar{z}_{w^{ik}} - \bar{z}_{w^{i}}}{m_k(s_{w^{i}} + s_{w^{0}})}. \tag{16}$$

In Eq. (16), $m_k = \sqrt{1/n_k - 1/n}$ is the standard error adjustment term and s_{w^0} is a small positive constant value [11,24,30]. These difference scores are shrunken towards zero with soft-thresholding method:

$$d'_{w^{ik}} = \text{sign}(d_{w^{ik}})\max(|d_{w^{ik}}| - \lambda, 0), \tag{17}$$

where λ is a tuning parameter that needs to be optimized. Finally, $d_{w^{ik}}$ in Eq. (16) is replaced with shrunken value $d'_{w^{ik}}$ and shrunken centroid $\tilde{z}_{w^{ik}}$ is calculated using the equality $\tilde{z}_{w^{ik}} = \bar{z}_{w^i} + m_k(s_{w^i} + s_{w^0})d'_{w^{ik}}$ for each class. As λ inreases, more features are shrunken towards common centroid. Discriminating function of this classifier is as follows:

$$\delta_k^{\text{NSC}}(x^*) = -\sum_{i=1}^{p} \frac{(z_i^* - \tilde{z}_{w^{ik}})^2}{(s_{w^i} + s_{w^0})} + 2\log(\widehat{\pi}_k). \tag{18}$$

Note that a feature satisfying that $d'_{w^{ik}} = 0$ for all $k = 1, 2, \ldots, K$ will be removed from model since it does not contribute to discriminant function. More details about these classifiers can be found in Zararsiz et al. [11].

2.5. Preprocessing and feature selection

The aim in a statistical modelling is to obtain the simplest model which gives the highest predictive ability. Preprocessing and feature selection are two important steps in statistical modelling to improve model performances, obtain simpler model and better understand the effect of features on the response variable when working with high dimensional data [32]. In the preprocessing step, we filtered features with near zero variances, i.e. the variables that have exactly one or very few unique values relative to the number of samples [28]. In RNA-sequencing studies, the number of mapped reads are highly dependent on how deep a sample is sequenced. Hence, we first obtained normalized counts using one of normalization methods and performed near zero variance filtering to remove nuisance features. It is also possible to remove low quality features (i.e poorly sequenced features) by using minimum count filtering [33]. If a feature has total mapped read counts below a given threshold, it may be filtered.

The complexity of a model is related with the number of model parameters to be estimated and the number of features included in the model. Let us consider starting with the simplest model in which we have one predictor variable. Although the simplest model can be preferred, we might add more features to obtain better accuracies. The question here is that how many features should be included in the model? As the number of features increases, the classifier's performance increases until the optimal number of features is reached. However, increasing the number of features after that point will only increase the complexity of the model and probably decrease the model performance as a result of "curse of dimensionality". In such cases, the trained model is overfitted and the results for a test set is significantly lower comparing to training set performances. Hence, selecting an optimum subset of a feature space is crucial to overcome overfitting problem and obtain higher test set performance.

Feature selection is preferred to decrease complexity and increase performance of a classifier by selecting an optimal subset of features. This can be done, for example, using a differential expression analysis [8,9] or built-in algorithms of the classifiers. Among selected classifiers, SVM is non-sparse which means it does not have built-in algorithm for feature selection. The remainders are sparse classifiers. In this study, we skip differential expression analysis steps. Hence, we let either the classifier to use all features or select optimal subset using built-in feature selection criteria. We measured the amount of selected features with "sparsity" parameter for each classifier. As the value of sparsity decreases towards 0, the less number of features are selected in the classifier.

2.6. Model building and parameter optimization

After specifying a classification model, users can build and optimize the model parameters using a single function `classify(...)` in **MLSeq**. Parameter optimization is done using repeated *k*-folds cross-validation. The total number of parameters being optimized differs among classifiers. Some of the classifiers have no tuning parameter while some others have 2 or more parameters to be optimized. A grid search over the number of levels for each tuning parameter, which is controlled by the `tuneLength` argument within `classify`, is performed for selecting the best value of tuning parameters. A complete list of model parameters for continuous classifiers from **caret** package can be found at https://topepo.github.io/caret/index.html.

The model building and optimization procedures for discrete and voom-based classifiers are similar to caret-based classifiers. PLDA and voomNSC, for example, have a single tuning parameter λ to be optimized. This parameter is called as *threshold* parameter which controls the number of features included in the classifier. Finally, the optimized model is returned from `classify(...)` function. Returned object stores input data, preprocessed and transformed data, trained model and cross-validation results.

