*Briefings in Bioinformatics*, 2023, **24(3)**, 1–10

https://doi.org/10.1093/bib/bbad159

Advance access publication date 24 April 2023

**Case Study**

# Benchmarking of analytical combinations for COVID-19 outcome prediction using single-cell RNA sequencing data

Yue Cao, Shila Ghazanfar, Pengyi Yang and Jean Yang

Corresponding author. Jean Yang, School of Mathematics and Statistics, The University of Sydney, NSW 2006, Australia. Tel.: 9351 3012; Fax: 9351 4534; E-mail: jean.yang@sydney.edu.au

## Abstract

The advances of single-cell transcriptomic technologies have led to increasing use of single-cell RNA sequencing (scRNA-seq) data in large-scale patient cohort studies. The resulting high-dimensional data can be summarized and incorporated into patient outcome prediction models in several ways; however, there is a pressing need to understand the impact of analytical decisions on such model quality. In this study, we evaluate the impact of analytical choices on model choices, ensemble learning strategies and integrate approaches on patient outcome prediction using five scRNA-seq COVID-19 datasets. First, we examine the difference in performance between using single-view feature space versus multi-view feature space. Next, we survey multiple learning platforms from classical machine learning to modern deep learning methods. Lastly, we compare different integration approaches when combining datasets is necessary. Through benchmarking such analytical combinations, our study highlights the power of ensemble learning, consistency among different learning methods and robustness to dataset normalization when using multiple datasets as the model input.

**Keywords:** single-cell, COVID-19, benchmark, disease outcome prediction, patient analysis

## INTRODUCTION

Single-cell RNA-sequencing (scRNA-seq) is a powerful tool that measures the transcriptomes of individual cells. As the technology advances, a typical dataset size has grown from a few thousand of cells in 2014 to hundreds of thousands of cells in 2022 [1]. We are now in the era where the technology enables us to collect large pools of cells from multiple patients across multiple conditions. The current single-cell literature has mostly focused on analyzing gene and cell level changes [2, 3], for example dissecting the transcriptional heterogeneity in the population of single cells and identifying genes that mark the cell types [4]. Recently, there is an increasing number of studies designed at an individual level, such as between normal and patients (i.e. case versus control). These studies create the opportunity to examine disease mechanisms from multiple aspects, such as cell-type-specific changes in gene expression, pathway regulation [5, 6] and cell–cell interaction [7] to gain a deeper understanding of disease mechanism. The analysis of such data requires the development of methods that can extract information from multi-sample multi-condition disease study designed at an individual level rather than at a cell level [8].

To date, the majority of the questions at individual level have focused on the identification of differentially expressed genes between cell types and states [9], and differential abundance of cells between states [10, 11] of individuals. A natural next question is to develop models that explore at a higher level how the outcome associated with each individual can be predicted in multi-sample multi-condition scRNA-seq dataset. As the number of individuals increases, there will be a demand to develop models that accurately predict the outcome of each patient in such data. To meet this demand with an interpretable focus, it will be necessary to first extract informative features from complex single-cell data structures that represent each individual and then understand which approaches are most effective for utilizing the summary statistics for downstream analysis.

To date, a large repertoire of approaches has been developed to model for the prediction task, which prompts the question: ‘What are the optimal approaches?’. Since the past decade, modern deep learning has gained tremendous success compared with classical machine learning [12] in analyzing complex data. However, it is worth noting that deep learning models often involve millions of parameters [13] and require larger amounts of data and computational resources to train compared with classical machine learning. This raises the question of whether it is necessary to use deep learning models. In parallel, when extracting information from data, we may obtain multiple pieces of information, that is,

**Yue Cao** is an early career biomedical data scientist. Her interests are in the computational analysis of high-dimensional omics data in the era of precision medicine with a particular focus on single-cell data.

**Jean Yang** (professor) is an applied statistician with expertise in statistical bioinformatics. Her research has centred on the development of machine learning and statistical methods to problems in omics and biomedical research.

**Pengyi Yang** is an associate professor. His research lies at the interface of computational and systems biology. He develops machine learning and statistical models to reconstruct signalling, epigenomic/transcriptional and proteomic networks in cellular systems.

**Shila Ghazanfar** is a biomedical data scientist with interest in developing approaches for integration of complex and high-dimensional biological datasets across various technological or omics modalities.

**Received:** January 17, 2023. **Revised:** March 30, 2023. **Accepted:** April 3, 2023

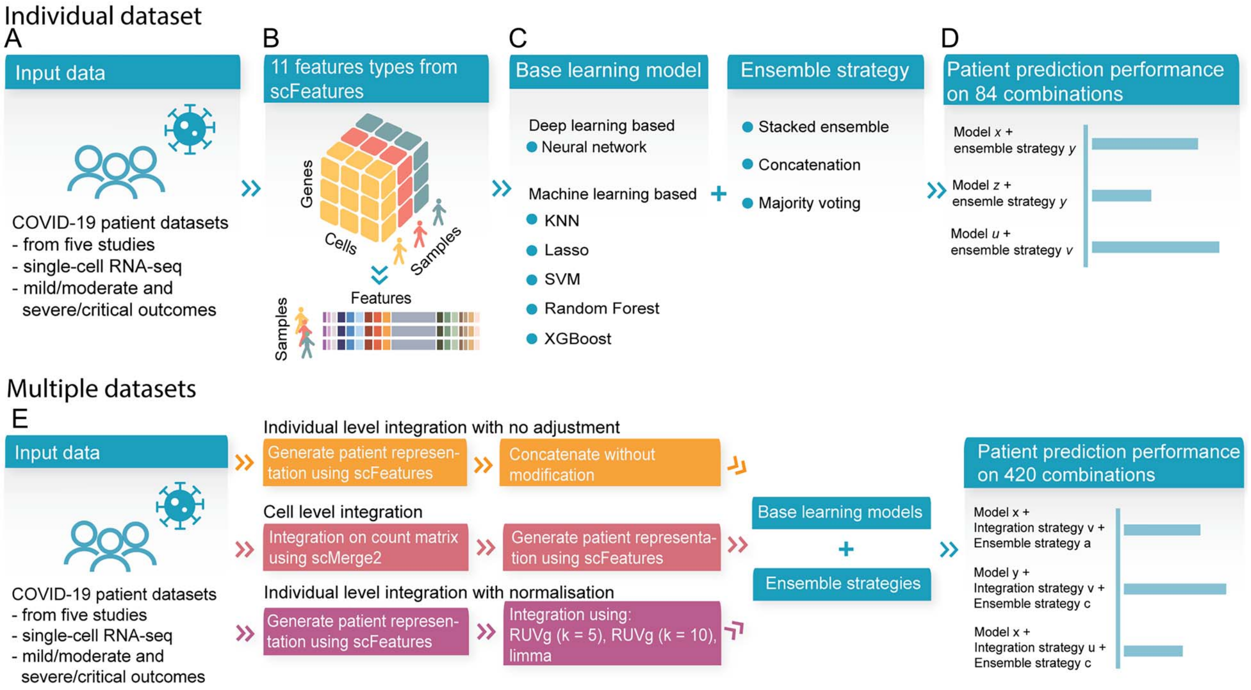
Individual dataset
A
Input data
COVID-19 patient datasets
- from five studies
- single-cell RNA-seq
- mild/moderate and
severe/critical outcomes
B
11 features types from scFeatures
Genes
Cells
Samples
Features
Samples
C
Base learning model
Deep learning based
● Neural network
Machine learning based
● KNN
● Lasso
● SVM
● Random Forest
● XGBoost
Ensemble strategy
● Stacked ensemble
● Concatenation
● Majority voting
D
Patient prediction performance on 84 combinations
Model x + ensemble strategy y
Model z + ensemle strategy y
Model u + ensemble strategy v
Multiple datasets
E
Input data
COVID-19 patient datasets
- from five studies
- single-cell RNA-seq
- mild/moderate and
severe/critical outcomes
Individual level integration with no adjustment
Generate patient representation using scFeatures
Concatenate without modification
Cell level integration
Integration on count matrix using scMerge2
Generate patient representation using scFeatures
Individual level integration with normalisation
Generate patient representation using scFeatures
Integration using: RUVg (k = 5), RUVg (k = 10), limma
Base learning models
+
Ensemble strategies
Patient prediction performance on 420 combinations
Model x + Integration strategy v + Ensemble strategy a
Model y + Integration strategy v + Ensemble strategy c
Model x + Integration strategy u + Ensemble strategy c

