# Privacy Preserving RNA-Model Validation Across Laboratories

**Talal Ahmed[1], Mark A Carty[1], Stephane Wenric[1], Jonathan R Dry[1], Ameen Abdulla Salahudeen[1], Aly A. Khan[1], Eric Lefkofsky[1], Martin C Stumpe[1], Raphael Pelossof[1#]**

**[1] Tempus Labs Inc., Chicago, IL 60654, USA**

**[#] Correspondence to Raphael Pelossof <rafi.pelossof@tempus.com>**

## Abstract

Reproducibility of results obtained using 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. While current RNA correction algorithms may overcome these differences, they require access to patient-level data which carries inherent risk of loss of privacy. Here, we describe SpinAdapt, a novel unsupervised domain adaptation algorithm that enables the transfer of molecular models across laboratories without access to patient-level sequencing data thereby minimizing privacy risk. SpinAdapt computes data corrections via aggregate statistics of each dataset, rather than requiring full sample-level data access, thereby maintaining patient data privacy. Furthermore, decoupling the model from its training data allows the correction of new streaming prospective data, enabling model evaluation on validation cohorts. SpinAdapt outperforms current correction methods that require patient-level data access. We expect this novel correction paradigm to enhance research reproducibility, quality, and patient privacy.

## 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 inter-institutional 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 is not transferred.

Correction of batch-specific biases in RNA-Seq 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 [2–4]. 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. Since the reference remains unchanged in batch correction, the asymmetry enables the transfer (application) of models trained on reference dataset to batch-corrected dataset(s).

Machine learning models such as disease or subtype classifiers are often developed using RNA expression data [5]. Batch integration methods may not be suitable for transfering classifiers trained on gene expression datasets, since integration methods do not necessarily output expression profiles. These include methods based on gene-wise linear models like Limma[6], mutually nearest neighbors (MNNs) like MNN Correct [7] and Scanorama [8], mutually nearest clusters (MNCs) like ScGen [9], pseudoreplicates like ScMerge [10], and multi-batch clusters like Harmony [11]. In contrast, batch correction methods can correct a source library to a target reference library, like Combat [2], Seurat3 [12] and the proposed SpinAdapt algorithm, and thus can be used for transferring classifiers 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 datasets 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 transfer of data may not be possible due to data ownership, GDPR, or similar regulations. SpinAdapt 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.

**Results**

In this study we demonstrate the transfer and validation of molecular predictors across transcriptomic datasets while preserving data privacy (**Figure 1A)**. Data factors, which are aggregate statistics of each dataset, neither convey Protected Health Information (PHI) nor allow reconstruction of sample-level data (**supplementary note**), and thus can be transferred between labs. SpinAdapt learns corrections between data factors of each dataset, followed by application of corrections on source expression data, even when the source data is prospective.

SpinAdapt corrections are learnt using a regularized linear transformation between the data factors of source and target, which comprise of the PCA (Principal Component Analysis) basis

of source and target, respectively (**methods, Figure 1B**). Once the transformation has been learned, it can be applied on the source dataset for correction, followed by application of the target-trained classifier on the corrected source dataset (**Figure 1C)**. Therefore, SpinAdapt only requires access to the PCA basis of each dataset to learn the transformation for the source dataset. Furthermore, in contrast to current methods, the transformation can also be applied to new prospective data that was held-out in the learning step. The ability to transform data held-out from model training is deemed necessary for machine learning algorithms to avoid overfitting, by ensuring the evaluation step (transform) is independent of the learning step (fit). This fit-transform paradigm is extended by SpinAdapt to RNA adaptation of new prospective data.

To evaluate batch correction methods for transferring molecular predictors across datasets, we transferred tumor subtype classifiers across four pairs of publicly available cancer datasets (bladder, breast, colorectal, pancreatic), covering 4,076 samples across 17 different tumor subtypes and three technological platforms (RNA-Seq, Affymetrix U133plus2 Microarray, and Human Exon 1.0 ST Microarray) (**Supplementary Table 1**). For each dataset pair, we trained one-vs-rest tumor subtype classifiers on the target dataset (**methods**).

A common approach for validating target-trained classifiers across datasets is to correct source to target, and then evaluate the classifier on corrected source dataset. However, such an approach requires the batch correction model to train on the test set for the classifier, which may lead to information leakage. We propose a validation framework that holds out the classifier test set from both correction (adaptation) model and classifier training.

The validation framework proceeds by creating two mutually exclusive sets from source (Source-A and Source-B). First, we fit the adaptation model between Source-A and target, followed by transformation and prediction on Source-B (**Supplementary Figure 1A**). Second, we fit the adaptation model between Source-B and target, followed by transformation and prediction on Source-A (**Supplementary Figure 1B**). Finally, the held-out predictions were concatenated, and performance was evaluated using F-1 score (**methods**). SpinAdapt's performance was evaluated using this framework. Existing correction methods, like ComBat and Seurat, have currently not implemented a transformation method for held-out out-of-sample data. Therefore, these methods were trained on the classifier test set (**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 17 tumor

subtypes including Pancreatic subtypes: Progenitor, ADEX, Immunogenic, Colorectal subtypes: CMS3, Breast subtypes: Luminal A, Bladder subtypes: Squamous and Stroma. SpinAdapt also significantly outperformed ComBat on 11 out of 17 subtypes including Pancreatic subtypes: Progenitor, ADEX, Immunogenic, Colorectal subtypes: CMS1, CMS2, CMS3, Breast Subtypes: Her2, Bladder subtypes: Lump P, Lum U, Lum NS, and Stroma. SpinAdapt was not significantly outperformed by either Seurat or ComBat for any subtype (**Figure 2A-D**, **Supplementary Figure 2**, **Supplementary Tables 2 and 4, methods**).

Dataset integration, an RNA-homogenization task that requires access to sample-leve data, is commonly adopted for single-cell RNA homogenization. To test the tradeoff between privacy preservation and full data access, we compared SpinAapat to Seurat, Combat, Limma[6], and Scanorama[8] for integration of our bulk-RNA datasets. For high integration performance, we want to maximize dataset (batch) mixing while maintaining subtype-wise separability (no mixing of tumor subtypes) within integrated datasets (**methods, Supplementary Figures 3,4**).

To quantify batch mixing or/and subtype-wise separation, we compute silhouette score for each sample, and then report the average silhouette score across all integrated samples (**methods**, **Figure 2E, Supplementary Table 3**). Even though SpinAdapt did not have access to sample-level data when fitting, it significantly outperformed each of the other methods for colorectal cancer and breast cancer (P < 0e-5 and *P*<10e-34, respectively). For pancreatic cancer, SpinAdapt outperformed ComBat, Limma, and Scanorama (*P*<0.05 for each method). For bladder cancer, Scanorama outperformed SpinAdapt (*P* < 10e-5), whereas SpinAdapt outperformed ComBat and Limma (*P* < 10e-6) (**Figure 2E, Supplementary Table 5**). To analyze batch mixing and subtype-wise clustering independently, we also employ batch LISI (bLISI) and tissue LISI (tLISI) metrics to quantify mixing and subtype-wise segregation performance, respectively (**methods, Supplementary Figure 5**).

Finally, we analyzed the performance of batch correction algorithms on paired expression datasets. Paired datasets are ideal for comparing correction methods since any sample differences are known to be technical batch biases. First, we simulated batch effect between two datasets (source and target), corrected source to target, and then evaluated the performance of SpinAdapt, ComBat and Seurat in correcting the simulated source data (see results in **Supplementary Figure 6, Supplementary note**). Next, we analyzed a real-world 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. We adapted the source to target and quantified SpinAdapt, Seurat, and Combat's performance in constructing

embeddings, correcting gene expression profiles, and transferring multiple linear regression models across datasets (**methods**).

RNA-expression similarity between the paired RNA-seq and uncorrected microarray datasets showed low concordance. All batch correction methods, correcting the microarray data to paired RNA-seq data, achieved improved performance (**Figure 3A, Supplementary Figure 7B**). Corrected source basis and embeddings of the microarray data showed high concordance to that of the RNA-seq, using SpinAdapt, compared with uncorrected source basis and embeddings (**Figure 3A, Supplementary Figure 7A**). Differences between methods in terms of gene-wise expression error between target and source-corrected data were quantified with a t-test. SpinAdapt outperformed Seurat ($P$ < 10e-11) and Combat ($P$ < 10e-11) (**methods**, **Figure 3B, Supplementary figure 7B**). Finally, we validated the transfer of a wider range of functions across the paired datasets. We synthetically generated 500 randomly sparse linear models and evaluated them on target and source-corrected data for increasing sparsity levels. SpinAdapt outperformed Seurat ($P$ < 10e-27) and Combat ($P$ < 10e-27) in predicting the target model outcomes from source-corrected data (two-sided t-test, **Figure 3C**).

**Discussion**

To achieve the reported performance, the aforementioned correction and integration methods require sharing patient-level data. However, since privacy-preserving laboratories cannot share patient-level data, they may require a common trusted broker to have 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 [13]. Similar trust limitations still exist between healthcare organizations that actively limit data sharing due to privacy and security concerns.

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. Even though there is an inherent tradeoff between performance and privacy, SpinAdapt shows state of the art performance both in prediction and integration tasks, outperforming similar algorithms that require access to private sample-level data for correction of technical dataset biases. By only sharing data factors that preserve privacy alongside a given RNA model, SpinAdapt allows external validation and reuse of pre-trained RNA models on novel datasets, hence improving research reproducibility across multiple laboratories.

**Authors' Contributions**

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

**Competing interests**

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

**Acknowledgements**

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

**Bibliography**

1. Leek, J. T. *et al.* Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat. Rev. Genet.* **11**, 733–739 (2010).
2. Johnson, W. E., Li, C. & Rabinovic, A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* **8**, 118–127 (2007).
3. Leek, J. T. svaseq: removing batch effects and other unwanted noise from sequencing data. *Nucleic Acids Res.* **42**, (2014).
4. Tran, H. T. N. *et al.* A benchmark of batch-effect correction methods for single-cell RNA sequencing data. *Genome Biol.* **21**, 12 (2020).
5. Hoadley, K. A., Yau, C., Wolf, D. M. & Cherniack, A. D. Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin. *Cell* (2014).
6. Ritchie, M. E. *et al.* limma powers differential expression analyses for RNA-sequencing and

microarray studies. *Nucleic Acids Res.* **43**, e47 (2015).