2.7. Evaluation of model performances

In a classification problem, the predicted and actual class labels can be summarized on a cross table. For example, Table 1 shows a 2-by-2 classification table of a binary classification problem where predicted class labels are represented in the rows. A variety of performance metrics for a confusion matrix is available through **MLSeq** for evaluating model performances. These metrics include accuracy rate (ACC), Kappa statistic (κ), sensitivity (SE), specificity (SP), positive and negative predictive values (PPV, NPV), prevalence (P_r), detection rate (DR), detection prevalence (DP_r) and balanced accuracy (bAR) measures. Calculations of these measures for a binary classification problem are shown in Table 2. For a multi-class problem, these metrics, except overall accuracy, are calculated using *one-versus-all* approach.

Table 1
A confusion matrix.

Predicted labels	Actual class labels		Total
	Positive	Negative	
Positive	TP	FP	TP + FP
Negative	FN	TN	FN + TN
Total	TP + FN	FP + TN	n
TP: True positive FP: False positive	TN: True negative FN: False negative		n: Sample size

Table 2
Calculation of the performance metrics.

Performance metric	Formula
Accuracy	$ACC = (TP + TN)/2$
Kappa	$\kappa = \frac{ACC - p_e}{1 - p_e}$ $p_e = \frac{(TP+FN)(TP+FP)+(FP+TN)(FN+TN)}{n^2}$
Sensitivity (recall)	$SE = TP/(TP + FN)$
Specificity	$SP = TN/(TN + FP)$
Positive predictive value (precision)	$PPV = TP/(TP + FP)$
Negative predictive value	$NPV = TN/(TN + FN)$
Prevalence	$P_r = (TP + FN)/2$
Detection rate	$DR = TP/n$
Detection prevalence	$DP_r = (TP + FP)/n$
Balanced accuracy	$bAR = (SE + SP)/2$

2.8. Prediction of the test cases

Class labels of test cases are predicted based on the model characteristics of the trained model, e.g. discriminant function in discriminant-based classifiers. However, an important point is that test set must have passed the same analysis steps as training set. This is especially true for the normalization and transformation stages for RNA-sequencing based classification studies. Same preprocessing parameters should be used for both training and test sets to affirm that both sets are on the same scale and homoscedastic. Suppose that the training set is normalized with TMM normalization method. Then, test set will be normalized based on reference sample obtained from training data. Similarly, if we use deseq median ratio normalization, then the size factor of a test should be estimated as

$$\hat{s}^* = \frac{m^*}{\sum_{j=1}^{n} m_j}, \quad m^* = \text{median}_i \left\{ \frac{x_i^*}{(\prod_{j=1}^{n} x_{ij})^{1/n}} \right\}, \tag{19}$$

where m^* and s^* are estimated using geometric means of training set because we assume that test and training samples are drawn from the same population.

A similar procedure is applied for the transformation of test data. If vst is selected as the transformation method, then the test set will be transformed based on the dispersion function of the training data. Otherwise, if rlog is selected as the transformation method, then the test set will be transformed based on the dispersion function, beta prior variance and the intercept of the training data. Transformed test set and normalization factors are used within discriminant function and class label of test samples are predicted.

3. Results

3.1. Case study on real datasets

In this section, we will implement **MLSeq** using several real RNA-sequencing datasets. The first data set is *cervical cancer data* which was collected by Witten et al. [22]. The authors focused on both novel miRNA discovery and detection of the differentially expressed miRNAs between tumour and non-tumour conditions. They sequenced 29 cervical tumour tissues (i.e samples) and 29 matched non-tumour tissues using Solexa/Illumina platform, and captured the expression profiles of 714 miRNAs including known and unknown miRNAs. Afterwards, the authors proposed a Poisson log-linear model and detected 36 miRNAs significantly changed between groups based on the highest absolute scores obtained from this model. We used a count matrix in our analysis which consists of 58 samples and 714 miRNAs.

The second data set is *lung cancer* data which is downloaded from The Cancer Genome Atlas (TCGA, https://portal.gdc.cancer.gov) platform. This data set contains mapped read counts of 20531 known human RNAs belonging to 1128 lung cancer patients. The patients were diagnosed into two distinct classes of lung cancer which are lung adenocarcinoma (LUAD) and lung squamous cell with carcinoma (LUSC) with sample sizes 576 and 552, respectively.

Finally, we used *aging data* which were collected by Singh et al. [34]. This dataset contains the raw transcript abundances of single-cell mRNA sequencing which are obtained from the beta-cells of zebrafish species. The aim of this study was to investigate the transcriptional response of beta-cells with aging. This dataset includes 212 zebrafish samples classified into three chronological stages by their ages in month post-fertilization (mpf) which are juvenile (1 mpf, $n = 83$), adolescent (3 mpf, $n = 73$) and adult (14 mpf, $n = 56$).

