

Supplementary Material

A Deep Learning Architecture for Combining and Imputing Heterogeneous Metabolomics Datasets

S1 RELATED WORK

Metabolomics data usually exhibits a combination of various kinds of missing data patterns [1]. For instance, a recent categorization considers three main pattern types, i.e., missing not at random (MNAR), missing at random (MAR), and missing completely at random (MCAR) [2]. In the context of metabolomics, “missing not at random” values are influenced by an unobserved factor, such as when metabolite signals fall below a particular instrument’s detection threshold. On the other hand, “missing at random” values are affected by observed factors, due to insufficient data processing or technical problems like overlapping peaks in chromatography-based analysis. This can lead to chemically similar metabolites having closely matching retention times, making their peaks indistinguishable and resulting in only one metabolite being detected in each sample. Finally, “missing completely at random” values exhibit a random distribution (i.e., the likelihood of being missing is consistent across all variables) and are affected by neither the observed nor unobserved missing values. In practice, however, the true missingness mechanism is rarely known—especially in large-scale, heterogeneous metabolomics datasets collected from diverse sources. While statistical tests such as Little’s MCAR test can help detect MCAR, and logistic regression models may suggest MAR, these methods are limited in scope and reliability. MNAR is particularly difficult to confirm, as it depends on unobserved values. In metabolomics, MNAR is often suspected due to left-censoring effects from instrument detection limits. Given this uncertainty, our framework is designed to be *agnostic* to the missingness mechanism. Rather than relying on potentially inaccurate pre-classification, we employ deep generative models that learn robust latent representations across datasets, enabling effective imputation regardless of the underlying missingness type.

There are several works describing approaches for imputing missing values in metabolomics datasets [1], [2], [3], [4], [5], [6]. Some researchers conduct simulations and evaluations of the currently existing missing imputation strategies [1], [2], while others come up with novel imputation approaches to handle missing values in real-world

metabolomics datasets [3], [4], [5], [6]. The existing techniques can be categorized into three subgroups: (i) statistical model-based, (ii) machine learning-based, and (iii) generative model-based. Our proposed work in this paper falls under the category of generative model-based approaches.

S1.1 Statistical model-based

Wei et al. [1] benchmark a total of eight imputation methods for various missing value types using four real-world GC/MS-LC/MS metabolomics datasets. They study zero, half minimum (HM), K-Nearest Neighbors (KNN), Singular Value Decomposition (SVD), Random Forest (RF), and Quantile Regression Imputation of Left-Censored Data (QRILC) for MNAR, and mean, median, KNN, SVD, and RF for MAR/MCAR. The RF algorithm demonstrates superior performance in handling missing data with MAR/MCAR, whereas the QRILC algorithm excels in dealing with left-censored data (i.e., caused by the device detection limit), which can be categorized as MNAR.

Do et al. [2] perform another comparative study of existing missing value imputation methods on metabolomics datasets. The researchers assess 31 imputation techniques inside a simulation framework to create synthetic data that replicates the missing data patterns found in actual datasets and employ these techniques on clinical metabolomics data for biological validation. The study’s primary findings indicate that missing data are caused by the limit of detection (LOD) effects, which vary depending on the day of analysis. The study also reveals that multivariate imputation techniques, such as KNN-based imputation and MICE, outperform other methods in predicting missing data. Since MICE is computationally challenging, the authors suggest KNN to be a more robust imputation method for untargeted metabolomics datasets.

S1.2 Machine learning-based

Dekermanjian et al. [3] introduce a two-step methodology, MAI (a missing mechanism-aware imputation), that takes into account the missingness type when estimating missing values in metabolomics datasets. A random forest classifier was used to classify the missingness type for each absent

value in the dataset. In order to predict the missing values, a combination of imputation methods tailored to the anticipated type of missingness mechanism (i.e., MNAR or MAR/MCAR) is performed. Their simulations suggest that utilizing a combination of single imputation, a method they had developed for estimating the expected value of a metabolite with left-censoring for MNAR values, and RF imputation [7] for MAR/MCAR values, works best in most situations. An inherent limitation of the MAI framework is its dependence on the precise classification of missing values. Furthermore, the evaluation of different multi-class classifiers has not been attempted. Since artificially classifying missing data may lead to neglecting the genuine underlying causes, we choose not to pre-classify missing values in our proposed frameworks.

Jin et al. [4] proposed an algorithm, NetSVR, that employs support vector regression (SVR) to predict missing values in LC/MS metabolomics data by considering features in the network's neighborhood and adduct ion interactions. The *NetSVR* method is compared with other commonly used (KNN, Bayesian principal component analysis (BPCA), simple linear regression (SLR), singular value decomposition (SVD), and imputation by inserting single values such as mean, median, and half-minimum) imputation algorithms, and is shown to outperform them in terms of accuracy based on NRMSE.

There are also what are called mosaic-integration (MI) methods ([8], [9], [10], [11], [12], [13], [14]). MI approaches aim to find a common embedding for multiple omics datasets, usually in single-cell domain. The common goal in these studies is to cluster cells or batches with similar profiles together in the same class. Furthermore, cell type prediction is another application. Likewise, another class of methods operates on the same type of omics datasets with a similar goal of finding a common embedding for the combined datasets ([15], [16], [17], [18], [19]).

S1.3 Generative model-based

Another group of studies adopts the generative model approach and attempts to reconstruct novel data from trained samples. Gomari et al. [5] employ a more advanced version of autoencoders, variational autoencoder (VAE) [20]. VAE is a type of neural network architecture with fully connected layers. The authors exploit VAE as an unsupervised dimensionality reduction strategy to capture nonlinear changes and impute missing values in metabolomics datasets using a model trained with 217 metabolite measurements obtained from 4,644 patients in the TwinsUK study [21]. A comparison is made between the employed VAE model and different variations of the PCA [22] and KPCA [23] models, including cosine, polynomial, RBF, and sigmoid KPCA. The authors show that non-linear dimensionality reduction techniques outperform the linear PCA method in accurately reconstructing the original data. Gomari et al. utilize a uniform, singular dataset including a known set of metabolites. That is, it does not consider cases where datasets come from diverse sources. It also does not consider that datasets may have different metabolite sets. Our work, on the other hand, integrates different datasets from different research labs to enhance the diversity of our training database.

