Human Vaccines & Immunotherapeutics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/khvi20

Predictive overfitting in immunological applications: Pitfalls and solutions

Jeremy P. Gygi, Steven H. Kleinstein & Leying Guan

To cite this article: Jeremy P. Gygi, Steven H. Kleinstein & Leying Guan (2023) Predictive overfitting in immunological applications: Pitfalls and solutions, Human Vaccines & Immunotherapeutics, 19:2, 2251830, DOI: 10.1080/21645515.2023.2251830

To link to this article: https://doi.org/10.1080/21645515.2023.2251830

Published online: 12 Sep 2023.

HUMAN VACCINES & IMMUNOTHERAPEUTICS
2023, VOL. 19, NO. 2, 2251830
https://doi.org/10.1080/21645515.2023.2251830

Taylor & Francis
Taylor & Francis Group

REVIEW ARTICLE

OPEN ACCESS Check for updates

Predictive overfitting in immunological applications: Pitfalls and solutions

Jeremy P. Gygi[a], Steven H. Kleinstein[a,b,c], and Leying Guan[a,d]

[a]Program in Computational Biology & Bioinformatics, Yale University, New Haven, CT, USA; [b]Department of Pathology, Yale School of Medicine, New Haven, CT, USA; [c]Department of Immunobiology, Yale School of Medicine, New Haven, CT, USA; [d]Department of Biostatistics, Yale School of Public Health, New Haven, CT, USA

ABSTRACT

Overfitting describes the phenomenon where a highly predictive model on the training data generalizes poorly to future observations. It is a common concern when applying machine learning techniques to contemporary medical applications, such as predicting vaccination response and disease status in infectious disease or cancer studies. This review examines the causes of overfitting and offers strategies to counteract it, focusing on model complexity reduction, reliable model evaluation, and harnessing data diversity. Through discussion of the underlying mathematical models and illustrative examples using both synthetic data and published real datasets, our objective is to equip analysts and bioinformaticians with the knowledge and tools necessary to detect and mitigate overfitting in their research.

ARTICLE HISTORY
Received 1 May 2023
Revised 27 July 2023
Accepted 21 August 2023

KEYWORDS
Overfitting; regularization; dimension reduction; model evaluation; data diversity; distributionally robust optimization

Introduction

Machine learning (ML) and statistical modeling have become important tools in modeling medical data and prediction of disease outcomes and vaccination responses in immunological research.[1–5] However, the application of ML algorithms necessitates caution. A common occurrence in misusing ML algorithms is overfitting, which arises when a predictive model fits well to training data but performs poorly on new data due to excessive model complexity.[6,7] The implications of overfitting in medical research can result in the erroneous publication of immunological markers that are highly predictive on the training data but generalize poorly to untouched test datasets. Put another way, overfitted models perform well in their respective studies, but do not generalize to novel datasets. To this end, recognizing and avoiding common pitfalls that can lead to overfitting in contemporary immunological research is of paramount importance.

This review delves into the prevalent scenarios that contribute to overfitting and then presents strategies to counteract its effects. We structure the content into three primary themes:

- The double-edged nature of model complexity.
- Reliable model evaluation.
- Harnessing data diversity.

Our objective is to provide analysts and bioinformaticians with the concepts and tools required to detect and circumvent overfitting in their modeling and analysis endeavors, informed by the latest advances in statistics and machine learning.

Double-edged nature of model complexity

Model complexity, which quantifies the complexity of a model and its fit to the training data, is a crucial concept in understanding the phenomenon of overfitting. A predictive model's complexity increases with increased number of independent features, such as analytes. Model complexity also increases when a more intricate model architecture is used, such as comparing linear regression to a deep neural network. Modern immunological studies have enabled access to a vast array of tens of thousands of analytes. In addition, off-the-shelf machine learning tools have facilitated easy access to nonlinear modeling for capturing the interactive effects among biological processes.[8] Both of these trends have given rise to more complex and flexible models and reduce the training errors which measure the estimation error of the response on the training data. However, a flexible model does not always lead to improved prediction accuracy on new data (referred to as test data) not used during model fitting.

On the one hand, a more complex model might have a smaller test error which measures the prediction error of the response on the untouched test data. This is achieved by reducing the model bias which measures the distance between the expected/average estimated models and the underlying true model. Here, the averaging of the estimated models is done across models constructed using repeated regenerations of training data. On the other hand, it is important to note that fitting a more complex model also introduces a higher model variance in the prediction function. That is, if the training data were regenerated independently, the resulting fitted model could differ substantially. This interplay between model complexity, model bias, and model variance is commonly referred to as the bias-variance

CONTACT Leying Guan leying.guan@yale.edu Department of Biostatistics, Yale School of Public Health, 60 College Street, New Haven, CT, USA.

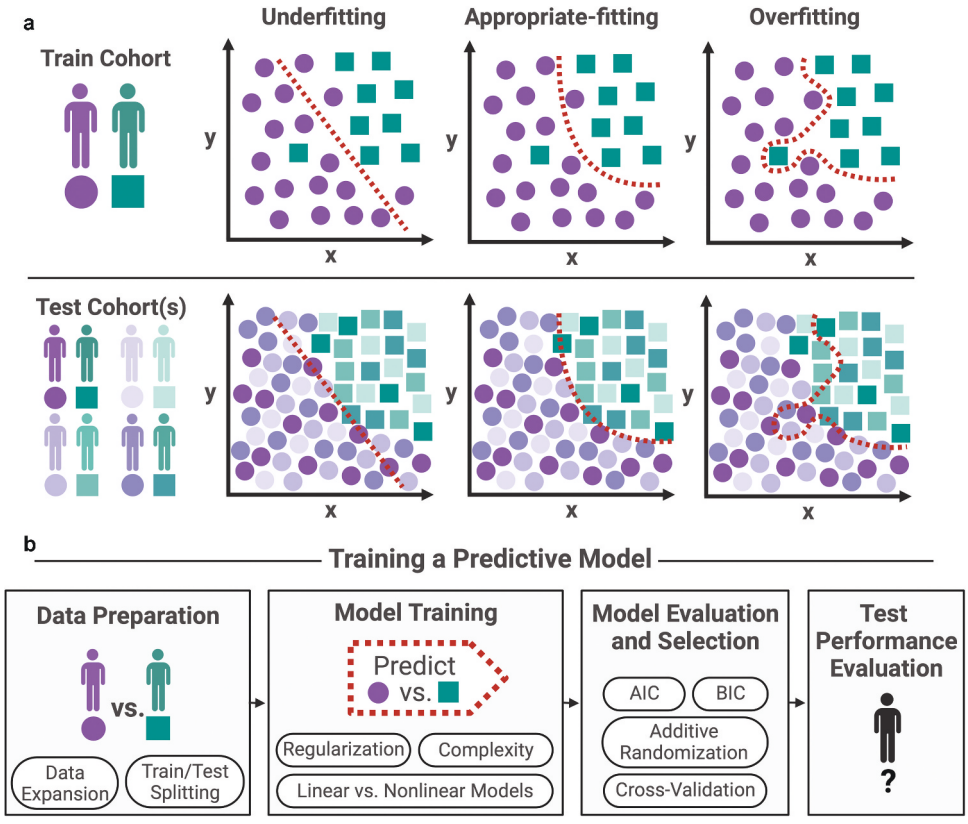

Figure 1. (a) Examples of underfitting, appropriate-fitting, and over-fitting a machine learning model to a training cohort. Underfitting oversimplifies the relationship between predictive features whereas over-fitting fails to generalize to novel test cohorts that were not used to train the machine learning model. (b) Schema for effective training of a predictive model, including data preparation, model training, model evaluation and selection, and finally test performance evaluation. Tools for good machine learning practice provided are further explored in the manuscript.

tradeoff. This offers a clear mathematical perspective on how model complexity affects the prediction performance on novel test data: while an overly simplified model might fail to capture strong predictive relationships and lead to underfitting, excessively high model complexity can cause the model to overfit by excessively fitting to the noise in the training data, thereby compromising its performance (Figure 1a).

