


RESEARCH

Open Access



Colorectal cancer subtype identification from differential gene expression levels using minimalist deep learning

Shaochuan Li¹, Yuning Yang¹, Xin Wang², Jun Li³, Jun Yu⁴, Xiangtao Li^{5,6*}  and Ka-Chun Wong^{6*}

*Correspondence:

kc.w@cityu.edu.hk;

lixt314@jlu.edu.cn

⁵School of Artificial Intelligence, Jilin University, Changchun, Jilin, China

⁶Department of Computer Science, City University of Hong Kong, Hong Kong SAR, China

Full list of author information is available at the end of the article

Abstract

Background: Cancer molecular subtyping plays a critical role in individualized patient treatment. In previous studies, high-throughput gene expression signature-based methods have been proposed to identify cancer subtypes. Unfortunately, the existing ones suffer from the curse of dimensionality, data sparsity, and computational deficiency.

Methods: To address those problems, we propose a computational framework for colorectal cancer subtyping without any exploitation in model complexity and generality. A supervised learning framework based on deep learning (DeepCSD) is proposed to identify cancer subtypes. Specifically, based on the differentially expressed genes under cancer consensus molecular subtyping, we design a minimalist feed-forward neural network to capture the distinct molecular features in different cancer subtypes. To mitigate the overfitting phenomenon of deep learning as much as possible, L_1 and L_2 regularization and dropout layers are added.

Results: For demonstrating the effectiveness of DeepCSD, we compared it with other methods including Random Forest (RF), Deep forest (gcForest), support vector machine (SVM), XGBoost, and DeepCC on eight independent colorectal cancer datasets. The results reflect that DeepCSD can achieve superior performance over other algorithms. In addition, gene ontology enrichment and pathology analysis are conducted to reveal novel insights into the cancer subtype identification and characterization mechanisms.

Conclusions: DeepCSD considers all subtype-specific genes as input, which is pathologically necessary for its completeness. At the same time, DeepCSD shows remarkable robustness in handling cross-platform gene expression data, achieving similar performance on both training and test data without significant model overfitting or exploitation of model complexity.

Keywords: DeepCSD, Cancer subtype identification, Differential gene expression



© The Author(s). 2022 **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>. The Creative Commons Public Domain Dedication waiver (<http://creativecommons.org/publicdomain/zero/1.0/>) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

Introduction

The gene expression-based consensus molecular subtyping has been demonstrated as a promising paradigm for patient stratification and therapy [1]. Molecular subtyping of cancer is an indispensable step toward individualized treatments while providing important biological insights into cancer heterogeneity [2]. However, traditional cancer diagnosis heavily relies on manual inspections and human clinician expertise [3, 4]. Moreover, the traditional diagnosis methods of cancer subtypes are too expensive that many patients give up treatment [5]. To address those problems, computational methods have been developed to diagnose cancer subtypes from molecular data. Unfortunately, since those molecular data are collected with the high dimensionality of the gene space and only a few cancer subtypes are available for different cancers, the sufficient data collection issues for cancer classification after reducing the dimensionality of gene space become particularly important. Meanwhile, the limited number of cancer subtypes can expose the computational methods vulnerable to model overfitting.

In the past, application-specific supervised cancer diagnosis methods have been developed to address the previously mentioned challenges; for instance, Yang et al. [6] constructed four models based on support vector machine (SVM) for predicting the metastasis of esophageal squamous cell carcinoma. Huang et al. [7] discussed the performance of the SVM-based model for predicting the response of cancer patients to drugs. Wang et al. [8] proposed an improved method based on random forest (RF) for lung cancer classification [9]. However, those methods usually suffer from realistic restrictions such as high dimensionality and computational scalability.

Recently, deep learning has been proposed to deal with those problems. It automatically generates informative features from low-dimensional features to discover the essential data representations [10]. Several supervised models based on deep learning have been employed to address cancer classification task successfully; for instance, Zeng et al. [11] applied DeepCues, a convolutional neural network, to classify seven major cancers based on DNA sequences. Islam and Poly [12] proposed a feedforward neural network to model breast cancer risk. Karabulut and Ibriki [13] illustrated deep belief networks to address cancer classification problem and demonstrated impressive performance over SVM and RF [14, 15]. Sveen et al. [1] proposed a model called PDX (patient-derived xenograft) to focus on CMS-specific drug sensitivity. DeepCC, a cancer molecular subtype classification based on deep learning framework, has been developed for predicting consensus molecular subtypes based on high-dimensional gene expression profiles [2]. Unfortunately, most of the classifiers are subject to certain limitations. Firstly, the gene signature method only emphasizes the role of individual genes, while not fully considering the pathological impacts. Moreover, the overfitting phenomenon is recurring due to model complexity and data redundancy [10]. Therefore, in this study, we proposed DeepCSD to address those problems.

At the beginning, to emphasize the pathological importance of different genes in identifying cancer subtypes, differential gene expression analysis is computed to identify different subtype-specific genes in each pair of subtypes. Secondly, we proposed a novel deep learning framework for cancer-subtype diagnosis, named DeepCSD. To avoid overfitting to the maximum extent possible, the dropout layer, L_1 and L_2 regularization are added into the network architecture to enhance the robustness of DeepCSD. Finally, to demonstrate the performance, we collected eight independent datasets to train and

validate DeepCSD. The experimental results reveal that DeepCSD can achieve competitive performance over the current state-of-the-art cancer-subtype diagnosis methods including Random Forest (RF), deep forest (gcForest), support vector machine (SVM), XGBoost, and DeepCC. In addition, we directly applied the DeepCSD model to independent TCGA data and can characterize differential gene expressions among diverse marker genes. Meanwhile, gene ontology enrichment, and KEGG pathology analysis are conducted to reveal novel insights into the cancer subtype identification and characterization mechanisms with validations.

Methods

Datasets

In this study, we collected eight independent colorectal cancer datasets [16] including GSE13067, GSE13294, GSE14333, GSE17536, GSE20916, GSE2109, GSE37892, and GSE39582 (Supplementary Table S1). All those datasets can be downloaded from the official repository of an international CRC subtyping consortium on Synapse (<https://www.synapse.org/#!Synapse:syn2623706/wiki/>) (downloaded on 1 Oct 2020).

Gene expression analysis

Since the tumor microenvironment makes an important contribution to gene expression [1], we identify all differential genes with discriminative gene expression values among the cancer subtypes (namely, subtype-specific genes).