Zhao et al. [6] propose a Multi-View Variational Autoencoder (MVAE) framework that combines genomics and metabolomics datasets to extract meaningful relationships between multi-omics data and impute missing values in the metabolomics datasets. They collected both whole-genome sequencing (WGS) and metabolomics data from the same individuals. Their cohort includes 1,110 subjects from the Louisiana Osteoporosis Study (LOS) [24]. MVAE integrates three different genomic aspects: burden scores derived from template metabolites, polygenic risk scores (PGS), which offer a quantification of an individual's susceptibility to a certain disease based on their genetic makeup, and linkage disequilibrium (LD)-pruned single-nucleotide polymorphisms (SNPs), which eliminate repetitive genetic mutations from collected samples. The MVAE model's performance is evaluated using the mean absolute percentage error (MAPE) and the R^2 score, using various pre-determined cut-off levels. Then, the authors compare the robustness of their metabolic data imputation method with the existing approaches such as KNN, ridge regression, RF, gradient boosting regression (GBT), and SVM. Zhao et al. integrate multiple types of omics data to increase the imputation method's power. Integrating different aspects of human biology is a novel research area. However, obtaining both genomic WGS data and metabolomic data from the same individuals is often impractical, costly, and hard to achieve in many clinical settings.

In addition to the metabolomics domain, generative models are also employed for data imputation in a variety of other fields. Zheng et al. [25] employ a diffusion model approach called TabCSDI for the imputation of time-series data and demonstrate the effectiveness of this method for imputing numerical variables. Gondora and Wang [26] develop an overly complete version of denoising autoencoders (unlike normal auto-encoders, in a structure that expands and shrinks) in a multiple imputation framework (MIDA) to reconstruct noise-free data. Their results demonstrate MIDA's superiority over MICE. Yoon et al. [27] build Generative Adversarial Imputation Nets (GAIN) with an extra hint generator to effectively represent the original data distribution, and compare its variations.

S2 MATERIALS AND METHODS

In this section, the overall data acquisition workflow, our training schemes, and three different merging approaches are presented.

S2.1 Database Construction

Metabolomics Workbench [28] is NIH's public repository for metabolomics datasets. Metabolomics datasets in this repository are generated by researchers worldwide from various laboratories using a range of instruments and extraction techniques. The URL-based REST API of Metabolomics Workbench enables researchers to explore studies, analyses, genes, and compounds with metadata analysis, phenotype filtering, and metabolite identification. In this study, we retrieved all human metabolomics studies from the Metabolomics Workbench (as of January 2024). Table S1 presents the statistics regarding the obtained metabolomics datasets.

TABLE S1: Our dataset summary statistics.

Database	Statistics		
	Studies	Patients	Metabolites
Workbench (Original)	490	43045	87774
Workbench (RefMet)	490	43045	18552
Workbench (RefMet + Filter)	437	40005	18045

S2.2 Data Processing

S2.2.1 Creating Datasets

A research study may involve the analysis of different types of data, including positive mode (LC/MS+) and negative mode (LC/MS-) reverse phase LC-MS metabolite measurements. Due to the lack of consistency in the investigations and analyses, it was necessary to combine the analysis files in order to obtain reproducible study datasets. Study files were constructed by merging these sub-measurements based on patients, with an initial filtering applied during the process. In cases where the same metabolite appeared multiple times within a study—such as being detected in both positive and negative ionization modes—we applied the following strategy:

- If the metabolite had identical names but different values (e.g., due to different m/z values in positive and negative modes), we retained a single representative value by taking the **element-wise maximum** across modes.
- If multiple identical entries existed within the same mode, we **averaged** their values.
- Metabolites that were uncaptured or blank were discarded.
- Studies with fewer than 50 metabolites after preprocessing were excluded from the dataset.

As a result of this pipeline, each study yielded a single, unified dataset with one value per metabolite, ensuring consistency across ionization modes while preserving the most informative signal.

Adduct Consideration and Ionization Modes

Our analysis is based on pre-processed and annotated metabolomics datasets obtained from public repositories, rather than raw spectral data. As such, we did not explicitly account for adduct formation or related ion species during preprocessing. We acknowledge that adduct formation can lead to different m/z values for the same compound, particularly across ionization modes. However, due to the lack of consistent raw data and adduct annotations across studies, we relied on the metabolite-level annotations provided by the original data contributors. These annotations may have already incorporated adduct resolution to varying degrees. We recognize this as a limitation of our study and highlight it as an area for future improvement when more detailed spectral metadata becomes available.

S2.2.2 Metabolite Mapping

Having metabolomics data across studies generated by different institutions and platforms poses a significant challenge for comparative analysis. The primary reason is that many metabolites lack a universally recognized common

name used by all researchers. To reconcile the heterogeneity across metabolomics datasets contributed by different laboratories, we performed a metabolite name harmonization step. Specifically, we mapped all metabolite identifiers to a standardized nomenclature using the RefMet (Reference Metabolite) database [29]. This mapping process ensures that chemically identical metabolites, which may be labeled differently across studies, are aligned under a common identifier. By doing so, we enable interoperability across datasets and facilitate the identification of shared metabolites, which is essential for effective dataset merging and model training. This step is particularly critical in metabolomics, where inconsistent naming conventions can severely hinder data integration and downstream analysis.

S2.2.3 Imputing real missing values and normalization

Some metabolite measurements may be missing in each dataset. Typically, researchers choose to exclude absent metabolites from the data or fill them with more accurate estimated values. Both approaches can potentially create bias and decrease the statistical power in subsequent analyses, such as examining the relationships between metabolites and clinical factors [3]. Metabolomics Workbench studies exhibited a median missingness value of 1.86%, with the range spanning from a minimum of 0.00% to a maximum of 75.22%. Missing values in the studies were imputed using Scikit Learn’s k-nearest-neighbour (KNN) imputation method. Data that was originally missing is masked while calculating the loss values during VAE model training to prevent the propagation of any bias introduced by KNN imputation.