**Figure 1.** Schematic of the benchmark workflow. (**A**) Five COVID-19 scRNA-seq datasets containing mild/moderate and severe/critical outcome patients were used in this benchmark study. (**B**) We used scFeatures to generate 11 types of molecular representations of each individuals (i.e. the patients). (**C**) We implemented six models containing both deep learning and machine learning, as well as three ensemble strategies. (**D**) The analytical strategies resulted in a total of 84 combinations for evaluating patient outcome prediction in each individual dataset. (**E**) We also evaluated the performance of analytical strategies on the combined dataset. To combine the dataset, we implemented three unique integration strategies with a total of five settings. We used the same base learning models and ensemble strategies as shown in (**C**). This resulted in another 420 combinations.

multi-view feature space on the same data. The fusion of multiple information, or ensemble learning [14], is a common technique to improve the performance of prediction model. There are various ways to fuse the information [15], including at the input feature level, at the model level and at the predicted outcome level. The question here is whether ensemble of features improves performance and which ensemble strategy is the most optimal. The increasing availability of single-cell datasets has led to the availability of multiple datasets focusing on the same interest, such as a particular disease. This strategy naturally lends itself to combining multiple datasets and enabling investigation that may not be possible with a single dataset. An important question to address is how to optimally integrate these datasets to achieve the best performance.

In this study, we examine the question of the optimal approaches as mentioned above using uniquely collected COVID-19 scRNA-seq datasets. To generate derived statistics for each patient sample, we utilize the recently developed scFeatures [8] method that constructs a multi-view representation across various feature types. We implement and compare different learning models from classical machine learning to modern deep learning models. We compare the performance of individual feature types as well as the ensemble of feature types by implementing a number of common ensemble strategies [14, 16]. In addition, using multiple COVID-19 datasets, we investigate the optimal data integration approach that maximizes prediction outcomes. Overall, through a comparison framework, we assess the combined impact of these key data analytical components (i.e. model choices, ensemble learning strategies and integration approaches) on COVID-19 outcome prediction.

# MATERIALS AND METHODS

## Design of a benchmarking study

Any benchmarking or comparison study typically involves three key elements. First, a collection of datasets is needed to evaluate the performance of methods without bias. Second, well-designed evaluation strategies are needed to compare methods or workflows. Third, evaluation metrics from multiple aspects are needed to quantify the performance. These elements are described in more detail in the following sections.

## Evaluation datasets collection

As the world has been heavily impacted by COVID-19 for over 3 years, the global effort in understanding this pandemic has made COVID-19 patient data perhaps one of the largest collections of multi-sample multi-condition single-cell datasets to date. Therefore, to examine the data analytics strategy for cohort analysis, we selected five large publicly available COVID-19 datasets containing individuals with mild, moderate, severe and critical disease progression (Figure 1A). This collection includes a total of 2 215 517 cells from 223 mild and moderate patients and 245 severe and critical patients. All datasets are composed of peripheral blood mononuclear cells (PBMCs) or whole blood samples. Additional details are provided in Table 1.

## Evaluation strategies for analytical combinations

The comparison study aims to examine the impact of various analytical strategies on individual level outcome prediction. To accomplish this, we utilized our recently developed feature engineering tool, scFeatures, to generate multi-view molecular

**Table 1.** Collection of COVID-19 PBMC datasets sequenced using scRNA-seq providing a total number of 2 million cells and 471 individuals

| Dataset name | Reference | Accession ID | Number of mild/moderate individuals | Number of severe/critical individuals | Number of mild/moderate and severe/critical individuals | Number of cells in mild/moderate and severe/critical individuals |
|---|---|---|---|---|---|---|
| Combat | [23] | EGAS00001005493 | 30 | 61 | 91 | 524 557 |
| Ren | [24] | GSE158055 | 68 | 85 | 153 | 872 663 |
| Schulte-Schrepping | [25] | EGAS00001004571 | 44 | 51 | 95 | 212 023 |
| Stephenson | [26] | E-MTAB-10026 | 58 | 32 | 90 | 493 685 |
| Wilk | [27] | GSE174072 | 23 | 19 | 42 | 112 589 |
| Total | | | 223 | 245 | 471 | 2 215 517 |

**Table 2.** Summary of the analytical choices implemented in this comparison study

| Analytical component | Analytical choice |
|---|---|
| Base learning model | (1) KNN<br>(2) Lasso<br>(3) Random forest<br>(4) SVM<br>(5) XGBoost<br>(6) Neural network |
| Ensemble strategy | (1) Concatenation<br>(2) Majority voting<br>(3) Stacking |
| Level of integration | (A) Cell level integration<br>(B) Individual level integration with no adjustment<br>(C1) Individual level integration with RUVg normalization (using $k = 5$), (C2) Individual level integration with RUVg normalization (using $k = 10$)<br>(D) Individual level integration with limma batch correction |

representation of each individual that served as input for downstream analytical models (Figure 1B).

The evaluation is composed of three key components (Table 2): (1) comparing the performance of multiple learning models using the generated features as input, (2) comparing single-view (i.e. using each of the feature types individually) and multi-view features (i.e. integrating multiple feature types) through ensemble strategies and (3) comparing integration strategies when using multiple datasets as the input. In Component 1, we surveyed and implemented multiple learning models from classical machine learning to modern deep learning methods (Figure 1C). In Component 2, we examined the difference in performance between using single-view feature space versus multi-view feature space via implementing multiple ensemble strategies (Figure 1D). In Component 3, given that we collected multiple COVID-19 datasets, we compared the performance of analytical choices on the combined dataset.