7. Haghverdi, L., Lun, A. T. L., Morgan, M. D. & Marioni, J. C. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat. Biotechnol.* **36**, 421–427 (2018).
8. Hie, B., Bryson, B. & Berger, B. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. *Nat. Biotechnol.* **37**, 685–691 (2019).
9. Lotfollahi, M., Wolf, F. A. & Theis, F. J. scGen predicts single-cell perturbation responses. *Nat. Methods* **16**, 715–721 (2019).
10. Lin, Y. *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.* **116**, 9775–9784 (2019).
11. Korsunsky, I. *et al.* Fast, sensitive and accurate integration of single-cell data with Harmony. *Nat. Methods* **16**, 1289–1296 (2019).
12. Stuart, T. *et al.* Comprehensive Integration of Single-Cell Data. *Cell* **177**, 1888–1902.e21 (2019).
13. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Manubot* (2019).
14. Duncan, G. T. & Pearson, R. W. Enhancing Access to Microdata While Protecting Confidentiality: Prospects for the Future. *SSO Schweiz. Monatsschr. Zahnheilkd.* **6**, 219–232 (1991).
15. Zhou, S., Lafferty, J. & Wasserman, L. Compressed and Privacy-Sensitive Sparse Regression. *IEEE Trans. Inf. Theory* **55**, 846–866 (2009).
16. Kamoun, A. *et al.* A Consensus Molecular Classification of Muscle-invasive Bladder

Cancer. *Eur. Urol.* **77**, 420–433 (2020).

17. Hoadley, K. A. *et al.* Cell-of-Origin Patterns Dominate the Molecular Classification of 10,000 Tumors from 33 Types of Cancer. *Cell* **173**, 291–304.e6 (2018).
18. Saal, L. H. *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.* **7**, 20 (2015).
19. Zhao, L., Zhao, H. & Yan, H. Gene expression profiling of 1200 pancreatic ductal adenocarcinoma reveals novel subtypes. *BMC Cancer* **18**, 603 (2018).
20. Jorissen, R. N. *et al.* Metastasis-Associated Gene Expression Changes Predict Poor Outcomes in Patients with Dukes Stage B and C Colorectal Cancer. *Clin. Cancer Res.* **15**, 7642–7651 (2009).
21. Buechler, S. A. *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.* **10**, 12123 (2020).
22. Prat, A., Parker, J. S., Fan, C. & Perou, C. M. PAM50 assay and the three-gene model for identifying the major and clinically relevant molecular subtypes of breast cancer. *Breast Cancer Res. Treat.* **135**, 301–306 (2012).
23. Guinney, J. *et al.* The consensus molecular subtypes of colorectal cancer. *Nat. Med.* **21**, 1350–1356 (2015).
24. Bailey, P. *et al.* Genomic analyses identify molecular subtypes of pancreatic cancer. *Nature* **531**, 47–52 (2016).
25. Dietterich, T. G. Approximate Statistical Tests for Comparing Supervised Classification

Learning Algorithms. *Neural Comput.* **10**, 1895–1923 (1998).

26. Rousseeuw, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987).
27. Rish, I. & Grabarnik, G. *Sparse Modeling: Theory, Algorithms, and Applications*. (CRC Press, 2014).

## Main Figures

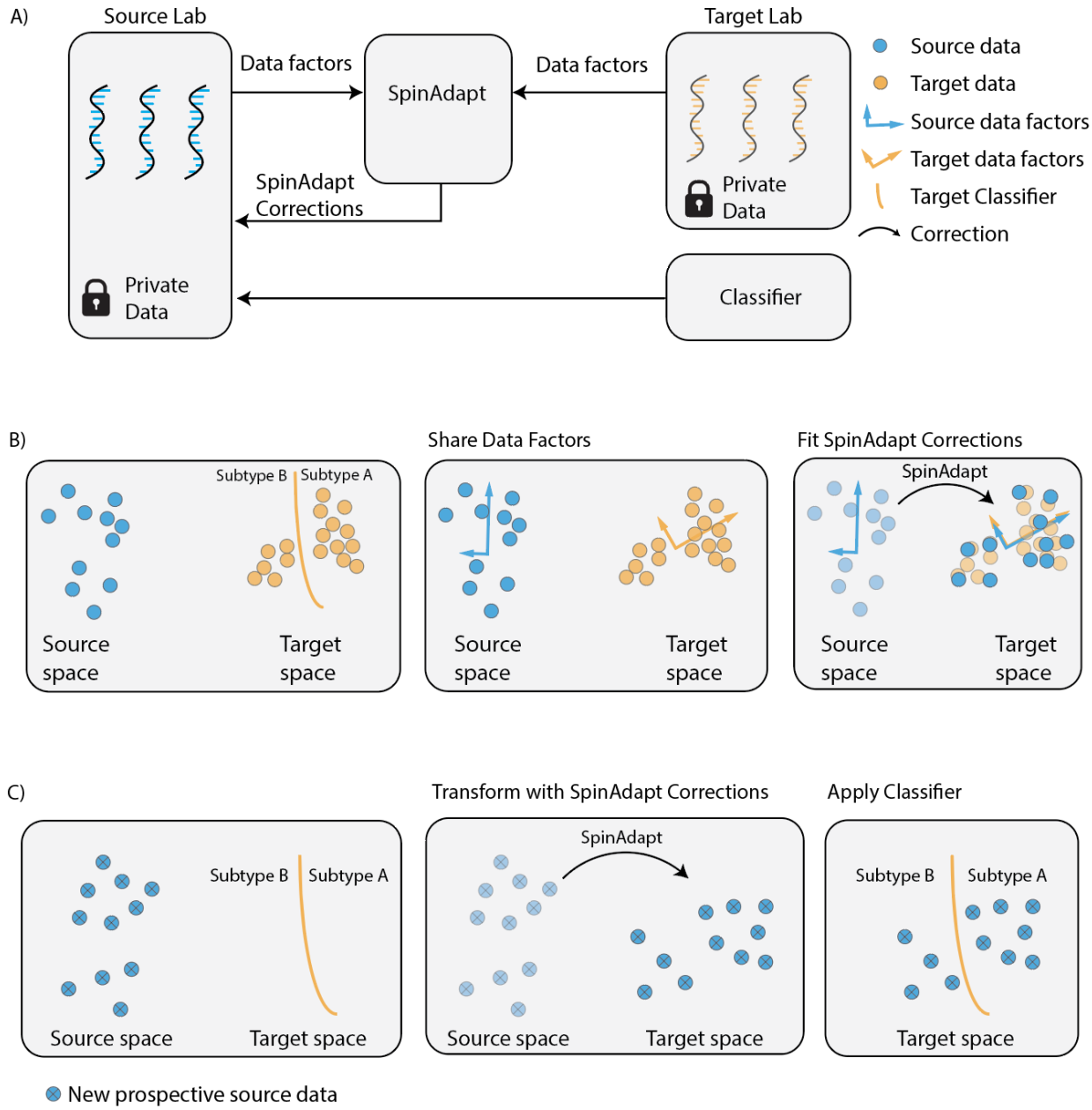


**Figure 1.** Privacy-preserving transfer of molecular models between a target lab and a source lab. **A**) A target dataset with a trained classifier and protected RNA data provides its privacy-preserving RNA factors and a molecular classifier to SpinAdapt. A source dataset used for validation provides 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.