In this study, we focus on the well-established consensus molecular subtypes (CMS), i.e. CMS1-CMS4. Each subtype is characterized by its own unique feature: CMS1 (microsatellite instability immune), CMS2 (canonical), CMS3 (metabolic), and CMS4 (mesenchymal). The definitions can be found in [16]. Such high-dimensional data often brings difficulties in computational resources (i.e., computer memory). Therefore, the primary task is to select meaningful features from such high-dimensional gene expression data. Recognizing that there were four consensus molecular subtypes in this study, we combined these subtypes into six groups: CMS1 vs CMS2, CMS1 vs CMS3, CMS1 vs CMS4, CMS2 vs CMS3, CMS2 vs CMS4, and CMS3 vs CMS4. The subtype-specific genes are identified based on the fold-change and adjust P value (also called Q value). Mathematically, the fold-change can be defined as follows:

$$fold-change = \frac{\bar{A}}{\bar{B}} \quad (1)$$

while A and B denote the genes expression values of the same gene in different subtypes, \bar{A} is the average value of A and \bar{B} is the average value of B . In common, the \log_2 fold-change is widely used to normalize the range of fold-change. The \log_2 fold-change can be defined as follows:

$$\log_2 fold-change = \log_2 \bar{A} - \log_2 \bar{B} \quad (2)$$

However, the fold change has a drawback that the misclassification of differentially expressed genes with large differences may result in poor identification of changes in high expression levels.

To address it, the P value is implemented as an alternative to the rejection point and provided the minimum level at which the initial hypothesis will be rejected. The P value

we used is adjusted by the Benjamini-Hochberg (BH) program [17], also called *Q* value. In this study, we take the *T*-test to compute every gene’s *P* value. More formally, the *T*-test is defined as follows:

$$T = \frac{\bar{A} - \bar{B}}{\sqrt{S_A^2/n + S_B^2/n}} \tag{3}$$

Then the significance *P* value is calculated according to the *T* distribution to measure the significance of this difference. Next, we took BH to obtain a *Q* value. More formally, the *Q* value in this study is defined as follows:

$$Q = BH(P) \tag{4}$$

After that, we calculated the fold-change and *Q* value for each CMS group. Then, the genes with $|\log_2 \text{fold-change}| > 1$ and *Q* value < 0.05 are retained and identified. Finally, we input those subtype-specific genes into deep learning and then construct our DeepCSD model.

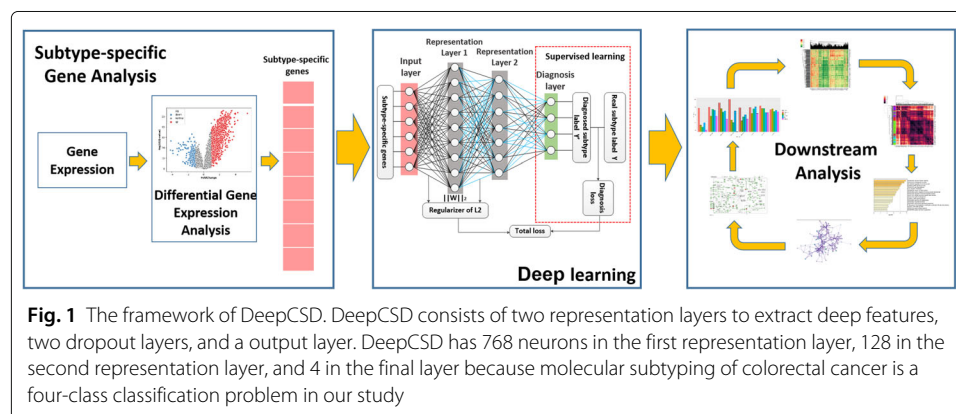
Deep learning for cancer subtype diagnosis (DeepCSD)

Our deep neural network (DNN) is a feed-forward neural network assembled by a sequential layer-by-layer structure that realizes a series of functional transformations. Each layer is fed with the previous layer’s outputs as its inputs is to execute its transformation function. Specifically, every layer consists of multiple “neurons”. The input data will pass every layer by the “connections” with the weight parameters between “neurons”. The basic layers in DeepCSD are the representation layers and dropout layer (as shown in Fig. 1). DeepCSD features the sequential alternating representation layers with nonlinear activation functions (i.e. ReLU) and dropout layers, followed by representation layers with softmax activation functions for computing probability of each CMS (more detail can be seen in Fig. 1). Mathematically, it is recursively defined as follows:

$$f^l(X) = \alpha_l(W_l R_{l-1} f^{l-1}(X) - \theta_l) \quad l = 1, 2, 3 \tag{5}$$

while *X* is the model’s inputs, $f^{l-1}(X)$ is the (*l* – 1)-th layer’s output, θ_l is the *l*-th layer’s threshold, and α_l is the *l*-th layer’s activate function. It is worth noticing that R_0 does not exist. In other words, the dropout layer is not employed in DeepCSD’s first layer.

The natural method is to process the original inputs layer by layer to obtain “deep features”. In other words, every layer in the framework of deep learning is to extract deep



features from the previous layer’s outputs. The more layers the DNN has, the deeper features the model can extract. In theory, the addition of layers and parameters can increase the model expressiveness for addressing complex learning tasks. However, the possibility of overfitting is also increased [10]. To reduce the influence of overfitting, L_2 regularization and dropout layers are employed in our model. For the dropout layer, it is to abandon a part of neurons to mitigate the overfitting. The brief implementation step is defined as follows:

$$R = \text{Bernoulli}(p) \tag{6}$$

$$f'(x) = R * f(x) \tag{7}$$

where the Bernoulli function is to randomly generate a binary mask vector with Bernoulli trial probability p , $f(x)$ is the input of dropout layer, and $f'(x)$ is the output [18].

After that, DeepCSD applies L_1 and L_2 regularization to representation layers 1 and 2 by adding a penalty term to the empirical loss. It is worth to be noticed that the output layer in DeepCSD does not employ such regularization.

Training of the DeepCSD model

For such a multi-class classification task, we minimized the objective function of DeepCSD defined as follow:

$$\min \text{ loss} = \sum_{i=0}^n \sum_{t=0}^3 y_{i,t} \ln(y'_{i,t}) + \frac{\lambda}{2} \|W\|_1 + \frac{\lambda}{2} \|W\|_2 \tag{8}$$

while n is the number of samples, $y'_{i,t}$ is the t -th neuron’s output of i -th sample, λ is the learning rate, $\frac{\lambda}{2} \|W\|_1$ is the term of L_1 , and $\frac{\lambda}{2} \|W\|_2$ is the term of L_2 . With one-hot label vector encoding, the objective function can be simplified as follow:

$$\begin{aligned} \min \text{ loss} &= \sum_{i=0}^n \ln(y^*_{i,t}) + \frac{\lambda}{2} \|W\|_2 \\ \text{subject to } &y_{i,t} = 1, \sum_{j=0}^3 y_{i,j} = 1 \end{aligned} \tag{9}$$

while $y^*_{i,t}$ indicates the output label index of the element 1 in one-hot vector for i -th sample.

