



Privacy preserving validation for multiomic prediction models

Talal Ahmed , Mark A. Carty, Stephane Wenric, Jonathan R. Dry , Ameen A. Salahudeen, Aly A. Khan, Eric Lefkofsky, Martin C. Stumpe and Raphael Pelossof

Corresponding author: Raphael Pelossof, Data Science, Tempus Labs Inc, Chicago, IL 60654, USA. Tel./Fax: (833) 514-4187; E-mail: rafi.pelossof@tempus.com

Abstract

Reproducibility of results obtained using ribonucleic acid (RNA) data across labs remains a major hurdle in cancer research. Often, molecular predictors trained on one dataset cannot be applied to another due to differences in RNA library preparation and quantification, which inhibits the validation of predictors across labs. While current RNA correction algorithms reduce these differences, they require simultaneous access to patient-level data from all datasets, which necessitates the sharing of training data for predictors when sharing predictors. Here, we describe SpinAdapt, an unsupervised RNA correction algorithm that enables the transfer of molecular models without requiring access to patient-level data. It computes data corrections only via aggregate statistics of each dataset, thereby maintaining patient data privacy. Despite an inherent trade-off between privacy and performance, SpinAdapt outperforms current correction methods, like Seurat and ComBat, on publicly available cancer studies, including TCGA and ICGC. Furthermore, SpinAdapt can correct new samples, thereby enabling unbiased evaluation on validation cohorts. We expect this novel correction paradigm to enhance research reproducibility and to preserve patient privacy.

Keywords: privacy, translational research, reproducibility, model validation, machine learning, transcriptomics

Introduction

The advent of high-throughput gene expression profiling has powered the development of sophisticated molecular models to capture complex biological patterns. To ensure the generalization of molecular patterns across independent studies, molecular predictors require validation across platforms and laboratories. However, the transfer of predictors across laboratories still remains a technical obstacle. Batch-specific effects that dominate the biological signal exist between different technologies, laboratories and even library preparation protocols within the same laboratory [1]. Furthermore, often these interinstitutional datasets are siloed due to human subject privacy concerns. There is an unmet need for a technology that enables the transfer of molecular predictors across labs in a privacy-preserving manner such that sample-level patient data are not transferred.

Correction of batch-specific biases in ribonucleic acid (RNA) expression datasets has been an active field of research in the past two decades. Numerous methods are

proposed to correct batch effects, and these mostly fall into two categories: batch integration and batch correction. Batch integration entails joint embedding of batch-biased expression data in a shared embedding space where batch variations are minimized [2, 3]. Batch correction removes batch biases in the gene expression space, harmonizing batch-biased dataset(s) to a reference dataset. For batch correction, we refer to the reference dataset as the target and the batch-biased dataset as the source.

Machine learning models in clinical research are often developed using RNA expression data [4] and need to be validated on external datasets with acquisition biases. We therefore consider the model immutable and avoid further training or fine tuning of the existing model to new datasets. Batch correction methods enable validation of existing immutable predictive models on external datasets that are corrected to the reference set. This is achieved by setting the training dataset as the reference dataset. Thereby, batch correction avoids the necessity to

Talal Ahmed, Tempus Labs Inc, NYC, NY 1016, USA.

Mark A. Carty, Tempus Labs Inc, NYC, NY 10169, USA.

Stephane Wenric, Tempus Labs Inc, NYC, NY 10169, USA

Jonathan R. Dry, Tempus Labs Inc, Chicago, IL 60654, USA

Ameen Abdulla Salahudeen, Tempus Labs Inc, Chicago, IL 60654, USA

Aly A. Khan, Tempus Labs Inc, Chicago, IL 60654, USA

Eric Lefkofsky, Tempus Labs Inc, Chicago, IL 60654, USA

Martin C. Stumpe, Tempus Labs Inc, Redwood City, CA 94065, USA

Raphael Pelossof, Tempus Labs Inc, NYC, NY 10169, USA

Received: October 26, 2021. Revised: February 17, 2022. Accepted: March 5, 2022

© The Author(s) 2022. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial License (<https://creativecommons.org/licenses/by-nc/4.0/>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact journals.permissions@oup.com

retrain the model for every validation dataset. Furthermore, data-leakage is a cause of overfitting in machine learning, and hence, we propose to extend the definition of batch correctors to include the capability to apply the learned correction on unseen data (transform).

In contrast, batch integration methods have no reference set by definition. Therefore, the paradigm of validating immutable predictors cannot be executed using batch integration. Some examples of batch integration include methods based on gene-wise linear models like Limma [5], mutually nearest neighbors (MNNs) like MNN Correct [6] and Scanorama [7], mutually nearest clusters like ScGen [8], pseudoreplicates like ScMerge [9] and multibatch clusters like Harmony [10]. Batch correction methods can correct a source library to a target reference library, like Combat [11], Seurat3 [12] and the proposed SpinAdapt algorithm, and thus can be used for transferring molecular predictors across expression datasets.

Prior integration and correction methods require full sample-level access for integration or correction of datasets. Therefore, the transfer of molecular predictors between laboratories necessitates the transfer of patient-level training data for the molecular predictor. This data access requirement can inhibit the transfer of models between laboratories, given the transfer of data may not be possible due to data ownership, GDPR or similar regulations. To overcome these challenges, we present SpinAdapt, a method that enables the transfer of molecular predictors between laboratories without disclosure of the sample-level training data for the predictor, thereby allowing laboratories to maintain ownership of the training data and protect patient privacy. Instead of sharing sample-level data, privacy-preserving aggregate statistics of the training data are shared along with the molecular predictor. Our approach is based on the concept of matrix masks from privacy literature [13], where the sample-level data and matrix mask are kept private, while the output of the matrix mask is shared publicly.

This study demonstrates the transfer and validation of diagnostic and prognostic models across transcriptomic datasets, using SpinAdapt, while drawing comparisons with other batch correction methods. The common task of integration (homogenization) across multiple transcriptomic datasets is also evaluated for multiple cancer types, comparing various integration methods. In our experiments, SpinAdapt outperforms other batch correction methods in the majority of these diagnostic, prognostic and integration tasks without requiring direct access to sample-level data. Therefore, SpinAdapt may also be preferable for sharing molecular predictors across labs where the training dataset can be shared and data privacy is not an issue.

Results

Algorithm overview

We aimed to develop a framework for transfer and validation of molecular predictors across platforms,

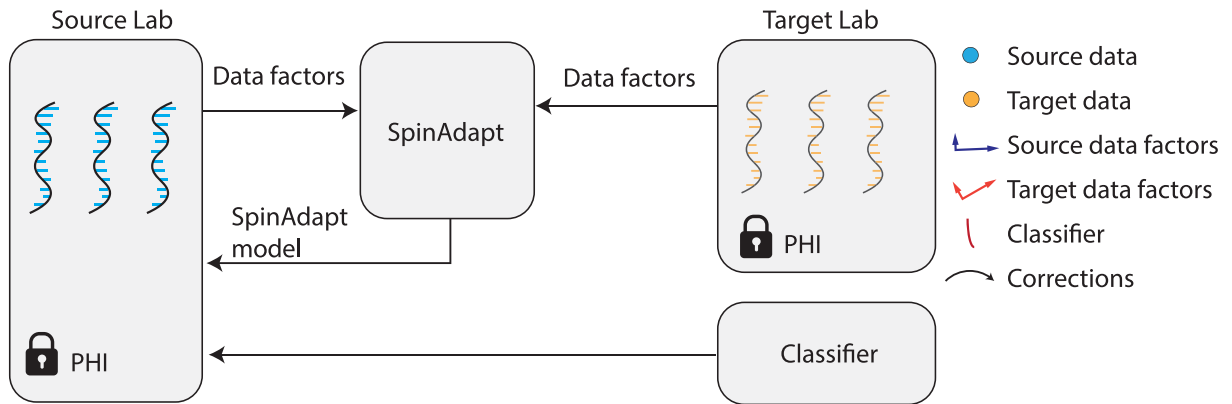
laboratories and varying technical conditions. Furthermore, we aimed to remove the requirement of sharing training data in order to evaluate and validate predictors across labs. To this end, we developed SpinAdapt that enables the validation of predictors while preserving data privacy (Figure 1A). Data factors, which are aggregate statistics of each dataset, neither convey protected health information nor allow reconstruction of sample-level data (Supplementary Note), and thus can be shared externally. SpinAdapt learns corrections between data factors of each dataset, which is followed by application of corrections on the biased expression dataset (source). Note that our framework also enables the correction of new data samples, which has important implications as discussed later.

SpinAdapt corrections are learned using a regularized linear transformation between the data factors of source and target, which comprise the principal component analysis (PCA) basis, gene-wise means and gene-wise SDs of source and target, respectively (Methods, Figure 1B). The linear transformation is the solution of a non-convex objective function, which is optimized using an efficient computational approach based on projected gradient descent. Once the transformation has been learned, it can be applied on the source dataset for correction, followed by the application of the target-trained predictor on the corrected source dataset (Figure 1C). Therefore, the learning module requires access only to the data factors of each dataset to learn the correction for the source dataset.