On each of the individual datasets, we examined a total of 84 analytical combinations from 11 feature representations, 6 base models and 3 ensemble strategies (as detailed in Table 2). On the combined dataset, we applied the same 84 combinations to each of the 5 integration settings, resulting in a further 420 combinations. Further details on each of the three components are given in the following subsections.

## Feature generation

We used scFeatures to generate the molecular representation for each individual (referred to as the patient representation hereafter) in each of the COVID-19 datasets. A total of 11 feature types from five feature categories were generated to reflect different views of the molecular property and were used for downstream analysis. In detail, the following feature representations were generated for each patient: (1) proportion ratio, (2) proportion raw, (3) proportion logit, (4) gene mean celltype, (5) gene proportion celltype, (6) pathway gsva, (7) pathway mean, (8) pathway proportion, (9) CCI, (10) gene mean aggregate and (11) gene proportion aggregated. Information regarding each of the feature types can be found in Table 3.

As detailed in Table 3, scFeatures extracts features from a given dataset independently for each patient. It does not rely on information from the entire dataset during the feature extraction step, thereby avoiding any potential issues of seeing the whole dataset (i.e. both the training and testing data) and data leakage in model training.

## Base model selection

To examine the effect of different learning models on individual level outcome prediction, we examined a selection of approaches from classical machine learning methods to the more recent deep learning approach. In the rest of the paper, we used the word 'machine learning' in its broadest definition to refer to both classical machine learning and deep learning methods.

For classical machine learning approach, we included a range of models including k-nearest neighbours (KNN), Lasso, Random Forest, support vector machine (SVM) with linear kernel and XGBoost with a linear booster using the implementation in the R package Caret [17]. Each feature type was used individually as the input to compare the performance of each feature type. The severity (mild/moderate and severe/critical) of the individuals' conditions was used as the outcome variable. For Lasso, which outputs the prediction in terms of probability instead of discrete outcome, we used 0.5 as the threshold.

For the representative deep learning approach, we implemented a neural network structure containing four fully connected layers. For each feature type, we used the same network structure but varied the number of nodes in the layers, depending on the number of features in the feature type. In detail,

**Table 3.** Implementation details on the 11 feature types utilized in this study

| Feature types | Implementation details |
|---|---|
| Proportion raw | Calculates the proportion of each cell type in each COVID-19 patient sample. |
| Proportion logit | Performs logit transformation of the cell type proportion as it is one of the most common transformations for proportional data. |
| Proportion ratio | Computes the pairwise ratio of two cell types' proportions, i.e. cell type 1 divided by cell type 2. This was calculated for each paired cell type combination. To avoid dividing by zero when a cell type is not present in a COVID-19 patient, we added 1 to both the numerator and denominator. The range of values was then scaled using log2 transformation. |
| Cell type specific gene expressions | Calculates the mean expression of genes within each cell type. We restricted to the top variable genes to reduce the dimensions of the feature. We calculate two sets of highly variable genes, (1) across all cells within each cell type and (2) across all cells.<br>First, for each cell type, the genes of interest were obtained by selecting the top 50 variable genes per sample, followed by taking the union of the genes across all samples.<br>Then, the top variable 50 genes across all cells were calculated per sample, followed by taking the union of the genes across all samples.<br>The final output is a vector of mean expression for the variable genes. |
| Gene proportion cell type | For each gene, we calculated the proportion that this gene is expressed across all cells. This was performed separately for each cell type of each patient. We then restricted to the top 50 variable genes using the same procedure defined in 'gene mean celltype'. The final output is a vector of proportion expressed for the subset of cell type specific genes. |
| Pathway gsva | We used 50 hallmark pathways from MSigDB [28] and the implementation provided by AUCell [29] to obtain the gene set enrichment score for every single cell of a patient. The enrichment score was then summarized for each cell type by averaging the scores from all the single cells within a cell type. As a result, this approach converts the matrix of gene expression by single cells into pathways by cell types for each patient. The matrix of pathways by cell types was further converted into a single vector by concatenating the scores from each cell type. |
| Pathway mean | For each of the 50 hallmark pathways, we averaged the gene expression values for all the genes in the pathway across all cells. This was done separately for each cell type of each patient. |
| Pathway proportion | For each of the 50 hallmark pathways, we averaged the gene expression values for all the genes in the pathway for each cell and used the third quantile of this value as a threshold. We then calculated the proportion of cells that have a higher average expression greater than the threshold. This is done separately for each cell type so that the final output for a patient is a cell-type-specific vector. |
| CCI | We implemented methods from the CellChat [30] package to calculate the cell–cell interaction probability between ligand and receptor pairs in every patient. This feature type is cell-type-specific, as the interaction between ligand and receptor is quantified separately for each cell type. The final output is a vector of interaction probabilities for each patient. |
| Gene mean aggregated | First, the mean expression of genes across all cells was computed for each sample. We then restricted to the top 1000 variable genes using the same procedure defined in 'gene mean celltype'. |
| Gene proportion aggregated | For each gene, we calculate the proportion that this gene is expressed across all cells for each patient. We then restricted to the top 1000 variable genes using the same procedure defined in 'gene mean celltype'. |

the input layer had a number of nodes equal to the number of features in the respective feature type. The second layer and third contained different numbers of nodes depending on the feature types. We describe the detailed implementation below:

(i) All feature types in the category 'cell type proportions' contained <100 features. For these feature types, we set both the first layer and second layer to 20 nodes.
(ii) All feature types in the category 'cell type specific pathway expressions', 'overall aggregated gene expressions' and 'cell–cell communications' contained <1000 features. For these feature types, we set the first layer to 500 nodes and the second layer to 100 nodes to reduce the dimension.
(iii) All feature types in 'cell type specific gene expression' contained <10 000 features. To reduce the dimensions for these feature types, we set the first layer to 1000 nodes and the second layer to 100 nodes.

The number of nodes in the output layer was the same for all feature types, with two nodes that output the probability of mild/moderate and severe/critical conditions, respectively. The condition with higher probability was considered the predicted condition.