S2.3 Merging Datasets

S2.3.1 General Merging Framework

In this section, we present our general merging framework that will be employed by all the proposed merging algorithms for common tasks. Algorithm S1 makes sure that each dataset is extended with metabolites that are not included in it and these new metabolite columns are filled with the provided imputation values (*imputationVals* in Algorithm S1) that are determined by the employed merging algorithm. More specifically, first, the union of metabolites in all datasets is obtained (line 1 in Algorithm S1). Subsequently, the feature space is expanded based on the provided missing values for each dataset (lines 8 and 10). After incorporating all new metabolites into a dataset, the metabolites are sorted (line 11).

S2.3.2 Cross-validation Framework

The merged dataset is split for training and validation of the studied dataset merging strategies. The splitting is done horizontally at the sample/patient level as outlined in Algorithm S2. Each dataset that is included in the merged data is divided into K folds (line 2). The minimum and maximum values from the training data of each fold are obtained, and then the training and validation data are normalized and yielded for training and evaluation purposes (lines 5, 6, 7, 8, and 9).

Then, imputation models are trained based on the above-specified data splitting. Algorithm S3 describes the model

Algorithm S1: Merge Datasets

Input : List of datasets to be merged: *datasetList*,
A dictionary that contains imputation values for each new metabolite column *m* added into a dataset *D*, where *D* and *m* are used as keys: *imputationVals*

Output: One fold of the merged dataset, where the data has been normalized

```

1 UnionMetabolites  $\leftarrow \bigcup_{i=1}^N$  metabolites in Si;
2 mergedDataset  $\leftarrow$  empty list;
3 foreach dataset S1 in datasetList do
4   foreach dataset S2 in datasetList do
5     foreach metabolite m in S2 do
6       if m  $\notin$  S1 then
7         if imputationVals is not null then
8            $S_1[m] \leftarrow imputationVals(S_1, m)$ ;
9         else
10           $S_1[m] \leftarrow$  null;
11   Sort metabolites of S1 wrt. UnionMetabolites;
12   mergedDataset.append(DataProcess(S1));
13 return mergedDataset

```

Algorithm S2: k-Fold Merged Datasets

Input : Merged dataset: *mergedDataset*,
Number of folds: *K*

Output: One fold of the merged dataset, where the data has been normalized

```

1 foreach dataset S in mergedDataset do
2   folds  $\leftarrow$  Split(S, K);
3   foreach fold k  $\in \{1, 2, \dots, K\}$  do
4     Strain, Sval  $\leftarrow$  folds[k];
5     min  $\leftarrow$  Find min value of Strain columns;
6     max  $\leftarrow$  Find max value of Strain columns;
7     Strain  $\leftarrow$  normalize(Strain, min, max);
8     Sval  $\leftarrow$  normalize(Sval, min, max);
9     yield Strain, Sval

```

creation. Different models are trained and evaluated for each split (lines 6 and 8). That is, all samples/patients from all datasets are used during the training and evaluation phases.

As an example, assume that the merge function is applied to datasets *S₁*, *S₂*, and *S₃*. Let *S₁* have metabolites *M₁*, *M₂*; *S₂* have metabolites *M₂*, *M₃*; and *S₃* have metabolites *M₆*, with 15, 10, and 20 patients, respectively. Suppose that the empty cells are initially filled with null values and a 5-fold cross-validation is performed. In this setup, five separate models are trained—one for each fold—and evaluated on their respective validation sets. The final performance is reported as the average across these folds. Here, the union of metabolites includes *M₁*, *M₂*, *M₃*, and *M₆*. Thus, during merging, all datasets will have metabolites *M₁*, *M₂*, *M₃*,

Algorithm S3: Generate VAE Model

Input : Merged dataset: *mergedDataset*

Output: Imputation models: *M₀*, *M₁*, ..., *M_K*

```

1 K  $\leftarrow$  define number of folds;
2 epoch  $\leftarrow$  define number of epoch;
3 modelList  $\leftarrow$  initialize a list for models to store;
4 foreach Epoch e  $\in \{1, 2, \dots, epoch\}$  do
5   foreach Strain, Sval in kFoldMergedDatasets(mergedDataset, K) do
6     M  $\leftarrow$  Train from Strain for fold;
7     Store M in modelList;
8     Evaluate M by Sval;
9 return modelList

```

and *M₆*. Each dataset's missing values are filled iteratively, and the merging process is completed following the data-processing step in line 12 of Algorithm S1. When the model creation process in Algorithm S3 is initiated, each epoch will iterate through folds of *S₁*, *S₂*, and *S₃*, respectively. For each fold, a separate model is created. For example, for the first fold, *S₁*'s metabolites *M₁*, *M₂*, *M₃*, and *M₆* will be used with 12 patients for training and 3 patients for validation. Similarly, *S₂* and *S₃* will be used with 8 and 16 patients for training, and 2 and 4 patients for validation, respectively. Thus, a total of 36 patients will be used for training and 9 patients for evaluation at the end of each epoch.

S2.3.3 Pseudocode for Approach 1: Union-based Merging

Union-based merging simply puts together all the individual datasets to create a single dataset (Algorithm S4). The metabolite set of the merged dataset is simply the union of the metabolite sets in merged datasets. The non-intersecting metabolites in each merged dataset are filled with 0s (lines 1 and 2). Next, samples/patients that have less than 50 metabolites, and metabolites that occur in less than 50 samples/patients are filtered out (line 3) from the merged dataset. Finally, an imputation model is trained on the merged dataset (line 4).

Algorithm S4: Union-based Merging

Input : Metabolite datasets in database:
datasetList

Output: Union-based model: *M_{union}*

```

1 vals  $\leftarrow$  initialize an array of 0s;
2 SN  $\leftarrow$  MergeDatasets(datasetList, vals);
3 Filter: SN  $\leftarrow$  based on patients and metabolites;
4 Munion  $\leftarrow$  GenerateVAEModel(SN);

```

S3 EXPERIMENTAL EVALUATION**S3.1 Baseline Selection Rationale**

To ensure a fair and comprehensive evaluation of our proposed methods, we selected baseline imputation techniques from three major methodological categories: (i) statistical

models (e.g., KNN, MICE), (ii) machine learning-based models (e.g., Random Forest), and (iii) generative models (e.g., GomariVAE). These methods were chosen based on their prevalence in the metabolomics literature and their performance in prior benchmarking studies.