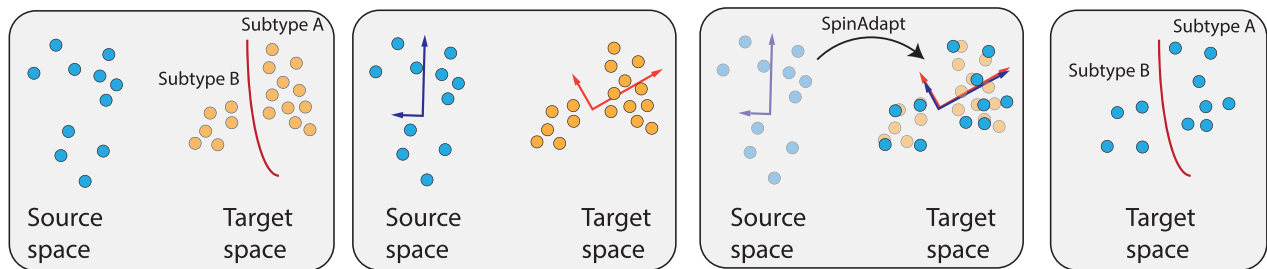
Since the learning step (Figure 1B) is separated from the transform step (Figure 1C), the transform step can be applied to new prospective data that were held out in the learning step. The ability to transform data that are held out from model training is deemed necessary for machine learning algorithms to avoid overfitting, which is ensured in this case by keeping the test data for the predictor separate from training of any data model. Including predictor test data in batch correction model training can lead to information leakage and overly optimistic performance metrics. To avoid this, the evaluation data in transform step (transform) are kept independent of the train data in the learning step (fit). This fit-transform paradigm is extended by SpinAdapt to transcriptomic datasets.

The training step of the algorithm is based on the idea of aligning the PCA basis of each dataset. To demonstrate the concept, we apply SpinAdapt on a transcriptomic dataset of paired patients, employing TCGA-BRCA cohort consisting of 481 breast cancer patients, where RNA was profiled both with RNA-seq and microarray. We assigned the RNA-seq library as target and the microarray as source. The application of SpinAdapt aligns the PCA basis of source to target, which resulted in the alignment of embeddings as well as gene expression profiles across the paired datasets (Figure 2A). The paired patients were composed of four cancer subtypes: Luminal A (LumA), Luminal B (LumB), Her2 and Basal. When the corrected source dataset is

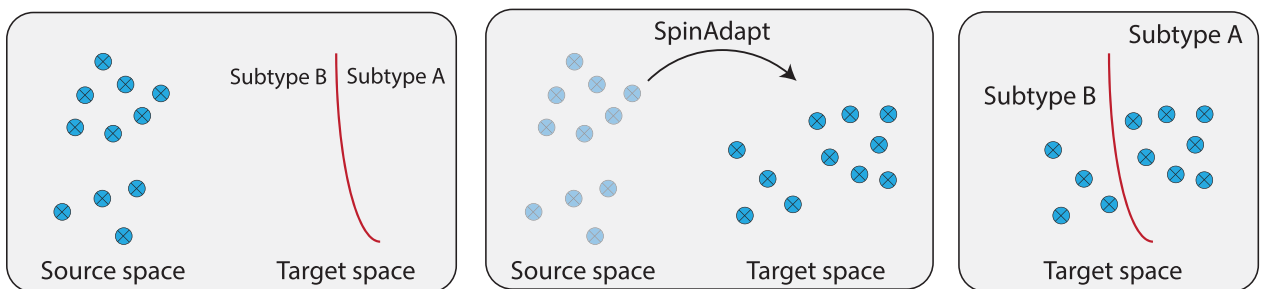
A) Concept



B) Fit adapt



C) Transform



◆ New prospective data

Figure 1. Privacy-preserving transfer of molecular models between a target lab and a source lab. (A) A target dataset with a trained classifier and protected RNA data provides its privacy-preserving RNA factors and a molecular classifier to SpinAdapt. A source dataset used for validation provides its own privacy-preserving RNA factors to SpinAdapt. Given the factors, SpinAdapt returns a correction model to source, where the source data is corrected. Target classifier without modification can then be validated on source-corrected. (B) Source and target factors are calculated as the principal components of RNA data. Next, SpinAdapt learns a correction model from source to target eigenvectors (factors). (C) Evaluation of the SpinAdapt correction model on the held-out prospective source data. Finally, the target-trained classifier is applied to the corrected source data.

visualized with the target library in a two-dimensional space with Uniform Manifold Approximation and Projection (UMAP), we observe each of the four subtypes to be harmonized across the two libraries as compared with before correction (Figure 2B). The subtype-wise homogenization is achieved without the use of subtype labels in the training step, demonstrating that alignment of basis in the PCA space achieves efficient removal of technical biases in the gene expression space. We also conduct a simulation experiment to explore SpinAdapt for removal of batch effects, where batch effect is simulated between synthetic datasets (source and target)

and the source dataset is corrected using SpinAdapt (see results in Supplementary Figure 6 available online at <https://academic.oup.com/bib>).

Transfer of diagnostic predictors

We demonstrate the transfer of multiple distinct tumor subtype classifiers on four pairs of publicly available cancer datasets (bladder, breast, colorectal and pancreatic), covering 4076 samples and three technological platforms (RNA-seq, Affymetrix U133plus2 Microarray and Human Exon 1.0 ST Microarray) (Supplementary Table 1

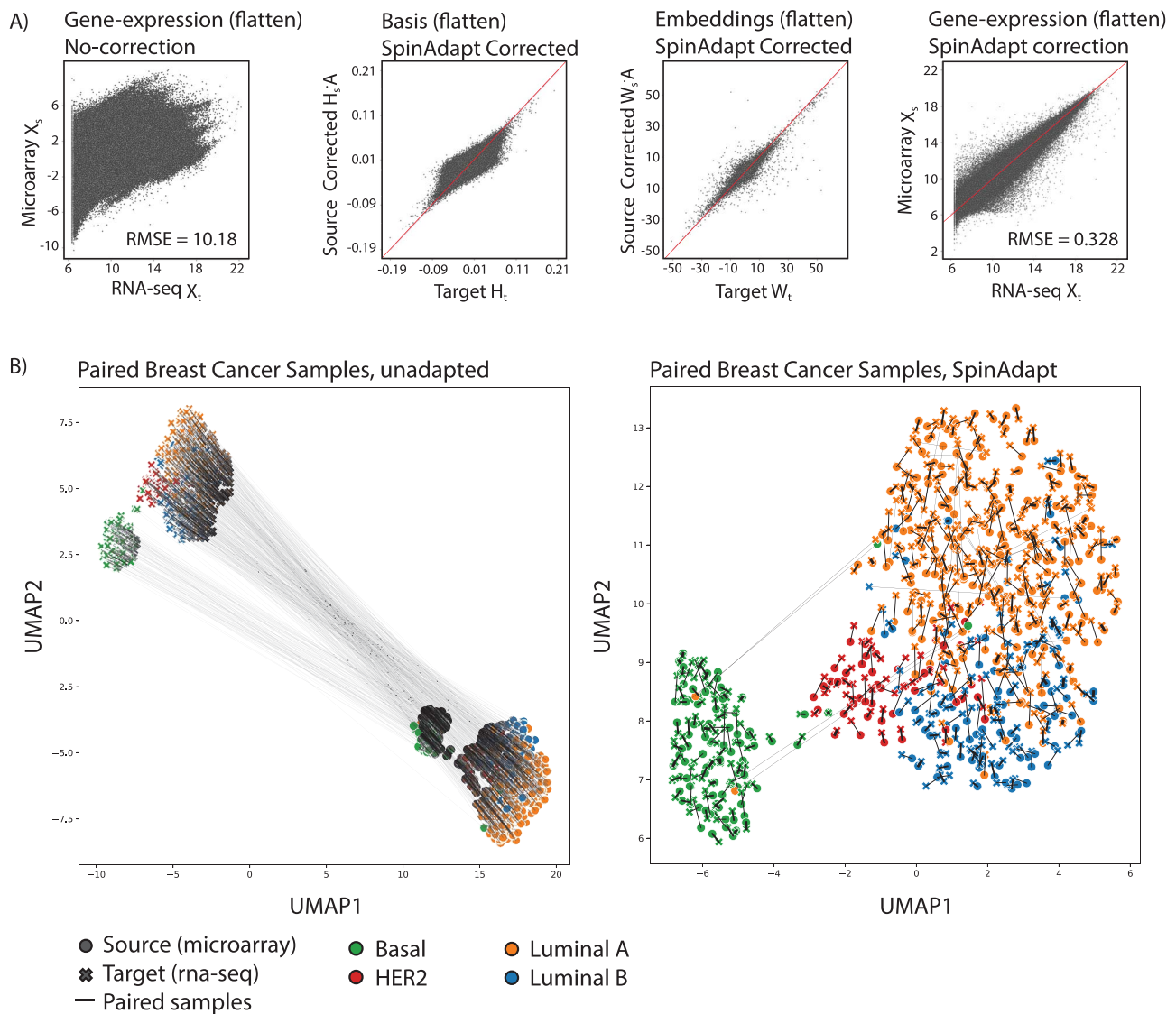


Figure 2. Batch correction performance on paired 481 TCGA-BRCA patients profiled with RNA-seq (target) and microarray (source). **(A)** Scatter plots of gene expression values in target with uncorrected source expression, target basis with corrected source basis, target embeddings with corrected source embeddings and finally gene expression values in target with corrected source expression, where all corrections are performed using SpinAdapt. **(B)** Source expression dataset before and after correction, plotted with the reference target library (visualized in 2D with UMAP embeddings). The samples are labeled by cancer subtype and the paired samples are connected with a solid line. Left panel shows cancer subtypes before correction. Right panel shows subtype homogeneity and matching of the paired samples across datasets after SpinAdapt correction.