## Ensemble strategies for multi-view features

We considered three types of ensemble strategies. scFeatures generates multiple feature types for a given patient, representing different and possibly complementary biological information (views). It is therefore of interest to examine the performance of multi-view versus single-view through ensemble learning. Here, we employed three types of ensemble strategies to obtain a 'multi-view' prediction. We used the term 'ensemble' in its broadest definition to refer to integrated learning at either feature or model level. Specifically, the implementation of these ensemble strategies is as follows:

(i) Early fusion using concatenation, which involved concatenating features across all feature types as the input. The implementation was the same for both machine learning and deep learning models as this strategy operates on the feature level.
(ii) Late fusion using stacked ensemble. This involved training each base learner on a single-view of the feature space followed by training a meta-learner to best combine the individual base learners. The implementation was different for machine learning and deep learning models. For machine learning models, base learners were trained and evaluated

on each of the individual feature types, resulting in 11 predictions for each patient. The predictions were then used as the input to build a logistic regression model, which serves as the meta-learner that combines the base learners to produce the final predicted outcome. For deep learning models, we implemented a network (Supplementary Figure 1) containing 11 subnetworks that took each of the 11 feature types as input. The subnetwork performed feature extraction for each of the feature types individually. We used the same network structure as the network described in the previous section that was used for extracting features from each feature type individually. The extracted features from each feature type were then concatenated, resulting in a vector of 860 features for each individual. This feature vector was then passed through a fully connected layer containing 50 nodes, followed by the output layer containing two nodes to produce the final prediction.

(iii) Score fusion using majority voting. We first obtained the predicted outcome from each of the 11 feature types, resulting in 11 predictions of either mild/moderate or severe/critical for each patient. Then the outcome with the most votes was considered to be the final predicted outcome for the patient. The implementation was the same for both machine learning and deep learning models.

## Levels of integration strategy

We examined different levels of integration to explore the optimal choice for predicting patient states when multiple datasets need to be combined and used as a whole in building a prediction model. The approaches are described in the following:

(i) Cell level integration—this approach refers to integration on count matrix: we used scMerge2 (personal communication) to perform data integration on the scRNA-seq count matrix. We then generated the patient representation using scFeatures on the integrated count matrix and used this as input for learning model.

(ii) Individual level integration with no adjustment: we simply concatenated the patient representation without any adjustment or normalization, and used this as input for learning model.

(iii) Integration level integration with normalization: we used a well-known batch correction method RUVg [18] as well as the removeBatchEffect function implemented in the limma package [19] to correct for the batch effect in the patient representation. In RUVg, $k$, the number of unwanted variations is a tunable parameter. We explored two settings of $k = 5$ (i.e. where the number of batches is equal to the number of datasets) and $k = 10$ (i.e. to introduce a stronger batch correction). For limma's removeBatchEffect function, we used the default setting. The batch-corrected patient representation was used as input for learning models.

Altogether, the three distinct integration strategies comprise a total of five settings.

## Evaluation metric

### Accuracy metric

To quantify the performance of the methods, we recorded the prediction accuracy of the severity outcome (Figure 1E). For the purpose of this comparison study, we combined the mild and moderate patients and referred to them as 'mild/moderate' throughout this study, as well as grouping the severe and critical patients and referring to them as 'severe/critical'. We then considered the classifiability between the 'mild/moderate' and 'severe/critical' patients.

To capture the variability in model performance, all classical machine learning and deep learning models were trained and tested with repeated 3-fold cross-validation using 20 repeats. The datasets were split into training and testing by completely random sampling. Therefore, to control for the potential impact of 'good' or 'bad' training/testing set splits, where a 'bad' split can result in extreme class imbalance in the modelling phase and affect model performance, we used the same training and testing splitting index across all machine learning and deep learning model to ensure a fair comparison. F1 score was used as the evaluation metric, as not all datasets are balanced.

### Aggregation of accuracy metric

Given the number of results from all analytical combinations, we aggregated the results to better quantify and interpret the results. First, we took the median F1 score across the 20 repeated cross-validation folds. This was then followed by different aggregation strategies depending on whether the input used individual or combined datasets.

For the result section where we dealt with the five datasets individually, we further aggregated the median F1 score across datasets by taking the median. Then, we ranked the feature types across each model choice as well as the model choice across each feature type to derive the ranking of feature types and the ranking of model choice.

## Computational resource metric

Apart from assessing the performance in terms of accuracy, we also assess the performance in terms of the computational resources. This was measured through running time and memory usage averaged over three repeats. All processes were executed using a research server with dual Intel(R) Xeon(R) Gold 6148 Processor with 40 cores, 768 GB of memory and two NVIDIA GeForce RTX 2080 Ti graphics cards.

The running time of each combination was measured using the Sys.time function built in R and the time.time function built in Python. For combinations involving machine learning models, the memory usage was quantified in terms of CPU memory. For combinations involving deep learning models, the memory usage was quantified as the sum of CPU and GPU memory, as the deep learning models were executed on GPU.

# RESULTS

## Certain ensemble strategies improve model performance

In this study, we examined the impact of using an ensemble of feature sets for predictive modelling in large cohort single-cell data by using scFeatures to extract 11 feature types for each individual. We compared the performance using prediction accuracy on patient outcomes across five COVID-19 patient datasets (Table 1, Supplementary Figure 2). Across the six learning approaches, we observed that certain ensemble strategies performed better than models based on individual features. In particular, majority voting consistently achieved the best performance, outperforming the other two ensemble strategies, as well as all individual features (Figure 2). This was followed by concatenation, which also performed better than using any of the individual features. These results highlight the effectiveness