For the generative model baseline, we re-implemented the VAE-based architecture proposed by Gomari et al. [5] and adapted it to our dataset using a union-based merging strategy. We also included comparisons using our own VAE model trained under the same merging conditions. All baseline models were evaluated using the same cross-validation splits and performance metrics ($\text{Sim-}R^2$ and R^2) as our proposed methods. This setup ensures that performance differences are attributable to methodological innovations rather than experimental inconsistencies.

S3.2 Comparison of Different Initial Imputation Methods

Our database includes multiple metabolomics datasets with various missingness ratios. These datasets were imputed to have a ground truth value so that we can evaluate our models' performance in reconstructing novel data. Therefore, we employed multiple imputation techniques to fill the original missing values in the datasets. For this experiment, we chose datasets that were originally complete and introduced artificially random gaps (i.e., missing values), ranging from a minimum of 5% missingness to a maximum of 50% missingness, and evaluated the accuracy of commonly used traditional imputation methods. In particular, we study the performance of the following imputation approaches:

- *Mean Imputation (mean)*: Missing values in each incomplete variable are substituted with the mean of the observed values for that metabolite.
- *Median Imputation (median)*: Missing values in each incomplete variable are substituted with the median of the observed values for that metabolite.
- *Half-minimum (HM)*: Missing values in each incomplete variable are substituted with half of the non-missing minimum value.
- *Random Imputation (random)*: Missing values in each incomplete variable are substituted with a random floating number between the minimum and maximum of the metabolite value.
- *K-nearest-neighbor (KNN) imputation* [30]: Missing values in each incomplete variable are substituted with the weighted average of a predetermined number of nearest non-missing variables for the specific metabolite. The origin of this imputation approach comes from microarray gene expression data. Neighbor distances were calculated using Euclidean distance, where closer neighbors of a missing metabolite will exert more effect than neighbors that are farther away. The Euclidean distance e between two points (x_1, y_1) and (x_2, y_2) can be calculated using equation 1. We applied the Python package `sci-kit-learn`'s `KNNImputer` for the imputation approach with a neighborhood size of 5.

$$e = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

- *Random Forest (RF) imputation* [7]: Missing values in each incomplete variable are substituted with the prediction of the trained random forest model. First, missing values are filled with the mean/mode values of the corresponding metabolite. Then for each feature containing missing values, an RF regressor was trained on other features, and missing values were predicted using the trained model. The iteration continued until a convergence value was reached. We applied the Python package `scikit-learn` `IterativeImputer` with `RandomForestRegressor` estimator for the imputation approach.

An initial experiment was conducted using the dataset ST000284 (Colorectal Cancer Detection) from Metabolomics Workbench, which has 224 individuals and 103 metabolites. A total of 10 unique seeds were used at every missingness percentage rate, ranging from 5% to 50% with increments of 5%. The results are shown in Figure S1. While *RF* demonstrated superior performance, it was comparable to *knn*. However, *RF*'s performance suffered from its complexity due to the training of numerous regressors for the imputation.

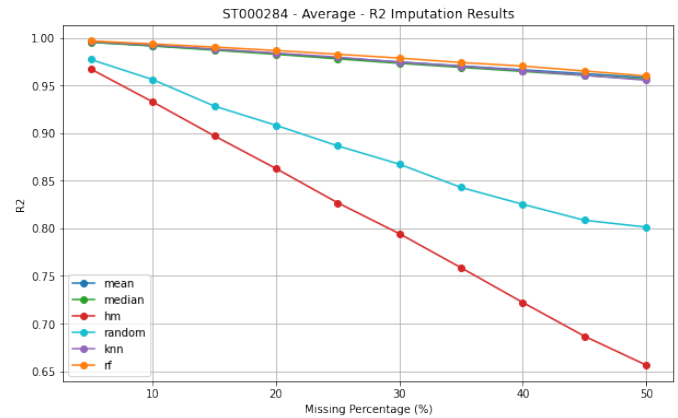


Fig. S1: R^2 scores for different imputation strategies on the ST000284-CRC (Colorectal Cancer Detection) dataset.

To further evaluate the robustness of the imputation approaches, we selected representative non-empty datasets of different sizes (small, medium, and large) and repeated the same experiment. Table S2 shows the mean $\text{Sim-}R^2$ values over all missingness percentage rates in the studies. The complexity drawback of *RF* was more evident in larger datasets due to the training of multiple regressors for the study. For this reason, *knn* imputation is chosen as the standard missing filling method in our proposed merging approaches for imputing the already existing missing values, as it is cheaper and provides comparable performance to *RF*.

S3.3 Comparison of Different Normalization Schemes

In order for the *MetaboliteVAE* model to learn effectively, the data needs to be normalized, scaled, or transformed after the imputation of initial missing values. Otherwise, the larger raw values in a dataset could hinder the model's ability to backpropagate and optimize the weights and biases of the layers present in the encoder and decoder parts of the VAE.

TABLE S2: Average Sim- R^2 scores for different imputation techniques

Dataset	Imputation Technique					
	Mean	Median	HM	Random	KNN	RF
ST000284 ($n = 23072$)	0.9255	0.9219	0.3827	0.5940	0.9266	0.9372
ST000918 ($n = 4118$)	0.9378	0.9385	0.4267	0.7659	0.9322	0.9330
ST001432 ($n = 11826$)	0.4793	0.4446	-0.0162	-3.6209	0.4517	0.5127
ST002555 ($n = 28368$)	0.1380	0.0782	-0.0140	-5.2625	0.4313	0.3366
MTBLS2615 ($n = 6643$)	0.8319	0.8169	0.3162	-0.0461	0.8204	0.8290
MTBLS7584 ($n = 13279$)	0.8351	0.8208	0.1189	0.5097	0.8443	0.8934
MTBLS4684 ($n = 20880$)	0.9159	0.9151	0.2541	0.6911	0.9315	0.9323
Average	0.7233	0.7051	0.2097	-0.9098	0.7625	0.7677

To this end, twelve different data pre-treatment methods (i.e., min-max normalization, centering, unit variance, Pareto, range, vast, level scaling, and log, Box-Cox, Yeo-Johnson transformation) are compared with that of a baseline VAE model trained on the six datasets from the previous experiment. A random missingness rate between 10% and 20% to mimic the missingness in the real clinical datasets is considered, and 10-fold cross-validation is applied for evaluation. Due to the range of the data, some pretreatment methods fail to create feasible models. The average Sim- R^2 scores are shown in Table S3. While the Yeo-Johnson transformation outperformed the other approaches, the drawback of increased complexity was apparent due to the optimization of the λ parameter. Thus, the $\log_{10}$ transformation was chosen as the standard pre-treatment method for our proposed merging approaches.