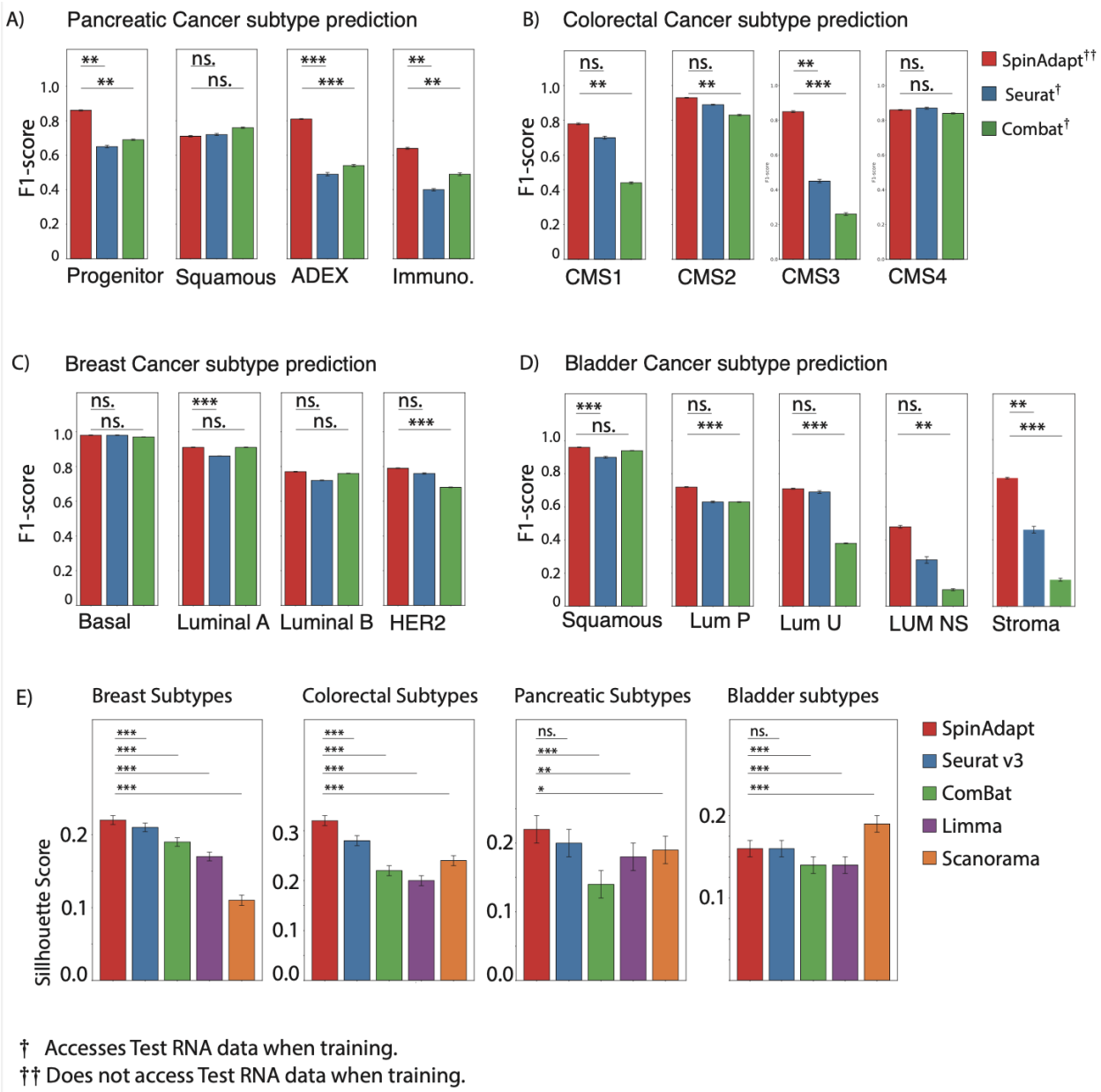



**Figure 2. A-D**) Subtype prediction performance on held-out source data. We train subtype predictors on target data and evaluate them on source data. Source data is split into two disjoint subsets such that the correction model is trained on one subset and the predictor performance is evaluated on the other held-out subset. Seurat and ComBat do not support a fit-transform paradigm, and therefore they require access to predictor held-out evaluation data to learn the correction model. 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**). **E**) Integration of source data to target

data. Quantification of integration performance using silhouette score finds SpinAdapt to provide significantly better integrations in breast, colorectal, and pancreatic cancer datasets. On bladder cancer dataset, Scanorama significantly outperforms SpinAdapt, but SpinAdapt significantly outperforms ComBat and Limma. Significance testing by two-sided paired Wilcoxon test (**methods**). (ns: $P \geq 0.05$, $*P < 0.05$, $**P < 0.01$, $***P < 0.001$). Statistical significance is defined at $P < 0.05$.

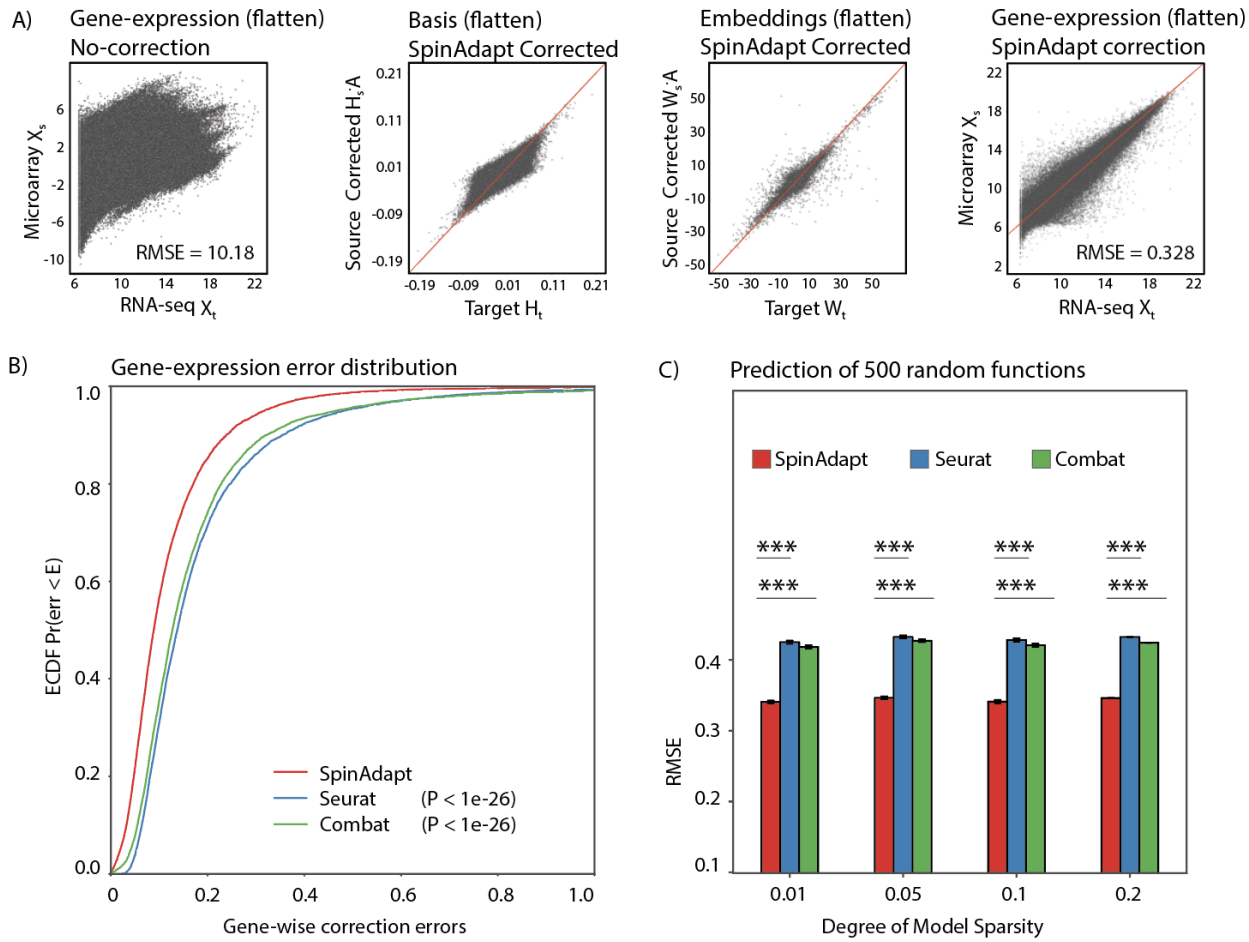
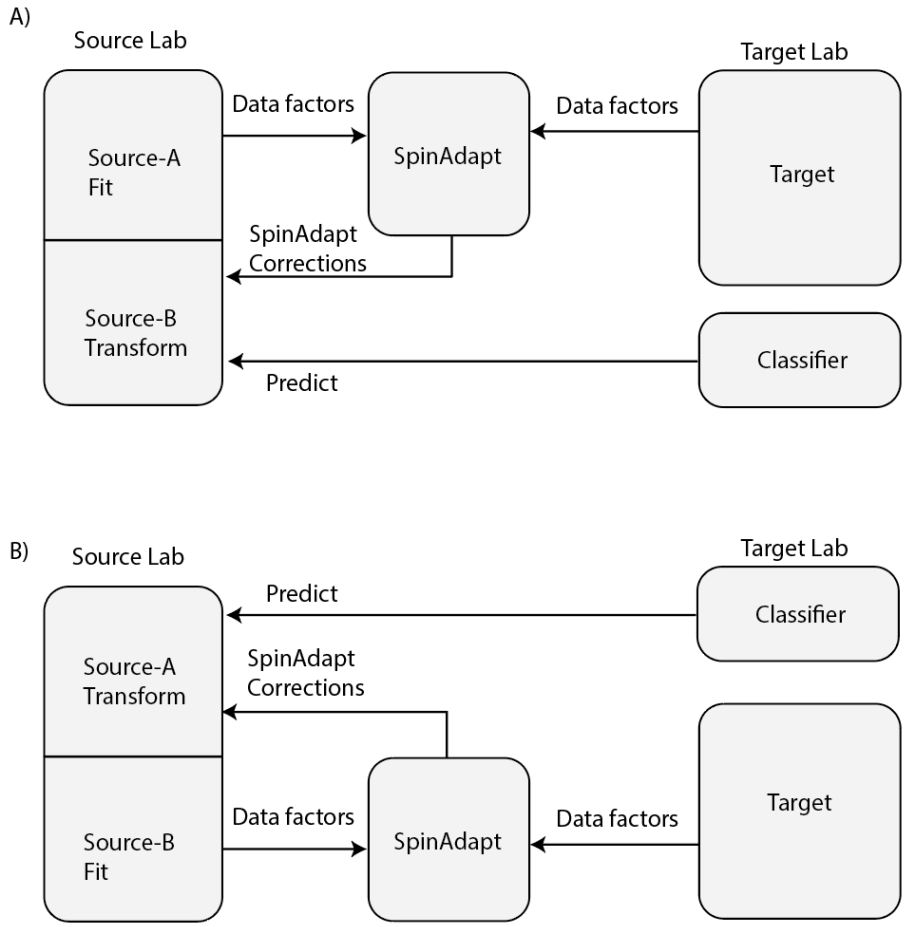