available online at <https://academic.oup.com/bib>). Cumulatively, we validated the transfer of 17 tumor subtypes across the four experiments, drawing comparisons of SpinAdapt with other batch correction methods like ComBat and Seurat. For each dataset pair and tumor subtype, we trained a one-versus-rest tumor subtype classifier on the target dataset. The hyperparameters for each subtype classifier were chosen in a cross-validation experiment on the target dataset, while the source dataset was held out from classifier training (Methods).

A common approach for validating target-trained classifiers across datasets is to correct all source data to target and then evaluate the classifier on corrected source dataset. However, such an approach requires the batch correction (adaptation) model to train on the source dataset, which is also the test set for the classifier. Training the adaptation model on the test set may lead to

information leakage, which may lead to overly optimistic performance results. The risk of overfitting on the test set has been sparingly discussed in the batch correction literature. We propose a validation framework that holds out a subset of the source data from training of both the adaptation model and subtype classifier. Since the source subset is completely held out from training of both models, it can be used as the test set for the subtype classifier without the risk of information leakage.

Specifically, the validation framework proceeds by creating two mutually exclusive sets from source (Source A and Source B). We first fit the adaptation model between Source A and target, transform Source B using the adaptation model, followed by prediction on transformed Source B using the target-trained classifier (Supplementary Figure 1A available online at <https://academic.oup.com/bib>). Similarly,

we fit the adaptation model between Source B and target, followed by transformation and prediction on Source A (Supplementary Figure 1B available online at <https://academic.oup.com/bib>). Finally, we concatenate the held-out predictions on Source A and Source B, followed by performance evaluation using F-1 score (Methods). SpinAdapt's performance was evaluated using this framework, so the test set is always held out from training modules. Existing correction methods, like ComBat and Seurat3, have currently not implemented a transformation method for out-of-sample data that is held out from their training. Therefore, these methods had to be trained on the classifier test set in the aforementioned framework (Methods).

We repeated the above experimental framework 30 times and reported the mean F-1 score for each tumor subtype. SpinAdapt significantly outperformed Seurat3 on 7 out of the 17 tumor subtypes, including pancreatic subtypes: progenitor, aberrantly differentiated endocrine exocrine (ADEX), immunogenic, colorectal subtypes: CMS3, breast subtypes: LumA, bladder subtypes: Squamous and Stroma. SpinAdapt also significantly outperformed ComBat on 11 out of the 17 subtypes, including pancreatic subtypes: progenitor, ADEX, immunogenic, colorectal subtypes: CMS1, CMS2, CMS3, breast subtypes: Her2, bladder subtypes: luminal papillary (LumP), luminal unstable (LumU), luminal nonspecified (LumNS) and Stroma. SpinAdapt was not significantly outperformed by either Seurat or ComBat for any subtype (Figure 3A–D, Supplementary Figure 2, Supplementary Tables 2 and 4, available online at <https://academic.oup.com/bib>, Methods).

Integration analysis

Dataset integration, an RNA homogenization task that requires access to sample-level data, is commonly adopted for single-cell RNA homogenization. To evaluate the trade-off between privacy preservation and full data access, we compared SpinAdapt to Seurat, ComBat, Limma [5], Scanorama [7] and ComBat-seq [14] for the integration of bulk-RNA datasets in the four dataset pairs used previously (Supplementary Table 1 available online at <https://academic.oup.com/bib>). ComBat-seq was evaluated on dataset pairs with RNA-seq data only. For high integration performance, we want to maximize dataset mixing while maintaining subtype-wise separability (no mixing of tumor subtypes) within integrated datasets.

To evaluate the various integration methods, we employ UMAP transform in conjunction with the average silhouette width (ASW) and local inverse Simpson's index (LISI) (Methods). For each of the four cancer dataset pairs, the silhouette score is computed for each integrated sample in source and target, and then the average silhouette score is reported across all samples (Methods, Figure 4A, Supplementary Table 3 available online at <https://academic.oup.com/bib>). Even though SpinAdapt did not have access to sample-level data

when learning the transformation between source and target, it significantly outperformed each of the other methods for colorectal cancer and breast cancer ($P < 10^{-5}$ and $P < 10^{-6}$, respectively). For pancreatic cancer, SpinAdapt outperformed ComBat, Limma, Scanorama and ComBat-seq ($P < 0.05$ for each method). For bladder cancer, Scanorama outperformed SpinAdapt ($P < 10^{-5}$), whereas SpinAdapt outperformed ComBat and Limma ($P < 10^{-6}$) (Supplementary Table 5 available online at <https://academic.oup.com/bib>). Even though the integration performance of SpinAdapt can be improved via direct access to samples, it significantly outperforms most of the competing methods in each experiment.

To analyze batch mixing and subtype-wise separation independently, we also employ the LISI metric for integration evaluation. For good integration performance, we sought high batch diversity and low subtype diversity in local sample neighborhoods, which associates with high batch LISI (bLISI) and low tissue LISI (tLISI) score, respectively. For each of the four cancer datasets, we report the average bLISI and tLISI scores across all integrated samples in source and target (Methods, Supplementary Figures 4, 9 and 10 available online at <https://academic.oup.com/bib>).

Transfer and validation of prognostic predictors

Finally, we demonstrate the transfer of Cox regression models across distinct datasets for three cancer types (breast, colorectal and pancreatic; Supplementary Table 5 available online at <https://academic.oup.com/bib>). Specifically, a Cox proportional hazards (PH) model is trained on the target dataset using a gene signature determined through application of an ensemble method on the target dataset (Methods). The risk thresholds for the survival model are determined based on the upper and lower quartiles of the distributions of log partial hazards of the target dataset such that samples with a predicted log partial hazard $>75\%$ percentile of said distribution are predicted to be high risk, and samples with a predicted log partial hazard $<25\%$ percentile of said distribution are predicted to be low risk. These model prediction thresholds were fixed across all evaluations.

We then compare multiple batch correction (adaptation) methods for transfer of the prognostic models from target to the source dataset. For each cancer type, the source dataset is adapted to the target using SpinAdapt, Seurat and ComBat. The target-trained Cox PH model is used to generate predictions (log partial hazards) on all samples from the source dataset, both for the uncorrected source dataset and the three correction methods. The risk thresholds determined on the target dataset are used to classify samples from the source dataset as low risk, high risk or unclassified based on their predicted log partial hazards values.

The performance of the prognostic models is quantified by computing the *c*-indices as well as the 5-year log-rank *P*-value and 5-year hazard ratio (HR) of the combined predicted high-risk and low-risk groups of