The effects of underfitting and overfitting become evident when evaluating the model on novel datasets. For instance, if we employ ordinary least squares (OLS) to fit a linear regression model with an equal or greater number of features than the features in the training dataset, the model can usually perfectly explain the response in the training cohort. However, such models often fail to generalize well when predicting outcomes for new test samples. The same overfitting issue may happen when adopting a highly non-linear model. Another example described in a study by Peng et al.[9] involves the use of support vector regression to forecast COVID-19 cases in severely affected countries, employing both linear and non-linear predictive models. While the non-linear fit demonstrated superior performance during training, it was the linear fit that achieved the best results on the test data. These two examples emphasize the need for caution when employing machine learning models in medical decision-making.

Finally, to illustrate the issue of overfitting in the immunological application, we present Example 2.1 where xgboost[10,11]

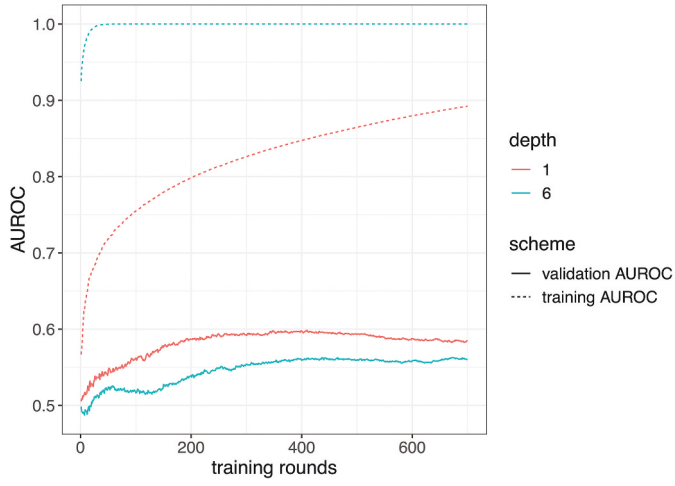

Figure 2. Training AUROC and validation AUROC for predicting high/low responders (y-axis) using xgboost against the training rounds (x-axis). It shows the model performances in training or cross-validation as the training rounds increased (increased model complexity) and compares the prediction performance when using trees with depth being 1 (low non-linearity) or 6 (high non-linearity).

was employed to identify common immune signatures that predict antibody responses using a recently curated database.

Example 1. Consider the identification of common PBMC transcriptomics signatures for predicting antibody responses

across 13 different vaccines by Fourati et al.,[7] covering live viruses, inactivated viruses, and glycoconjugate vaccines, collected by the Human Immunology Project Consortium (HIPC) as the Immune Signature Data Resources.[14] We demonstrate the phenomenon of overfitting by constructing models with varying complexity using the 500 most variable genes to separate high-responder (267 participants) from low-responder (265 participants) groups, defined from discretizing maximum fold change antibody responses (day 28/day 0). To evaluate how well the model generalizes to new test samples, we split the data into the training set and the validation set, with the model trained using only the training and classification accuracies recorded for both the training and the validation. This scheme was repeated five times, and results were averaged to increase the evaluation stability. The classification accuracy is measured by area under the receiver operating characteristic (AUROC), with higher value indicating better classification. Figure 2 displays the training AUROC and validation AUROC of prediction models trained from xgboost over the training rounds, with the tree depth being 1 or 6. At each training round, xgboost constructs a tree of depth 1 or 6 to improve the current fit and aggregate the newly constructed tree into the prediction model to increase model complexity. As the training rounds increased, the predictions within training samples achieved high accuracy, with AUROC improving for both tree depths. However, the validation AUROC is much lower. Selecting a model with highest training AUROC can lead to overfitting where the validation AUROC is worse compared to earlier training rounds. Furthermore, the model with a tree depth of 6 quickly overfitted from the inclusion of highly non-linear. It achieved almost perfect training AUROC but achieved worse validation AUROC than the simpler xgboost model with a tree depth of 1.

Ideally, the objective is to construct a model that strikes a favorable tradeoff between model bias and model variance, thereby achieving an appropriate level of fitting for good prediction performance on the test samples (Figure 1a). In this review, we discuss three important aspects of good practices for appropriate model fitting: model training, model selection, and evaluation (Figure 1b). Finally, we review the utilization of data diversity and its application to avoiding overfitting.

Model complexity reduction

Learning the available strategies to effectively control or reduce model complexity is an important step in taking full advantage of the wide array of machine learning tools. Here, we review model complexity reduction approaches based on regularization and dimensionality reduction.

Regularization

Regularization is one of the most widely used techniques for reducing the model complexity of prediction models. Given a loss function, which is the user-specified objective that measures the goodness of model fit, e.g., mean-squared error is the loss function for OLS, the standard regularization involves adding a penalty term to the loss function to discourage the model from learning overly complex patterns. As a concrete example, let x_i and y_i be the observed p analytes and response for training sample i, with $i = 1, \ldots, n$, for n total samples. Regularized linear regression considers the following regularized loss function:

$$L_\lambda(\beta) = \frac{1}{2}\sum_{i=1}^{n}(x_i\beta - y_i)^2 + \lambda J(\beta),$$

where β is a vector contains the coefficients of the linear model and β_j is the coefficient for analyte j for $j = 1, \ldots p$, and $J(\beta)$ is the regularization or penalty term to encourage simpler β, thus, a less complex model. λ is the amount of penalty included, with larger λ favoring a simpler model. Some of the popular forms for $J(\beta)$ are listed below.

- Best subset selection defines $J(\beta) = \sum_{j=1}^{p}\left|\beta_j\right|^0$ with $\left|\beta_j\right|^0$ being 0 if β_j is 0 and 1 otherwise. It penalizes the number of non-zero entries in β. The problem of best subset selection can be difficult to solve in general.[12] There have been many approximation algorithms for this purpose. For example, forward stepwise selection can be seen as a greedy algorithm for the best subset selection.[13]
- Ridge (or l_2) regularization defines $J(\beta) = \sum_{j=1}^{p}\left|\beta_j\right|^2$ as the square sums of entries in β.[14]
- Lasso (or l_1) regularization defines $J(\beta) = \sum_{j=1}^{p}\left|\beta_j\right|$ to penalize sum of the absolute values of entries in β[15]

In the family of penalty loss $J(\beta) = \sum_{j=1}^{p}\left|\beta_j\right|^\alpha$ that penalizes the α^{th} power of $\left|\beta_j\right|$,the lasso penalty with $\alpha = 1$ is a special case because it is the turning point where we can encourage sparsity (a small number of non-zero entries in β) while being able to solve the problem effectively. That is, when $\alpha \leq 1$, sparsity is encouraged in β, but it is difficult to solve the general problem exactly for all $\alpha < 1$.

There are other types of regularization penalties proposed by statisticians. For example, the elastic net penalty[16] considers a mixture of lasso penalty and ridge penalty to encourage sparsity and the co-selection of highly correlated features, which can be desirable in immunological applications for interpretation.[17] The grouped lasso penalty encourages the co-selection of features from the same group specified by the users.[18]

The idea of regularization through penalty losses is also applicable to other non-linear machine learning techniques, such as boosting and neural networks. In addition, regularization does not only come in the form of penalty losses. For example, dropout is a popular and effective technique to prevent overfitting in neural networks.[19] At each updating step during training, a percentage of nodes and connections from the neural network are randomly dropped. This can also be viewed as training with an adaptive regularizer that incorporates the effect of artificial feature noising.[20]

Early stopping is another technique used to train machine learning models, including boosting and neural networks, to prevent overfitting by stopping the training process before the model starts to overfit the training data.[7,21] This procedure can

be viewed as a form of implicit regularization with effects similar to ridge regularization in some cases.[22,23]

Dimension reduction

Dimension reduction can also be an effective way to reduce model complexity and is commonly used in contemporary immunological studies. It has been one of the main techniques for working with high-dimensional immune profiles, such as high-dimensional transcriptomics or complex multi-omics observations.[24–26] Dimension reduction addresses the challenge of high dimensionality by constructing low-dimensional factors to capture data variation from available omics. These factors can be used for both biological interpretation and prediction tasks of interest (Figure 3).[27,28] Working with low-dimensional factors helps to alleviate potential overfitting issues because the number of features, and therefore model complexity, is reduced.