3.2. Implementation of the MLSeq package

We perform both microarray-based (continuous) and count-based (discrete) classifiers on cervical data. Discrete classifiers are selected as PLDA with and without power transformation and NBLDA. Continuous classifiers are selected as SVM, voomNSC and NSC. We compare each classifier in terms of accuracy and sparsity. The dataset is split into two parts as training and test sets including 70% and 30% of all samples, respectively. Model validation and parameter optimization are evaluated using 5-folds 10-repeats cross validation. We should point out two steps in the workflow that (i) the samples in each fold are pre-defined in order to make a fair comparison between fitted models and (ii) minimum cell count is set to 1 by adding an offset value 1 to data matrix. Although most of the BIOCONDUCTOR packages which are developed for differential expression analysis are able to handle zero cell counts using independent filtering, it is likely to have convergence problem while fitting machine learning algorithms to such data sets. Hence, adding an offset value might be a useful strategy in predictive modelling.

All the analyses were performed by following a workflow given in Fig. 1. First, counts are normalized and filtered using near zero variance filtering in order to exclude nuisance features. When raw counts of cervical cancer data were used for near zero variance filtering, for example, 78 out of 714 features were removed due to near zero variation in the mapped read counts. However, this result was incorrect since the effect of sequencing depth was not considered. Hence, the normalized counts were used within near zero variance filtering and 0 out of 714 features were removed from cervical cancer dataset. It can be seen that the number of removed features is sensitive to normalization process. Hence, we suggest that the nuisance features should be detected using normalized counts in step 3 (Fig. 1). Furthermore, we used *maximum variance filtering* to select given number of features by their variances sorted in descending order. This strategy might be preferred, and can be useful when the number of features was very large. As the number of features increases, the computational cost and model complexity also increase. For example, lung cancer dataset has 20,531 features, i.e mRNAs. We selected top 2000 features by sorting feature-wise variances in descending order. As similar to near zero variance filtering, feature-wise variances were also calculated from normalized counts.

3.3. Performance comparison of classifiers

In this section, we discuss and compare the performance of the fitted models in details. As we mentioned in the earlier sections, several measures are considered for comparing the model performances. We reported the results using overall accuracy and sparsity measures since the number of samples in each class were approximately equal, i.e classes are slightly imbalanced. According to the overall accuracies, NBLDA performed better in cervical cancer data comparing to other discrete classifiers (Table 3). This might be due to highly overdispersed nature of cervical cancer data because NBLDA takes overdispersion effects into account in the discriminant function. Fig. 2 shows the distribution of gene-wise overdispersions for cervical cancer dataset which are estimated by the method-of-moments for untransformed and transformed counts. Because the cervical cancer data is highly overdispersed, NBLDA model outperforms PLDA model even after power transformation. Furthermore, the overall accuracy of PLDA increases as expected when power transformation is applied.

As an alternative to discrete classifiers (e.g NBLDA and PLDA), counts might be transformed into continuous scale and classified using microarray-based algorithms such as SVM and NSC. In