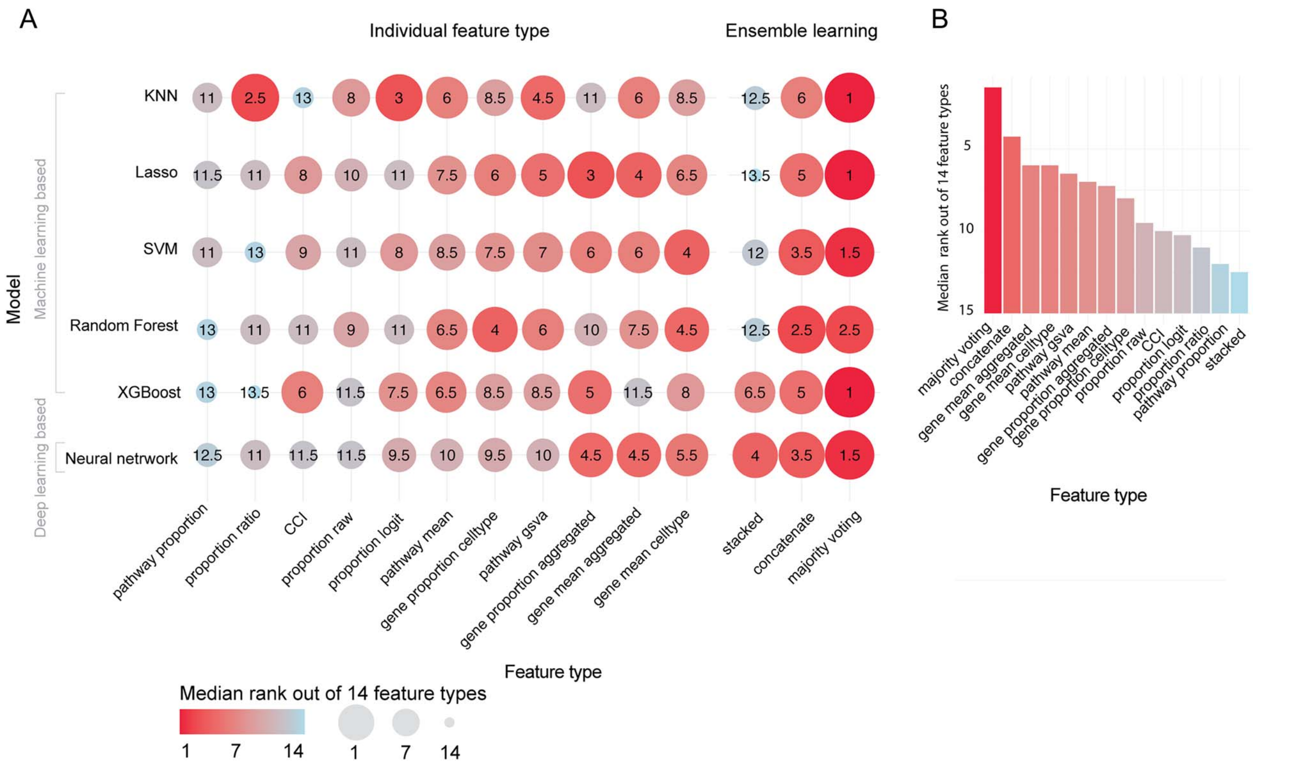

**Figure 2.** Performance of feature types for each model summarized across all datasets. (**A**) The dotplot shows the relative rank of each feature type to each other for each model, with 1 being the best and 14 being the worst. Ranks are summarized across the five datasets using the median and therefore do not necessarily range from 1 to 14 within each model. (**B**) Further summarizes the ranks of each feature type across all models using the median.

of ensemble learning and also suggest that the feature types are diverse, such that different feature types make different errors, combining them leads to improved model performance. Further examination of the top combinations of model and feature type (Figure 3) revealed that the top eight combinations all involves ensemble learning. Interestingly, the more complicated implementation of ensemble learning called stacked ensemble, in which a meta-learner is trained on the base learners trained from individual feature types, performed worse than using any of the individual feature types except for when deep learning was used.

We then took a closer examination at whether this observation is consistent irrespective of the learning model choice or dataset. We ranked the feature types on each of the six types of models and each of the five datasets. We observed that no individual feature type consistently ranked better or worse than others across all models and datasets (Figure 4). Almost all individual feature types had ranks that varied from 1 (the best rank) to 14 (the worst rank). This suggests that different feature types are useful for different models and different datasets, despite them all being COVID-19 datasets with mild/moderate and severe/critical individuals. In contrast, majority voting achieved a rank of 1 across multiple models and multiple datasets, again illustrating the power of ensemble strategy.

## Deep learning performs similarly to classical machine learning

Ranking the learning methods, we noted that there was no distinct difference between deep learning and some of the machine learning models (Figure 5A). In particular, both neural network achieved a median rank of 2 out of the 6 learning methods across the 14 feature types and 5 datasets, followed closely by random forest and SVM both with a median rank of 2.5 (Figure 5B). Within neural network, random forest and SVM, we then examined the difference between the maximum and minimum F1 score achieved by the three top-performing methods and observed a small median difference of 0.02 (Supplementary Figure 3). These result all suggest that deep learning do not significantly outperform certain machine learning models in this context.

We then compared the computational resource requirement to see whether the difference in performance came at a cost. Focusing on the feature type 'majority voting', we observed that both neural network and random forest took around 4 h on the largest Ren *et al.* dataset with 153 patients (Supplementary Figures 4 and 5). On the other hand, although SVM was ranked after neural network and random forest, it was more computationally efficient, taking <1 h on the Ren *et al.* dataset. Accounting for the significant difference in computational efficiency and the relatively small difference between model performance, one may consider SVM to be the optimal choice.

## Normalization is not necessary when combining multiple datasets as the input

Using multiple datasets as input data raises a number of questions, such as whether to integrate the raw data or the derived features. Here, we explored three categories of analytical combinations comprising of five settings. More specifically, different approaches to data integration, including individual level integration with no adjustment, cell level integration and individual level integration with normalization were explored. Our results are based on the examination of a total of 420 analytical combinations [5 integration settings × 14 feature types (11 individual feature types with 3 ensemble feature types) × 6 model choices].

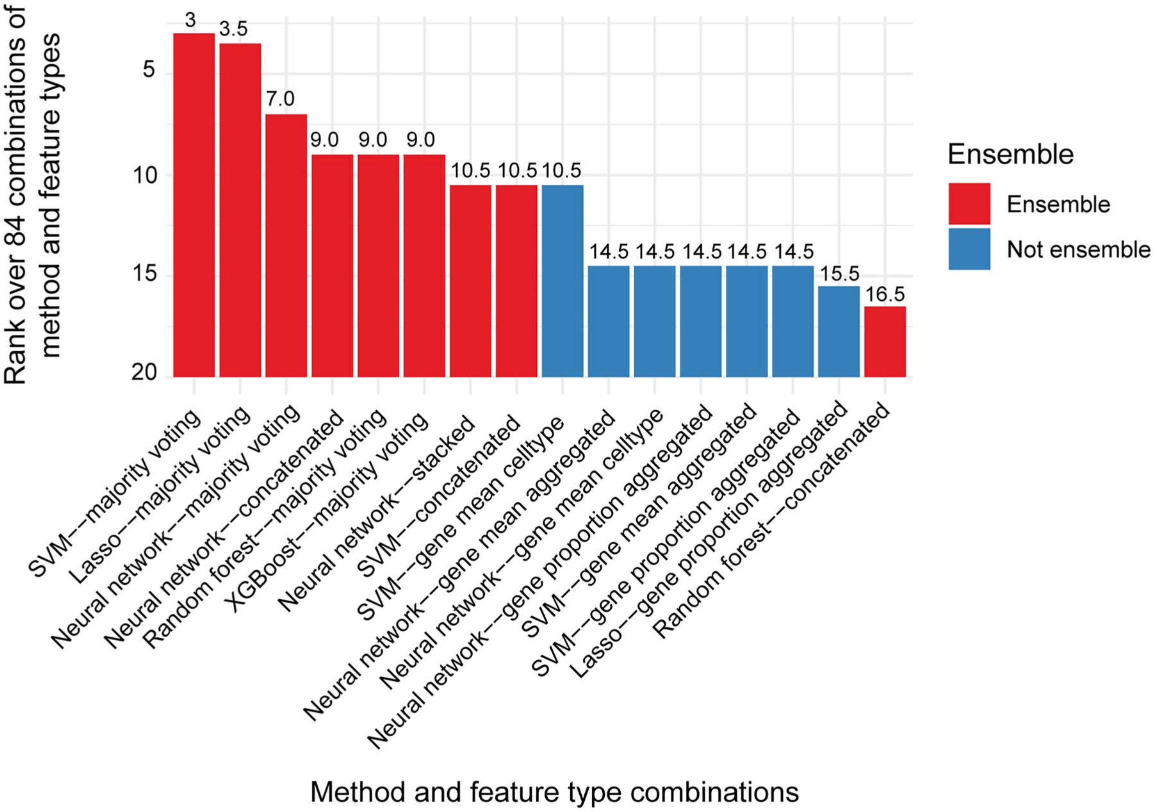

**Figure 3.** Top 16 combinations of model and feature type. Barplot shows the ranks of model and feature type. Given that the ranks are summarized across all five datasets using the median, the values do not necessarily range from 1 to 16.