Matrix factorization provides core concepts for dimension reduction with high-dimensional data and serves as the foundation of many dimension reduction tools for omics and multi-omics analyses in immunological studies. Suppose we have collected observations for p analytes across n samples and denote the observed matrix as $X \in \mathbb{R}^{n\times p}$ for these n samples and p analytes. The number of analytes, p, is often large. The central goal of matrix factorization is to approximate X by the product of*a* factor matrix $U \in \mathbb{R}^{n\times K}$ and a loading matrix $V \in \mathbb{R}^{p\times K}$ for some K much smaller then p :

$$X \approx UV^{\top}.$$

The factor matrix is constructed for capturing the variation across n samples using K factors (columns in U), and the loadings matrix describes each factor's influence on the analytes (columns in V). For example, in gene expression analysis, it can identify groups of genes that may be involved in underlying biological processes or functions, with factors capturing the variation of these processes across samples.

Widely used matrix factorization techniques principal component analysis (PCA) and single-value decomposition (SVD)[29,30] derive U such that U can explain the maximum amount of variance across features in X. Non-negative matrix factorization (NMF) requires additionally that U and V contain only non-negative values, which was originally proposed to work with image data where this non-negativity constraint leads to better interpretability.[31] The sparse PCA algorithm imposes that columns in V have sparse non-zero entries, which can lead to higher quality estimation in high dimensions and often improved interpretability by highlighting a smaller set of features for each factor.[32] There are many other variants of PCA for dimension reduction that could prove useful in analyzing omics data, such as hub-feature identification which identifies a small subset of hub-features as drivers of the systematic level changes[33,34] and autoencoder methods for dimension reduction with non-linear model architectures.[35]

Of note, the concept of matrix factorization is frequently combined with co-expression networks in omics analysis.[36,37] For instance, the widely used weighted correlation network analysis (WGCNA)[38] approach first identifies modules as non-overlapping subsets of co-expressed genes or other features and uses the eigenvectors for genes from each module to capture variability of features in the given module.

Given the increasing availability of large-scale multi-omics data, it is crucial to perform dimension reduction appropriately with multi-omics observations. While matrix factorization approaches can be directly applied after concatenating features from different omic assays, this is generally not advisable due to the varying dimensionality and data properties from different technologies. For example, instead of using WGCNA on concatenated data, it has been suggested to perform a two-step procedure where we first apply WGCNA to

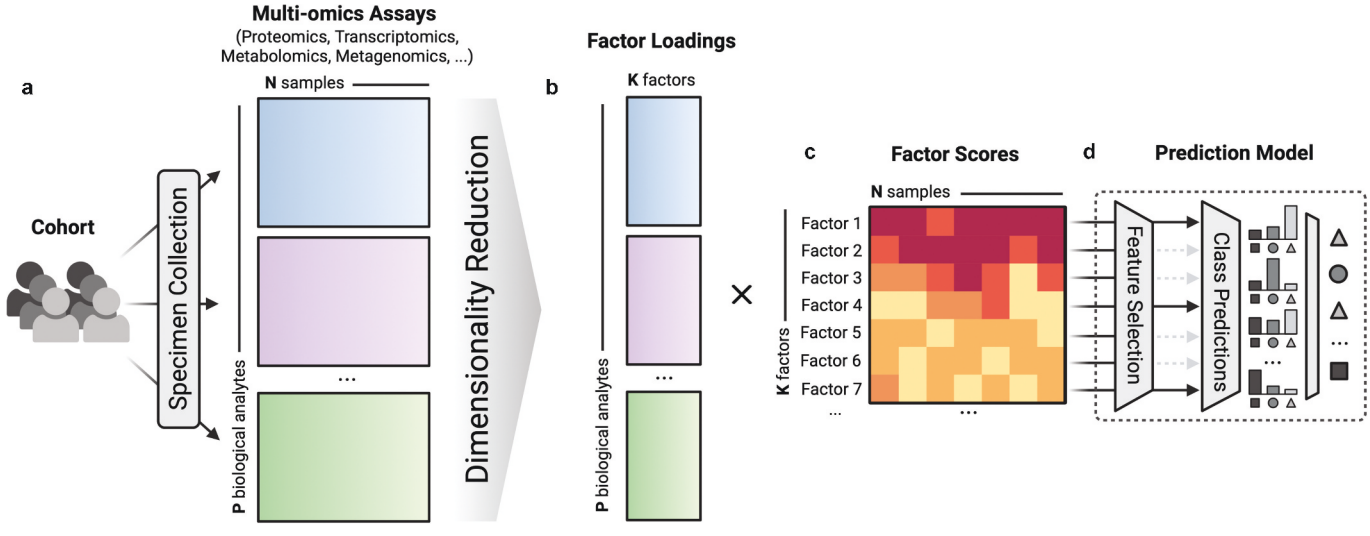

Figure 3. An illustration workflow of multi-omics dimensionality reduction. (a) Multi-omics assays are profiled from the same cohort, resulting in multi-omics profiles for the same N samples. (b) P biological analytes are condensed via dimensionality reduction into the construction of low-dimensional factors, consisting of factor loadings and factor scores. Factor loadings are coefficients that indicate which biological analytes are contributing to construction of the each factor. (c) All N samples from the cohort are assigned a score for each of the K factors, resulting in the factor scores matrix. (d) The resulting factor scores can be used as machine learning features to predict responses of interest in a prediction model.

individual omics assays and align different assays as a second step.[39]

Many popular matrix-factorization-based approaches have been developed for dimension reduction with multi-omics data, such as JIVE[40] as a generalization of PCA, jNMF and iNMF as generalizations of NFM.[41–43] Multi-omics integration methods based on Bayesian factor models have also gained popularity, which enables flexible modeling of different omic data types and introduction of suitable priors. Multi-omics integration tools like iCluster,[44] iClusterBayes,[45] and MOFA[46,47] employ such Bayesian factor models.

A different perspective for dimension reduction is taken by Canonical correlation analysis (CCA),[48] which extracts factors to capture co-varying patterns between two matrices instead of aiming for the maximization of explained variance as in matrix factorization. This idea can be useful by connecting different types of omics together to form a more comprehensive picture of the underlying biology. For instance, Li et al.[28] linked metabolomic profiles with transcriptomic profiles using CCA to gain an improved understanding of vaccination response. This concept has been generalized to work with multiple data blocks simultaneously, and we consider all methods based on the generalization of CCA as belonging to the generalized CCA (GCCA) family. GCAA has motivated the development of several multi-omics integration methods such as multiple co-inertia analysis (MCIA),[49,50] regularized GCCA (rGCCA),[51,52] and multi-block sparse CCA (msCCA).[53]

In this direction, supervised dimension reduction techniques have been proposed to identify factors predictive of responses of interest.[54–59] In multi-omics analysis, for example, DIABLO[58] includes the response matrix as another assay in dimension reduction using GCCA, while SPEAR[59] prioritizes the construction of factors in a Bayesian multi-omics factor model. To some extent, we can view supervised dimension reduction as preventing overfitting by using regularization to deviate from the meaningful data variation directions.

Reliable model evaluation and selection

In the previous section, we discussed many popular techniques for model complexity reduction, which can effectively avoid overfitting. Equally important is the reliability of model evaluation, crucial for parameter tuning, model selection, and understanding the prediction model and its usefulness. In this section, we will review model evaluation methods and discuss potential pitfalls.

AIC, BIC, and additive randomization

Akaike Information Criterion (AIC)[60] jointly considers the goodness-of-fit and model complexity, measured by the number of parameters in the model:

$$\mathrm{AIC} = -\frac{2}{n}\cdot \mathrm{loglik} + 2\cdot\frac{s}{n},$$

where *loglik* is the achieved log-likelihood summed over all n training samples, and s is the number of parameters used to characterize model training. For instance, if we consider a linear regression model where the noise follows a standard normal distribution, then, $\frac{2}{n}\cdot$ loglik becomes the mean-squared error in OLS and s is the number of features used in the linear model. In this case, the AIC statistic is equivalent to Mallow's C_p statistic,[61] whose expectation is unbiased for the prediction error. In general, the AIC aims to achieve this unbiasedness. Hence, by design, lower AIC statistics indicate better fitting of models, which can be very useful for selecting among competing models.

However, AIC may not always achieve this goal in practice. One significant issue in using AIC is determining s, as it is common that the number of parameters appearing in the trained model, does not always reflect the actual complexity of the model's training process.