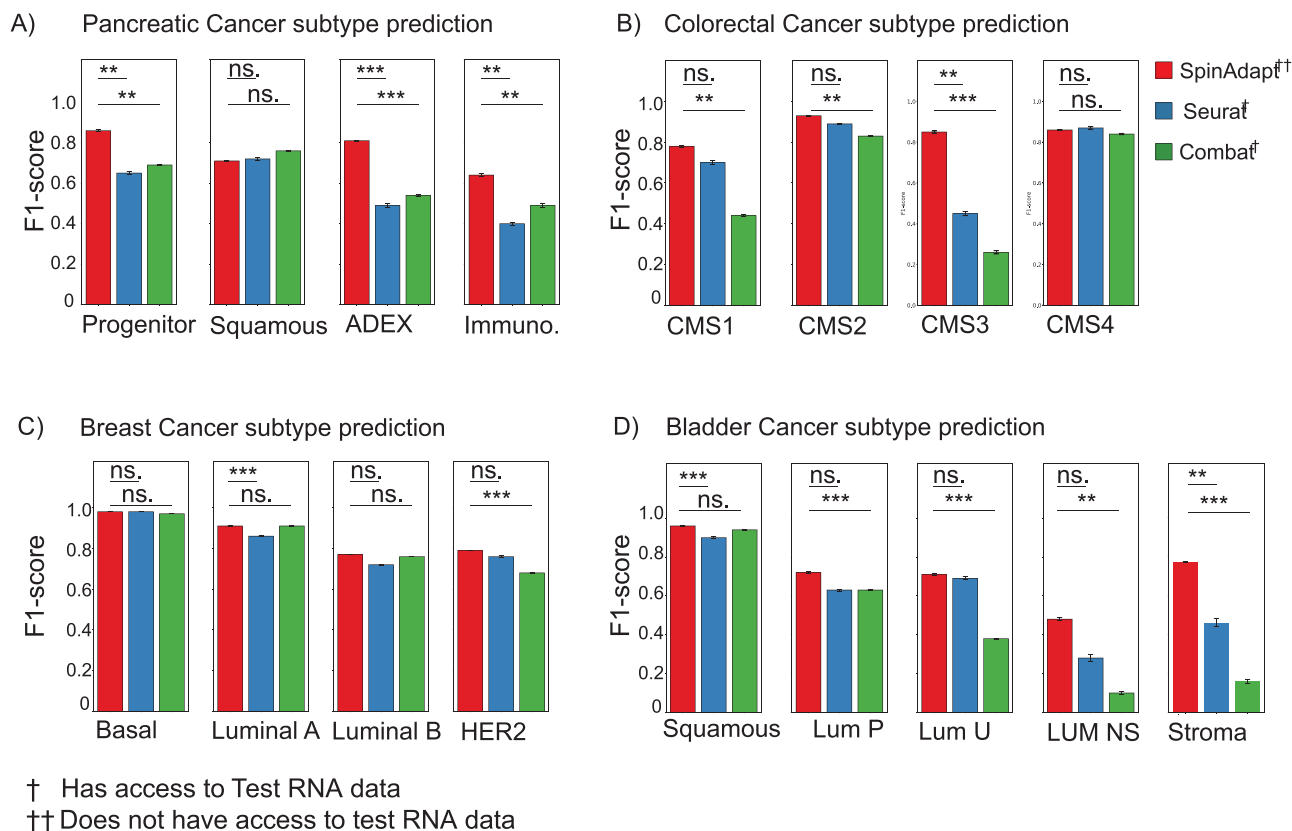


Figure 3. (A–D) Subtype prediction performance on held-out source subsets. We train subtype predictors on target data and evaluate them on source data. Source data is split into two disjoint subsets such that the correction model is trained on one subset and the predictor performance is evaluated on the other held-out subset. Seurat and ComBat do not support a fit-transform paradigm, and therefore, they are trained and evaluated on each of the disjoint subsets. For each subtype, the vertical bar represents the mean F1-score and the error bar represents the standard error over 30 repetitions of the experiment. SpinAdapt either ties or outperforms Seurat and ComBat on: pancreatic cancer, colorectal cancer, breast cancer and bladder cancer subtypes. Significance testing by two-sided paired McNemar test (Methods).

source samples for each cancer type and each adaptation method (Figure 5, Supplementary Table 7 available online at <https://academic.oup.com/bib>). SpinAdapt demonstrates high survival prediction accuracy for all datasets (*c*-index, log-rank *P*-value, HR): colorectal (0.63, 4e-2, 4.24), breast (0.66, 1e-5, 3.2) and pancreatic (0.65, 2e-6, 3.51), in contrast with uncorrected source dataset: colorectal (0.52, 4e-1, 0.54), breast (0.62, 7e-4, 2.2) and pancreatic (0.50, 7e-1, 1.29). Furthermore, SpinAdapt outperforms Seurat: colorectal (0.51, 7e-1, 0.85), breast (0.59, 2e-2, 2.0) and pancreatic (0.62, 2e-1, 2.66) as well as ComBat: colorectal (0.56, 2e-1, 0.57), breast (0.61, 3e-4, 2.4) and pancreatic (0.62, 7e-5, 2.64) in terms of all performance metrics for all cancer types.

Discussion

In the traditional paradigm of sharing data models across laboratories, the discussion presumes simultaneous transfer of molecular models and the associated training datasets. Therefore, all existing RNA correction methods require concurrent access to patient-level samples across all datasets. However, some health care organizations may prefer to not share patient-level data, and they would require a common trusted broker to have simultaneous access to their private datasets. Such trusted brokers are quite common in transactional

domains such as banking where privacy and trust play a major role. Recently, the use of cryptography and distributed computing has allowed the emergence of a secure, trustless financial transaction system that eliminates the use of such brokers [15]. Similar trust limitations still exist between health-care organizations that actively limit data sharing due to privacy and security concerns.

To allow the validation of models in a privacy-preserving setting, we identified three key desirable capabilities for a batch correction model: (i) the ability to work in a privacy-preserving setting; (ii) correct all datasets with respect to a reference set, which is used to train an immutable molecular model and (iii) transform/correct unseen samples after the correction model has been learned. SpinAdapt eliminates the need for a trusted broker, as it only operates on privacy-preserving aggregate statistics of each dataset and allows the application of a target model on privately held source data. The key idea of the algorithm is to match gene-wise means, variances and gene-gene covariances between a reference and other datasets. Despite an inherent trade-off between performance and privacy, SpinAdapt shows state-of-the-art performance on diagnostic, prognostic as well as integration tasks, outperforming similar correction algorithms that require access to private sample-level data. By only sharing data

factors of the training dataset alongside the RNA-based data model, SpinAdapt allows external validation and reuse of pretrained RNA models on novel datasets. The ability to share RNA models without the necessity of sharing model training data would improve research reproducibility across laboratories and pharmaceutical partners that cannot share patient data.

This paper demonstrates the application of SpinAdapt for transfer of diagnostic and prognostic models across distinct transcriptomic datasets, profiled across various laboratories and technological platforms, entailing RNA-Seq as well as various microarray platforms. Since the correction paradigm does not require sample-level data access, SpinAdapt enables correction of new prospective samples not included in training. The ability to correct held-out data is deemed necessary for validation frameworks, where the validation data need to be completely held out from training of any data models including classifiers, regressors or batch correctors to avoid overfitting. SpinAdapt enables rigorous validation of molecular predictors across independent studies by holding out the validation data from training of the predictor as well as the batch corrector.

Methods

Datasets

Gene expression dataset pairs are generated across various cancer types (bladder, breast, colorectal and pancreatic) pertaining to microarray platforms and RNA-sequencing (Supplementary Table 1 available online at <https://academic.oup.com/bib>). Bladder cancer datasets pertain to Seiler and TCGA cohorts [16], which are downloaded from GSE87304 and ICGC under the identifier BLCA-US, respectively. Colorectal cancer datasets are downloaded from GSE14333 [17] and ICGC under the identifier COAD-US, which are subsets of cohorts A and C, respectively, described in the ColoType prediction [18]. Breast cancer datasets pertain to TCGA [19] and SCAN-B cohorts [20], where the former are downloaded from ICGC under the identifier BRCA-US and the latter are obtained from GSE60789. Pancreatic cancer datasets are generated using the Bailey and TCGA cohorts [21], which are downloaded from ICGC under the identifiers PACA-AU and PAAD-US, respectively.

The subtype labels for patients across various cancer types are generated using well-accepted subtype annotations (Supplementary Table 1 available online at <https://academic.oup.com/bib>). Bladder cancer subtypes are labeled as luminal papillary (LumP), luminal non-specified (LumNS), luminal unstable (LumU), stroma-rich (Stroma), and basal/squamous (Ba/Sq), generated using a consensus subtyping approach [16]. Breast cancer subtypes are labeled as LumA, LumB, HER2-enriched (Her2) and Basal-like (Basal) [22]. Colorectal cancer subtypes are labeled as CMS1, CMS2, CMS3, CMS4, as published by the Colorectal Cancer Subtyping Consortium [23]. Pancreatic cancer subtypes are generated

using expression analysis and are labeled as squamous, pancreatic progenitor, immunogenic and ADEX [24].

Preprocessing

For each cancer type, we only keep patients with both expression data and subtype annotation labels available. Molecular subtypes with less than five patients in any cancer cohort are removed. For the microarray expression datasets (GSE14333 and Seiler; Supplementary Table 1 available online at <https://academic.oup.com/bib>), multiple probe sets may map to the same gene. The expression values were averaged across such probe sets to get gene expression value. Furthermore, for each cancer type, we remove genes with zero variance, and we only keep genes common between source and target, which are sorted in alphabetical order. Finally, we normalized the RNA-seq datasets using the variance stabilizing transform from DeSeq2, whereas microarray data were not normalized beyond their publication.

Benchmarking methods

For Seurat, we used the default package parameters, except when the number of samples in either dataset is <200, where the default value of `k.filter` does not work. Therefore, when the number of samples in either dataset is <200, we set the `k.filter` parameter as 50. For ComBat, a design matrix is created using the batch labels, and the method is implemented using the SVA package version 3.34.0. Similarly, Limma is implemented using limma package version 3.42.2. For Seurat and Scanorama, we deployed package versions 3.2.2 and 1.6, respectively. Finally, for ComBat-seq, we employed SVA with package version 3.38.