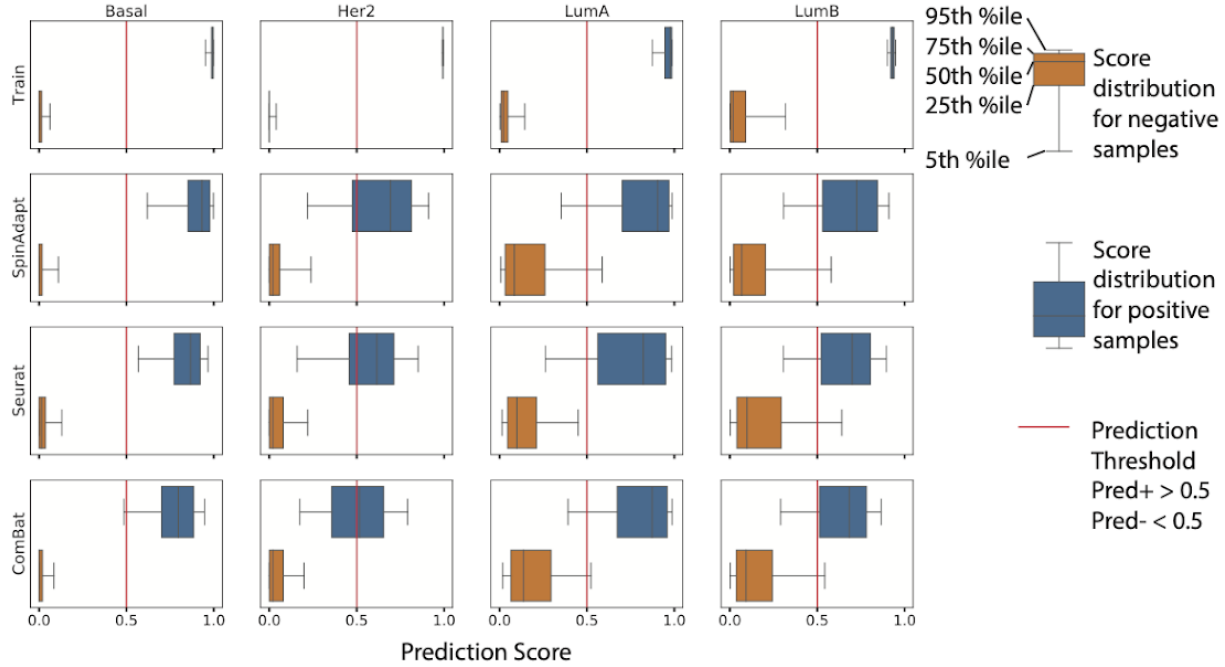
**Figure 3.** Batch correction performance on paired 481 TCGA-BRCA patients profiled with RNA-seq (target) and microarray (source). **A**) Scatter plots of paired expression values in target with uncorrected source, target basis with corrected source basis, target embeddings with corrected source embeddings, paired expression values in target and corrected source, where corrections are performed using SpinAdapt. **B**) Scatter plot of gene-wise mean square estimation error between paired expression values in target and corrected source, for each correction method. SpinAdapt significantly outperforms Seurat, and ComBat (t-test, $P < 1e-10$, **Supplementary Figure 7C** for all genes). **C**) Performance of 500 randomly generated sparse linear models that were evaluated on RNA-seq and then on corrected microarray data. The experiment is repeated for each method across varying sparsity levels, and SpinAdapt significantly outperforms Seurat and ComBat for each sparsity level (***$P < 0.001$, t-test).

## Supplementary figures

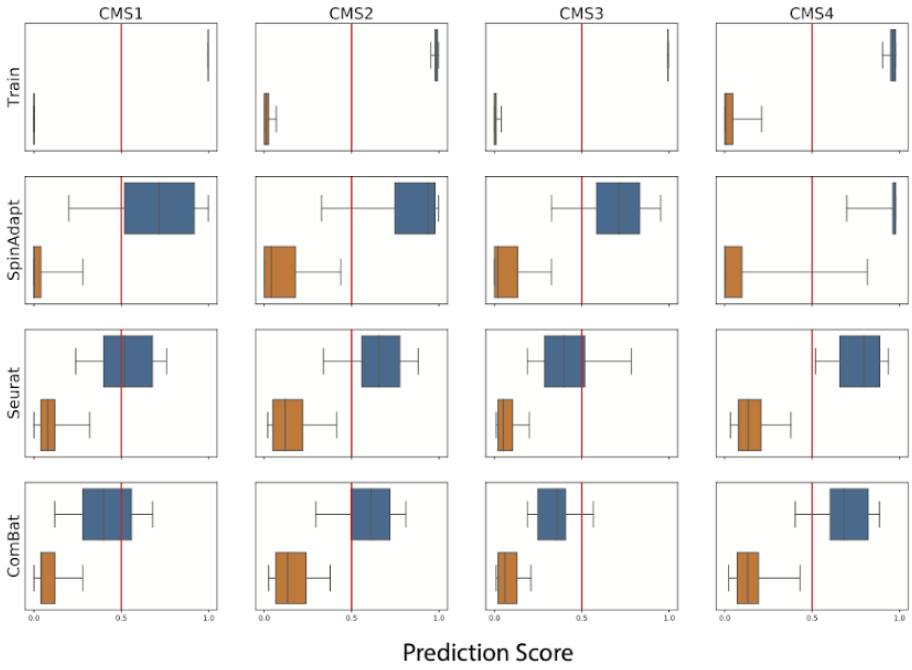



**Supplementary Figure 1.** Experimental framework for the validation of the target-trained molecular predictors on batch-corrected source dataset, such that the correction model never trains on the test data for the predictor.

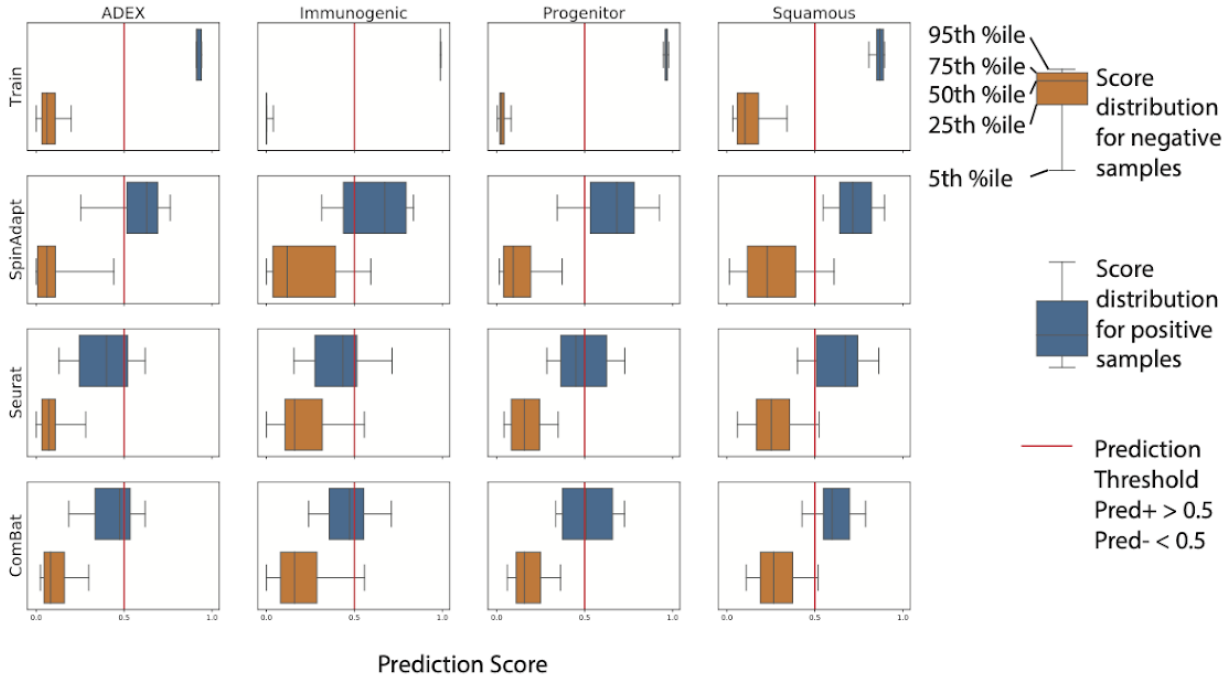
A) BRCA


Basal
Her2
LumA
LumB
Train
SpinAdapt
Seurat
ComBat
0.0
0.5
1.0
Prediction Score
95th %ile
75th %ile
50th %ile
25th %ile
5th %ile
Score distribution for negative samples
Score distribution for positive samples
Prediction Threshold
Pred+ > 0.5
Pred- < 0.5