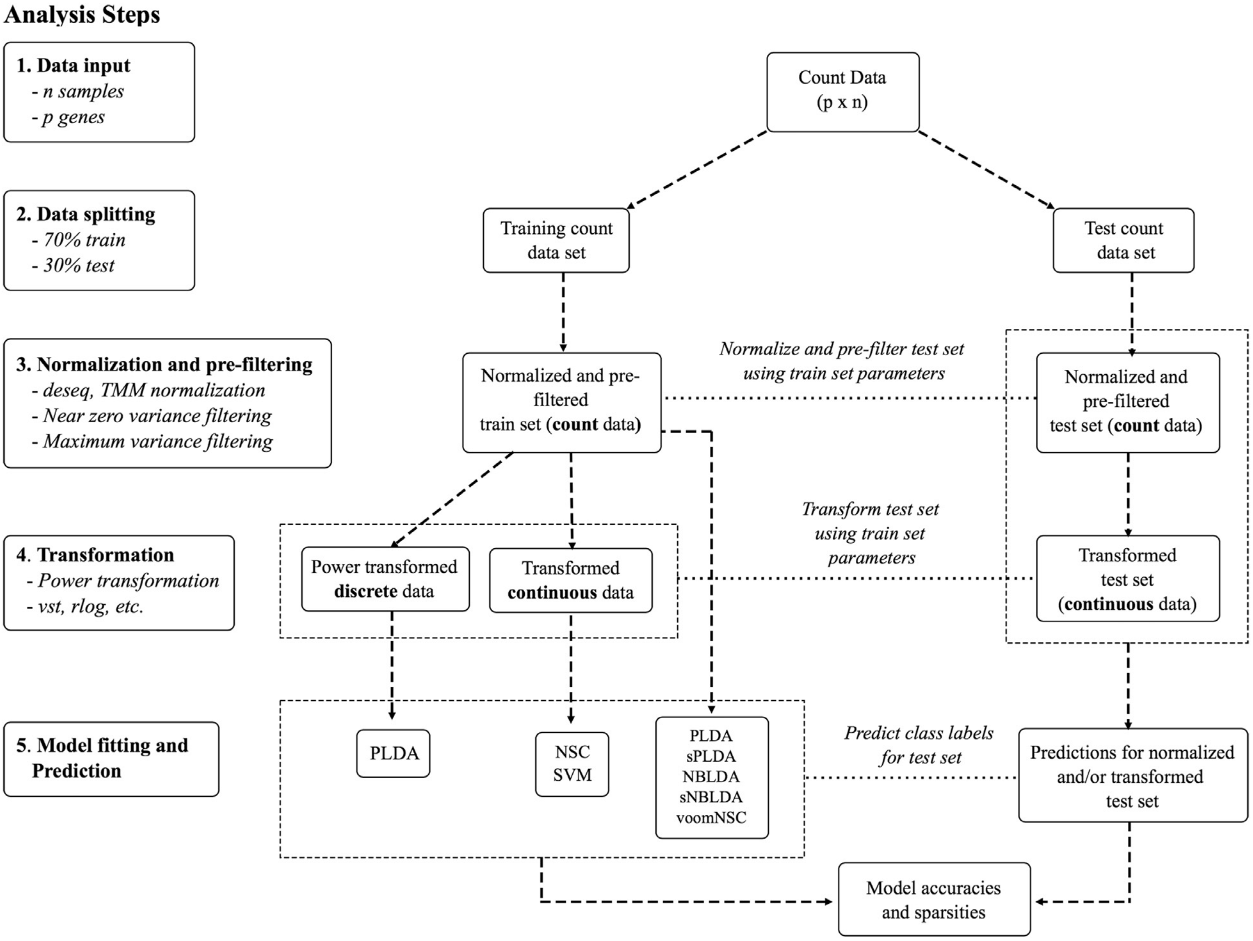

Analysis Steps
1. Data input
- n samples
- p genes
2. Data splitting
- 70% train
- 30% test
3. Normalization and pre-filtering
- deseq, TMM normalization
- Near zero variance filtering
- Maximum variance filtering
4. Transformation
- Power transformation
- vst, rlog, etc.
5. Model fitting and Prediction
Count Data (p x n)
Training count data set
Test count data set
Normalized and pre-filtered train set (count data)
Normalize and pre-filter test set using train set parameters
Normalized and pre-filtered test set (count data)
Power transformed discrete data
Transformed continuous data
Transform test set using train set parameters
Transformed test set (continuous data)
PLDA
NSC SVM
PLDA sPLDA NBLDA sNBLDA voomNSC
Predict class labels for test set
Predictions for normalized and/or transformed test set
Model accuracies and sparsities

Fig. 1. Analysis steps of the classification of real RNA-sequencing datasets.

Table 3
Classification results for real datasets.

	Cervical cancer		Lung cancer		Aging	
Number of features						
Raw data	714		20531		20651	
Pre-filtered data	714		2000		2000	
Class sizes	29/29		576/552		83/73/56	
Class ratios	1:1		1.043:1		1.48:1.30:1	
Models	Accuracy	Sparsity	Accuracy	Sparsity	Accuracy	Sparsity
PLDA	0.667	0.020	0.888	0.255	0.781	1.000
PLDA (Transformed)	0.889	0.217	0.882	1.000	0.969	1.000
NBLDA	0.944	-	0.888	-	0.875	-
sNBLDA	0.944	0.792	0.876	1.000	0.922	0.964
SVM	0.889	-	0.938	-	0.860	-
voomNSC	0.944	0.022	0.891	0.061	0.937	1.000
NSC	0.889	0.066	0.885	0.390	0.843	0.938