TABLE S3: Average Sim- R^2 scores of 10-folds *MetaboliteVAE* reconstructions for each data pretreatment technique.

Method	Dataset						
	ST284	ST918	ST1432	ST2555	MT2615	MT7584	MT4684
Min-Max	0.44	0.19	0.21	0.33	0.28	0.16	0.27
Centering	-	-0.29	0.14	0.58	-0.19	0.03	0.00
Standard Scaler	0.13	0.16	0.04	0.69	0.05	-0.41	0.40
Pareto Scaler	-	-0.15	0.07	0.69	0.09	0.29	0.28
Range Scaler	0.17	0.13	0.04	0.66	0.03	-0.45	0.36
Vast Scaler	0.18	0.17	0.09	0.67	0.08	-0.52	0.33
Level Scaler	-	-0.02	-16.35	0.63	-0.01	-0.04	0.41
$\log_e$	0.86	0.92	0.58	0.75	-	0.86	0.89
$\log_2$	0.86	0.92	0.56	0.76	-	0.86	0.88
$\log_{10}$	0.89	0.93	0.61	0.76	-	0.86	0.90
Box-Cox	-	0.90	0.63	0.72	-	0.85	0.88
Yeo-Johnson	0.75	0.95	0.65	0.62	0.96	0.85	0.88

S3.4 Comparison of Dataset Merging Approaches

We next evaluate our proposed dataset merging strategies for missing metabolite value imputation.

The first approach, *Union-based Merging*, aims to join all available datasets in public databases to create one universal large model. This approach represents the state-of-the-art in the metabolomics data imputation space [5]. However, under the *Union-based Merging* scheme, *MetaboliteVAE* fails to learn and optimize parameters. To form a comparable baseline, we made an attempt to make union-based merging work by introducing an improved version of it. We call the improved version *Best Effort Union-based Merging*. It selects random seed datasets from our database as a starting point and joins them based on the Jaccard similarity score until the merged dataset does not lead to a trainable model (i.e., the

loss value is not undefined or NaN). The number of most similar datasets that are merged in this approach is a parameter, represented as n_{sim} . Different n_{sim} values (e.g., 16, 12, 10, 8) are evaluated, and we determine that 8 is the most optimal number to create trainable models under the best-effort union-based merging approach before trained models lead to undefined loss values. A random selection of 10 seed datasets was made from the Metabolomics Workbench database, and each of them was combined with their 8 most similar datasets. A model is trained for each merged dataset using a 10-fold cross-validation setup. In cross-validation, datasets are first shuffled, then divided into 90% training and 10% test splits, repeated by the number of folds. Trained models were saved to be used in prediction. For 437 studies in the Metabolomics Workbench database, the test data that was reserved in the cross-validation split was used as unseen data. These test dataset splits are fully imputed with the *KNNImputer*. Then, artificial missing values are created at random missingness rates between 10%-20%, and predictions are made using different numbers of the most similar models (parametrized as n_{best}). We experiment with different n_{best} values of 1, 5, and 10. The results are shown in Figures S2 and S3. For cases where we have predictions by multiple models (i.e., $n_{best} > 1$), Jaccard similarity-based weighted averages were used. Under best-effort variation, since only part of the data repository could be used during model training, the models under this merging strategy were unable to provide predictions for all of the unseen Metabolomics Workbench data. More specifically, the best-effort union-based approach was able to come up with missing value predictions for only 63 datasets.

For the second approach, *Iterative Similarity-based Merging*, for each of 437 studies in our Metabolomics Workbench database, an optimal merge set is created with the proposed algorithm. After duplicates are eliminated, we obtain 264 unique Jaccard-based merge sets. For each merge set, studies inside the set are merged, and a model is trained using a 10-fold cross-validation setup on the merged dataset as explained above.

After saving our models in a repository, we tested them with the test data from each dataset that we split with cross-validation. Finally, the artificially created missing values are imputed with our newly proposed *IterativeSimilarityImputer*. The test data of 437 studies are imputed with different values of n_{best} as 1, 5, and 10. The results are presented in Figure S4 and Figure S5.

The third dataset merging approach, *Model-guided Agglomerative Merging*, merges datasets two by two until a single model is produced. When we look at the training losses of each individual model up until the final trained joint models, we observe that the loss value of the joint model becomes NaN (not a number) after joining 64 models, similar to the scenario of *Union-based Merging*. As a result, the approach's original prediction scheme as explained in the methods section, was slightly revised to tackle this issue. That is, instead of going all the way through the root node, the joining process is early-stopped just before loss values become undefined, i.e., NaN. With this revision, the performance of the model-guided agglomerative merging on unseen Metabolomics Workbench data is presented in Figure S6.