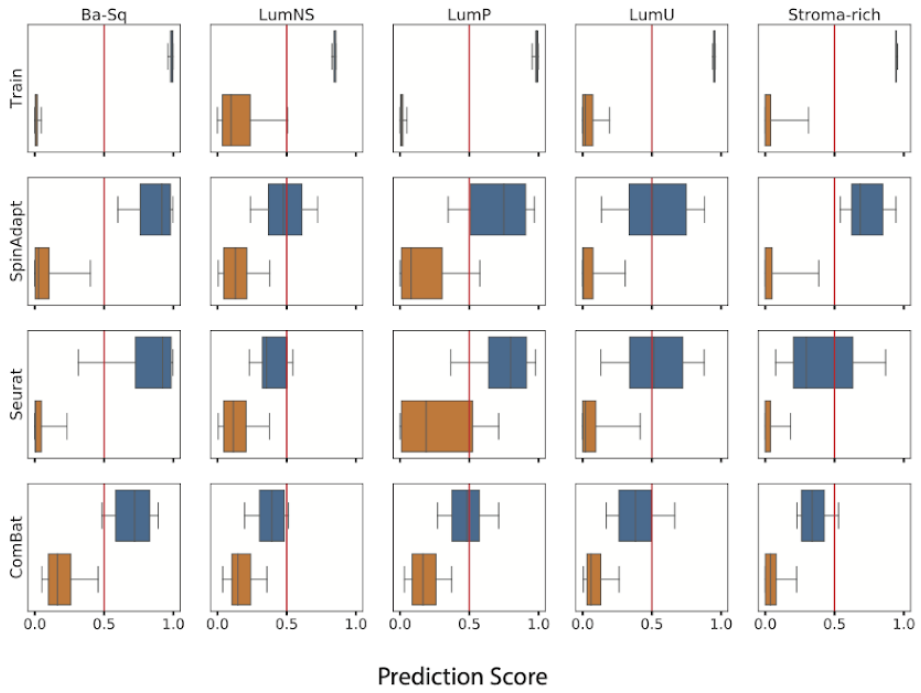
B) CRC


CMS1
CMS2
CMS3
CMS4
Train
SpinAdapt
Seurat
ComBat
Prediction Score

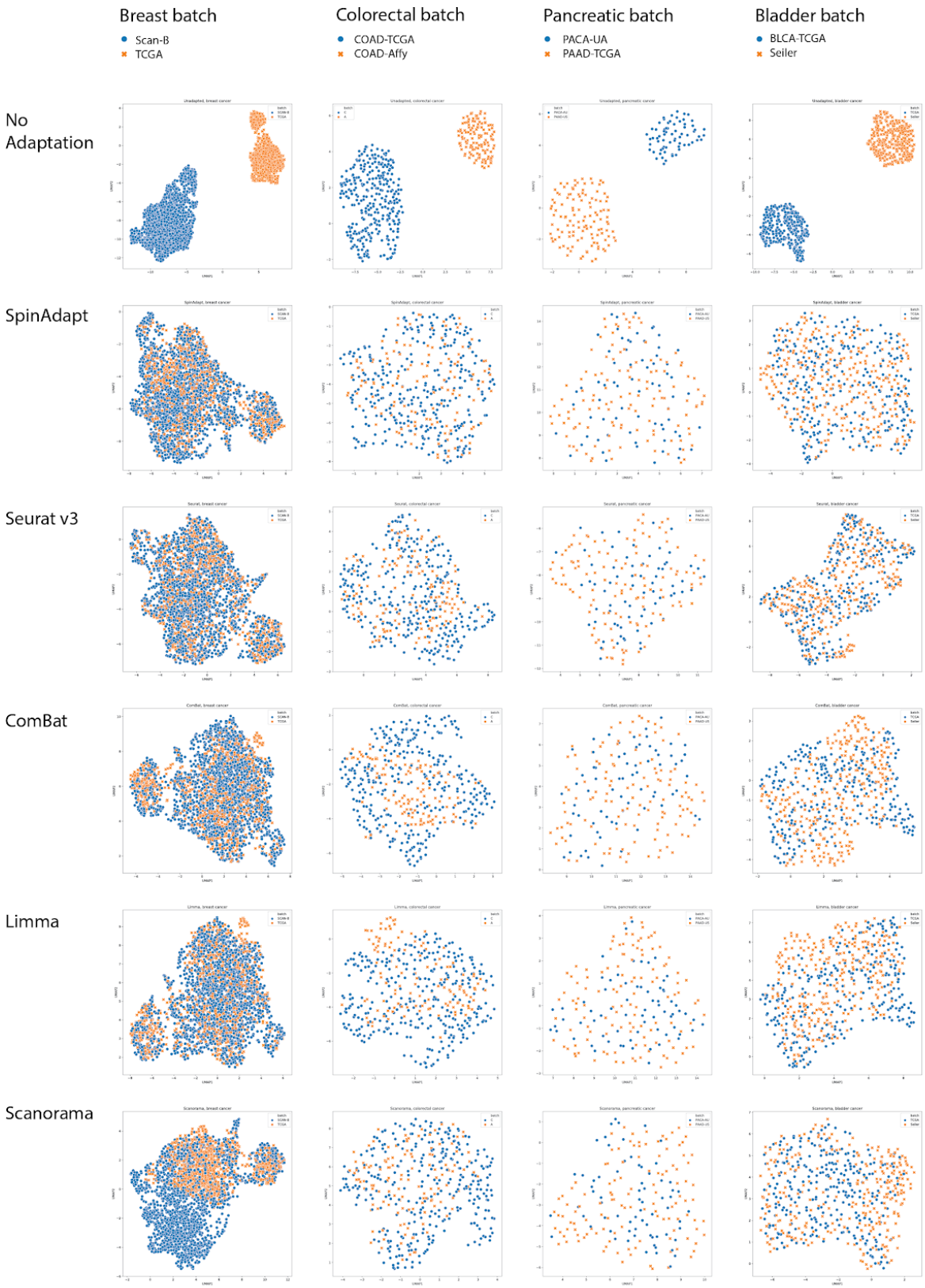
C) PANC




D) Bladder



**Supplementary Figure 2.** Boxplots for predictor scores on batch-corrected source dataset for each cancer subtype. For each cancer subtype (column) and correction method (row), the boxplots for positive and control samples are plotted separately, and the vertical line represents the decision threshold at 0.5 (**methods**). Better performance is achieved when control and positive sample score distributions are shifted to the left and right, respectively. **A**) Breast: SpinAdapt obtains higher median test scores on the positive samples, **B**) Colorectal: the SpinAdapt test score distributions for positive samples are shifted to the right, but CMS4 subtype suffers from lower specificity, **C**) Pancreatic: the SpinAdapt test score distributions for positive samples are shifted to the right, compared with Seurat and ComBat. **D**) Bladder: SpinAdapt obtains lower median test scores on the control samples for LumP subtype, while obtaining higher median test scores on the positive samples for the LumNS and Stroma-rich subtypes, compared with Seurat and ComBat.

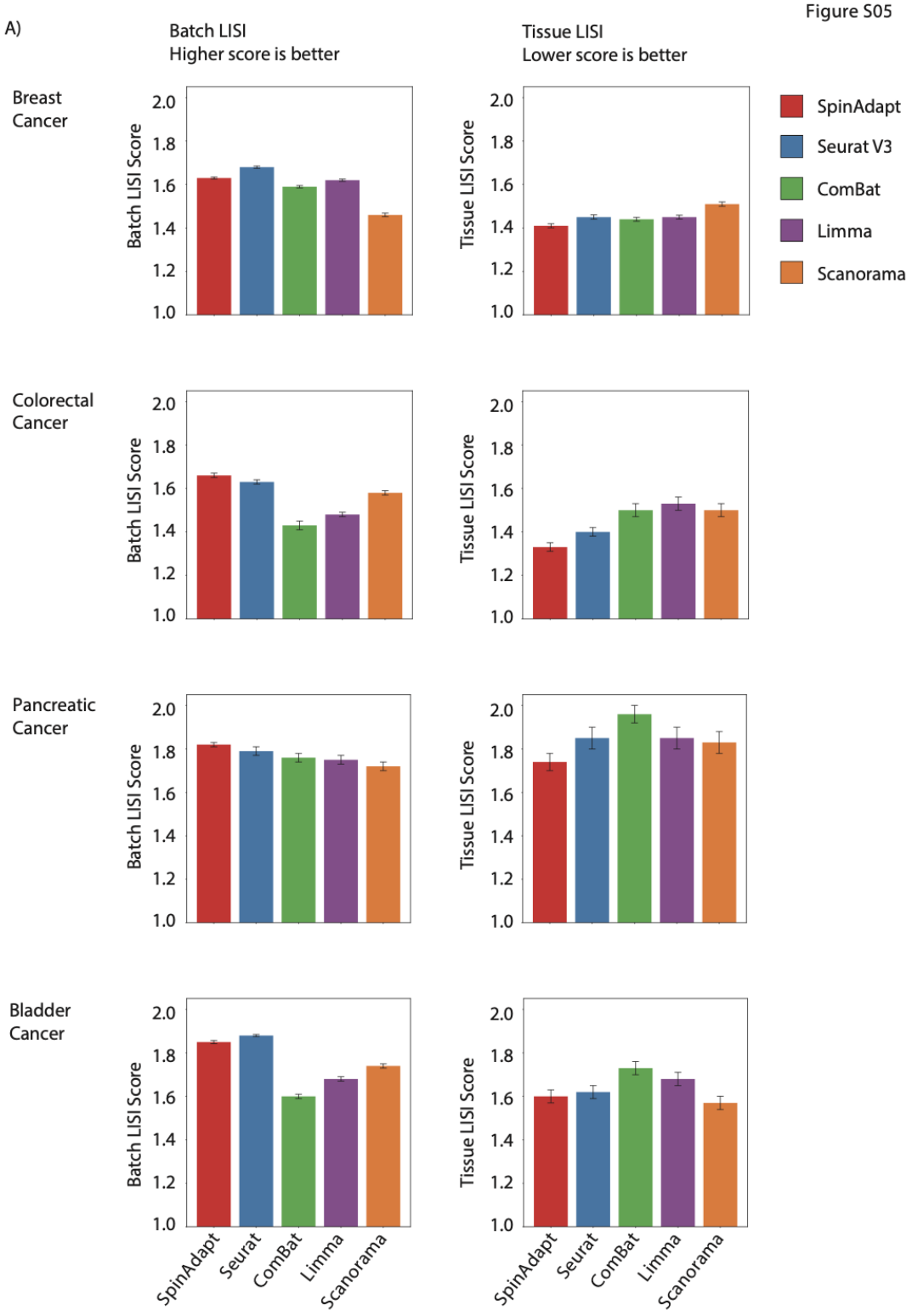

Breast batch
Scan-B
TCGA
Colorectal batch
COAD-TCGA
COAD-Affy
Pancreatic batch
PACA-UA
PAAD-TCGA
Bladder batch
BLCA-TCGA
Seiler
No Adaptation
SpinAdapt
Seurat v3
ComBat
Limma
Scanorama

**Supplementary Figure 3. A**) UMAP plots for dataset integration, labeling samples by dataset. Top panel shows dataset-based clustering in each cancer dataset before integration. Integration requires good batch mixing within integrated datasets, which is achieved by most methods. ComBat, Limma, Scanorama are outperformed by Seurat, SpinAdapt in terms of batch mixing in Colorectal, while Scanorama achieves poor batch mixing in Breast.