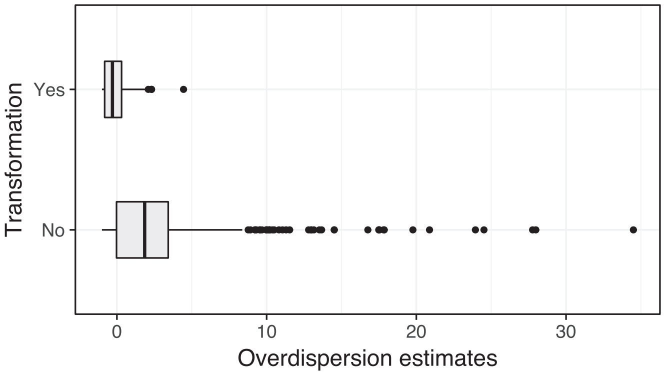
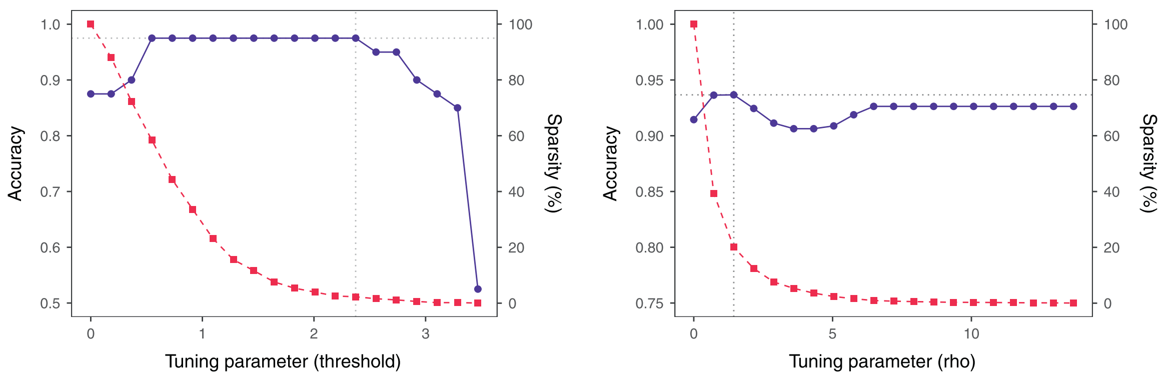


Fig. 2. Distribution of estimated gene-wise overdispersions - cervical cancer.

lung cancer dataset, for example, most of the selected classifiers performed similar except that the SVM performed the best. This might be due to underlying probability distribution in lung cancer dataset; hence, the underlying distribution may not be Poisson or negative binomial as we assumed. Furthermore, raw counts can be transformed using voom transformation within voomNSC algorithm. Here, we graphically presented the results of voomNSC and PLDA algorithms among sparse classifiers for cervical cancer data set (Fig. 3). Results showed that voomNSC performed the best among sparse classifiers. The effect of tuning parameter on cross-validated model performances is given in Fig. 3 where accuracies were given with blue solid and sparsities were given with red dashed lines. As can be seen from the figure, the accuracy of



(a) voomNSC

(b) Sparse PLDA

Fig. 3. Cross-validated training performances for various tuning parameters - accuracy: solid lines, sparsity: dashed lines.

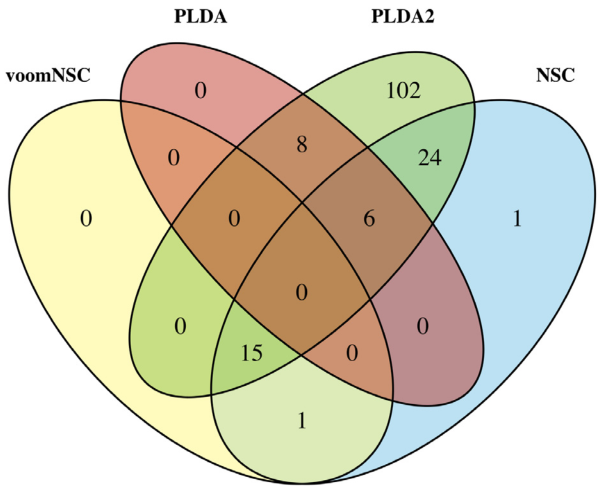


Fig. 4. Venn-diagram of selected features from sparse classifiers - cervical cancer dataset.

trained model increases until the optimal value of tuning parameter is reached.

In addition to overall accuracy, we consider sparsity measure for model comparisons. We omit SVM since this model is not sparse. Among sparse classifiers, voomNSC and NSC performs well using only 2.24% and 6.58% of 714 features, respectively for cervical cancer dataset. However, for aging dataset, sparse classifiers were not able to select an optimal subset of given features. This might be because the most of all features equally contribute to classification, and these features are included in the model. Furthermore, sparse models can be used to extract significant genes to be used as biomarkers for diagnosis. We showed selected features for cervical cancer dataset on a venn-diagram in Fig. 4. Some of the features are common between sparse classifiers. voomNSC, PLDA2 (Power transformed) and NSC, for example, commonly discover 15 features as possible biomarkers.