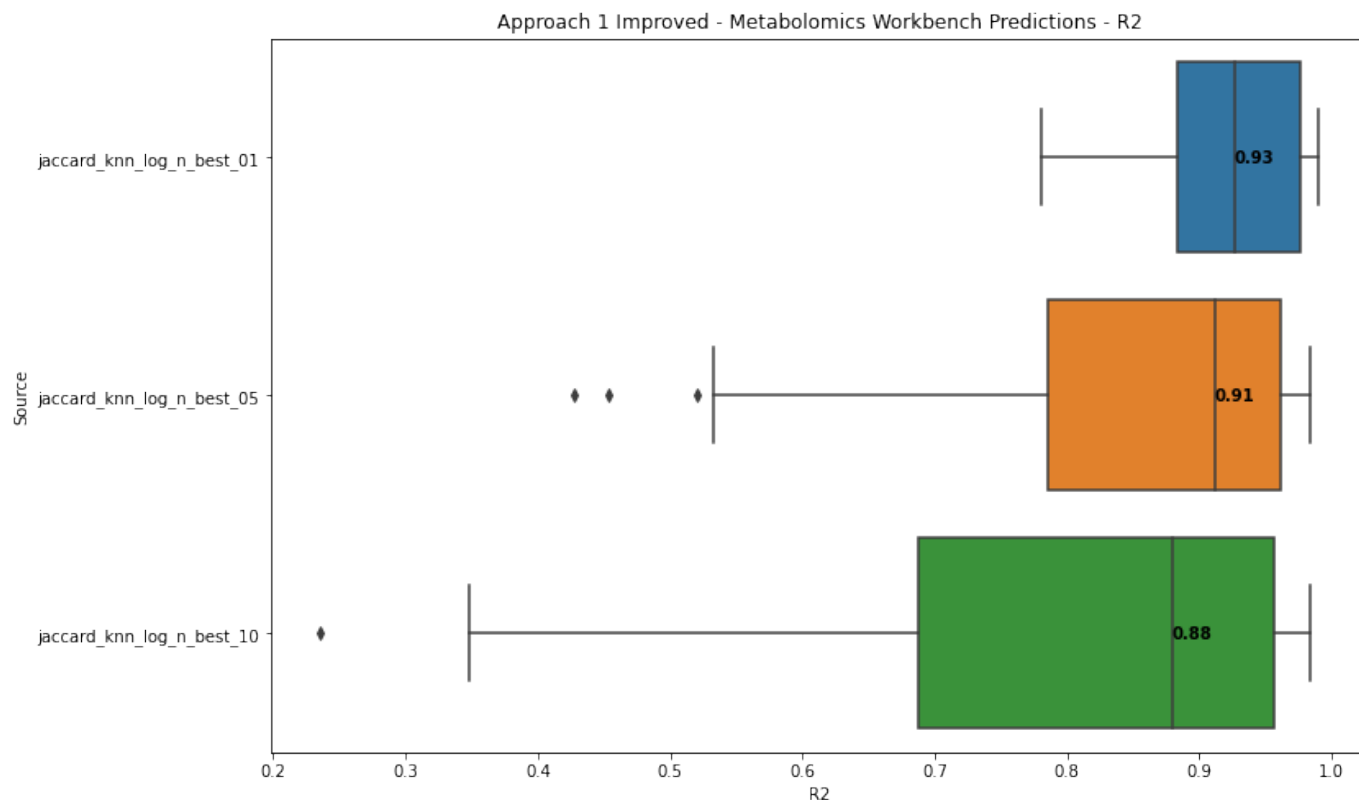


Fig. S2: Average R^2 scores of *Best-effort Union-based Merging* on Metabolomics Workbench studies.

S3.5 Comparison of Different Similarity Metrics

In this section, we compare the original Jaccard similarity score to our proposed weighted Jaccard similarity scoring scheme. For Iterative Similarity-based Merging, each of the 437 studies in our Metabolomics Workbench database was used to create new optimal merge sets with a weighted Jaccard score. After duplicates are eliminated, we obtain 363 unique weighted Jaccard-based merge sets. The same cross-validation setup was used to train all merge sets. Also, the weighted score was used for predictions. The average 10-fold prediction results on unseen Metabolomics Workbench data are presented in Figure S7 with R^2 scores at the top and $Sim-R^2$ scores at the bottom. The increase in performance in terms of R^2 scores was modest, but in terms of $Sim-R^2$ scores, the performance improvement was more evident. To sum up, the weighted Jaccard score enables us to build more stable and concise models for our model pool, increasing the multi-dataset prediction power.

REFERENCES

- [1] R. Wei, J. Wang, M. Su, E. Jia, S. Chen, T. Chen, and Y. Ni, "Missing value imputation approach for mass spectrometry-based metabolomics data," *Scientific Reports*, vol. 8, no. 1, p. 663, Jan 2018. [Online]. Available: <https://doi.org/10.1038/s41598-017-19120-0>
- [2] K. T. Do, S. Wahl, J. Raffler, S. Molnos, M. Laimighofer, J. Adamski, K. Suhre, K. Strauch, A. Peters, C. Gieger, C. Langenberg, I. D. Stewart, F. J. Theis, H. Grallert, G. Kastenmüller, and J. Krumsiek, "Characterization of missing values in untargeted ms-based metabolomics data and evaluation of missing data handling strategies," *Metabolomics*, vol. 14, no. 10, p. 128, Sep 2018. [Online]. Available: <https://doi.org/10.1007/s11306-018-1420-2>
- [3] J. P. Dekermanjian, E. Shaddox, D. Nandy, D. Ghosh, and K. Kechris, "Mechanism-aware imputation: a two-step approach in handling missing values in metabolomics," *BMC Bioinformatics*, vol. 23, no. 1, p. 179, May 2022. [Online]. Available: <https://doi.org/10.1186/s12859-022-04659-1>
- [4] Z. Jin, J. Kang, and T. Yu, "Missing value imputation for LC-MS metabolomics data by incorporating metabolic network and adduct ion relations," *Bioinformatics*, vol. 34, no. 9, pp. 1555–1561, 12 2017. [Online]. Available: <https://doi.org/10.1093/bioinformatics/btx816>
- [5] D. P. Gomari, A. Schweickart, L. Cerchietti, E. Paietta, H. Fernandez, H. Al-Amin, K. Suhre, and J. Krumsiek, "Variational autoencoders learn transferrable representations of metabolomics data," *Communications Biology*, vol. 5, no. 1, p. 645, Jun 2022. [Online]. Available: <https://doi.org/10.1038/s42003-022-03579-3>
- [6] C. Zhao, K.-J. Su, C. Wu, X. Cao, Q. Sha, W. Li, Z. Luo, T. Qin, C. Qiu, L. J. Zhao, A. Liu, L. Jiang, X. Zhang, H. Shen, W. Zhou, and H.-W. Deng, "Multi-view variational autoencoder for missing value imputation in untargeted metabolomics," *ArXiv*, Oct. 2023.
- [7] D. J. Stekhoven and P. Bühlmann, "MissForest—non-parametric missing value imputation for mixed-type data," *Bioinformatics*, vol. 28, no. 1, pp. 112–118, 10 2011. [Online]. Available: <https://doi.org/10.1093/bioinformatics/btr597>
- [8] Z. He, S. Hu, Y. Chen, S. An, J. Zhou, R. Liu, J. Shi, J. Wang, G. Dong, J. Shi *et al.*, "Mosaic integration and knowledge transfer of single-cell multimodal data with midas," *Nature Biotechnology*, pp. 1–12, 2024.
- [9] C. Fu, J. Lu, N. Zhao, and W. Ling, "Mosaic: A pipeline for microbiome studies analytical integration and correction," *bioRxiv*, 2024. [Online]. Available: <https://www.biorxiv.org/content/early/2024/11/11/2024.11.07.622561>
- [10] T. B. Verhey, H. Seo, A. Gillmor, V. Thoppey-Manoharan, D. Schriemer, and S. Morrissy, "mosaicmpi: a framework for modular data integration across cohorts and -omics modalities," *Nucleic Acids Research*, vol. 52, no. 12, pp. e53–e53, 05 2024. [Online]. Available: <https://doi.org/10.1093/nar/gkae442>
- [11] S. Ghazanfar, C. Guibentif, and J. C. Marionni, "Stabilized mosaic single-cell data integration using unshared features," *Nature biotechnology*, vol. 42, no. 2, pp. 284–292, 2024.