The Adam algorithm has been chosen as our model’s optimizer [19]. The update rule is defined as follow:

$$\Theta_t = \Theta_{t-1} - \alpha \frac{m_t}{1 - \beta_1^t} / (\frac{v_t}{1 - \beta_2^t} + \epsilon) \tag{10}$$

while

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{11}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{12}$$

$$g_t = \nabla_{\Theta} f_t(\Theta_{t-1}) \tag{13}$$

We can see that if we compute the parameter of g_t , the Adam algorithm automatically updates the corresponding parameter. Next, we take the update process of W_3 as a sample

to explain the detail of our model update process. Mathematically, the output is defined as follow:

$$y_{i,t}^* = \text{softmax}(W_3 R_2 f^2(X_i) - \Theta_3) \tag{14}$$

We focus on the parameter updating process of W_3 . The Adam algorithm makes some adjustment to W_3 if the g_t is computed. The g_t for W_3 is calculated as follow:

$$g_t = \frac{\partial \text{loss}}{\partial W_3} \tag{15}$$

while

$$\frac{\partial \text{loss}}{\partial W_3} = \frac{\partial \ln y_{i,t}^*}{\partial y_{i,t}^*} \frac{\partial y_{i,t}^*}{\partial (W_3 R_2 f^2(x) - \Theta_3)} \frac{\partial W_3 R_2 f^2(x) - \Theta_3}{\partial W_3} \tag{16}$$

$$\frac{\partial \text{loss}}{\partial W_3} = y_{i,t}^{*-1} \frac{\partial y_{i,t}^*}{\partial (W_3 R_2 f^2(x) - \Theta_3)} R_2 f^2(x) \tag{17}$$

Next, we focus on partial derivative calculation of activation function for $(W_3 R_2 f^2(x) - \Theta_3)$. Firstly, the partial derivative calculation of softmax activation function for k -th neuron is calculated as follow:

$$\frac{\partial y_{i,t}^*}{\partial k} = \frac{\frac{\partial e^{i,t}}{\partial k} \sum_{j=0}^3 e^{i,j} - e^{i,t} \sum_{j=0}^3 \frac{\partial e^{i,j}}{\partial k}}{(\sum_{j=0}^3 e^{i,j})^2} \tag{18}$$

It can result in two different outputs according to the value of k , i.e.

$$\frac{\partial y_{i,t}^*}{\partial k} = \begin{cases} \frac{e^{i,t} \sum_{j=0}^3 e^{i,j} - (e^{i,t})^2}{(\sum_{j=0}^3 e^{i,j})^2} = y_{i,t}^* (1 - y_{i,t}^*) & \text{if } (t = k) \\ \frac{e^{i,t} e^{i,k}}{(\sum_{j=0}^3 e^{i,j})^2} = -y_{i,t}^* y_{i,k}^* & \text{if } (t \neq k) \end{cases} \tag{19}$$

Therefore, we view W_3 as $(w_1, w_2, w_3, w_4)^T$ (expand matrix W_3 by row). Then, the partial derivative calculation of activation function for $(W_3 R_2 f^2(x) - \Theta_3)$ is divided in two situations as follows:

$$\frac{\partial y_{i,t}^*}{\partial (w_k R_2 f^2(x) - \Theta_{3,k})} = \begin{cases} y_{i,t}^* (1 - y_{i,t}^*) & \text{if } (k = t) \\ -y_{i,t}^* y_{i,k}^* & \text{if } (k \neq t) \end{cases} \tag{20}$$

The final g_t is given by the following formula:

$$\frac{\partial \text{loss}}{\partial w_k} = \begin{cases} (1 - y_{i,t}^*) R_2 f^2(x) & \text{if } (k = t) \\ -y_{i,k}^* R_2 f^2(x) & \text{if } (k \neq t) \end{cases} \tag{21}$$

Combining those column vectors into a matrix in order, the g_t appeared. The $y_{i,t}^*$ is the neuron's output that its position is consistent with the real label, which means the possibility of the real label. We can conclude that w_k will be strengthened if " $k = t$ "; it will be punished if " $k \neq t$ ".

Model parameters and running time

In this study, those eight molecular datasets are employed for experimental comparisons. For the DeepCSD, the parameters of Adam follows the default setting [20] and the minimum learning rate was set to 10^{-5} . For SVM and RF, the parameter setting follows the default parameter values of Python Scikit-learn 0.21.2. For gcForest, we fixed this model to a multi-class mission with the literature default setting [22]. For XGBoost, $\text{gamma}=0.1, \text{maxdepth}=12, \text{subsample}=0.7, \text{colsamplebytree}=0.7, \text{earlystoppingrounds}=15$. For DeepCC, all parameter setting follows the reference [2]. Moreover, we

also provide the running time of our model. DeepCSD is written in Python. We have relied on a server with CPU = Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz, GPU = GTX1080 Ti. Meanwhile, we design our deep learning framework based on Keras. The versions of those software and packages are Python =3.7, Anaconda=3.7, Keras=2.1.0, Tensorflow=1.14.0. After running on those molecular datasets, the average running time is 457 seconds.

Results

Competing methods

To comprehensively demonstrate the DeepCSD performance, five state-of-the-art cancer-subtype diagnosis methods including Random Forest (RF), multi-Grained Cascade Forest (gcForest), support vector machine (SVM), XGBoost, and DeepCC are employed in this study. Firstly, two traditional frameworks were constructed based on RF and SVM. The fundamental idea of SVM is to find a hyperplane in a given sample space to distinguish samples with different classes [10]. Although the initial aim of SVM is to address the two-class task, Hsu and Lin [21] improved it to the multi-class task. RF is an ensemble learning algorithm in which component learner is a decision tree known for its random feature selection method. Then, deep forest (gcForest) framework is a deep forest ensemble to address classification task and the final output is generated by a vote of each tree [22]. XGBoost is a kind of boosting method based on the decision tree, which is a powerful machine learning method known for “regularized boosting”. DeepCC is a deep learning method as one of the comparative models with five hidden layers. Moreover, we also compared our DeepCSD with other deep learning methods (DeepCC) and deep forest (gcForest) to demonstrate the effectiveness of differential gene expression analysis in our model.