Parameters for SpinAdapt

For any given pair of source and target datasets, let p be the number of genes, n_s be the number of samples in source, and n_t be the number of samples in target. Across all experiments performed in this study, the parameters in SpinAdapt (Algorithm 1) are set as follows: $\alpha = 0.01$, $\lambda = (2/3) * \min(n_s, n_t)$, and `variance_norm` is set to True when the source is microarray but False otherwise. However, for integration analysis of source and target datasets, we always set `variance_norm = True`.

Algorithm 1 SpinAdapt algorithm

```
function Main( $X_t, X_s, X_{sh}$  (optional),  $F_t, \alpha, \lambda, \text{variance}_{norm}$ )
1. Compute data factors for source:
    $U_s, m_s, s_s = \text{Factors}(X_s)$ .
2. Compute data factors for target:
    $U_t, m_t, s_t = \text{Factors}(X_t)$ .
3. Fit (train) SpinAdapt correction model:
    $A \leftarrow \text{SpinAdapt.Fit}(U_s, U_t, \alpha, \lambda)$ .
4. Transform source dataset:
    $X_{sc} \leftarrow \text{SpinAdapt.Transform}$ 
( $X_s, X_{sh}$  (optional),  $A, U_s, m_s, s_s, U_t, m_t, s_t, \text{variance}_{norm}$ ).
5. Apply target-trained predictor  $F_t$ :
    $y_{sc} \leftarrow F_t(X_{sc})$ .
6. Return predictions  $y_{sc}$ .
```

Evaluation methods for transfer of diagnostic models

For each of the 17 cancer subtypes (Supplementary Table 1 available online at <https://academic.oup.com/bib>), we train a one-versus-rest random forest classifier on the target dataset such that the classifier learns to discriminate the selected subtype against all other subtypes in the target. Specifically, all target samples annotated with the selected subtype are given a positive label, while the rest of the target samples are assigned a negative label. The hyperparameters for the random forest classifier are learnt in a 3-fold cross-validation experiment on the target dataset.

We compare multiple batch correction (adaptation) methods for transfer of the subtype classifier from target to the source dataset. The transfer requires adaptation of the source dataset to the target reference. For unbiased performance evaluation of a batch correction method, the test set for the classifier and the training set for the correction method need to be disjoint, and thus, the correction model does not train on the classifier test set. We propose a framework for validating transfer of classifiers across datasets that avoids such information leakage.

The validation framework randomly splits the source dataset into two mutually exclusive subsets: Source A and Source B. First, the adaptation model is trained from Source A to target (fit), and applied to Source B (transform). The target classifier generates predictions on corrected Source B (Supplementary Figure 1A available online at <https://academic.oup.com/bib>). Second, the adaptation model is fit from Source B to target, followed by transformation of Source A and generation of predictions on corrected Source A (Supplementary Figure 1B available online at <https://academic.oup.com/bib>). Finally, the classification performance is quantified by computing F-1 scores for all samples in the held-out corrected source-A and source-B subsets. We repeat the entire procedure for 30 times, choosing a different partitioning of source into source-A and source-B, and we report the mean F-1 score for each subtype over the 30 iterations.

We evaluate SpinAdapt using the aforementioned framework for validating transfer of subtype classifiers. However, since Seurat and ComBat cannot transform out-of-sample data, they can only correct samples included in training of these correction methods. Therefore, for these two methods, we train the correction model on the classifier test set, followed by application of target-trained classifier on the corrected test set. Specifically, for ComBat and Seurat, we fit-transformed source-A to target, fit-transformed source-B to target and computed F-1 scores for all samples in the transformed source-A and source-B subsets. As before, we repeat the procedure for 30 times using the same data splits as used for SpinAdapt validation, which enabled pairwise performance comparisons between SpinAdapt, Seurat and ComBat.

For each subtype, we performed the two-sided, paired McNemar test to identify if the differences between any pair of adaptation methods are statistically significant [25]. Due to the rarity of positives for a selected subtype in each dataset, we perform the McNemar test only on samples with positive ground truth. Each positive sample is assigned a correct or incorrect classification label. Then, for each pair of correction methods, the McNemar test statistic is evaluated on the disagreements between correction methods on the positive samples. We report the median P-value across the 30 repetitions of the validation framework (Supplementary Table 4 available online at <https://academic.oup.com/bib>).

Evaluation methods for dataset integration

A common task for RNA-based algorithms is dataset integration (batch mixing). There is an inherent trade-off between batch mixing and preservation of the biological signal within integrated datasets. To quantify preservation of the biological signal, we quantify subtype-wise separability (no mixing of tumor subtypes) in the integrated datasets. Therefore, for high data integration performance, we want to minimize subtype mixing while maximizing batch mixing.

To compare SpinAdapt with other batch integration methods, we assess the goodness of batch mixing and tissue type separation. First, we employed the ASW to quantify batch mixing and tissue segregation. The silhouette score of a sample is obtained by subtracting the average distance to samples with the same tissue label from the average distance to samples in the nearest cluster w.r.t. the tissue label and then dividing by the larger of the two values [26]. Therefore, the silhouette score for a given sample varies between -1 and 1 , such that a higher score implies a good fit among samples with the same tissue label and vice versa. In other words, a higher AVW implies mixing of batches within each tissue type or/and separation of samples from distinct tissue types.

To explicitly quantify batch mixing and tissue segregation, independently, we employ the LISI. The LISI metric assigns a diversity score to each sample by computing the effective number of label types in the local neighborhood of the sample. Therefore, the notion of diversity depends on the label under consideration. When the label is set to batch membership, the resulting metric is referred to as bLISI since it measures batch diversity in the neighborhood of each sample. When the label is set to tissue type, the resulting metric is referred to as tLISI since it measures tissue type diversity in sample neighborhood. For good integration, we sought sample neighborhoods with high batch diversity and low subtype diversity, which correlates with high bLISI and low tLISI score, respectively. For each integration method and cancer dataset, we report average bLISI and tLISI scores across all samples in source and target datasets (Supplementary Table 3, Supplementary Figures 4, 9 and

10 available online at <https://academic.oup.com/bib>). When comparing methods using average bLISI, which measures dataset mixing, Seurat outperforms SpinAdapt on breast and bladder cancer datasets ($P < 10^{-3}$), whereas SpinAdapt outperforms ComBat, Limma and Scanorama on colorectal and pancreatic cancer datasets ($P < 10^{-3}$) (Supplementary Tables 3 and 5 available online at <https://academic.oup.com/bib>). SpinAdapt also outperforms ComBat-seq on both breast and pancreatic cancer datasets ($P < 10^{-4}$). When comparing methods using average tLISI, SpinAdapt significantly outperforms all other methods on colorectal ($P < 10^{-7}$) and pancreatic ($P < 0.05$) cancer datasets, while significantly outperforming Seurat, ComBat, Limma and Scanorama on breast cancer dataset ($P < 10^{-13}$), implying SpinAdapt best preserves molecular structures for dataset integration (Supplementary Tables 3 and 5 available online at <https://academic.oup.com/bib>). When comparing methods using tLISI on the bladder cancer dataset, SpinAdapt outperforms Seurat, ComBat and Limma ($P < 10^{-6}$), whereas Scanorama outperforms SpinAdapt without significance.

The various integration metrics, including silhouette, bLISI and tLISI scores are computed on the UMAP embeddings of the integrated datasets for each cancer type (Supplementary Table 1 available online at <https://academic.oup.com/bib>). Specifically, the scores in each experiment are computed on the first 50 components of the UMAP transform, where the UMAP embeddings are computed using default parameters of the package. The ASW, bLISI and tLISI scores are reported along with the standard errors (Supplementary Table 3 available online at <https://academic.oup.com/bib>). For each metric, significance testing between methods is performed by a two-sided paired Wilcoxon test (Supplementary Table 5 available online at <https://academic.oup.com/bib>).

Evaluation methods for transfer of prognostic models

For each of the four cancer datasets (colorectal, breast and pancreatic, Supplementary Table 6 available online at <https://academic.oup.com/bib>), we trained a Cox PH model on the target dataset using a gene signature determined through an ensemble method performed on the target dataset. The ensemble method for feature selection uses four ranked lists of genes, based on different statistical tests or machine learning models: chi-square scores, F-scores, random forest importance metrics and univariate Cox PH P -value. We tested the predictive values of various permutations of genes of increasing length ($n = 10, 50, 75, 100, 200, 300, 500$ genes) as signatures of a Cox PH model trained and tested on random splits of the target dataset, in a 5-fold cross-validation setting, where 50% of the target dataset was assigned to the training set and the remainder was assigned to the test set. The best-performing signature was determined based on the c-index determined on the five random test sets. We then

used this signature to train a final Cox PH model on the target dataset.