Bayesian Information Criterion (BIC)[62] adopts a similar form to AIC, with more penalization on s:

$$\mathrm{BIC} = -\frac{2}{n}\cdot \mathrm{loglik} + p\cdot\frac{s}{n},$$

where p is the total number of parameters. Since $p>2$ in most settings, it penalizes complex models more heavily and favors the selection of simpler models compared to AIC. Despite the similarity between BIC and AIC, BIC has a different motivation. Approximately, selection of the model with minimum BIC is equivalent to choosing the model with the largest posterior probability in the Bayesian framework.[63]

Determining s to reflect the model complexity can be challenging at times, even in the linear regression setting as demonstrated in Example 3.1.

Example 2: Suppose that $x \in \mathbb{R}^{1000}$ is a vector contains 1000 features, where each feature x_j is generated from a standard normal distribution for $j = 1, \ldots, 1000$, and the response $y = x_1\beta_1 + \varepsilon$ depends only on the value of the first feature. Independent noise (ε)is then generated from a standard normal distribution. Figure 4a,b show training errors, AIC, BIC, and prediction errors as well as error estimates from the additive randomization method (see later) for forward stepwise selection with subset size $0 \le s \le 10$, averaged over 10 random repetitions.

The issue with using AIC and BIC in Example 2 arises from the fact that the model at each subset size s is not fixed but adaptively constructed and selected from the data, which is common with high-dimensional data. Following the endeavor of AIC to perform a fair evaluation of the bias-variance trade-off, the additive randomization method[64] was proposed to measure the prediction performance among competing models regardless of complex training and selection procedures. The core idea of additive randomization is to construct randomized responses y^+ and y^- as shown below,

$$y^+ = y + \alpha\omega = \mu + \varepsilon + \alpha\omega,\ y^- = y - \frac{1}{\alpha}\omega = \mu + \varepsilon - \frac{1}{\alpha}\omega,$$

where $\mu = \mu(x)$ is the underlying signal depending on the feature value x, and ω is additional additive noise from a standard normal distribution generated by the analyst, which is used in the construction of responses y^+ and y^-

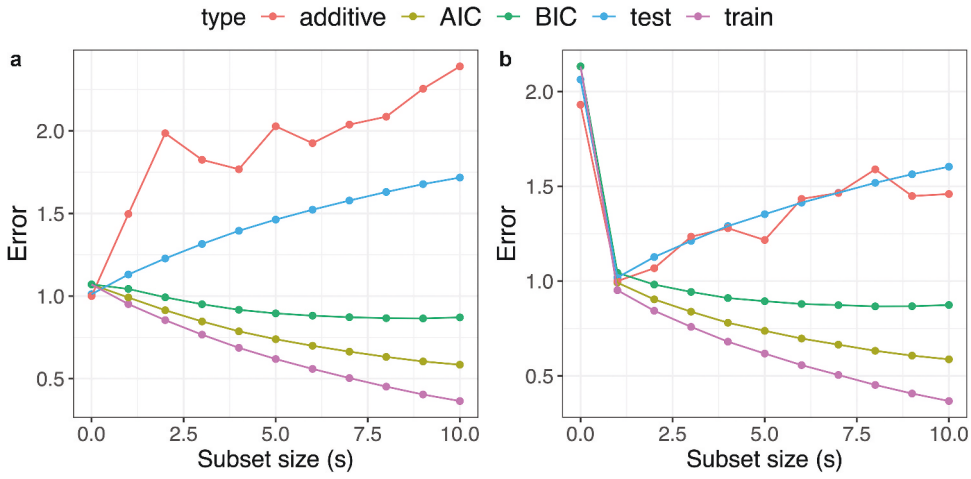

Figure 4. A comparison of error estimate from additive randomization (denoted as "additive"), AIC, BIC, the true prediction error (denoted as "test"), and training error (denoted as "train") in example 2. Panels a and B shows the results for $\beta = 0$ and $\beta = 1$, respectively as we vary the subset size (s). The prediction performance in this example is not able to be tracked by the training error, AIC or BIC.

after multiplying by the scalers α and $\frac{1}{\alpha}$ respectively. This particular form of construction guarantees the mutual independence between y^+ and y^-. Motivated by such an observation, the additive randomization method uses y^+ for all model training, while using y^- to measure the model predictive performance and select a final model with the smallest loss using y^-. It is worth noting that only the relative loss values here are meaningful, as the absolute loss calculated from using y^- does not resemble the true prediction error. In Figure 4, we have shifted the loss estimated with additive randomization such that the best model corresponds to a value of 1. Unlike AIC and BIC, the evaluations from using additive randomization successfully identified the model with the best prediction performance in both experiment settings. The drawbacks of the additive randomization are threefold: (1) it is currently only applicable to linear regression problems with Gaussian data, (2) it requires knowledge of the noise distribution, and (3) the randomization renders higher variability, as seen in Figure 4.

Cross-validation

Cross-validation (CV) is a widely used technique for assessing the performance of machine learning models and model selection to prevent overfitting.[65,66] Its popularity can be attributed to its conceptual simplicity, improvement of data utilization efficiency over sample-splitting, and wide applicability to all types of supervised machine learning models.

CV starts by dividing the training samples into K disjoint subsets (folds) randomly, usually of roughly equal size. For each fold K, CV evaluates the prediction accuracy of a training procedure by fitting it on the remaining $(K-1)$ folds. For example, we may fit a Lasso regularized linear regression with penalty λ to the remaining $(K-1)$ folds, with $\widehat{\beta^{-k}}$ being the estimated regression coefficient. Then, for each sample i, suppose it in fold $k(i)$, let (x_{n+1}, y_{n+1})be its associated feature value and response. We then calculate the squared error loss with $\mathrm{l}_i^{-k(i)} = \left(y_i - x_i^\top \widehat{\beta^{-k(i)}}\right)^2$ for all $k = 1, \ldots, K$. The CV error is defined by pooling together all cross-validated errors $\mathrm{l}_i^{-k(i)}$:

$$\widehat{\mathrm{Err}^{CV}} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{l}_i^{-k(i)}.$$

The CV error $\widehat{\mathrm{Err}^{CV}}$ is used as the evaluation for the trained model. By selecting a model with a small CV error from a set of competing models, we can select a relatively good model among them and mitigate overfitting. While AIC, BIC, and additive randomization estimate the prediction error with fixed feature values, CV aims to estimate the expected test error on new data as the both the features and response are independently generated together.[7,67]

A common choice for the number of folds is $K = 5$ or $K = 10$ as suggested by Kohavi et al.[68] However, practitioners may want to use a larger K, even leave-one-out cross-validation which treats each sample as its own fold in CV, when the sample size n is small. The fold number K indicates a different kind of bias-variance tradeoff: when K is large, the per-fold model in CV has a comparable training sample size as the original model, thus, CV induces less learning bias due to reduced sample size; however, when K is large, the model similarity across folds becomes increased, and $\widehat{\mathrm{Err}^{CV}}$ has increased variability for estimating the expected test error.[7]

Practitioners should exercise caution when applying CV. Firstly, the estimated CV error of the selected model can be biased and an underestimation of the actual test error due to selection, especially when the space of candidate models is large.[69] Guan and Tibshirani[70] proposed a randomized CV, which combines CV with the idea of additive randomization, enabling unbiased test error estimation after arbitrary selection in the CV procedure. Secondly, careless application of CV can fail even for model selection purposes. This often happens if there is unintended information leakage, in which data information from the validation set is not perfectly hidden from practitioners during the training and leaked to practitioners during model training. Here, we demonstrate that

information leakage invalidates CV under two most encountered settings in practice: (1) Explicit information leakage due to using response (e.g., feature filtering) outside the CV loop, and (2) Implicit information leakage due to unaccounted sample structure.

We first demonstrate that information leakage can happen when preprocessing steps occur before CV. Example 3 is a simple demonstration built upon section 7.10.2 from Hastie et al.[7] which highlights the danger of severe overfitting caused by analysis steps (e.g., feature filtering) outside of the CV loop.