Evaluation metrics

Four evaluation metrics have been measured in this study. For each dataset, according to the relationship between the real subtype and the diagnosis of DeepCSD, each diagnosis of DeepCSD was divided into true positive (TP), true negative (TN), false positive (FP), and false negative (FN). These four-measure metrics are defined as follow:

$$Sensitivity = \frac{TP}{TP + FN} * 100\% \quad (22)$$

$$Specificity = \frac{TN}{TN + FP} * 100\% \quad (23)$$

$$Precision = \frac{TP}{TP + FP} * 100\% \quad (24)$$

$$Accuracy = \frac{1}{n} \sum_{i=1}^n I(y_i = y_i^*) * 100\% \quad (25)$$

which y_i is the sample's true label, y_i^* is the diagnosis of DeepCSD, $I(x)$ is the indicator function, and n is the samples of each molecular data.

Application to cancer gene expression data

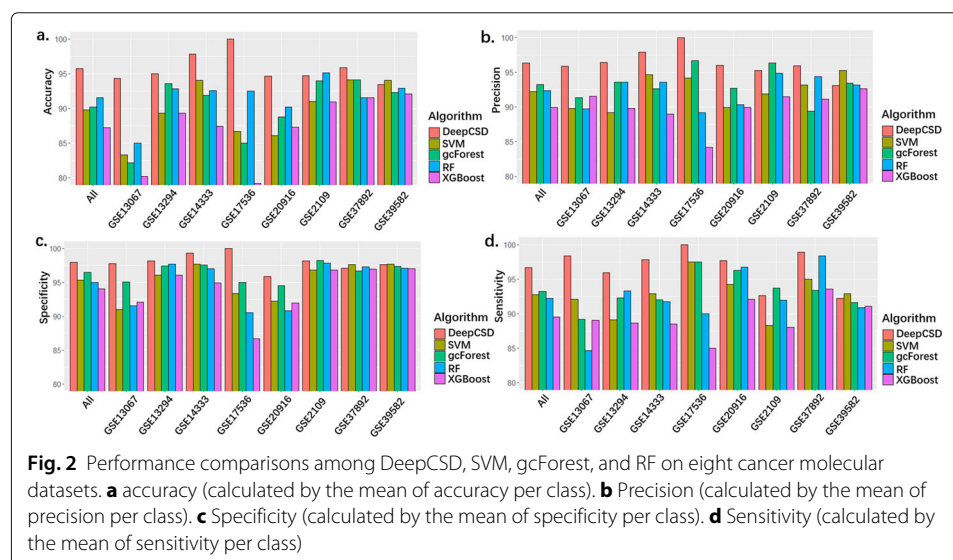
To comprehensively demonstrate the performance of DeepCSD, all datasets (i.e. GSE13067, GSE13294, GSE37892, GSE39582, GSE2109, GSE14333, GSE17536, GSE20916) are adopted for model training with 10-fold cross-validation. We identified the differential

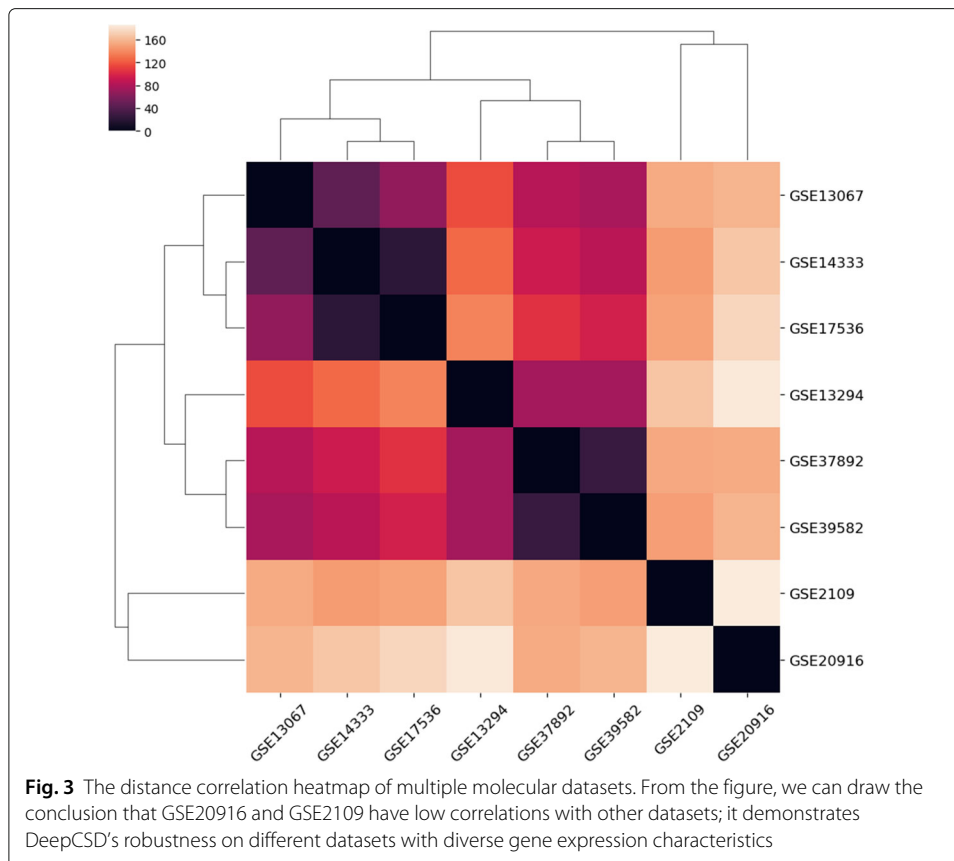
genes as the inputs of SVM, GCF, RF, and XGBoost by subtype-specific gene analysis. The experimental results are summarized in Fig. 2. We can conclude several interesting phenomenons:

- Firstly, DeepCSD provides competitive performances in specificity, precision, sensitivity, and accuracy: 1) the accuracy of DeepCSD is higher than other classifiers in GSE13067, GSE17536, and GSE20916 by 5%, while the specificity and sensitivity is not less than 94%. 2) although the performance of DeepCSD is not the best in GSE39582 and GSE2109, the difference in accuracy is only lower than the champion by $\sim 1\%$ while the sensitivity is higher than 92% and specificity is higher than 96%. 3) the average accuracy, precision, specificity, and sensitivity are higher than the second place by 2% \sim 5%.
- Secondly, 1) gcForest and RF provide good performance since those methods are boosting methods. Its accuracy, specificity, and sensitivity are higher than 90% in 5 out of 8 datasets. The accuracy of RF is higher than gcForest on GSE20916, GSE14333, and GSE39582. However, the specificity of RF is lower than gcForest for those datasets. 2) gcForest, RF, SVM, XGBoost and DeepCSD can obtain better results than other compared algorithms on GSE39582 due to the influence of subtype-specific genes.
- Thirdly, the average accuracy of DeepCSD, gcForest, and RF are above 90% while SVM and XGBoost cannot provide such performance.
- Last but not least, we can find that its accuracy on GSE20916 and GSE2109 is not bad at all compared with other datasets. Moreover, we also computed the distance correlations [23] between those eight datasets as shown in Fig. 3. We can draw the conclusion that GSE20916 and GSE2109 have low correlation with other datasets, demonstrating DeepCSD's robustness on different datasets with diverse gene expression characteristics.