Visualization

We employ the UMAP transform to visualize the batch integration results for each cancer type (Figures 2 and 4, Supplementary Figure 3 available online at <https://academic.oup.com/bib>). Specifically, we perform visualization in each experiment using the first two components of the UMAP embeddings, where the number of neighbors are set to 10 and the min_dist parameter is set to 0.5. These parameters are fixed for all visualizations in the study that employ UMAP embeddings.

Algorithm details

SpinAdapt inputs source and target expression datasets for training, corrects the batch-biased source expression data, even when the source data is held out from training, followed by evaluation of the target-trained predictor on the corrected source data. The algorithm, as outlined in Algorithm 1, can be broken down into several main steps: computation of source and target data factors from source and target datasets in Steps 1 and 2, respectively, estimation of a low-rank affine map between source and target PCA basis in Step 3, adaptation (correction) of the source dataset in Step 4, and finally, evaluation of the target-trained predictor on adapted source dataset in Step 5. Notably, Step 4, Algorithm 1 can adapt source dataset X_{sh} that is held-out from the training source dataset X_s . Steps 1 and 2 are executed using Algorithm 2, whereas Steps 3 and 4 are executed using Algorithms 3 and 4, respectively. These steps are explained next in further detail.

Algorithm 2 Compute data factors, gene-wise means and variances

function Factors(X_e):

1. Define $n_e :=$ number of columns in X_e , $p :=$ number of rows in X_e , $d_e := \min(n_e, p)$.
 2. Compute empirical mean vector.

$$m_e \leftarrow \frac{1}{n_e} \sum_{i=1}^{n_e} X_{e,i}$$
 3. Compute empirical variance vector.

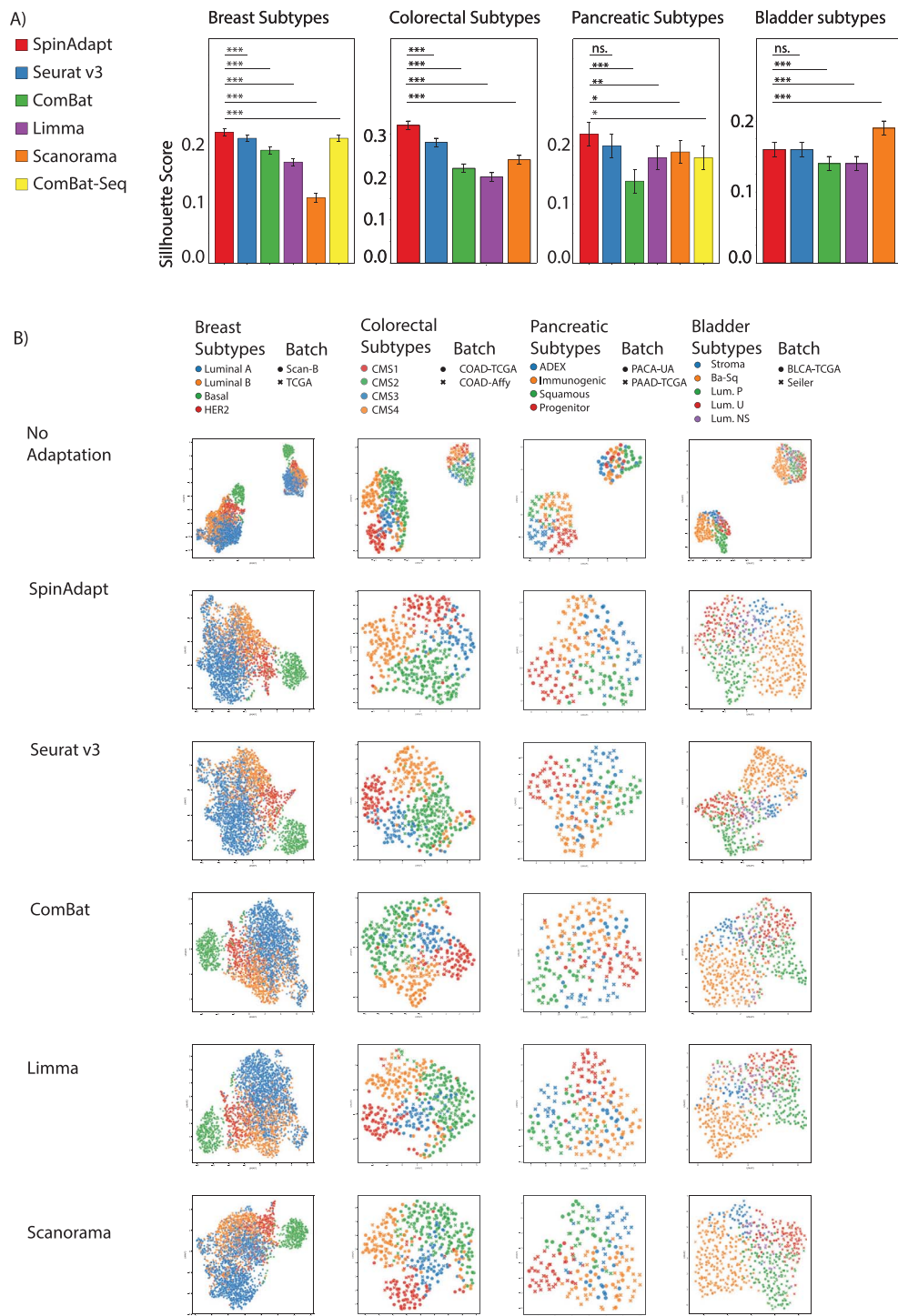
$$s_e \leftarrow \frac{1}{n_e - 1} \sum_{i=1}^{n_e} (X_{e,i} - m_e)^2$$
 4. Compute empirical covariance matrix.

$$C_e \leftarrow \frac{1}{n_e - 1} (X_e - m_e) (X_e - m_e)^T$$
 5. Compute eigenvectors of the empirical covariance matrix.

$$U_e \leftarrow \text{Eigenvectors}(C_e)$$
 6. Retain top d_e eigenvectors.

$$U_e \leftarrow U_e[1 : d_e]$$
 7. Return U_e, m_e, s_e
-

In Step 1, Algorithm 1, data factors are computed for the source dataset X_s , where the data factors comprise of the PCA basis U_s , gene-wise means m_s and gene-wise variances s_s of the source dataset. The details for the computation of these data factors are outlined in Algorithm 2, where the gene-wise means and variances are computed



in Steps 2 and 3, whereas the PCA basis are computed in Steps 4–6. Similarly, data factors are computed for the target dataset in Step 2, Algorithm 1, where the data factors entail the PCA basis U_t , gene-wise means m_t and

gene-wise variances s_t for the target dataset X_t . The gene-wise means and variances m_s , m_t , s_s and s_t are used in the correction step (Step 4, Algorithm 1), whereas the PCA basis U_s and U_t are used in the train and correction steps

(Steps 3 and 4, Algorithm 1). The usage of statistics s_s and s_t in Step 4, Algorithm 1 is optional depending on the Boolean value of $\text{variance}_{\text{norm}}$, as we explain later.

Notably, Algorithm 1 does not require simultaneous access to sample-level patient data in source and target datasets at any step. Computation of source data factors in Step 1 needs access to X_s only, whereas computation of target data factors in Step 2 needs access to X_t only. Training in Step 3 only requires access to the PCA basis of source and target datasets. Since the PCA basis cannot be used for recovery of sample-level patient data, the basis are privacy-preserving (Supplementary Note). Adaptation in Step 4 requires access to the source expression data X_s , linear map A , and the data factors of both datasets, without requiring access to the target dataset X_t .

Algorithm details: glossary

We define the data structures employed in Algorithm 1, with the dimensionality of each structure. The dimensionality is stated in terms of p , the number of genes; n_s , number of samples in source dataset; n_t , number of samples in target dataset; d_s , dimensionality of source latent space; d_t , dimensionality of source latent space.

- $X_s \in \mathbb{R}^{p \times n_s}$ The train source dataset.
- $X_t \in \mathbb{R}^{p \times n_t}$ The train target dataset.
- $X_{sh} \in \mathbb{R}^{p \times n_t}$ The held-out source dataset.
- $X_{s,i} \in \mathbb{R}^p$ The i th column of X_s .
- $X_{t,i} \in \mathbb{R}^p$ The i th column of X_t .
- $m_s \in \mathbb{R}^p$ The empirical gene-wise mean of source dataset.
- $m_t \in \mathbb{R}^p$ The empirical gene-wise mean of target dataset.
- $s_s \in \mathbb{R}^p$ The empirical gene-wise variance of source dataset.
- $s_t \in \mathbb{R}^p$ The empirical gene-wise variance of target dataset.
- $C_s \in \mathbb{R}^{p \times p}$ The empirical covariance of source dataset.
- $C_t \in \mathbb{R}^{p \times p}$ The empirical covariance of target dataset.
- $U_s \in \mathbb{R}^{p \times d_s}$ Principal component factors for source dataset.
- $U_t \in \mathbb{R}^{p \times d_t}$ Principal component factors for target dataset.
- $A \in \mathbb{R}^{d_s \times d_t}$ Transformation matrix.
- $X_{sc} \in \mathbb{R}^{p \times n_s}$ The corrected output source dataset.
- $X(i, j)$ The i th row and j th column of any matrix X .
- $v(i)$ The i th entry of any vector v .
- F_t Classifier trained on the target dataset.