4. Conclusion

RNA-sequencing is currently the most efficient technique in characterizing and quantifying transcriptomes. With its major advantages, this technique has replaced microarrays as the technology of choice and has revolutionized gene expression profiling. This kind of raw data is standard for various RNA-sequencing data analysis software and can be extracted from a large number of software such as htseq [35], featureCounts [36] and bedtools [37]. One major task using gene expression data is to identify a small subset of genes and classify the data for various purposes such as cancer classification, development of RNA-sequencing based diagnostic assays, identification of types of species, separation of developmental differences, etc. Since data structure is different, microarray-based classifiers are not directly applied. Although there are recent discrete-based classifiers available, there is not a single environment that users can process their raw RNA-sequencing data, simply apply variety of classification algorithms and make predictions based on the built models.

In this paper, we present **MLSeq**, as the most comprehensive and user-friendly R package, for classification of RNA-sequencing data. **MLSeq** currently includes over 80 microarray-based classifiers including the novel voom-based classifiers, voomDLDA, voomDQDA and voomNSC. Moreover, two discrete RNA-sequencing classifiers, PLDA and NBLDA, are also available in this package. **MLSeq** is a platform which brings a variety of classification algorithms together, and provides a pipeline for classification of RNA-sequencing data. Although it is possible to perform all the steps using separate R packages, this process might be confusing and time consuming for researchers unfamiliar with R programming language. **MLSeq**, on the other side, is easier to understand and faster since it initially performs preprocessing, normalization, transformation and classification task. Hence, **MLSeq** is an easy-to-use analysis tool for inexperienced researchers, and it is possible to perform classification of RNA-sequencing data by following the directives provided with package manuals and vignettes.

Besides RNA-sequencing, **MLSeq** can also be applied to other -omics data, e.g. single-cell sequencing, metagenome sequencing or ChIP sequencing. The only requirement is that the data should be high-dimensional and contain the raw counts where samples in the columns and features in the rows.

The sparse algorithms in **MLSeq** can be used for biomarker discovery. voomNSC, for example, detected 16 features as possible biomarkers for cervical cancer dataset. Among these features, one feature which is named as **Candidate-12-3p** might be considered as a novel biomarker for cervical cancer diagnosis. However, this finding requires further clinical assessments and should be clinically verified.

Currently, **MLSeq** contains various microarray-based and discrete-based classifiers along with the normalization and transformation algorithms. The package will go on adding novel approaches as they are developed and will be regularly updated based on the changes. Future work will also include the availability of gene expression based clustering and survival analysis methods based on the RNA-sequencing data. All source codes of **MLSeq** can be found on GitHub repository at https://github.com/gokmenzararsiz/MLSeq.

Conflict of interest

The authors declare that there are no conflicts of interest.

Acknowledgements

We would like to thank S. Anders, M.I. Love and the BIOCONDUCTOR team for their useful comments, suggestions and support to improve the usefulness of the package. This study was supported by the Research Fund of Erciyes University [TDK-2015-5468]. We also would like to thank Sumeet Pal Singh for sharing the aging dataset which is used in the illustration of MLSeq package. Finally, we would like to thank anonymous reviewers for their contribution and valuable comments.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cmpb.2019.04.007.

References

[1] Z. Peng, Y. Cheng, B.C.-M. Tan, L. Kang, Z. Tian, Y. Zhu, W. Zhang, Y. Liang, X. Hu, X. Tan, J. Guo, Z. Dong, Y. Liang, L. Bao, J. Wang, Comprehensive analysis of RNA-Seq data reveals extensive RNA editing in a human transcriptome, Nat. Biotechnol. 30 (3) (2012) 253–260, doi:10.1038/nbt.2122.

[2] C. Klijn, S. Durinck, E.W. Stawiski, P.M. Haverty, Z. Jiang, H. Liu, J. Degenhardt, O. Mayba, F. Gnad, J. Liu, G. Pau, J. Reeder, Y. Cao, K. Mukhyala, S.K. Selvaraj, M. Yu, G.J. Zynda, M.J. Brauer, T.D. Wu, R.C. Gentleman, G. Manning, R.L. Yauch, R. Bourgon, D. Stokoe, Z. Modrusan, R.M. Neve, F.J. de Sauvage, J. Settleman, S. Seshagiri, Z. Zhang, A comprehensive transcriptional portrait of human cancer cell lines, Nat. Biotechnol. 33 (3) (2015) 306–312, doi:10.1038/476 nbt.3080.