Subtype-specific gene analysis

To explore the influence of subtype-specific genes, GSE39582 was chosen as an example to reveal the subtype-specific genes' contributions.



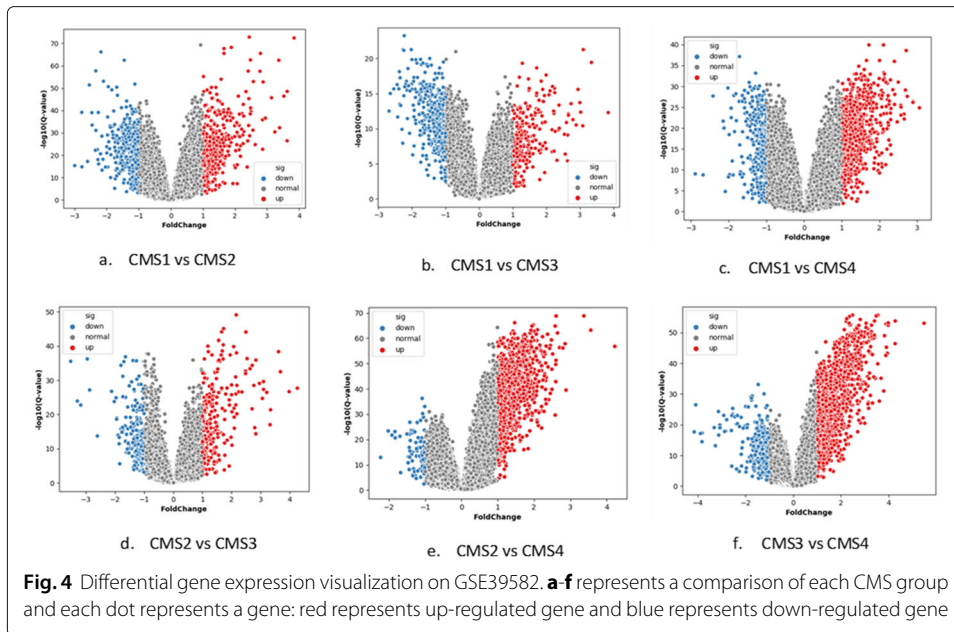


In GSE39582, 2917 subtype-specific genes were extracted and visualized as red and blue dots in Fig. 4. From Fig. 4(a)-(f), we can draw some observations: firstly, all subtypes illustrate its distinctiveness since each CMS type has distinct up-regulated and down-regulated genes compared with others; secondly, it is a hard job to distinguish CMS2 due to the fact that the numbers of down-regulated gene are much lower than up-regulated genes compared with CMS3 and CMS4; last but not least, CMS4 is easy to diagnose given its distinctive characteristics in up-regulated genes.

To demonstrate the distinctiveness of those 2917 subtype-specific genes, we employed spectral clustering on all samples as depicted in Fig. 5. Figure 5(h) illustrates the difference between samples, while Fig. 5(g) provides the genes' correlation to each sample. From Fig. 5, we can conclude that even with the differential gene analysis method, the distinguishability between the cancer patients remains very small.

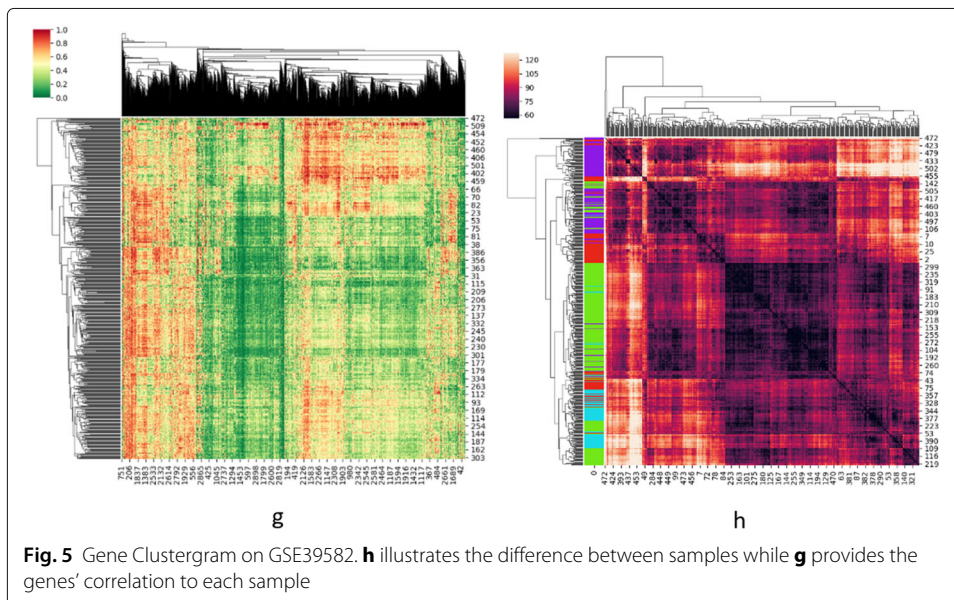
To further demonstrate the subtype-specific genes' performance, we compared our proposed DeepCSD with DeepCC. DeepCC combines samples' gene expressions with public data to process gene set enrichment analysis (GSEA) to select some gene expressions which the authors called Functional Spectrum [2]. As mentioned above, DeepCC consists of five representation layers to extract "deep features" while DeepCSD only employed two layers as a minimalist approach. Therefore, it may not be a fair comparison for our proposed DeepCSD.