Example 3: Suppose we have $n = 50$ samples with half from class 1 and the other half class 2. Suppose we have $p = 5000$ quantitative features, such as gene expression levels, that are independently generated from the standard Gaussian distribution and are independent of the class assignment. Consider the following invalid but typical CV analysis strategy: (1) perform feature filtering by selecting d features with the largest associations with the response, (2) construct a one-nearest neighbor classifier (1NN) using the selected features. This CV scheme is invalid because the feature filtering step has used all data before CV is carried out. Since the data contains pure noise, we expect the test error to be 50%. However, as we vary $d \in \{10, 50, 100, 500, 1000, 2000, 3000, 4000, 5000\}$, CV classification errors from step 2 are much lower than 50% when we adopt the feature filtering scheme and let d be smaller than 5000 (Figure 5a).

When applying CV, it is also crucial to understand sample structures that may contribute to potential information leakage sources and account for them. Example 4 illustrates the importance of accounting for the sample structure when performing CV. In this example, a completely random splitting of samples into different folds results in an invalid CV scheme.

Example 4: The lipidomic breast cancer data from the lab of Livia Schiavinato Eberlin at UT Austin consist of 806 features measured on 15,359 pixels in tissue images from 24 breast cancer patients, and this data is used by Guan et al.[37] The pixels are divided into two classes, the normal class and the cancer class, and we fit a regularized logistic regression model using each procedure. In this example, randomly splitting CV folds is an invalid scheme since pixels from the same patient reveal patient-specific information, and the resulting CV errors are over-optimistic for test error evaluation on samples from a new patient. Instead, a better CV scheme is to consider the stratified population structure and randomly split samples based on their patient ids. Indeed, we observe that the CV errors using random sample splitting to be much smaller than that using random patient-id splitting (Figure 5b), as a result of sample associations from the same patient.

Apart from such population stratification as demonstrated in Example 4, the random sample splitting scheme is also often invalid for time series data where the noises from nearby time points are often correlated. In this case, randomly sampling long blocks consisting of consecutive time points can usually alleviate this problem.[71,72]

These examples highlight the importance of the proper implementation of cross-validation. For practitioners who utilize CV in their research, it is beneficial to think about the following questions when designing the CV scheme instead of blindly using the default choice:

(1) Are the samples independent or are they correlated with each other? If samples are correlated with each

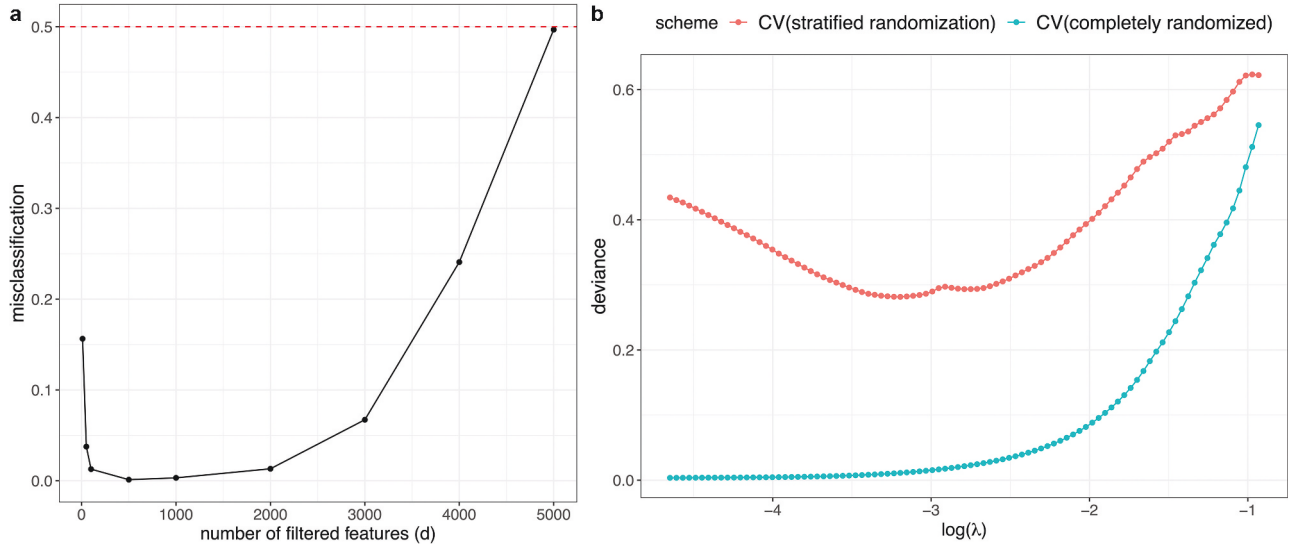

Figure 5. A) CV evaluation using 1NN with feature filtering as described in example 3. The x-axis shows the number of remaining features after filtering (d) with $d = 5000$ representing no-filtering, and the y-axis shows the misclassification error using CV. The actual test error should be 0.5 (red dashed line), which is much higher than the CV error in the presence of strong filtering (small d). B) CV evaluation with different randomization schemes using the lipidomic breast cancer dataset in example 4. The x-axis shows the logarithm of lasso penalty (λ), and the y-axis is the deviance loss. The achieved deviance when using CV with the stratified randomization grouped by patient id (in red) is considerably higher than that from using CV with the completely randomized scheme (in turquoise).

other, correlated samples should be grouped together when constructing the CV folds, such as samples from the same patient, samples from consecutive time, etc.

(2) Has the information from the response been used outside of the CV loop? Any analysis utilizing the response itself is not allowed outside the CV loop. For example, if we use feature filtering as a preprocessing step before feeding data into the machine learning model, this must also be done using only samples excluding the fold under evaluation.

To further guard against potential negligence of using CV, alternative model-free approaches such as permutation-based analysis can be employed.[73] Lu et al.[74] used a random permutation approach to assess the degree to which their modeling approach was susceptibly overfitting and raised the alert of over-training if the prediction accuracy is much better than random guess.

Finally, it is always beneficial to set aside a separate test dataset to mimic future independent observations. Similarly as in CV, the test dataset aims to contain independent samples from the remaining training data, for example, samples from the same patient are not shared by the test dataset and the training. Once set aside, the test dataset should not be used during the entire pipeline consisting of training, evaluation, model selection, and comparisons, and is only used for the final validation (Figure 1b). Evaluations on this independent test set can provide a reliable assessment of model performance, and huge discrepancy between CV error and the prediction error on the test set can help raise concerns about the adopted CV scheme.

Note that even if our cross-validation or other evaluations are not overfitted toward the noise, there is no guarantee that we will always achieve the same level of accuracy on a future dataset. For instance, the prediction errors on the test set could still be significantly smaller than the prediction error on a future dataset. This situation can arise if the future data introduce new sources of variation that were not present in the training data. For instance, measurements collected on COVID-19 infected patients in 2020 may not completely reflect the biology of new variants of interest, and a model trained on medical data collected from one hospital might not generalize well to another hospital.

Utilization of data diversity

The lack of reproducibility in generalizing discoveries to novel datasets is an issue that has been well recognized by the scientific community.[75] Here, we ask the question: Can we mitigate overfitting toward certain environments to improve the prediction model's generalizability to underrepresented or new environments?

In many instances, the prediction accuracy can increase significantly by increasing the volume of the training dataset. The power of data expansion goes beyond prediction improvement solely based on the volume boosting. Previously, it has been shown that a suitable meta-analysis of multi-cohorts' study can enhance the reproducibility of signature discovery via hypothesis testing.[76] Similarly, the diversity of data could play a critical role in improving the generalization of the model to under-represented environments. For example, Fourati et al.[5] and Hagan et al.[77] considered the problem of identifying common immune signatures predictive of antibody response among 13 different vaccinations, leading to signatures with increased generalization potential.