[3] Q. Xu, J. Chen, S. Ni, C. Tan, M. Xu, L. Dong, L. Yuan, Q. Wang, X. Du, Pan-cancer transcriptome analysis reveals a gene expression signature for the identification of tumor tissue origin, Mod. Pathol. 29 (6) (2016) 546–556, doi:10.1038/modpathol.2016.60.

[4] R. Bi, P. Liu, Sample size calculation while controlling false discovery rate for differential expression analysis with RNA-sequencing experiments, BMC Bioinformat. 17 (2016) 146, doi:10.1186/s12859-016-0994-9.

[5] D.G. Robinson, J.Y. Wang, J.D. Storey, A nested parallel experiment demonstrates differences in intensity-dependence between RNA-seq and microarrays, Nucleic. Acids Res. 43 (20) (2015) e131, doi:10.1093/nar/gkv636.

[6] S.A. Byron, K.R. Van Keuren-Jensen, D.M. Engelthaler, J.D. Carpten, D.W. Craig, Translating RNA sequencing into clinical diagnostics: opportunities and challenges, Nat. Rev. Genet. 17 (5) (2016) 257–271, doi:10.1038/nrg.2016.10.

[7] S. Anders, W. Huber, Differential expression analysis for sequence count data, Genome Biol. 11 (10) (2010) R106, doi:10.1186/gb-2010-11-10-r106.

[8] M.I. Love, W. Huber, S. Anders, Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2, Genome Biol. 15 (12) (2014) 550, doi:10.1186/s13059-014-0550-8.

[9] M.D. Robinson, D.J. McCarthy, G.K. Smyth, Edger: a bioconductor package for differential expression analysis of digital gene expression data, Bioinformatics 26 (1) (2010) 139–140, doi:10.1093/bioinformatics/btp616.

[10] C.W. Law, Y. Chen, W. Shi, G.K. Smyth, Voom: precision weights unlock linear model analysis tools for RNA-seq read counts, Genome Biol. 15 (2) (2014) R29, doi:10.1186/gb-2014-15-2-r29.

[11] G. Zararsiz, D. Goksuluk, B. Klaus, S. Korkmaz, V. Eldem, E. Karabulut, A. Ozturk, Voomdda: discovery of diagnostic biomarkers and classification of RNA-seq data., PeerJ 5 (2017) e3890, doi:10.7717/peerj.3890.

[12] G. Zararsiz, D. Goksuluk, S. Korkmaz, V. Eldem, G.E. Zararsiz, I.P. Duru, A. Ozturk, A comprehensive simulation study on classification of RNA-seq data., PLoS One 12 (8) (2017) e0182507, doi:10.1371/journal.pone.0182507.

[13] D.M. Witten, Classification and clustering of sequencing data using a poisson model, Annal. Appl. Stati. 5 (4) (2011) 2493–2518, doi:10.1214/11-AOAS493.

[14] K. Dong, H. Zhao, T. Tong, X. Wan, NBLDA: negative binomial linear discriminant analysis for RNA-Seq data, BMC Bioinformat. 17 (1) (2016) 369, doi:10.1186/s12859-016-1208-1.

[15] J. Zyprych-Walczak, A. Szabelska, L. Handschuh, K. Gorczak, K. Klamecka, M. Figlerowicz, I. Siatkowski, The impact of normalization methods on RNA-Seq data analysis., Biomed. Res. Int. 2015 (2015) 621690, doi:10.1155/2015/621690.

[16] M.D. Robinson, A. Oshlack, A scaling normalization method for differential expression analysis of RNA-Seq data., Genome Biol. 11 (3) (2010) R25, doi:10.1186/gb-2010-11-3-r25.

[17] J.C. Marioni, C.E. Mason, S.M. Mane, M. Stephens, Y. Gilad, RNA-Seq: an assessment of technical reproducibility and comparison with gene expression arrays., Genome Res. 18 (9) (2008) 1509–1517, doi:10.1101/gr.079558.108.

[18] J. Li, D.M. Witten, I.M. Johnstone, R. Tibshirani, Normalization, testing, and false discovery rate estimation for RNA-sequencing data., Biostatistics 13 (3) (2012) 523–538, doi:10.1093/biostatistics/kxr031.

[19] L. Wang, Z. Feng, X. Wang, X. Wang, X. Zhang, DEGseq: An r package for identifying differentially expressed genes from RNA-seq data., Bioinformatics 26 (1) (2010) 136–138, doi:10.1093/bioinformatics/btp612.