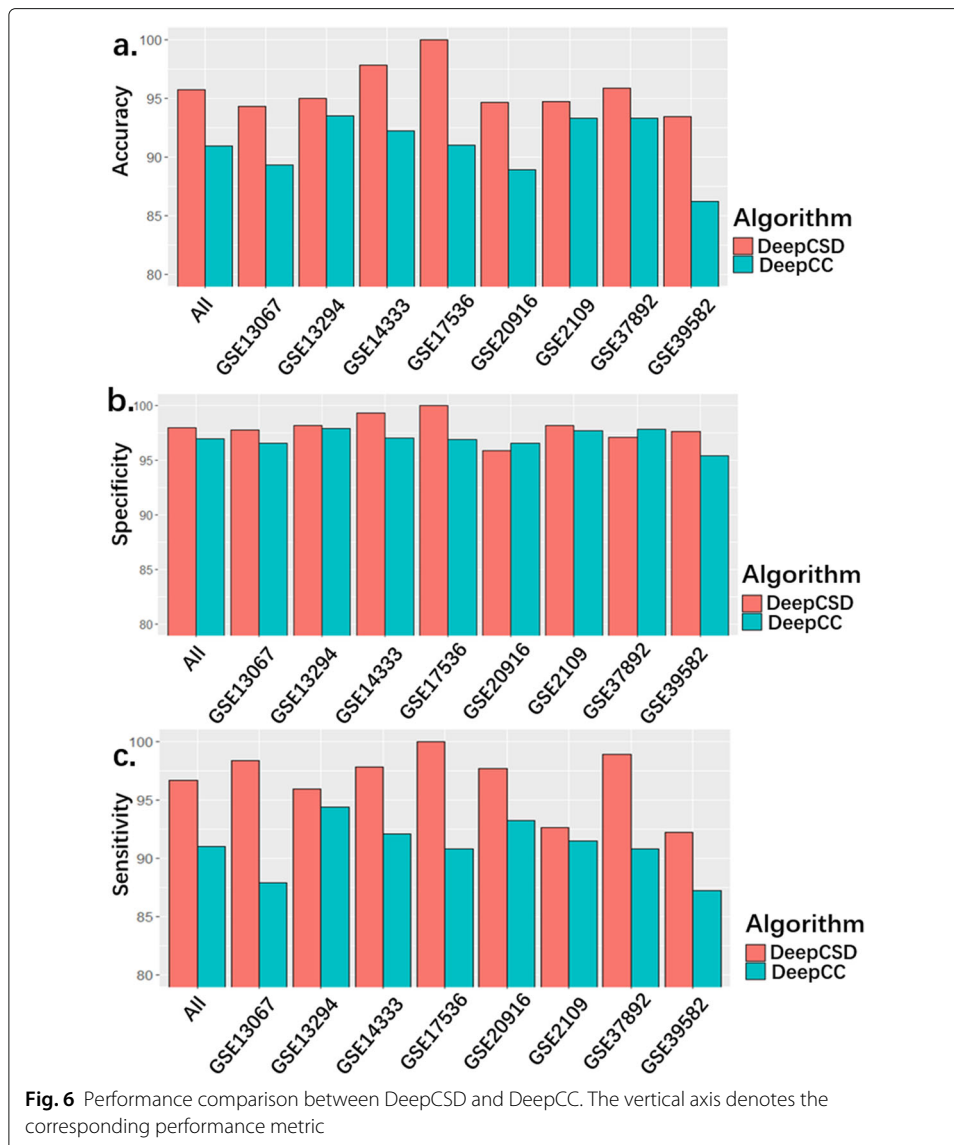
The experimental results are shown in Fig. 6. We can observe the followings: first of all, DeepCSD achieves higher performance than DeepCC in terms of accuracy, specificity, and sensitivity, while the average of accuracy and sensitivity is even higher than



DeepCC by 5%. Meanwhile, although the specificity of DeepCSD is lower than DeepCC in GSE37892, the accuracy, and sensitivity of DeepCSD are higher than DeepCC. Given that, the subtype-specific gene provides a significant contribution to the diagnosis. To demonstrate the necessity of differential gene expression analysis, ensembles of decision trees (EDT) and select k best features (SKB) (supported by scikit-learn) were employed as feature selection methods for comparisons. Equally, the top 2000 genes selected by each method were selected as the input of our model. The results are summarized in Fig. 7.

It can be found that the performance of DeepCSD is higher than EDT and SKB under the same condition. Meanwhile, we also compared DeepCSD with and without





differential genes. The results reveal that the differential genes analysis is very important in DeepCSD. Therefore, we can conclude that the subtype-specific differential gene expression analysis is an inevitable step towards cancer subtype diagnosis improvement.

Parameter analysis

In DeepCSD, the dropout parameter is 0.5. It indicates that all inputs delivered from previous layers will be abandoned for half of the neurons. Traditionally, the parameter is chosen in the range from 0.2 to 0.5. However, to reduce the influence of overfitting to the utmost extent and to strengthen the generalization ability, dropout=0.5 is chosen in DeepCSD [18, 20].

The real challenge lies in the parameter setting of the regularizer. It is well known that the regularizer of L_1 and L_2 can strengthen the important features and reduce the weights of redundant features. Therefore, if the parameter is too large, the learning ability of DeepCSD will be limited. On the opposite, if the parameter is too small, the phenomenon