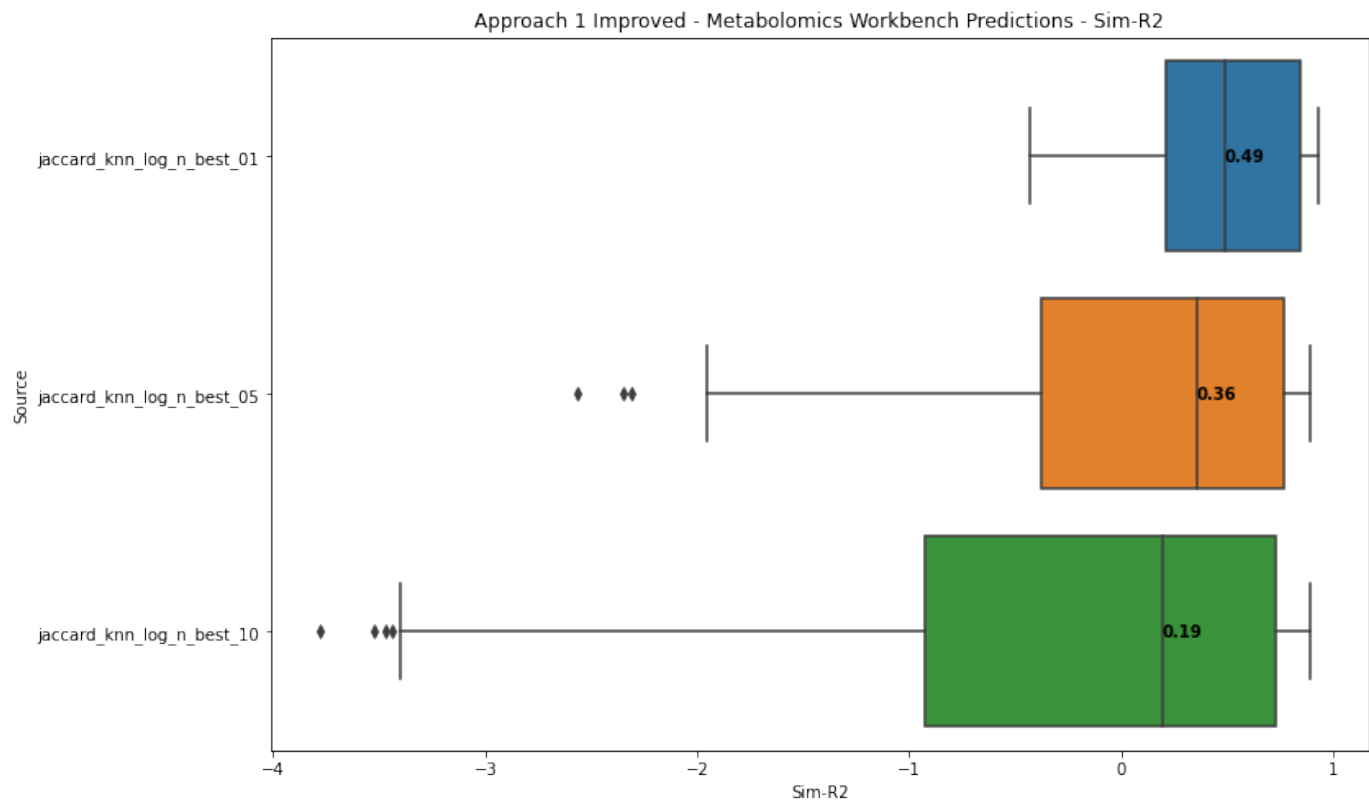


Fig. S3: Average $Sim-R^2$ scores of *Best-effort Union-based Merging* on Metabolomics Workbench studies.