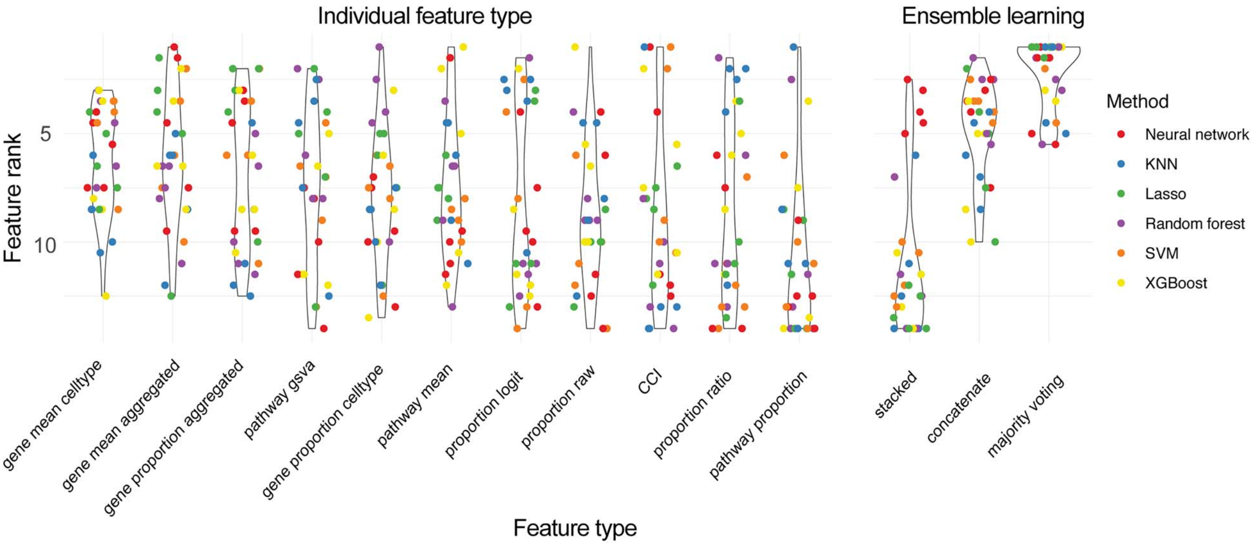

**Figure 4.** Performance of feature types for each model and each dataset. Violin plot shows the distribution of rank of each feature type for each model choice and each dataset choice. A total of 30 points are shown for each violin plot, as each feature type was evaluated on six models and five datasets.

Interestingly, there was only a slight difference between integration on the count matrix and individual level integration with no adjustment (Figure 6), which both achieved high F1 scores. On the other hand, individual level integration with normalization achieved lower F1 scores with both RUVg and limma's batch correction function. When utilizing RUVg, we further found that with the stronger the batch removal parameter setting, the worse the F1 score. Moreover, this observation is consistent across the choice of method and the type of feature used (Supplementary Figures 6 and 7). Using the data matrix from individual-level integration with no adjustment, we further selected any four of the datasets as the training datasets and tested the generalizability of the model performance on the remaining dataset. We showed that this external cross-validation resulted in similar performance (Supplementary Figure 8) as the internal cross-validation, thereby demonstrating not only the robustness of the features to dataset batch effect, but also the generalizability of the features.

One of the key strengths of data integration is the ability to examine condition-associated features for a subgroup of individuals. Due to the small number of individuals that typically fall

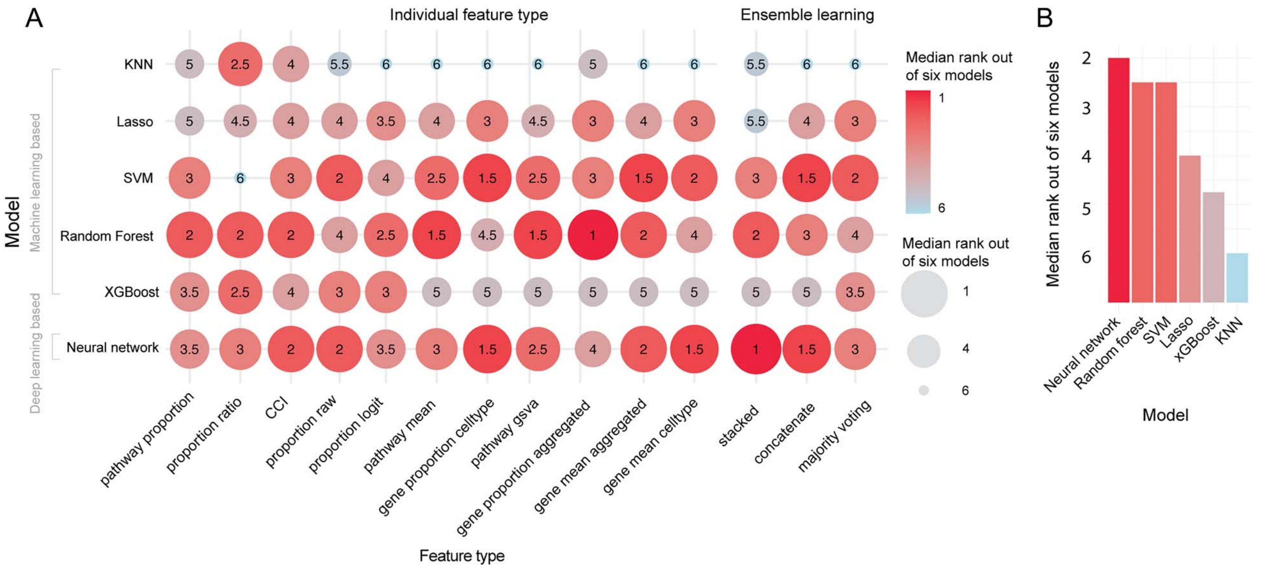

**Figure 5.** Performance of models. (**A**) Shows the relative rank of each model to each other for each feature type with 1 being the best and 6 being the worst. Ranks are summarized across the five datasets using the median and therefore do not necessarily range from 1 to 6 within each feature type. (**B**) Further summarizes the ranks of each model across all feature types using the median.