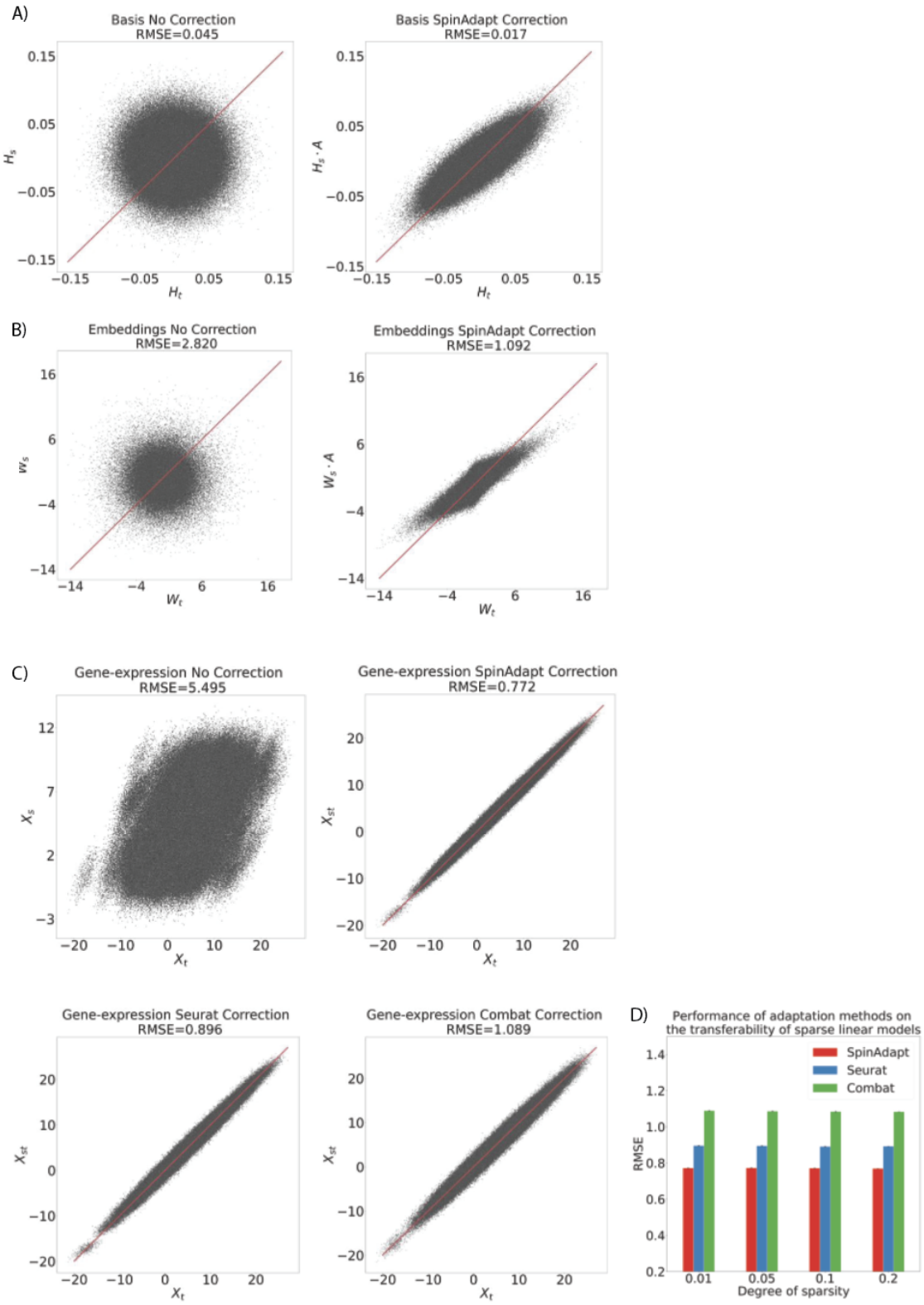
A)
Breast Subtypes
Luminal A
Luminal B
Basal
HER2
Batch
Scan-B
TCGA
Colorectal Subtypes
CMS1
CMS2
CMS3
CMS4
Batch
COAD-TCGA
COAD-Affy
Pancreatic Subtypes
ADEX
Immunogenic
Squamous
Progenitor
Batch
PACA-UA
PAAD-TCGA
Bladder Subtypes
Stroma
Ba-Sq
Lum. P
Lum. U
Lum. NS
Batch
BLCA-TCGA
Seiler
No Adaptation
SpinAdapt
Seurat v3
ComBat
Limma
Scanorama

**Supplementary Figure 4. A**) UMAP plots for dataset integration, labeling samples by cancer subtype. Top panel shows cancer subtypes in each dataset before correction. Subtype homogeneity is apparent in the majority of integration tasks regardless of library size. Subtype mixing is visible in regions where multiple subtypes cluster together. Subtype mixing was observed before and after correction in Breast between luminal subtypes, in Colorectal between CMS 2 and 4, in Pancreatic between ADEX and immunogenic, and in Bladder between luminal subtypes.

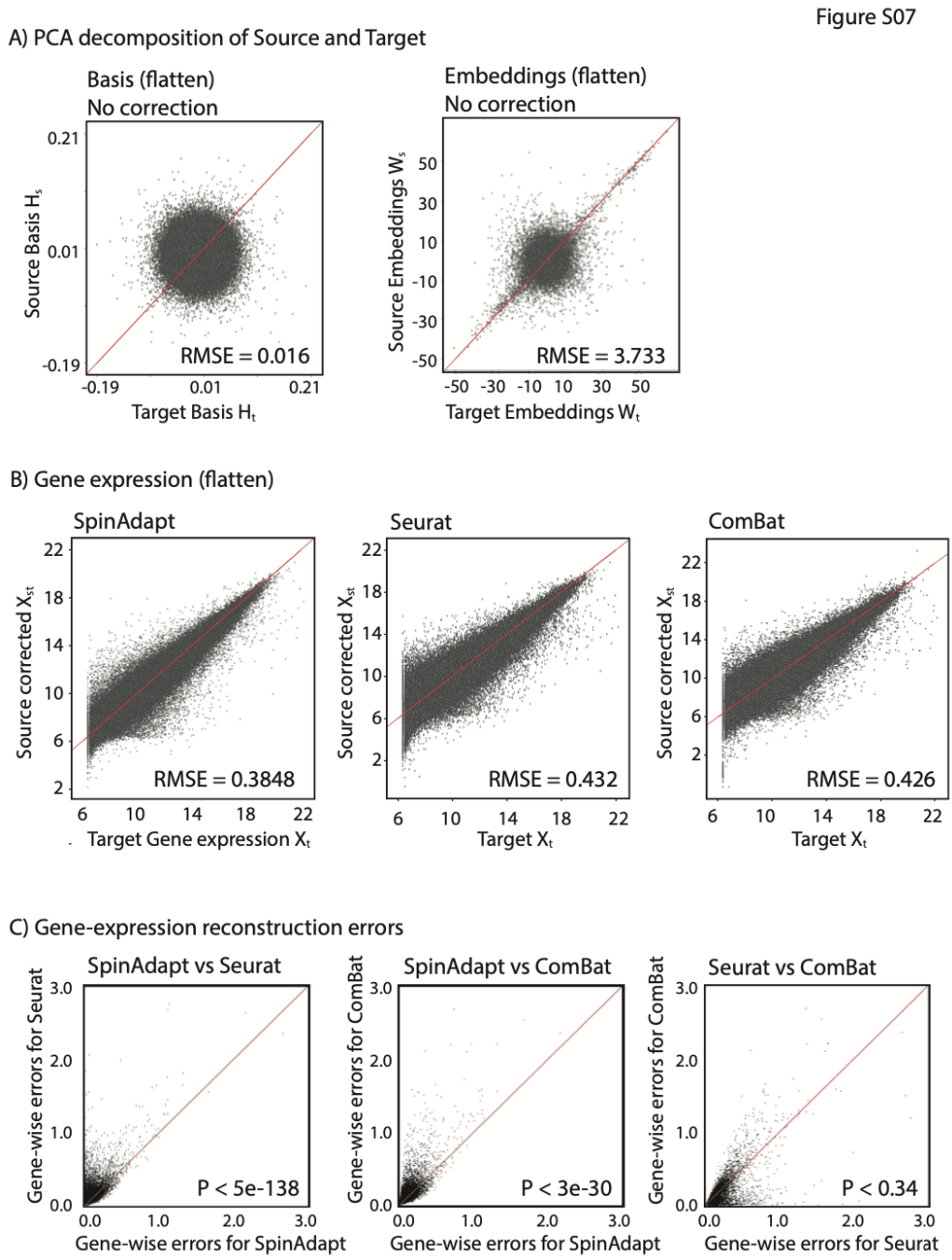
Figure S05


A)
Batch LISI
Higher score is better
Tissue LISI
Lower score is better
Breast Cancer
Colorectal Cancer
Pancreatic Cancer
Bladder Cancer
Batch LISI Score
Tissue LISI Score
2.0
1.8
1.6
1.4
1.2
1.0
SpinAdapt
Seurat V3
ComBat
Limma
Scanorama
SpinAdapt
Seurat
ComBat
Limma
Scanorama

**Supplementary Figure 5. A**) Quantification of dataset integration performance using batch LISI (bLISI) and tissue LISI (tLISI) metrics, where bLISI measures batch homogeneity (a higher score is better) and tLISI measures subtype heterogeneity in local sample neighborhoods (a lower score is better). For each dataset, correction method, and performance metric, the associated barplot reports the mean and standard error over all the integrated samples.


A)
Basis No Correction
RMSE=0.045
Basis SpinAdapt Correction
RMSE=0.017
B)
Embeddings No Correction
RMSE=2.820
Embeddings SpinAdapt Correction
RMSE=1.092
C)
Gene-expression No Correction
RMSE=5.495
Gene-expression SpinAdapt Correction
RMSE=0.772
Gene-expression Seurat Correction
RMSE=0.896
Gene-expression Combat Correction
RMSE=1.089
D)
Performance of adaptation methods on the transferability of sparse linear models
SpinAdapt
Seurat
Combat
RMSE
Degree of sparsity

**Supplementary Figure 6.** Paired simulated experiment. **A**) Scatter plots of the target basis and uncorrected source basis (RMSE = 0.045), and the target basis and the learned SpinAdapt corrected source basis (RMSE = 0.017). **B**) Scatter plots of target embedding and the uncorrected source embedding (RMSE = 2.82), and target embedding and SpinAdapt corrected source embedding (RMSE = 1.092). **C**) Scatter plots of paired expression values in the target and: un-corrected source (RMSE = 5.5) SpinAdapt corrected source (RMSE = 0.77), Seurat corrected source for Seurat (RMSE = 0.89), and Combat corrected source (RMSE = 1.09). **D**) Performance of 500 randomly generated sparse linear models evaluated on simulated target and on corrected source. Experiment repeated for each method across varying sparsity levels. SpinAdapt outperforms Seurat and ComBat for each sparsity level based on RMSE.