Case study

The performance of DeepCSD on TCGA

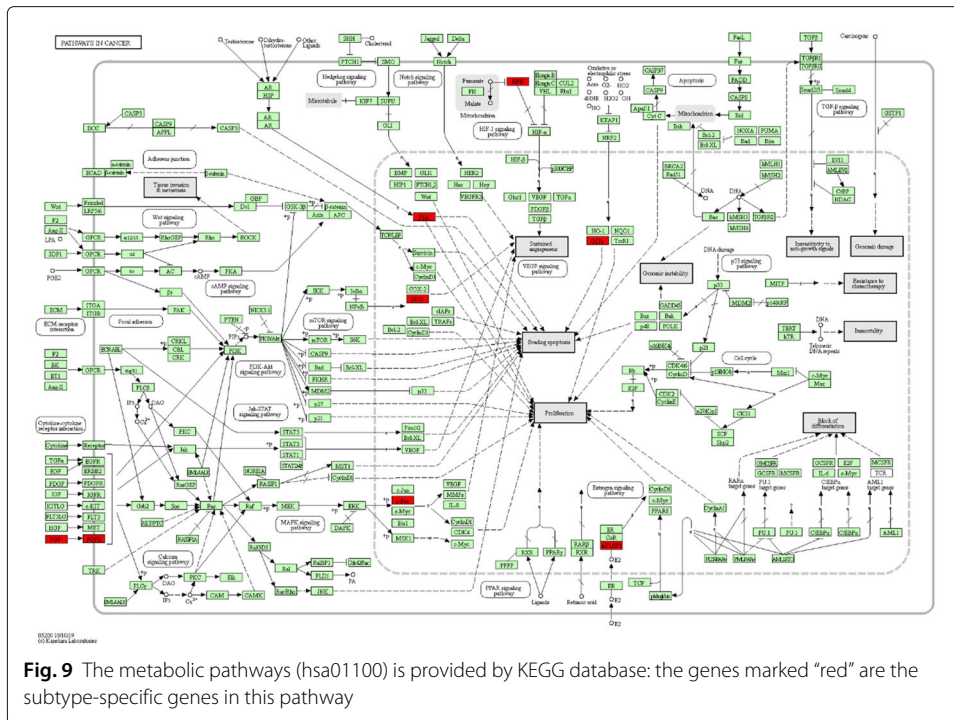
To further demonstrate the performance of DeepCSD, we applied it to annotate subtypes on an independent colorectal cancer dataset from The Cancer Genome Atlas (TCGA) [16, 24]. The TCGA expression data can be downloaded from (<https://www.synapse.org/#!/Synapse:syn2623706/wiki/>). The TCGA dataset has 512 patients with explicit CMS labels in total. Meanwhile, each patient sample in TCGA has 20293 features. In the first, the TCGA dataset was split randomly into a training set and test set with the 9:1 ratio. Next, we trained DeepCSD on a training set with 10-fold cross-validation and took the corresponding test set to demonstrate the performance of DeepCSD. To identify the differential genes, we applied the subtype-specific differential gene expression analysis, resulting in 4881 genes for such TCGA data. From [Supplementary Fig. S3](#), we can find that CMS4 also illustrated its distinctive difference compared with other groups. Moreover, the TCGA dataset shown its unique characterization from the previous molecular data. Intuitively, DeepCSD was trained on each training set ($n=460$) and then applied it to the diagnosis of the independent test set ($n=52$). The experimental results are tabulated in [Supplementary Tables S2](#) and [S3](#).

From [Supplementary Table S2](#), we can conclude that the performance of the test set is no worse than that of the training set, indicating that the impact from the overfitting phenomenon of DeepCSD may be minimized. [Supplementary Table S3](#) explicitly lists the performance of DeepCSD on TCGA: 1), 2 out of 19 CMS2 was diagnosed to other subtypes since the corresponding number of up- and down-regulated genes is low as we discussed in [Subtype-specific gene analysis](#) section; 2), the sensitivity of CMS3 and CMS4 are 100% since a large number of up- and down-regulated genes are available for diagnosis.

Biological interpretability from DeepCSD

In the previous section, we have demonstrated the competitive performance of DeepCSD based on subtype-specific genes. In this section, we investigated the biological significance of the subtype-specific genes.

Firstly, we identified the top 100 genes with the largest weights in each of the first 10 neurons in the representation layer 1 (the detail of those selected genes can be found in [Supplemental Data](#)). After that, we conduct gene ontology (GO) enrichment analysis based on Metascape [25] to analyze those selected genes. We input those selected genes into Metascape and then collect the GO enrichment in the first 10 neurons in the representation layer 1 of DeepCSD trained on the TCGA dataset. [Supplementary Fig. S4](#) summarizes the results of GO enrichment analysis. In those figures, the top category is the enriched biological process ordered by p -values. After that, we randomly took the 9-th neuron as example; for instance, the top three enriched GO biological processes in 9-th neuron are mucosal immune response (GO:0002385), prostaglandin secretion (GO:0032310), and regulation of secretion by cell (GO:1903530). Meanwhile, as depicted in [Fig. S2](#), the Cytoscape is employed to visualize a subset of enriched terms and present a network connected by edges to illustrate the relationship between terms. From the figure, each node denotes an enriched term and is colored first by its cluster ID as shown in the first figure of [Fig. 9](#). After that, the p -values of the enriched term are clustered in the second figure of [Fig. 9](#). Furthermore, some molecular pathways were



provided by the enrichment analysis. For example, MAPK signaling pathway (hsa04010), PID HIF1 TFPATHWAY (M255, a kind of canonical pathways), cAMP signaling pathway (hsa04024), and Pancreatic secretion (hsa04972).

In addition, to further analyze the molecular pathways behind the layer of DeepCSD, those selected genes were taken to the molecular pathway analysis with KEGG (<https://www.kegg.jp>). As shown in Fig. 9, we observe that 7 subtype-specific genes of the 9-th neurons can be found in the PATHWAYS IN CANCER (hsa05200). That provide evidences that our subtype-specific genes make an extinguished contribution to diagnosis.

Discussion and conclusions

Given the central importance of colorectal cancer subtyping, the consensus molecular subtype classification is always desirable for patient stratification. Specifically, the most important component to the consensus molecular subtyping is that the underlying classifier can diagnose every subtype correctly. In this study, we proposed a deep learning framework based on differential gene expression analysis to diagnose cancer subtypes.

Based on the results, we observe that DeepCSD can achieve better colorectal cancer subtype identification than RF, SVM, gcForest, XGBoost, and DeepCC. We also highlighted the significance of the differential gene expression analysis compared with other possible algorithms as shown in Fig. 7. Obviously, we can observe that the differential gene expression analysis is necessary and important for cancer subtype diagnosis. In particular, we conceive that DeepCSD can bring us the following performance advantages: 1) Biological interpretability: DeepCSD takes all of the subtype-specific genes into account as inputs, which are pathologically necessary for its completeness. 2) Robustness: DeepCSD demonstrated its remarkable robustness in processing cross-platform gene expression

data. 3) Generalization ability: DeepCSD achieves similar performance on both training and test data, suggesting that it may not suffer from severe model overfitting or model complexity exploitation..

In the future, we are optimistic that our minimalist approach can demonstrate its own value and applicability to cancer genomics in the face of high-throughput medical data.

Abbreviations

DeepCSD: A supervised learning framework based on deep learning; RF: Random Forest; gcForest: Deep forest; SVM: support vector machine; DeepCC: a cancer molecular subtype classification based on deep learning framework; TCGA: The Cancer Genome Atlas; KEGG: Kyoto Encyclopedia of Genes and Genomes; CMS: consensus molecular subtypes; DNN: deep neural network; TP: true positive; TN: true negative; FP: false positive; FN: false negative; GSEA: gene set enrichment analysis; EDT: ensembles of decision trees; SKB: select k best features

Supplementary Information

The online version contains supplementary material available at <https://doi.org/10.1186/s13040-022-00295-w>.