[20] U. Nagalakshmi, Z. Wang, K. Waern, C. Shou, D. Raha, M. Gerstein, M. Snyder, The transcriptional landscape of the yeast genome defined by RNA sequencing., Science 320 (5881) (2008) 1344–1349, doi:10.1126/science.1158441.

[21] M.D. Robinson, G.K. Smyth, Moderated statistical tests for assessing differences in tag abundance., Bioinformatics 23 (21) (2007) 2881–2887.

[22] D. Witten, R. Tibshirani, S.G. Gu, A. Fire, W.-O. Lui, Ultra-high throughput sequencing-based small RNA discovery and discrete statistical biomarker analysis in a collection of cervical tumours and matched controls., BMC Biol. 8 (2010) 58, doi:10.1186/1741-7007-8-58.

[23] J.H. Bullard, E. Purdom, K.D. Hansen, S. Dudoit, Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments., BMC Bioinformat. 11 (2010) 94, doi:10.1186/1471-2105-11-94.

[24] R. Tibshirani, T. Hastie, B. Narasimhan, G. Chu, Diagnosis of multiple cancer types by shrunken centroids of gene expression, Proc. Natl. Acad. Sci. USA 99 (10) (2002) 6567–6572, doi:10.1073/pnas.082099299.

[25] D. Goksuluk, G. Zararsiz, S. Korkmaz, A.E. Karaagaoglu, NBLDA: negative binomial linear discriminant analysis, 2018https://CRAN.R-project.org/package=NBLDA R package version 0.99.0

[26] G. Zararsiz, D. Goksuluk, S. Korkmaz, V. Eldem, I.P. Duru, A. Ozturk, A.E. Karaagaoglu, MLSeq: machine learning interface for RNA-Seq data, 2018. R package version 2.0.0.

[27] M.E. Ritchie, B. Phipson, D. Wu, Y. Hu, C.W. Law, W. Shi, G.K. Smyth, Limma powers differential expression analyses for RNA-sequencing and microarray studies, Nucleic Acids Res. 43 (7) (2015) e47.

[28] M. Kuhn, Building predictive models in r using the caret package, J. Stat. Softw. 28 (5) (2008) 1–26, doi:10.18637/jss.v028.i05.

[29] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, F. Leisch, e1071: misc functions of the department of statistics, probability theory group (Formerly: E1071), TU Wien, 2017. https://CRAN.R-project.org/package=e1071, R package version 1.6–8

[30] R. Tibshirani, T. Hastie, B. Narasimhan, G. Chu, Class prediction by nearest shrunken centroids, with applications to DNA microarrays, Stat. Sci. 18 (1) (2003) 104–117, doi:10.1214/ss/1056397488.

[31] S. Dudoit, J. Fridlyand, T.P. Speed, Comparison of discrimination methods for the classification of tumors using gene expression data, J. Am. Stat. Assoc. 97 (457) (2002) 77–87, doi:10.1198/016214502753479248.

[32] S. Korkmaz, D. Goksuluk, G. Zararsiz, S. Karahan, Genesurv: an interactive web-based tool for survival analysis in genomics research, Comput. Biol. Med. 89 (2017) 487–496, doi:10.1016/j.compbiomed.2017.08.031.

[33] P. Leidinger, C. Backes, S. Deutscher, K. Schmitt, S.C. Mueller, K. Frese, J. Haas, K. Ruprecht, F. Paul, C. Stähler, C.J. Lang, B. Meder, T. Bartfai, E. Meese, A. Keller, A blood based 12-miRNA signature of Alzheimer disease patients, Genome Biol. 14 (7) (2013) R78.

[34] S.P. Singh, S. Janjuha, S. Chaudhuri, S. Reinhardt, A. Kränkel, S. Dietz, A. Eugster, H. Bilgin, S. Korkmaz, G. Zararsiz, N. Ninov, J.E. Reid, Machine learning based classification of cells into chronological stages using single-cell transcriptomics, Sci. Rep. 8 (1) (2018) 17156, doi:10.1038/s41598-018-35218-5.

[35] S. Anders, P.T. Pyl, W. Huber, HTSeq–A python framework to work with high-throughput sequencing data., Bioinformatics 31 (2) (2015) 166–169, doi:10.1093/bioinformatics/btu638.

[36] Y. Liao, G.K. Smyth, W. Shi, Featurecounts: an efficient general purpose program for assigning sequence reads to genomic features, Bioinformatics 30 (7) (2014) 923–930, doi:10.1093/bioinformatics/btt656.

[37] A.R. Quinlan, I.M. Hall, BEDTools: A flexible suite of utilities for comparing genomic features., Bioinformatics 26 (6) (2010) 841–842, doi:10.1093/bioinformatics/btq033.