**Supplementary Figure 7.** Paired breast data experiment. **A**) RNA basis and embeddings (flattened) before and after correction from microarray to RNA-seq. **B**) RNA expression correction quantification. RNA-seq (target) on X axis, microarray (source) on Y-axis. SpinAdapt outperforms Seurat and Combat (RMSE = 0.385; 0.432; 0.426 respectively). **C**) Scatter plot of gene-wise mean square estimation error between paired expression values in target and corrected source. SpinAdapt significantly outperforms Seurat, and ComBat (t-test, P < 5e-138; P < 3e-30 respectively). Seurat and Combat are not significantly different (P < 0.34).

## Methods

### Datasets

Gene expression dataset pairs are generated across various cancer types (bladder, breast, colorectal, pancreatic) using microarray platforms and RNA-sequencing (**Table 1**). Bladder cancer datasets are generated using the Seiler and TCGA cohorts [16], breast cancer datasets are generated using the TCGA-BRCA [17] and SCAN-B cohorts [18], pancreatic cancer datasets are generated using the Bailey and TCGA cohorts [19]. Colorectal cancer datasets are generated using GSE14333 [20] and TCGA-COAD datasets, which are subsets of cohorts A and C, respectively, used in validation of the ColoType predictor [21].

The subtype labels for patients across various cancer types are generated using well-accepted subtype annotations (**Table 1**). Bladder cancer subtypes are labeled as luminal papillary (LumP), luminal nonspecified (LumNS), luminal unstable (LumU), stroma-rich, basal/squamous (Ba/Sq), and neuroendocrine-like (NE-like), generated using a consensus subtyping approach [16]. Breast cancer subtypes are labeled as Luminal A (LumA), Luminal B (LumB), HER2-enriched (Her2), and Basal-like (Basal) [22]. Colorectal cancer subtypes are labeled as CMS1, CMS2, CMS3, CMS4, as published by Colorectal Cancer Subtyping Consortium (CRCSC) [23]. Pancreatic cancer subtypes are generated using expression analysis, labeled as squamous, pancreatic progenitor, immunogenic, and aberrantly differentiated endocrine exocrine (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 5 patients in any cancer cohort are removed. For the microarray expression datasets (GSE14333 and Seiler, see **Table 1**), multiple probe sets may map to the same gene. The expression values were averaged across such probe sets to get expression value for the corresponding gene. Furthermore, for each cancer type, we remove genes with zero variance, and we only keep genes common between source and target, sorted in alphabetical order. Finally, we normalized the RNA-Seq datasets using the variance stabilizing transform (VST) 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 less than 200, where the default value of k.filter does not work. Therefore, when the number of samples in either dataset is less than 200, we set the k.filter parameter to 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.

**Parameters for SpinAdapt**

For a given pair of source and target datasets, let $p$ be the number of genes, $n_s$ be the number of samples in the source dataset, and $n_t$ be the number of samples in the target dataset. Across all experiments performed in this study, the parameters in SpinAdapt (Algorithm 1) are set as follows. We set $\alpha = 0.01$ and $\lambda = (2/3) * min(n_s, n_t)$. When transferring classifiers from target to source dataset, we set $variance_{norm} = True$ when source is microarray and $variance_{norm} = False$ when source is RNA-Seq data. Finally, when integrating source and target datasets, we set $variance_{norm} = True$.

**Evaluation approaches for prediction experiments**

For each of the 17 cancer subtypes (see **Table 1**), we train a one-vs-rest random forest classifier on the target dataset, such that the classifier learns to discriminate the selected subtype against all other subtypes present in the target. For each selected subtype, 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 three-fold cross-validation experiment on the target dataset.

We evaluate batch correction (adaptation) methods for application of the target-trained classifiers on the source dataset. The batch correctors are required to adapt the source dataset to the target reference. For unbiased evaluation of a batch correction method, the transform and training sets for the correction method need to be disjoint, so that the correction model does not train on the classifier test set during training step (fit). We propose a framework for validating transfer of classifiers across datasets, such that there is no information leakage from classifier test set to adaptation training data, as follows.

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). Second, the target classifier generates predictions on corrected source-B (**Supplementary Figure 1A**). Third, 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**). Finally, the classification performance is quantified by computing F-1 scores for all samples in the corrected source-A and source-B subsets. We repeated the entire procedure 30 times, choosing a new random partition of source dataset in each iteration, and we reported the mean F-1 score for each subtype.

We evaluate SpinAdapt using the aforementioned framework for validating transfer of classifiers. We also validate classifier transfer performance for Seurat and ComBat. However, since these correction methods have currently not implemented a transformation method for out-of-sample data, they can only correct samples included in the training data. Therefore, we trained each correction model on the evaluation set, and then applied the target-trained classifier on the corrected evaluation 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 repeated the procedure 30 times, using the same source data splits as used for SpinAdapt validation, which enabled pairwise 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 filter the positive samples and test them for classification consistency. Each positive sample is assigned a correct or incorrect classification label for each correction method. Then, for each pair of correction methods, the McNemar test statistic was evaluated on positive samples that were correctly classified by one method while being misclassified by the other method. In other words, the test statistic is evaluated on the disagreements between correction methods on positive samples, for each of the 17 cancer subtypes. We report the median *P*-value across the 30 repetitions of the experimental framework, as explained above, to compare methods across all iterations for each subtype (**Supplementary Table 4**).

**Evaluation approaches for integration experiments**

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 maximize batch mixing while minimizing subtype mixing.

To compare SpinAdapt with other batch integration methods, we assess the goodness of batch mixing and tissue type separation. First, we employed the average silhouette width (ASW) to quantify batch mixing and tissue segregation. The silhouette score of a given 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 average silhouette width implies that samples for each tissue label are tightly packed together or/and the various tissue types are well-separated from each other.

To explicitly quantify batch mixing and tissue segregation, independently, we employ the local inverse Simpson's index (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 batch LISI (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 tissue LISI (tLISI), since it measures tissue type diversity in the neighborhood of each sample. 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)**. When comparing methods using average bLISI, which measures dataset mixing, Seurat outperforms SpinAdapt on breast and bladder cancer datasets ($P$ < 10e-3), whereas SpinAdapt outperforms ComBat, Limma, and Scanorama on colorectal and pancreatic cancer datasets ($P$ < 10e-3) (**Supplementary Tables 3 and 5**). When comparing methods using average tLISI, SpinAdapt significantly outperforms all other methods on breast ($P$ < 10e-13), colorectal ($P$ < 10e-7), and pancreatic ($P$ < 0.05) cancer datasets (**Supplementary Tables 3 and 5**), implying SpinAdapt best preserves molecular structures in integrated datasets for these cancer types. When comparing methods using tLISI on the bladder cancer dataset, SpinAdapt outperforms Seurat, ComBat, Limma (P < 10e-6), whereas the performance difference with Scanorama was insignificant. We also visualize the dataset integration performance using UMAP embeddings (**methods**), annotating integrated samples by batch (**Supplementary Figure 3**) and subtype membership (**Supplementary Figure 4**).

In this study, the integration metrics including silhouette, bLISI, and tLISI scores are computed on the UMAP embeddings of the integrated datasets for each cancer type (**Table 1**). Specifically, these scores are computed on the first 50 components of the UMAP transform for each cancer type, where the UMAP embeddings are computed using default parameters of the package. The average silhouette width, bLISI, and tLISI scores are reported along with the standard errors (**Supplementary Table 3**). For each metric, significance testing between methods is performed by a two-sided paired Wilcoxon test (**Supplementary Table 5**).

**Visualization**

We employ the UMAP transform to visualize the batch integration results for each cancer type (**Supplementary Figure 3**,**4**). Specifically, we perform visualization 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, for computation of the UMAP transform for each cancer type.

**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 train 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 $X_s$. Within Algorithm 1, each of Steps 1 and 2 are executed using Algorithm 2, whereas Steps 3 and 4 are executed using Algorithms 3 and 4, respectively.

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 2-3, 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 $m_s$, $m_t$, $s_s$, $s_t$ 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 3-4, Algorithm 1). The usage of statistics $s_s$ and $s_t$ in Step 4, Algorithm 1 is optional depending on the boolean value of $variance_{norm}$, 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 requires access to the PCA basis of source and target datasets, which cannot be used for recovery of sample-level patient data (**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 $X_t$.

The main idea of SpinAdapt is to learn a low matrix-rank transformation between latent space representations of source and target datasets, respectively. Since the transformation is learned between low-dimensional representations, the algorithm evades the typical sample and computational requirements of explicitly learning gene-wise corrections between datasets. As follows from Steps 1-2 in Algorithm 1, we choose PCA basis as the latent space representations. The Step 3 in Algorithm 1 learns a transformation between the PCA basis of the source and target datasets, respectively. In other words, the algorithm matches the eigenvectors of the sample covariance matrix of the source dataset with those of the target dataset. Therefore, in contrast to many prior works that perform independent gene-wise matching, the algorithm matches covariance between gene-pairs.

**Algorithm 1** SpinAdapt algorithm

function Main($X_t$, $X_s$, $X_{sh}(optional)$, $F_t$, $\alpha$, $\lambda$, $variance_{norm}$)

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:

$$X_{sc} \leftarrow \text{SpinAdapt_Transform}(X_s,\ X_{sh}(optional),\ A,\ U_s,\ m_s,\ s_s,\ U_t,\ m_t,\ s_t,\ variance_{norm}).$$

5. Apply target-trained predictor $F_t$:

$$y_{sc} \leftarrow F_t(X_{sc}).$$

6. Return predictions $y_{sc}$.

**Algorithm 2** Compute data factors

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 sample mean vector.

$$m_e \leftarrow \frac{1}{n_e} \sum_{i=1}^{n_e} X_{e,i}$$

3. Compute sample variance vector.

$$s_e \leftarrow \frac{1}{n_e - 1} \sum_{i=1}^{n_e} (X_{e,i} - m_e)^2$$

4. Compute sample covariance matrix.

$$C_e \leftarrow \frac{1}{n_e - 1} (X_e - m_e)(X_e - m_e)^T$$

5. Compute eigenvectors of the sample covariance matrix.

$$U_e \leftarrow Eigenvectors(C_e)$$

6. Retain top $d_e$ eigenvectors.

$$U_e \leftarrow U_e[1{:}d_e]$$

7. Return $U_e$, $m_e$, $s_e$

**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 target latent space.

$X_s \in R^{p \times n_s}$ The train source dataset.
$X_t \in R^{p \times n_t}$ The train target dataset.
$X_{sh} \in R^{p \times n_t}$ The held-out source dataset.
$X_{s,i} \in R^p$ The i-th column of $X_s$.
$X_{t,i} \in R^p$ The i-th column of $X_t$.
$m_s \in R^p$ The sample gene-wise mean of source dataset.
$m_t \in R^p$ The sample gene-wise mean of target dataset.
$s_s \in R^p$ The sample gene-wise variance of source dataset.
$s_t \in R^p$ The sample gene-wise variance of target dataset.
$C_s \in R^{p \times p}$ The sample covariance of source dataset.
$C_t \in R^{p \times p}$ The sample covariance of target dataset.