The parameters α and λ correspond to step size and regularization parameters for the iterative algorithm Fit (Algorithm 3). The parameter $\text{variance}_{\text{norm}}$ is a Boolean variable, which determines if the adaptation step (Step 4, Algorithm 1) entails variance normalization of the source dataset (for details, see Algorithm 4).

Algorithm details: learning transformation between PCA factors

In Step 3, Algorithm 1, we learn a low matrix-rank transformation between PCA factors of the source dataset and

Algorithm 3 Learn transformation from source to target factors (Fit)

```
function SpinAdapt.Fit( $U_s, U_t, \alpha, \lambda$ )
   $k = 0$ 
  Initialize  $A^{(0)}$ 
  repeat
    a. {Gradient descent}
        $\hat{A}^{(k)} \leftarrow A^{(k)} - \alpha \nabla g(A^{(k)})$ 
    b. {Projection Step}
        $A^{(k+1)} \leftarrow P_\lambda(\hat{A}^{(k)})$ 
    c.  $k \leftarrow k + 1$ 
  until convergence
  return  $A^{(k)}$ 
```

Algorithm 4 Adapt the source dataset (Transform)

```
function SpinAdapt.Transform
( $X_s, X_{sh}$  (optional),  $A, U_s, m_s, s_s, U_t, m_t, s_t, \text{variance}_{\text{norm}}$ )
  1. Select the dataset for transformation:
     If  $X_{sh}$  is null:
        $X_o := X_s$ .
     Else:
        $X_o := X_{sh}$ .
  2. Gene-wise variance normalization:
     If  $\text{variance}_{\text{norm}}$  is True:
       For  $i = 1, 2, \dots, p$ 
         For  $j = 1, 2, \dots, n_s$ 
            $X_o(i, j) \leftarrow (\sqrt{s_t(i)} (X_o(i, j) - m_s(i)) / \sqrt{s_s(i)}) + m_s(i)$ .
  3. Compute source PCA embeddings:
        $\tilde{X}_s \leftarrow U_s^T (X_o - m_s)$ .
  4. Correct source PCA embeddings:
        $\tilde{X}_{sc} \leftarrow A^T \tilde{X}_s$ .
  5. Map corrected source PCA embeddings to the gene expression space:
        $X_{sc} \leftarrow U_t \tilde{X}_{sc} + m_t$ .
  6. Return  $X_{sc}$ .
```

PCA factors of the target dataset. We pose a non-convex optimization problem to learn the transformation, and then we present an effective computational approach to solve it, as we explain next.

Objective function for step 3, Algorithm 1

The objective function is based on Frobenius norm between transformed source PCA basis $U_s A$ and the target PCA basis U_t , as follows

$$A_r^* = \arg \min_A \|U_s A - U_t\|_F, \quad \text{s.t. rank}(A) \leq \lambda, \quad (1)$$

where A represents the transformation matrix, λ represents the matrix-rank constraint and $\text{rank}(A)$ represents the matrix-rank of A . In the main term, it can be seen that the i th column of the transformation matrix A determines what linear combination of the columns of U_s best approximates the i th column of U_t , where $i = 1, 2, \dots, d_t$. Therefore, the intuition behind the main term is to approximate each target factor using some linear combination of source factors.

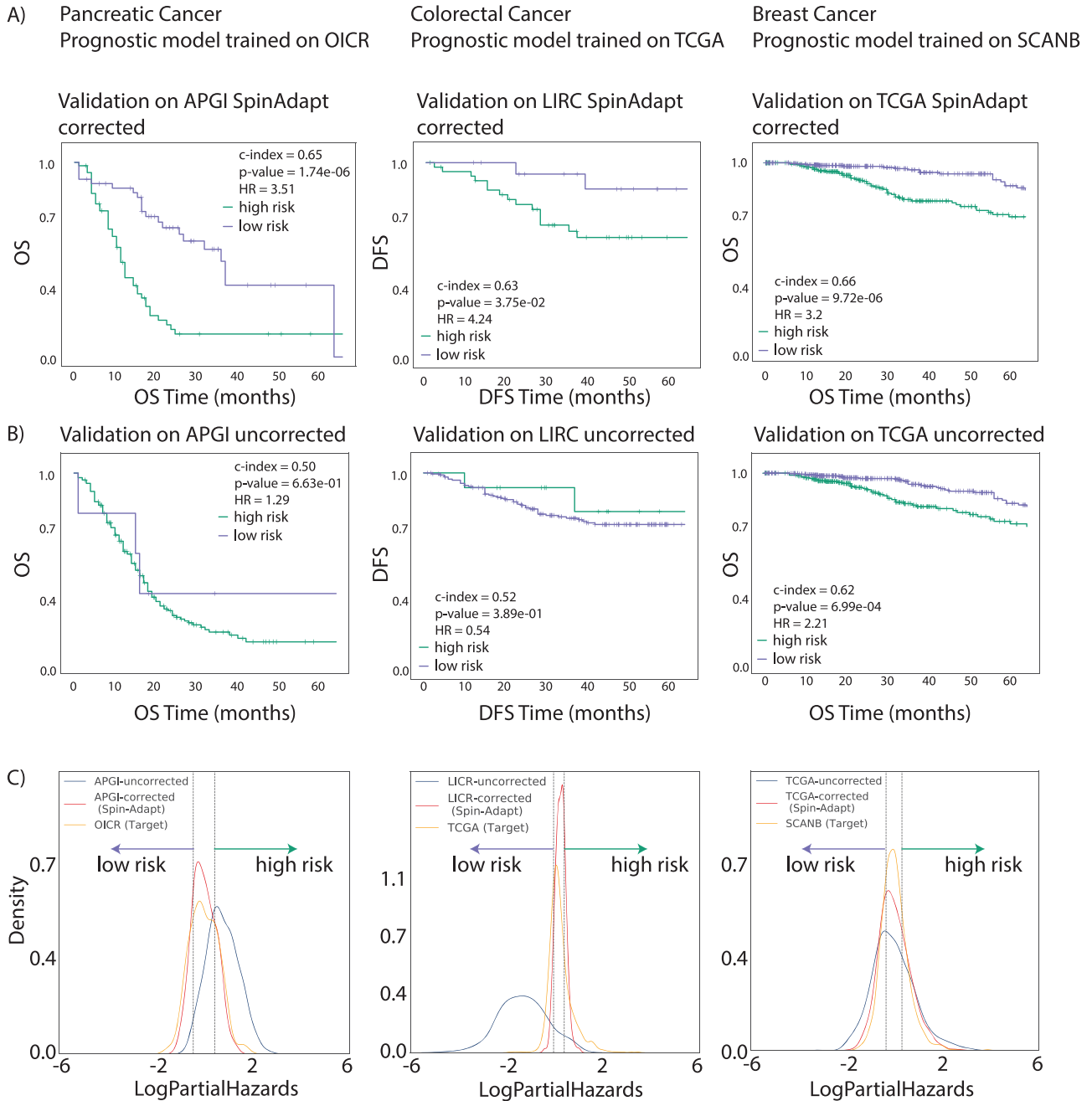


Figure 5. Survival curves for the predicted low-risk and predicted high-risk groups in the validation source dataset (A) before correction and (B) after correction using SpinAdapt, for each of the three cancer types (breast, colorectal and pancreatic). (C) Distribution of log-partial hazards for the target-trained Cox model on the target dataset, validation source dataset and corrected validation source dataset for each of the three cancer types.

The inequality constraint in Equation (1) is a matrix-rank penalization term, which restricts the solution space of A to matrices with matrix-rank is less than λ . The rank constraint is reminiscent of sparse constraint in sparse recovery problems, where the constraint restricts the maximum number of non-zero entries in the estimated solution, thereby reducing the sample complexity of the learning task. Similarly, in Equation (1), the constraint restricts the maximum matrix-rank of A , making the algorithm less prone to overfitting, while decreasing the sample requirement of learning the affine map from source to target factors. However, the problem

posed in Equation (1) turns out to be non-convex and thus hard to solve. We employ traditional optimization techniques and derive an efficient routine for computing A_r^* , as follows in the next subsection.