Apart from data expansion efforts, we may further consider adapting machine learning approaches to explicitly utilize data diversity. Standard model training criteria involve empirical risk minimization (ERM), which aims to achieve overall high accuracy on another independently and identically generated test cohort, e.g., the test cohort behaves similarly as the training data with some newly generated data noise. However, models from ERM may perform poorly on certain subgroups of samples due to heterogeneous subpopulation structure and inclusion of non-generalizable predictive relationships.[78–81] Alternatively, we can construct models that favor uniformly good performance across different subpopulations rather than focusing on overall accuracy, often referred to as distributional robust optimization (DRO).[81–86]

Below, we describe a standard DRO where we find the model parameter $\hat{\beta}$to minimize the loss ($\min_{\beta}$) in the worst subpopulation or group ($\max_{g\in G}$):

$$\hat{\beta} = \operatorname*{argmin}_{\beta} \max_{g\in G} \frac{1}{|\mathcal{J}_g|} \sum i \in \mathcal{J} \ell_{\beta}(x_i, y_i) + J(\beta),$$

where $g \in G$ represents the group assignment, $\mathcal{J}_g$ contains samples in group g, $|\mathcal{J}_g|$ is the size of $\mathcal{J}_g$, β is the model parameter, $J(\beta)$ is some regularization penalty on the parameter β, $\ell_{\beta}(x_i, y_i)$is the achieved loss at sample i with model parameter β. The above *minmax* formulation enables predictions with more uniform performance across groups. This DRO framework has been used in linear and other more flexible models such as neural networks and is effective for robust classification against distributional changes, including future changes in the test cohort, and helps identify invariant or generalizable predictive relationships across different populations. Example 5 demonstrates this desirable property of DRO over ERM in a simulated example.

Example 5: Consider a classification task where the response is $y \in \{0, 1\}$, the hidden group is $g \in \{0, 1\}$. Given the class label $y$and the group assignment g, the observed feature values x are from a 10-dimensional Gaussian distribution. The first dimension of $x$separates the two response classes in the same way and the second and third dimensions have opposite effects for samples from the two groups. We generate the samples with 90% of them from group 1 and 10% of them from group 2 and predict y using feature x. Figure 6a shows the boxplots of predicted probability of $y = 1$ separately for the two groups using an ERM model and a DRO model, which minimizes the worst group performance.[85] For samples from group 2, the ERM model achieves no better accuracy than random guessing, even though it achieves high classification accuracy in group 1. In contrast, the DRO model performs similarly well in

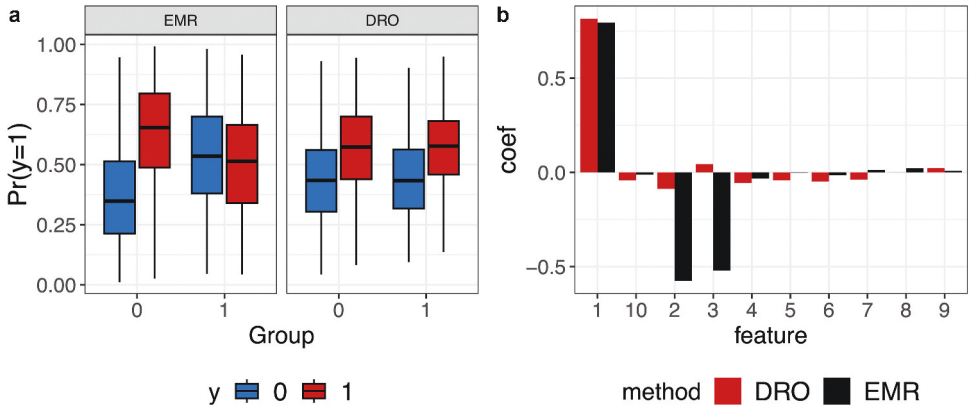

Figure 6. Illustrative example comparing ERM and DRO. Data are generated using example 5 with a signal-to-noise ratio of 1 for the prediction task. There are two groups, with 90% samples from group 1 and 10% from group 2. In the underlying model, the first feature separates the two classes invariantly, but the second and third features have opposite effects on samples from the two groups. A) Boxplots of predicted probability for $y = 1$ using the ERM and the DRO models separately for the two groups. B) Estimated model coefficients (y-axis) for both the DRO and ERM models.

both groups. The estimated model coefficients are shown in Figure 6b, with the ERM model depending heavily on all three features and the DRO model largely dependent on the first invariant feature.

The idea of explicitly utilizing population diversity for robust learning has not been widely exploited in medical studies, with only a few works adopting this modern learning scheme. For example, Yang et al.[87] considered robust COVID-19 risk predictions across sex and ethnicity, opening opportunities for novel analysis in this direction.

Discussion

When applying machine learning methods to predictive tasks in immunological and other biomedical applications, researchers need to be aware of both the strengths and limitations of these methods. Overfitting is one common issue encountered during the construction of prediction models in contemporary applications, which often deal with complex and high-dimensional data. Gaining a thorough understanding of the causes behind overfitting, the associated challenges, and cutting-edge strategies for diagnosis and mitigation is critical for appropriately applying various techniques.

Traditional perspectives on overfitting typically assume that training and test samples exhibit similar characteristics. In practice, however, test samples might follow different patterns than the training counterparts. In such situations, it becomes increasingly important to develop models that can capture invariant relationships, which can be more effectively achieved by leveraging the diversity of populations within the training data. Although discussions on this topic are currently limited, recognizing this crucial aspect of research could lead to significant advancements, particularly in light of the growing international efforts to conduct large-scale studies spanning multiple environments, as exemplified by various study centers, cohort populations, and responses of interest.[88–91]

Disclosure statement

SHK receives consulting fees from Peraton. All other authors declare that they have no competing interests.

Funding

The work was supported by the National Institute of Allergy and Infectious Diseases under the Award numbers U19AI089992 and U01AI167892, and the National Science Foundation under the Award Number DMS2310836.

References

1. Kononenko I. Machine learning for medical diagnosis: history, state of the art and perspective. Artif Intell Med. 2001;23(1):89–109. doi:10.1016/S0933-3657(01)00077-X.
2. Kourou K, Exarchos TP, Exarchos KP, Karamouzis MV, Fotiadis DI. Machine learning applications in cancer prognosis and prediction. Comput Struct Biotechnol J. 2015;13:8–17. doi:10.1016/j.csbj.2014.11.005.
3. Chaudhary K, Poirion OB, Lu L, Garmire LX. Deep learning–based multi-omics integration robustly predicts survival in liver CancerUsing deep learning to predict liver cancer prognosis. Clin Cancer Res. 2018;24(6):1248–59. doi:10.1158/1078-0432.CCR-17-0853.
4. Hagan T, Nakaya HI, Subramaniam S, Pulendran B. Systems vaccinology: enabling rational vaccine design with systems biological approaches. Vaccine. 2015;33(40):5294–301. doi:10.1016/j.vaccine.2015.03.072.
5. Fourati S, Tomalin LE, Mulè MP, Chawla DG, Gerritsen B, Rychkov D, Henrich E, Miller HE, Hagan T, Diray-Arce J, et al. Pan-vaccine analysis reveals innate immune endotypes predictive of antibody responses to vaccination. Nat Immunol. 2022;23(12):1777–87. doi:10.1038/s41590-022-01329-5.
6. Bishop CM, Nasrabadi NM. Pattern recognition and machine learning. New York: Springer; 2006.
7. Hastie T, Tibshirani R, Friedman JH, Friedman JH. The elements of statistical learning: data mining, inference, and prediction. Springer; 2009. doi:10.1007/978-0-387-84858-7.
8. Costa KD, Kleinstein SH, Hershberg U. Biomedical model fitting and error analysis. Sci Signal. 2011;4(192):tr9. doi:10.1126/scisignal.2001983.
9. Peng Y, Nagata MH. An empirical overview of nonlinearity and overfitting in machine learning using COVID-19 data. Chaos Soliton Fract. 2020;139:110055. doi:10.1016/j.chaos.2020.110055.

10. Chen T, Guestrin C. Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. San Francisco (CA). 2016. p. 785–94. doi:10.1145/2939672.2939785.
11. Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H, Chen K, Mitchell R, Cano I, Zhou T, et al. Xgboost: extreme gradient boosting. R package version 04-2. 2015;1:1–4.
12. Natarajan BK. Sparse approximate solutions to linear systems. Siam J Comput. 1995;24(2):227–34. doi:10.1137/S0097539792240406.
13. Draper NR, Smith H. Applied regression analysis. John Wiley & Sons; 1998. doi:10.1002/9781118625590.
14. Hoerl AE, Kennard RW. Ridge regression: biased estimation for nonorthogonal problems. Technometrics. 1970;12(1):55–67. doi:10.1080/00401706.1970.10488634.
15. Tibshirani R. Regression shrinkage and selection via the lasso. J R Stat Soc Series B Stat Methodol. 1996;58(1):267–88. doi:10.1111/j.2517-6161.1996.tb02080.x.
16. Zou H, Hastie T. Regularization and variable selection via the elastic net. J R Stat Soc Series B Stat Methodol. 2005;67(2):301–20. doi:10.1111/j.1467-9868.2005.00503.x.
17. Furman D, Davis MM. New approaches to understanding the immune response to vaccination and infection. Vaccine. 2015;33(40):5271–81. doi:10.1016/j.vaccine.2015.06.117.
18. Yuan M, Lin Y. Model selection and estimation in regression with grouped variables. J R Stat Soc. 2006;68:49–67. doi:10.1111/j.1467-9868.2005.00532.x.
19. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors. 2012. doi:10.48550/arXiv.1207.0580.
20. Wager S, Wang S, Liang PS. Dropout training as adaptive regularization. In: Proceedings of the 26th International Conference on Neural Information Processing Systems. 2013. p. 351–59. doi:10.5555/2999611.2999651.
21. Prechelt L. Early stopping–but when? In: Neural networks: tricks of the trade. 2nd ed. 2012. p. 53–67. doi:10.1007/978-3-642-35289-8_5.
22. Krogh A, Hertz J. A simple weight decay can improve generalization. In: Proceedings of the 4th International Conference on Neural Information Processing SystemsDecember. 1991. p. 950–57. doi:10.5555/2986916.2987033.
23. Bishop CM. Training with noise is equivalent to Tikhonov regularization. Neural Comput. 1995;7(1):108–16. doi:10.1162/neco.1995.7.1.108.
24. Conesa A, Madrigal P, Tarazona S, Gomez-Cabrero D, Cervera A, McPherson A, Szcześniak MW, Gaffney DJ, Elo LL, Zhang X, et al. A survey of best practices for RNA-seq data analysis. Genome Biol. 2016;17(1):1–19. doi:10.1186/s13059-016-0881-8.
25. Huang S, Chaudhary K, Garmire LX. More is better: recent progress in multi-omics data integration methods. Front Genet. 2017;8:84. doi:10.3389/fgene.2017.00084.
26. Hasin Y, Seldin M, Lusis A. Multi-omics approaches to disease. Genome Biol. 2017;18(1):1–15. doi:10.1186/s13059-017-1215-1.
27. Li S, Rouphael N, Duraisingham S, Romero-Steiner S, Presnell S, Davis C, Schmidt DS, Johnson SE, Milton A, Rajam G, et al. Molecular signatures of antibody responses derived from a systems biological study of 5 human vaccines. Nat Immunol. 2014;15:195–204. doi:10.1038/ni.2789.
28. Li S, Sullivan NL, Rouphael N, Yu T, Banton S, Maddur MS, McCausland M, Chiu C, Canniff J, Dubey S, et al. Metabolic phenotypes of response to vaccination in humans. Cell. 2017;169:862–77. doi:10.1016/j.cell.2017.04.026.
29. Pearson KL. LIII. on lines and planes of closest fit to systems of points in space. London, Edinburgh Dublin Phil Mag J Sci. 1901;2(11):559–72. doi:10.1080/14786440109462720.
30. Golub GH, Reinsch C. Singular value decomposition and least squares solutions. Linear Algebra. 1971;2:134–51.
31. Lee DD, Seung HS. Learning the parts of objects by non-negative matrix factorization. Nature. 1999;401(6755):788–91. doi:10.1038/44565.
32. Zou H, Hastie T, Tibshirani R. Sparse principal component analysis. J Comput Graph Stat. 2006;15(2):265–86. doi:10.1198/106186006X113430.
33. Tan KM, London P, Mohan K, Lee S-I, Fazel M, Witten D. Learning graphical models with hubs. J Mach Learn Res. 2014;15:3297–331. doi:10.5555/2627435.2697070.
34. Guan L, Fan Z, Tibshirani R. Supervised learning via the "Hubnet" procedure. Stat Sin. 2018;28:1225. doi:10.5705/ss.202016.0482.
35. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Sci. 2006;313(5786):504–7. doi:10.1126/science.1127647.
36. Zhang B, Horvath S. A general framework for weighted gene co-expression network analysis. Stat Appl Genet Mol Biol. 2005;4(1):4. doi:10.2202/1544-6115.1128.
37. Van Dam S, Vosa U, van der Graaf A, Franke L, de Magalhaes JP. Gene co-expression analysis for functional classification and gene–disease predictions. Brief Bioinform. 2018;19:575–92. doi:10.1093/bib/bbw139.
38. Langfelder P, Horvath S. WGCNA: an R package for weighted correlation network analysis. BMC Bioinform. 2008;9(1):1–13. doi:10.1186/1471-2105-9-559.
39. Zoppi J, Guillaume J-F, Neunlist M, Chaffron S. MiBiOmics: an interactive web application for multi-omics data exploration and integration. BMC Bioinform. 2021;22(1):1–14. doi:10.1186/s12859-020-03921-8.
40. Lock EF, Hoadley KA, Marron JS, Nobel AB. Joint and individual variation explained (JIVE) for integrated analysis of multiple data types. Ann Appl Stat. 2013;7(1):523. doi:10.1214/12-AOAS597.
41. Zhang S, Liu C-C, Li W, Shen H, Laird PW, Zhou XJ. Discovery of multi-dimensional modules by integrative analysis of cancer genomic data. Nucleic Acids Res. 2012;40(19):9379–91. doi:10.1093/nar/gks725.
42. Yang Z, Michailidis G. A non-negative matrix factorization method for detecting modules in heterogeneous omics multi-modal data. Bioinformatics. 2016;32(1):1–8. doi:10.1093/bioinformatics/btv544.
43. Chalise P, Fridley BL, Peddada SD. Integrative clustering of multi-level 'omic data based on non-negative matrix factorization algorithm. PLoS One. 2017;12(5):e0176278. doi:10.1371/journal.pone.0176278.
44. Shen R, Mo Q, Schultz N, Seshan VE, Olshen AB, Huse J, Ladanyi M, Sander C, Brusic V. Integrative subtype discovery in glioblastoma using iCluster. PLoS One. 2012;7(4):e35236. doi:10.1371/journal.pone.0035236.
45. Mo Q, Shen R, Guo C, Vannucci M, Chan KS, Hilsenbeck SG. A fully Bayesian latent variable model for integrative clustering analysis of multi-type omics data. Biostatistics. 2018;19(1):71–86. doi:10.1093/biostatistics/kxx017.
46. Argelaguet R, Velten B, Arnol D, Dietrich S, Zenz T, Marioni JC, Buettner F, Huber W, Stegle O. Multi-omics factor analysis-a framework for unsupervised integration of multi-omics data sets. Mol Syst Biol. 2018;14:e8124. doi:10.15252/msb.20178124.
47. Argelaguet R, Arnol D, Bredikhin D, Deloro Y, Velten B, Marioni JC, Stegle O. MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data. Genome Biol. 2020;21(1):111. doi:10.1186/s13059-020-02015-1.
48. Kettenring JR. Canonical analysis of several sets of variables. Biometrika. 1971;58(3):433–51. doi:10.1093/biomet/58.3.433.
49. Meng C, Kuster B, Culhane AC, Gholami AM. A multivariate approach to the integration of multi-omics datasets. BMC Bioinform. 2014;15(1):162. doi:10.1186/1471-2105-15-162.
50. Min EJ, Long Q. Sparse multiple co-inertia analysis with application to integrative analysis of multi-omics data. BMC Bioinform. 2020;21(1):1–12. doi:10.1186/s12859-020-3455-4.
51. Tenenhaus A, Tenenhaus M. Regularized generalized Canonical correlation analysis. Psychometrika. 2011;76(2):257–84. doi:10.1007/s11336-011-9206-8.
52. Tenenhaus M, Tenenhaus A, Groenen PJ. Regularized generalized canonical correlation analysis: a framework for sequential