$U_s \in R^{p \times d_s}$ Principal Component factors for source dataset.
$U_t \in R^{p \times d_t}$ Principal Component factors for target dataset.
$A \in R^{d_s \times d_t}$ Transformation matrix.
$X_{sc} \in R^{p \times n_s}$ 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 $v$.
$F_t$ Predictor trained on the target dataset.

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

**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 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.* 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

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

where A represents the transformation matrix, $\lambda$ represents the matrix-rank constraint, and 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 $i = 1, 2, ..., d_t$. Therefore, the intuition behind the main term is to approximate each target factor using some linear combination of source factors.

The inequality constraint in equation (1) is a matrix-rank penalization term, which restricts the solution space of A to matrices with matrix-rank less than $\lambda$. 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 [27]. 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.* Let g(A) = $\| U_s A - U_t \|_F$ , and let $S_\lambda = \{A : A \in R^{d_s \times d_t}, rank(A) \leq \lambda\}$ . Then, the objective function in equation (1) can be expressed as

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

If we ignore the constraint, the objective function reduces to minimization of $g(A)$, which is convex and differentiable, and thus it can be minimized on $R^{d_s \times d_t}$ using the gradient descent method. Given the gradient $\nabla g(A)$, an initial estimate $A^{(0)}$, and step size $\alpha$, we can minimize $g(A)$ using the iterative application of

$$A^{(k+1)} = A^{(k)} - \alpha \nabla g(A) \, , \tag{3}$$

for $k = 0, 1, 2, 3, \ldots$ till convergence. Let's consider an intuitive interpretation of this gradient descent step. Let $\tilde{g}(A^{(k+1)})$ be a quadratic approximation of $g(A^{(k+1)})$ from point $A^{(k)}$, such that the Hessian is replaced by an identity map and then scaled by $\frac{1}{\alpha}$ . Then, we have

$$\tilde{g}(A^{(k+1)}) \approx g(A^{(k)}) + \langle \nabla g(A^{(k)}), \; A^{(k+1)} - A^{(k)} \rangle + \frac{1}{2\alpha} \| A^{(k+1)} - A^{(k)} \|_F^2 \, . \tag{4}$$

Given a matrix $A^{(k)}$, we want to find another matrix $A^{(k+1)}$ such that $\tilde{g}(A^{(k+1)})$ is minimized. To find the optimal $A^{(k+1)}$, we solve $\nabla \tilde{g}(A^{(k+1)}) = 0$ and obtain $A^{(k+1)} = A^{(k)} - \alpha \nabla g(A)$, which is the gradient descent step in equation (3). Thus, the gradient descent step in equation (3) follows from the minimization of a quadratic approximation of the function $g$ at a point $A^{(k)}$.

Gradient descent can be used to minimize $g(A)$ because the function is convex and differentiable. In contrast, equation (2) cannot be evaluated using gradient descent, since the set of low-rank matrices $S_\lambda$ is non-convex. However, we note that the Euclidean projection onto the set $S_\lambda$ 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 $S_\lambda$ be denoted by $P_\lambda(A)$. Then, mathematically we have $P_\lambda(A) = argmin_Z \{\| A - Z \|_F : Z \in S_\lambda\}$. From the Eckart-Young Theorem, we know that $P_\lambda(A)$ can be efficiently evaluated by computing the top $\lambda$ 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 $U_\lambda$ contain the top $\lambda$ eigenvectors of $AA^T$ , columns of $V_\lambda$ contain the top $\lambda$ eigenvectors of $A^T A$, and entries of the diagonal matrix $\Sigma_\lambda$ are square roots of the top $\lambda$ eigenvalues of $AA^T$ . We are finally ready to present an algorithm for solving (1).

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

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

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

**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 $(X_s, X_{sh}(optional), A, U_s, m_s, s_s, U_t, m_t, s_t, variance_{norm})$

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 $variance_{norm}$ $is\ True$:

   For $i = 1,2,...,p$

   For $j = 1,2,...,n_s$

   $X_o(i,j) \leftarrow (\sqrt{s_t(i)}\,(X_o(i,j) - m_s(i)) / \sqrt{s_s(i)}) + m_s(i)$.

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 $X_{sc}$.

*Pseudocode for Transform (Algorithm 4)*. Finally, we present an 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 $variance_{norm}$ 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.

## Supplementary note

### Privacy

SpinAdapt is a batch correction method, which learns corrections between latent space representations that de-identify the sample-level information in each dataset. In this study, the PCA basis are chosen as the latent space representations for each dataset. SpinAdapt only requires access to data factors of each dataset for computation and application of dataset corrections, where the data factors comprise of the PCA basis, gene-wise means, and gene-wise variances of each dataset. Therefore, for the transfer of an RNA model from a training dataset to a validation dataset, only the data factors of the training dataset need to be transferred along the RNA model, thereby maintaining ownership of the training dataset.

Next, we show that the data factors are privacy-preserving, since an expression dataset cannot be recovered from the PCA basis of the dataset. Let $X$ be an expression matrix of size {genes x samples}, let the columns of a matrix $W$ contain the eigenvectors of $XX^T$, let the columns of a matrix $V$ contain the eigenvectors of $X^TX$, and let $\Sigma$ be a diagonal matrix such that the entries are square roots of the eigenvalues of $XX^T$. Note that $W$ and $V$ are PCA basis for the columns and rows of the expression matrix $X$, respectively. An SVD decomposition of the expression matrix $X$ can be written as $X = W \Sigma V^T$, which implies $W = X V \Sigma^{-1}$, since the columns of matrix $V$ are orthonormal and $\Sigma$ is a diagonal matrix. Let $H$ be another matrix $H = V \Sigma^{-1}$. Then, $W = X V \Sigma^{-1}$ can be rewritten as $W = X H$. From $W = X H$, it can be seen that the PCA basis matrix $W$ is obtained via an affine transformation $H$, which de-identifies the original expression matrix $X$. Such affine transformations are called matrix masks in the privacy literature 14. Since the matrix mask $H$ is kept private, while only the PCA basis matrix $W$ is made public, recovering original sample-level data $X$ from $W$ involves solving a highly underdetermined system 15. Hence, the affine transformation $H$ de-identifies the original data $X$, and the PCA factors $W$ are privacy-preserving representations of the expression data $X$.

**Simulation**

Let us set p = 1000 and n = 500, where p is number of genes and n is number of patients. To evaluate SpinAdapt, we simulated n source samples (denoted as $X_s$) from a p-dimensional multivariate normal distribution with mean $\mu_s$ and covariance matrix $\Sigma_s := CC'$, where $\mu_s$ is a 1000-dimensional vector of uniform(0,10) random variables, C' is the transpose of the matrix C, and C is a p-by-p matrix of standard normal random variables. To model the dataset bias, we generate a p-by-p matrix B, which is equal to the sum of an identity matrix and a p-by-p matrix of standard normal random variables. We generated target data $X_t$ using $X_t = X_s B + \varepsilon_t$, where $\varepsilon_t$ is a n-by-n matrix of normal random variables. The target data $X_t$ represents n instances of a p-dimensional multivariate normal distribution with mean $\mu_s B$ and covariance matrix $\Sigma_t = B'\Sigma_s B + 0.25 I_{1000}$ .

We train SpinAdapt on the PCA basis of Xs and Xt, which are Hs and Ht, respectively:

$$W_s, H_s = PCA(X_s)$$
$$W_t, H_t = PCA(X_t).$$

The basis correction (A is a linear basis change operator for SpinAdapt in this setting) to Hs approximates Ht with low error (RMSE = 0.017, **Supplementary Figure 6A**). We correct the source embeddings through Ws*A, which estimates Wt with RMSE = 1.092 (**Supplementary Figure 6B**). We evaluated the performance of SpinAdapt on correction of the simulated biased source dataset $X_s$, drawing comparisons with no correction, Seurat, and Combat. To compare the correction performance for each method, we computed the RMSE for each technique. All of the methods achieved lower error than no correction (**Supplementary Figure 6C**). SpinAdapt outperformed both Seurat and Combat with an RMSE = 0.772 vs RMSE = 0.896 and RMSE = 1.089, respectively.

Testing the transferability of an arbitrary function across target and source corrected data, we generated 500 sparse linear models, each consisting of a normalized p-dimensional coefficient vector of standard normal random variables. We repeat the experiment for sparsity levels of 1%, 5%, 10%, and 20%, such that 10, 50, 100, and 200 components of the coefficient vectors have non-zero elements, respectively. For each sparsity level, we denote this collection of models as $\left\{\beta^{(m)}\right\}_{m=1}^{500}$. For each sparsity level, we calculated the RMSE of the differences between observed target values ($y_i^{(m)} = x_{t_i}\beta^{(m)}$) and predicted values for the corrected source data ($\hat{y}_i^{(m)} = x_{st_i}\beta^{(m)}$), where $x_{st_i}$ represents source corrected dataset (**Supplementary Figure 6D**). SpinAdapt outperforms Seurat and ComBat in terms of RMSE, for any chosen level of sparsity.