Optimization solution for step 3, Algorithm 1

Let $g(A) = \|U_s A - U_t\|_F$, and let $S_\lambda = \{A : A \in R^{d_s \times d_t}, \text{rank}(A) \leq \lambda\}$. Then, the objective function in Equation (1) can be rewritten as

$$A_r^* = \underset{A}{\text{argmin}} g(A), \text{ s.t. } A \in S_\lambda. \tag{2}$$

Gradient descent can be used to minimize $g(A)$ w.r.t. A because the function is convex and differentiable. In contrast, Equation (2) cannot be evaluated using gradient descent, since the set of low-rank matrices S_λ is non-convex. However, we note that the Euclidean projection onto the set S_λ can be efficiently computed, which hints that Equation (2) can be minimized using projected gradient descent, as we explain next. Let the Euclidean projection of a matrix A onto set S_λ be denoted by $P_\lambda(A)$. Then, mathematically we have $P_\lambda(A) = \arg \min_Z \{\|A - Z\|_F : Z \in S_\lambda\}$. From the Eckart-Young Theorem, we know that $P_\lambda(A)$ can be efficiently evaluated by computing the top λ singular values and singular vectors of A . The closed-form solution of $P_\lambda(A)$ is given by the SVD transform $U_\lambda \Sigma_\lambda V_\lambda^T$, where columns of U_λ contain the top λ eigenvectors of AA^T , columns of V_λ contain the top λ eigenvectors of $A^T A$ and entries of the diagonal matrix Σ_λ are square roots of the top λ eigenvalues of AA^T . We are finally ready to present an algorithm for solving Equation (2).

Pseudocode for SpinAdapt.Fit (Algorithm 3)

We present the algorithm for executing Step 3, Algorithm 1, which is essentially a solution to the optimization problem in Equation (2). We propose the use of the projected gradient descent algorithm to evaluate Equation (2), which is an iterative application of the following descent step:

$$A^{(k+1)} = P_\lambda \left(A^{(k)} - \alpha \nabla g(A) \right)$$

for $k = 0, 1, 2, 3, \dots$ till convergence. Details are provided in Algorithm 3 below.

Pseudocode for SpinAdapt.Transform (Algorithm 4)

Finally, we present the algorithm for executing Step 4, Algorithm 1, where the batch-biased dataset is corrected using the transformation A . Details are outlined in Algorithm 4, as follows. In Step 1, Algorithm 4, the held-out source dataset X_{sh} is selected for correction, if provided. If held-out evaluation data (X_{sh}) is not provided, the train source dataset X_s is selected for correction. In Step 2, if the input parameter $\text{variance}_{\text{norm}}$ is set to True, the variance of each gene in the source dataset is matched to variance of the corresponding gene in the target dataset. In Step 3, the PCA embeddings of each source sample are computed. In Step 4, the computed PCA embeddings are corrected, using the transformation matrix A . In Step 5, the corrected PCA embeddings are transformed to the gene expression space. Finally, the corrected source gene expression profiles are returned in Step 6.

Key Points

- We propose SpinAdapt, which is an unsupervised batch correction algorithm that learns a correction model between privacy-preserving aggregate statistics of the

reference and the other datasets, while enabling correction of prospective new samples.

- SpinAdapt has the following desired characteristics: asymmetric correction to a reference, learning of the correction model without concurrent access to patient-level information across all laboratories and correction of new samples.
- Despite an inherent trade-off between performance and privacy, SpinAdapt outperforms other batch correction algorithms for transfer of various diagnostic and prognostic models across laboratories.
- SpinAdapt enables the application of existing published models on independent validation studies. This is in contrast to the common practice of retraining and modifying published algorithms for predicting on new independent datasets.
- We demonstrate the application of SpinAdapt in multiple omics domains: RNA, methylation and protein data, quantified by varying platforms: microarrays, sequencing and mass spectrometry.

Authors' contributions

T.A., A.A.K., E.L., M.C.S. and R.P. conceived the study, T.A., M.A.C., S.W. and R.P. designed and developed SpinAdapt, T.A., M.C., S.W. and R.P. performed computational analysis, T.A., M.C., S.W., J.R.D., A.A.S., A.A.K., M.C.S. and R.P. wrote the manuscript.

Competing Interests

All authors have a financial relationship as employees of Tempus Labs, Inc.

Supplementary Data

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

Acknowledgements

We thank Matthew Kase and Alexandria Bobe on the Tempus Scientific Communications team for review of the manuscript.

Data availability

For benchmarking, we employed gene expression dataset pairs (Supplementary Table 1 available online at <https://academic.oup.com/bib>) across various cancer types (bladder, breast, colorectal and pancreatic). Bladder cancer datasets pertain to Seiler and TCGA cohorts, where the former are downloaded from GSE87304 and the latter are obtained from ICGC under the identifier BLCA-US. Colorectal cancer datasets pertain to GSE14333 and TCGA cohorts, where the former are obtained from GSE14333 and the latter are acquired from ICGC under the identifier COAD-US. Breast cancer datasets pertain

to TCGA and SCAN-B cohorts, where the former are downloaded from ICGC under the identifier BRCA-US and the latter are obtained from GSE60789. Pancreatic cancer datasets pertain to the Bailey and TCGA cohorts, which are downloaded from ICGC under the identifiers PACA-AU and PAAD-US, respectively.

Code availability

Programming code related to data processing, algorithm design and evaluation will be made available for non-commercial use upon request to the corresponding author.

References

1. Leek JT, Scharpf RB, Bravo HC, et al. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat Rev Genet* 2010;**11**:733–9.
2. Leek JT. svaseq: removing batch effects and other unwanted noise from sequencing data. *Nucleic Acid Res* 2014;**42**:e161.
3. Tran HTN, Ang KS, Chevrier M, et al. A benchmark of batch-effect correction methods for single-cell RNA sequencing data. *Genome Biol* 2020;**21**:12.
4. Hoadley KA, Yau C, Wolf DM, et al. Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin. *Cell* 2014;**158**:929–44.
5. Ritchie ME, Phipson B, Wu D, et al. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acid Res* 2015;**43**:e47.
6. Haghverdi L, Lun ATL, Morgan MD, et al. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat Biotechnol* 2018;**36**:421–7.
7. Hie B, Bryson B, Berger B. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. *Nat Biotechnol* 2019;**37**:685–91.
8. Lotfollahi M, Wolf FA, Theis FJ. scGen predicts single-cell perturbation responses. *Nat Methods* 2019;**16**:715–21.
9. Lin Y, Ghazanfar S, Wang KYX, et al. scMerge leverages factor analysis, stable expression, and pseudoreplication to merge multiple single-cell RNA-seq datasets. *Proc Natl Acad Sci U S A* 2019;**116**:9775–84.
10. Korsunsky I, Millard N, Fan J, et al. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nat Method* 2019;**16**:1289–96.
11. Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* 2007;**8**:118–27.
12. Stuart T, Butler A, Hoffman P, et al. Comprehensive integration of single-cell data. *Cell* 2019;**177**:1888–902.e21.
13. Duncan GT, Pearson RW. Enhancing access to microdata while protecting confidentiality: prospects for the future. *Stat Sci* 1991;**6**:219–32.
14. Zhang Y, Parmigiani G, Johnson WE. ComBat-seq: batch effect adjustment for RNA-seq count data. *NAR Genom Bioinform* 2020;**2**:1–10.
15. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. *Econometrica*: Manubot, 2019.
16. Kamoun A, de Reyniès A, Allory Y, et al. A consensus molecular classification of muscle-invasive bladder cancer. *Eur Urol* 2020;**77**:420–33.
17. Jorissen RN, Gibbs P, Christie M, et al. Metastasis-associated gene expression changes predict poor outcomes in patients with dukes stage B and C colorectal cancer. *Clin Cancer Res* 2009;**15**:7642–51.
18. Buechler SA, Stephens MT, Hummon AB, et al. ColoType: a forty gene signature for consensus molecular subtyping of colorectal cancer tumors using whole-genome assay or targeted RNA-sequencing. *Sci Rep* 2020;**10**:12123.
19. Hoadley KA, Yau C, Hinoue T, et al. Cell-of-origin patterns dominate the molecular classification of 10,000 Tumors from 33 types of cancer. *Cell* 2018;**173**:291–304.e6.
20. Saal LH, Vallon-Christersson J, Häkkinen J, et al. The Sweden Cancerome analysis network—breast (SCAN-B) initiative: a large-scale multicenter infrastructure towards implementation of breast cancer genomic analyses in the clinical routine. *Genome Med* 2015;**7**:20.
21. Zhao L, Zhao H, Yan H. Gene expression profiling of 1200 pancreatic ductal adenocarcinoma reveals novel subtypes. *BMC Cancer* 2018;**18**:603.
22. Prat A, Parker JS, Fan C, et al. PAM50 assay and the three-gene model for identifying the major and clinically relevant molecular subtypes of breast cancer. *Breast Cancer Res Treat* 2012;**135**:301–6.
23. Guinney J, Dienstmann R, Wang X, et al. The consensus molecular subtypes of colorectal cancer. *Nat Med* 2015;**21**:1350–6.
24. Australian Pancreatic Cancer Genome Initiative, Bailey P, Chang DK, et al. Genomic analyses identify molecular subtypes of pancreatic cancer. *Nature* 2016;**531**:47–52.
25. Dietterich TG. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput* 1998;**10**:1895–923.
26. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* 1987;**20**:53–65.