multiblock component methods. Psychometrika. 2017;82 (3):737–77. doi:10.1007/s11336-017-9573-x.
53. Guan L l1-norm constrained multi-block sparse canonical correlation analysis via proximal gradient descent. 2022. doi:10.48550/arXiv.2201.05289.
54. Bair E, Hastie T, Paul D, Tibshirani R. Prediction by supervised principal components. J Am Stat Assoc. 2006;101(473):119–37. doi:10.1198/016214505000000628.
55. Fan J, Ke ZT, Liu H, Xia L. QUADRO: a supervised dimension reduction method via Rayleigh quotient optimization. Ann Stat. 2015;43:1498. doi:10.1214/14-AOS1307.
56. Zhang D, Zhou Z-H, Chen S Semi-supervised dimensionality reduction. In: Proceedings of the 2007 SIAM International Conference on Data Mining. SIAM. Minneapolis (MN). 2007. p. 629–34.
57. Zhang M, Li H, Su S. High dimensional Bayesian optimization via supervised dimension reduction. 2019. doi:10.48550/arXiv.1907.08953.
58. Singh A, Shannon CP, Gautier B, Rohart F, Vacher M, Tebbutt SJ, Cao K-AL. DIABLO: an integrative approach for identifying key molecular drivers from multi-omics assays. Bioinformatics. 2019;35:3055–62. doi:10.1093/bioinformatics/bty1054.
59. Gygi JP, Konstorum A, Pawar S, Kleinstein SH, Guan L. SPEAR: a sparse supervised Bayesian factor model for multi-omic integration. 2023. doi:10.1101/2023.01.25.525545.
60. Akaike H. Information Theory and an Extension of the Maximum Likelihood Principle. In: Selected Papers of Hirotugu Akaike. Springer Series in Statistics. New York (NY): Springer; 1998. p. 199–213.
61. Mallows CL. Some comments on Cp. Technometrics. 2000;42 (1):87–94. doi:10.1080/00401706.2000.10485984.
62. Schwarz G. Estimating the dimension of a model. Ann Stat. 1978;6 (2):461–4. doi:10.1214/aos/1176344136.
63. Wasserman L. Bayesian model selection and model averaging. J Math Psychol. 2000;44(1):92–107. doi:10.1006/jmps.1999.1278.
64. Tian X. Prediction error after model search. Ann Stat. 2020;48 (2):763–84. doi:10.1214/19-AOS1818.
65. Stone M. Cross-validatory choice and assessment of statistical predictions. J R Stat Soc. 1974;36(2):111–33. doi:10.1111/j.2517-6161.1974.tb00994.x.
66. Geisser S. The predictive sample reuse method with applications. J Am Stat Assoc. 1975;70(350):320–8. doi:10.1080/01621459.1975.10479865.
67. Bates S, Hastie T, Tibshirani R. Cross-validation: what does it estimate and how well does it do it? J Am Stat Assoc. 2021:1–12. doi:10.1080/01621459.2023.2197686.
68. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: IJCAI. Montreal, Canada; 1995. p. 1137–45.
69. Rao RB, Fung G, Rosales R On the dangers of cross-validation. An experimental evaluation. In: Proceedings of the 2008 SIAM international conference on data mining. SIAM. Atlanta, GA. 2008. p. 588–96.
70. Guan L, Tibshirani R. Post model-fitting exploration via a "next-door" analysis. Can J Stat. 2020;48:447–70. doi:10.1002/cjs.11542.
71. Varoquaux G, Raamana PR, Engemann DA, Hoyos-Idrobo A, Schwartz Y, Thirion B. Assessing and tuning brain decoders: cross-validation, caveats, and guidelines. NeuroImage. 2017;145:166–79. doi:10.1016/j.neuroimage.2016.10.038.
72. Bergmeir C, Hyndman RJ, Koo B. A note on the validity of cross-validation for evaluating autoregressive time series prediction. Comput Stat Data Anal. 2018;120:70–83. doi:10.1016/j.csda.2017.11.003.
73. Welch WJ. Construction of permutation tests. J Am Stat Assoc. 1990;85(411):693–8. doi:10.1080/01621459.1990.10474929.
74. Lu P, Guerin DJ, Lin S, Chaudhury S, Ackerman ME, Bolton DL, Wallqvist A. Immunoprofiling correlates of protection against SHIV infection in adjuvanted HIV-1 pox-protein vaccinated Rhesus Macaques. Front Immunol. 2021;12:625030. doi:10.3389/fimmu.2021.625030.
75. Baker M. 1,500 scientists lift the lid on reproducibility. Nature. 2016;533(7604):452–4. doi:10.1038/533452a.
76. Haynes WA, Vallania F, Liu C, Bongen E, Tomczak A, Andres-Terrè M, Lofgren S, Tam A, Deisseroth CA, Li MD, et al. Empowering multi-cohort gene expression analysis to increase reproducibility. Pac Symp Biocomput. 2017;22:144–53.
77. Hagan T, Gerritsen B, Tomalin LE, Fourati S, Mulè MP, Chawla DG, Rychkov D, Henrich E, Miller HER, Diray-Arce J, et al. Transcriptional atlas of the human immune response to 13 vaccines reveals a common predictor of vaccine-induced antibody responses. Nat Immunol. 2022;23(12):1788–98. doi:10.1038/s41590-022-01328-6.
78. Peters J, Bühlmann P, Meinshausen N. Causal inference by using invariant prediction: identification and confidence intervals. J R Stat Soc. 2016;78:947–1012. doi:10.1111/rssb.12167.
79. Arjovsky M, Bottou L, Gulrajani I, Lopez-Paz D. Invariant risk minimization. 2019. doi:10.48550/arXiv.1907.02893.
80. Meinshausen N, Bühlmann P. Maximin effects in inhomogeneous large-scale data. Ann Stat. 2015;43(4):1801–30. doi:10.1214/15-AOS1325.
81. Bühlmann P, Meinshausen N Magging: maximin aggregation for inhomogeneous large-scale data. In: Proceedings of the IEEE. Vol. 104. Boston (MA). 2015. p. 126–35.
82. Wen J, Yu C-N, Greiner R Robust learning under uncertain test distributions: relating covariate shift to model misspecification. In: International Conference on Machine Learning. Beijing (CN): PMLR. 2014. p. 631–9.
83. Yang F, Wang Z, Heinze-Deml C. Invariance-inducing regularization using worst-case transformations suffices to boost accuracy and spatial robustness. Adv Neural Inf Proc Sys. 2019;32 (10):5339–51. doi:10.1007/s00521-020-04704-1.
84. Sagawa S, Koh PW, Hashimoto TB, Liang P. Distributionally robust neural networks for group shifts: on the importance of regularization for worst-case generalization. 2019. doi:10.48550/arXiv.1911.08731.
85. Liu EZ, Haghgoo B, Chen AS, Raghunathan A, Koh PW, Sagawa S, Liang P, Finn C. Just train twice: improving group robustness without training group information. In: Proceedings of the 38th International Conference on Machine Learning. online. 2021. p. 6781–92.
86. Zhang J, Menon A, Veit A, Bhojanapalli S, Kumar S, Sra S. Coping with label shift via distributionally robust optimisation. arXiv Preprint arXiv: 201012230. 2020. doi:10.48550/arXiv.2010.12230.
87. Yang J, Soltan AA, Yang Y, Clifton DA. Algorithmic fairness and bias mitigation for clinical machine learning: insights from rapid COVID-19 diagnosis by adversarial learning. medRxiv. 2022. doi:10.1101/2022.01.13.22268948.
88. Diray-Arce J, Miller HE, Henrich E, Gerritsen B, Mulè MP, Fourati S, Gygi J, Hagan T, Tomalin L, Rychkov D, et al. The immune signatures data resource, a compendium of systems vaccinology datasets. Sci Data. 2022;9(1):635. doi:10.1038/s41597-022-01714-7.
89. Maecker HT, McCoy JP, Nussenblatt R. Standardizing immunophenotyping for the human immunology project. Nat Rev Immunol. 2012;12(3):191–200. doi:10.1038/nri3158.
90. Brusic V, Gottardo R, Kleinstein SH, Davis MM. Alkis 9 H steering committee DMM 5 HDA 8 QH 9 PAK 10 PGA 11 PB 12 REL 1 SKD 13 T. Computational resources for high-dimensional immune analysis from the human immunology project consortium. Nat Biotechnol. 2014;32(2):146–8. doi:10.1038/nbt.2777.
91. Rouphael N, Maecker H, Montgomery RR, Diray-Arce J, Kleinstein SH, Altman MC, Bosinger SE, Eckalbar W, Guan L, Hough CL, et al. Immunophenotyping assessment in a COVID-19 cohort (IMPACC): a prospective longitudinal study. Sci Immunol. 2021;6:eabf3733. doi:10.1126/sciimmunol.abf3733.