Additional file 1: Supplementary figures and tables.

Acknowledgments

Not applicable.

Authors' contributions

XT, SC and KC designed the study. XW performed the data analysis. JY, JL, and YN contributed to data interpretation. XT revised results and contributed to the analysis pipelines. XT and SC wrote the manuscript. All the co-authors reviewed the manuscript and approved the final version.

Funding

The work described in this paper was substantially supported by the grant from the Health and Medical Research Fund, the Food and Health Bureau, The Government of the Hong Kong Special Administrative Region [07181426].

Availability of data and materials

In this study, we collected eight independent colorectal cancer datasets [16] including GSE13067, GSE13294, GSE14333, GSE17536, GSE20916, GSE2109, GSE37892, and GSE39582 (Supplementary Table S1). All those datasets can be downloaded from the official repository of an international CRC subtyping consortium on Synapse (<https://www.synapse.org/#Synapse:syn2623706/wiki/>) (downloaded on 1 Oct 2020)

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Department of Information Science and Technology, Northeast Normal University, Changchun, Jilin, China. ²Department of Surgery, Chinese University of Hong Kong, Hong Kong SAR, China. ³Department of Infectious Diseases and Public Health, Jockey Club College of Veterinary Medicine and Life Sciences, and School of Data Science, City University of Hong Kong, Hong Kong SAR, China. ⁴Institute of Digestive Disease and Department of Medicine and Therapeutics, State Key Laboratory of Digestive Disease, Li Ka Shing Institute of Health Sciences, The Chinese University of Hong Kong, Sha Tin, Hong Kong SAR, China. ⁵School of Artificial Intelligence, Jilin University, Changchun, Jilin, China. ⁶Department of Computer Science, City University of Hong Kong, Hong Kong SAR, China.

Received: 5 October 2021 Accepted: 26 March 2022

Published online: 23 April 2022

References

1. Sveen A, Bruun J, Eide PW, et al. Colorectal cancer consensus molecular subtypes translated to preclinical models uncover potentially targetable cancer cell dependencies[J]. *Clin Cancer Res*. 2018;24(4):794–806.
2. Gao F, Wang W, Tan M, et al. DeepCC: a novel deep learning-based framework for cancer molecular subtype classification[J]. *Oncogenesis*. 2019;8(9):1–12.

3. Breugom AJ, et al. Adjuvant chemotherapy and relative survival of patients with stage II colon cancer-A EURECCA international comparison between the Netherlands, Denmark, Sweden, England, Ireland, Belgium, and Lithuania. *Eur J Cancer*. 2016;63:110–7.
4. Dotan E, Cohen SJ. Challenges in the management of stage II colon cancer. *Semin Oncol*. 2011;38:511–20.
5. Tannock IF, Hickman JA. Limits to personalized cancer medicine. *N Engl J Med*. 2016;375(13):1289–94.
6. Yang H, Feng W, Wei J, et al. Support vector machine-based nomogram predicts postoperative distant metastasis for patients with oesophageal squamous cell carcinoma. *Br J Cancer*. 2013;109:1109–16. <https://doi.org/10.1038/bjc.2013.379>.
7. Huang C, Clayton EA, Matyunina LV, et al. Machine learning predicts individual cancer patient responses to therapeutic drugs with high accuracy. *Sci Rep*. 2018;8:16444. <https://doi.org/10.1038/s41598-018-34753-5>.
8. Wang Q, Zhou Y, Ding W, Zhang Z, Muhammad K, Cao Z. Random Forest with Self-Paced Bootstrap Learning in Lung Cancer Prognosis. *ACM Trans Multimedia Comput Commun Appl*. 16(1s):1–12. <https://doi.org/10.1145/3345314>.
9. Yu K, Zhang C, Berry G, et al. Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features. *Nat Commun*. 2016;7:12474. <https://doi.org/10.1038/ncomms12474>.
10. Zhihua Z. *Machine learning*. Beijing: Tsinghua University Press; 2015.
11. Zeng Z, Mao C, Vo A, et al. Deep learning for cancer type classification[J]. *bioRxiv*. 2019;612762. <https://doi.org/10.1101/612762>.
12. Islam MM, Poly TN. Machine Learning Models of Breast Cancer Risk Prediction[J]. *BioRxiv*. 2019;723304. <https://doi.org/10.1101/723304>.
13. Karabulut EM, Ibrkci T. Discriminative deep belief networks for microarray based cancer classification. *Biomed Res*. 2017;28:1016–24.
14. Ibrahim R, Yousri NA, Ismail MA, El-Makky NM. Multi-level gene/MiRNA feature selection using deep belief nets and active learning. *Conf Proc IEEE Eng Med Biol Soc*. 2014;2014:3957–60.
15. Fakoor R, Ladhak F, Nazi A, Huber M. Using deep learning to enhance cancer diagnosis and classification. In: *Proceedings of the International Conference on Machine Learning*. New York: ACM; 2013.
16. Guinney J, Dienstmann R, Wang X, et al. The consensus molecular subtypes of colorectal cancer[J]. *Nat Med*. 2015;21(11):1350.
17. Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J R Stat Soc Ser B (Methodol)*. 1995;57(1):289–300.
18. Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. *J Mach Learn Res*. 2014;15(1):1929–58.
19. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. *CoRR abs/1412.6980*. 2015.
20. Snoek J, Larochelle H, Adams RP. Practical Bayesian optimization of machine learning algorithms. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'12)*. Red Hook: Curran Associates Inc.; 2012. p. 2951–59.
21. Hsu C-W, Lin C-J. A comparison of methods for multi-class support vector machines. *IEEE Trans Neural Netw*. 2002;13(2):415–25.
22. Zhou Z-H, Feng J. Deep forest: towards an alternative to deep neural networks. In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. AAAI Press; 2017. p. 3553–9.
23. Székely GJ, Rizzo ML, Bakirov NK. Measuring and testing dependence by correlation of distances[J]. *Ann Stat*. 2007;35(6):2769–94.
24. Brunet J-P, Tamayo P, Golub TR, Mesirov JP. Metagenes and molecular pattern discovery using matrix factorization. *Proc Natl Acad Sci USA*. 2004;101:4164–9.
25. Zhou Y, et al. Metascape provides a biologist-oriented resource for the analysis of systems-level datasets. *Nat Commun*. 2019;10(1):1523.
26. Breiman L. Random forests[J]. *Mach Learn*. 2001;45(1):5–32.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions