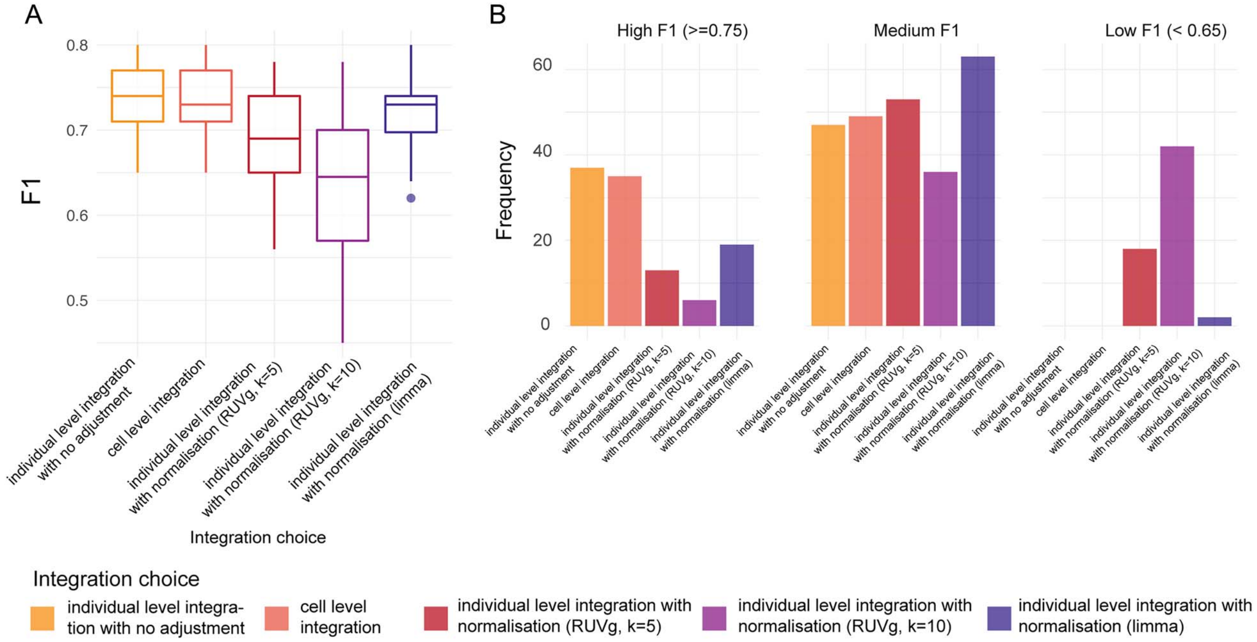

**Figure 6.** Performance of various approaches on combining multiple datasets for building prediction model. (**A**) Shows the F1 of these 420 analytical combinations, with the *x*-axis indicating the type of integration choice used in the combinations. (**B**) Further stratifies the F1 score based on high F1 (defined to be F1 ≥ 0.75), medium F1 score (defined to be 0.65 < F1 < 0.75) and low F1 score (defined to be F1 ≤ 0.65) and examines the proportion of each integration choice in the set of combinations that fall in the stratification.

into the subgroup of interest, this type of research is difficult to conduct using a single dataset. Here, we focused on a subgroup of patients in the 41–50 age group and investigated whether the identification of features is affected by different data integration strategies. First, we compared the rankings of the features obtained according to the feature importance score from the prediction model and found high consistency of the rankings between cell level integration and individual level integration with no adjustment (Supplementary Figure 9a). In comparison, the consistency was much lower between cell level integration and individual level integration with normalization. Clustering and dimension reduction on the features revealed that in both cases the clustering patterns and sources of variation of the patients were not driven by the dataset source (Supplementary Figure 9b and c). The lack of batch effect in the generated features suggests that the generated features may have self-adjusted in the feature extraction procedure, therefore explaining the minimal difference observed between the feature rankings and suggesting that there is no need for normalization on cell level or on individual level.

## DISCUSSION

In this comparison study, we explored different analytical approaches for predicting the severity of COVID-19 using multi-sample multi-condition scRNA-seq data. We used scFeatures to generate various feature representations for COVID-19 patients and examined the performance of individual feature types and ensemble feature types in classifying COVID-19 severity. By evaluating, using multiple datasets and multiple learning methods, from classical machine learning to modern deep learning methods, this study demonstrated that all machine learning methods perform similarly, with SVM being a slightly better method when accounting for the computational efficiency. Through implementing different ensemble strategies to incorporate multiple feature types as input into machine learning models, we revealed certain ensemble strategies, in particular majority voting, consistently led to increased performance compared with the non-ensemble strategy of using individual feature types alone. Stacked ensemble for example, often did not achieve better performance compared with using individual feature types. Finally, we suggest that when combining datasets is required for a prediction model, prior data integration is not necessary and does not necessarily improve prediction performance.

We observed that with the sets of COVID-19 datasets containing 42 to 153 patients, which is a realistic sample size in the current literature, the more complex approaches do not necessarily outperform simpler approaches. In particular, stacked ensemble can be considered the most complex implementation as it trains additional meta-learner on top of the base models. We observed that although the other two implementations (majority voting and concatenation) both performed better than individual features, stacked ensembled had worse performance compared with using the individual features. Furthermore, we observed minimal improvement when loading all single-cell data, a total of over 2 million cells for five datasets, as opposed to first generating patient-level features. With the extensive, and potentially prohibitive, computational resources required for such cell-level integration [20], such gain in model accuracy may not be worth the trade-off in computational resources.

In this study, the patient representation, being the explanatory variable, is derived from the scRNA-seq gene expression data. Other variables such as the clinical variables can also be useful in explaining the disease severity. In particular for COVID-19, age, gender and comorbidity are some of the clinical variables shown to be associated with severity outcomes [21]. These features can be easily concatenated to the patient representation derived from scFeatures given the availability of the data. Although this is beyond the scope of this study, having a focus on the utility of features extracted from the scRNA-seq data itself, future research could explore the use of clinical variables to provide additional insight into disease severity.

For clarity on the patient outcome classification, we focused on the binary classification of mild/moderate and severe/critical. A multi-class classification model would provide additional insights into disease severity and outcome, especially for the finer level of disease progression. We note that some of the benchmarked methods such as deep learning can naturally handle multi-class classification, and others such as SVM with linear kernel can achieve multi-class classification through using one-versus-all approach [22]. Thus, this benchmark framework can be easily extended to examine multi-class classification. Another area worth future investigation is the prediction of patients' future developmental trends based on their current molecular characteristics. This may enable early intervention and is highly important in healthcare. Although such time-resolved patient data is scarce, it nevertheless presents new opportunities.

Recently, there have been a growing number of multi-sample multi-condition datasets. Although this opens opportunities for patient level analysis such as case–control study, it also opens new questions on what are the representative methods for each question, what are the appropriate quantitative evaluation metrics to assess each method, what are the recommended approaches for answering a given question with data of certain characteristics and what are the guidelines for future methods development. In this study, we utilized five COVID-19 patient datasets to evaluate the choices of the method, ensemble strategy and integration strategy and obtained consistent trends. We envisage the current comparison framework will point valuable direction into an optimized analytical combination for outcome prediction using single-cell data in future where cohort study with more than a few hundred or over a thousand patients are readily available.

**Key Points**

- This work assesses and compares the performance of three categories of workflow consisting of 504 analytical combinations for outcome prediction using multi-sample, multi-conditions single-cell studies.
- We observed that using ensemble of feature types performs better than using individual feature types.
- We found that in the current data, all learning approaches including deep learning exhibit similar predictive performance. When combining multiple datasets as the input, our study found that integrating multiple datasets at the cell level performs similarly to simply concatenating the patient representation without modification.

## SUPPLEMENTARY DATA

Supplementary data are available online at https://academic.oup.com/bib.

## ACKNOWLEDGEMENTS

The authors thank their colleagues at The University of Sydney, Sydney Precision Data Science Centre, School of Mathematics and Statistics, Charles Perkins Center for intellectual engagement. The authors would like to recognize that some of the content in this paper was developed and previously included in the first author's PhD thesis.

## FUNDING

The following sources of funding for each author, and for the manuscript preparation, are gratefully acknowledged: Y.C. is supported by Research Training Program Tuition Fee Offset and University of Sydney Postgraduate Award Stipend Scholarship; S.G. is supported by an Australian Research Council Discovery Early Career Researcher Awards (DE220100964) and Chan Zuckerberg Initiative Single Cell Biology Data Insights (Grant DI-0000000027). J.Y.H.Y. and P.Y. are supported by the AIR@innoHK programme of the Innovation and Technology Commission of Hong Kong. P.Y. is supported by a National Health and Medical Research Council (NHMRC) Investigator Grant (1173469).

## DATA AVAILABILITY

The data that support the findings of this study are publicly available and the accession ID is reported in Table 1.

## CODE AVAILABILITY

The scFeatures package used for extracting patient representation is freely available at our Github page: https://github.com/SydneyBioX/scFeatures.

## REFERENCES

1. Svensson V, da Veiga BE, Pachter L. A curated database reveals trends in single-cell transcriptomics. *Database* 2020;**2020**: baaa073.
2. Zappia L, Theis FJ. Over 1000 tools reveal trends in the single-cell RNA-seq analysis landscape. *Genome Biol* 2021;**22**:301.
3. Zhang M, Guo FR. BSDE: barycenter single-cell differential expression for case–control studies. *Bioinformatics* 2022;**38**: 2765–72.
4. Yang P, Huang H, Liu C. Feature selection revisited in the single-cell era. *Genome Biol* 2021;**22**:321.
5. Sun G, Li Z, Rong D, *et al.* Single-cell RNA sequencing in cancer: applications, advances, and emerging challenges. *Mol Ther Oncolytics* 2021;**21**:183–206.
6. Zhu S, Qing T, Zheng Y, *et al.* Advances in single-cell RNA sequencing and its applications in cancer research. *Oncotarget* 2017;**8**:53763–79.
7. Armingol E, Officer A, Harismendy O, *et al.* Deciphering cell–cell interactions and communication from gene expression. *Nat Rev Genet* 2020;**22**:71–88.
8. Cao Y, Lin Y, Patrick E, *et al.* scFeatures: multi-view representations of single-cell and spatial data for disease outcome prediction. *Bioinformatics* 2022;**38**:4745–53.
9. Thurman AL, Ratcliff JA, Chimenti MS, *et al.* Differential gene expression analysis for multi-subject single-cell RNA-sequencing studies with *aggregateBioVar*. *Bioinformatics* 2021;**37**: 3243–51.
10. Millard N, Korsunsky I, Weinand K, *et al.* Maximizing statistical power to detect differentially abundant cell states with scPOST. *Cell Rep Methods* 2021;**1**:100120.
11. Zhao J, Jaffe A, Li H, *et al.* Detection of differentially abundant cell subpopulations in scRNA-seq data. *Proc Natl Acad Sci U S A* 2021;**118**:e2100293118.
12. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015;**521**: 436–44.
13. Sejnowski TJ. The unreasonable effectiveness of deep learning in artificial intelligence. *Proc Natl Acad Sci U S A* 2020;**117**:30033–8.
14. Yang P, Hwa Yang Y, Zhou BB, *et al.* A review of ensemble methods in bioinformatics. *Curr Bioinform* 2010;**5**:296–308.
15. Seeland M, Mäder P. Multi-view classification with convolutional neural networks. *PLoS One* 2021;**16**:e0245230.
16. Cao Y, Geddes TA, Yang JYH, *et al.* Ensemble deep learning in bioinformatics. *Nat Mach Intell* 2020;**2**:500–8.
17. Kuhn M. Building predictive models in R using the caret package. *J Stat Softw* 2008;**28**:1–26.
18. Risso D, Ngai J, Speed TP, *et al.* Normalization of RNA-seq data using factor analysis of control genes or samples. *Nat Biotechnol* 2014;**32**:896–902.
19. Ritchie ME, Phipson B, Wu D, *et al.* Limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res* 2015;**43**:e47.
20. Luecken MD, Büttner M, Chaichoompu K, *et al.* Benchmarking atlas-level data integration in single-cell genomics. *Nat Methods* 2021;**19**:41–50.
21. Williamson EJ, Walker AJ, Bhaskaran K, *et al.* Factors associated with COVID-19-related death using OpenSAFELY. *Nature* 2020;**584**:430–6.
22. Rifkin R, Klautau A. In defense of one-vs-all classification. *J Mach Learn Res* 2004;**5**:101–41.
23. Covid-19 Multi-omics Blood ATlas (COMBAT) Consortium. A blood atlas of COVID-19 defines hallmarks of disease severity and specificity. *Cell* 2022;**185**:916–38.e58.
24. Ren X, Wen W, Fan X, *et al.* COVID-19 immune features revealed by a large-scale single-cell transcriptome atlas. *Cell* 2021;**184**:5838.
25. Schulte-Schrepping J, Reusch N, Paclik D, *et al.* Severe COVID-19 is marked by a dysregulated myeloid cell compartment. *Cell* 2020;**182**:1419–1440.e23.
26. Stephenson E, Reynolds G, Botting RA, *et al.* Single-cell multi-omics analysis of the immune response in COVID-19. *Nat Med* 2021;**27**:904–16.
27. Wilk AJ, Lee MJ, Wei B, *et al.* Multi-omic profiling reveals widespread dysregulation of innate immunity and hematopoiesis in COVID-19. *J Exp Med* 2021;**218**: e20210582.
28. Liberzon A, Birger C, Thorvaldsdóttir H, *et al.* The molecular signatures database (MSigDB) hallmark gene set collection. *Cell Syst* 2015;**1**:417–25.
29. Aibar S, González-Blas CB, Moerman T, *et al.* SCENIC: single-cell regulatory network inference and clustering. *Nat Methods* 2017;**14**:1083–6.
30. Jin S, Guerrero-Juarez CF, Zhang L, *et al.* Inference and analysis of cell-cell communication using CellChat. *Nat Commun* 2021;**12**:1088.