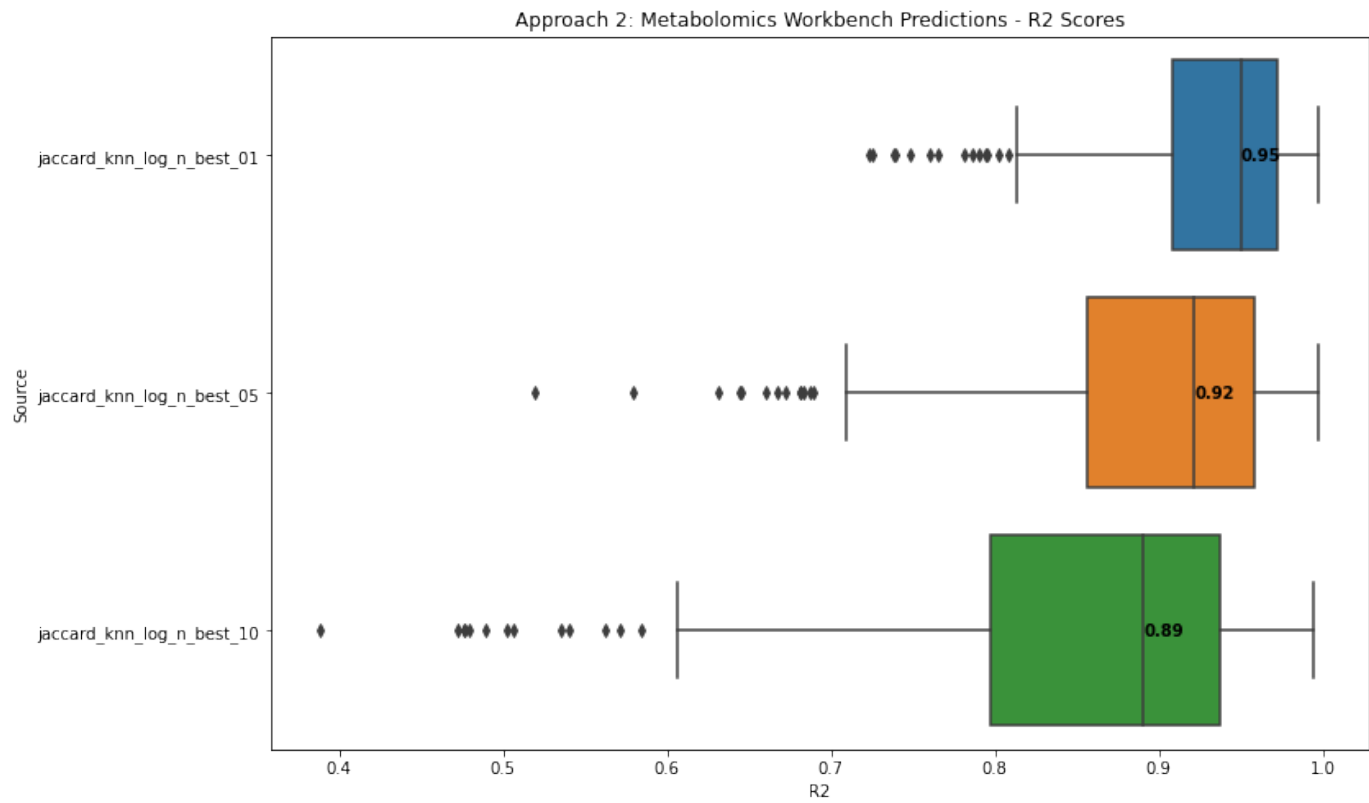


Fig. S4: Average R^2 scores of *Iterative Similarity-based Merging* on Metabolomics Workbench studies.

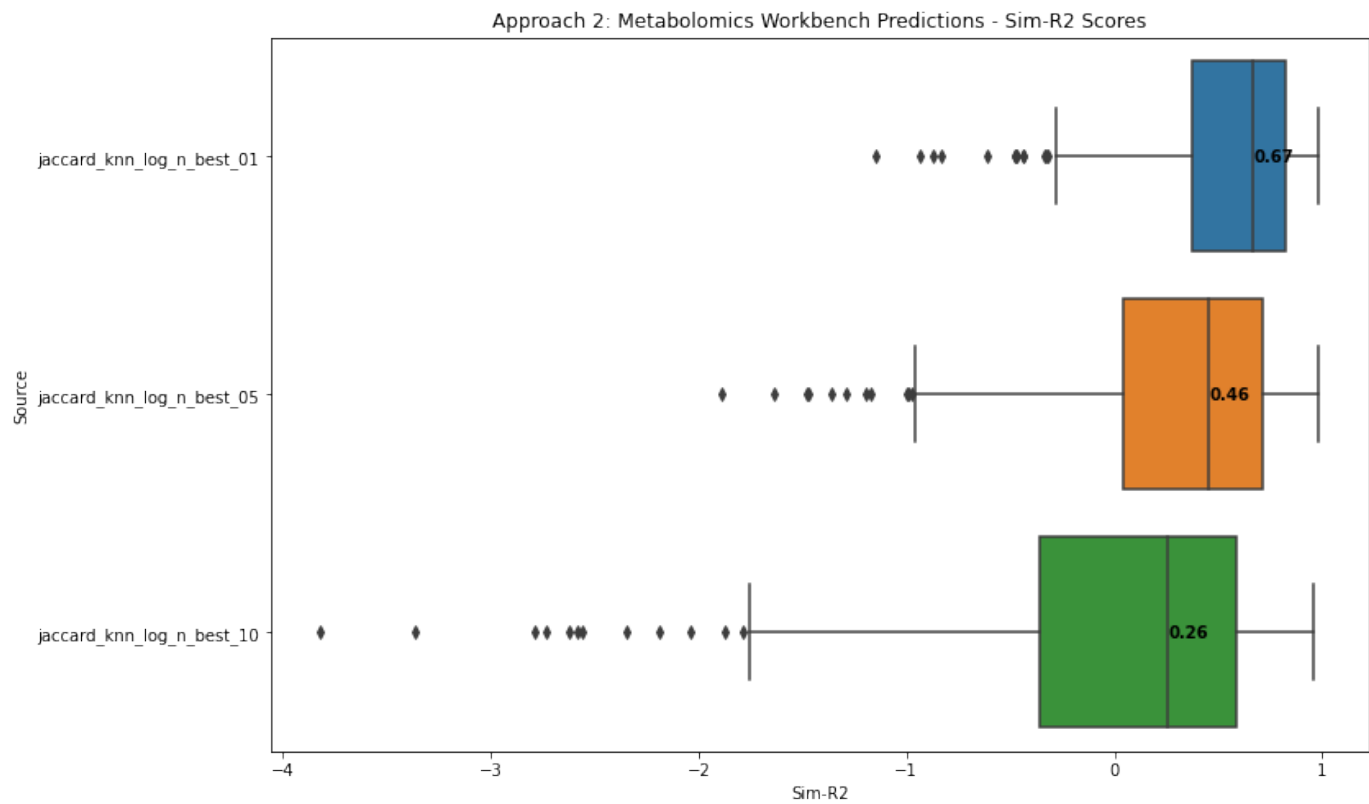


Fig. S5: Average *Sim-R²* scores of *Iterative Similarity-based Merging* on Metabolomics Workbench studies.

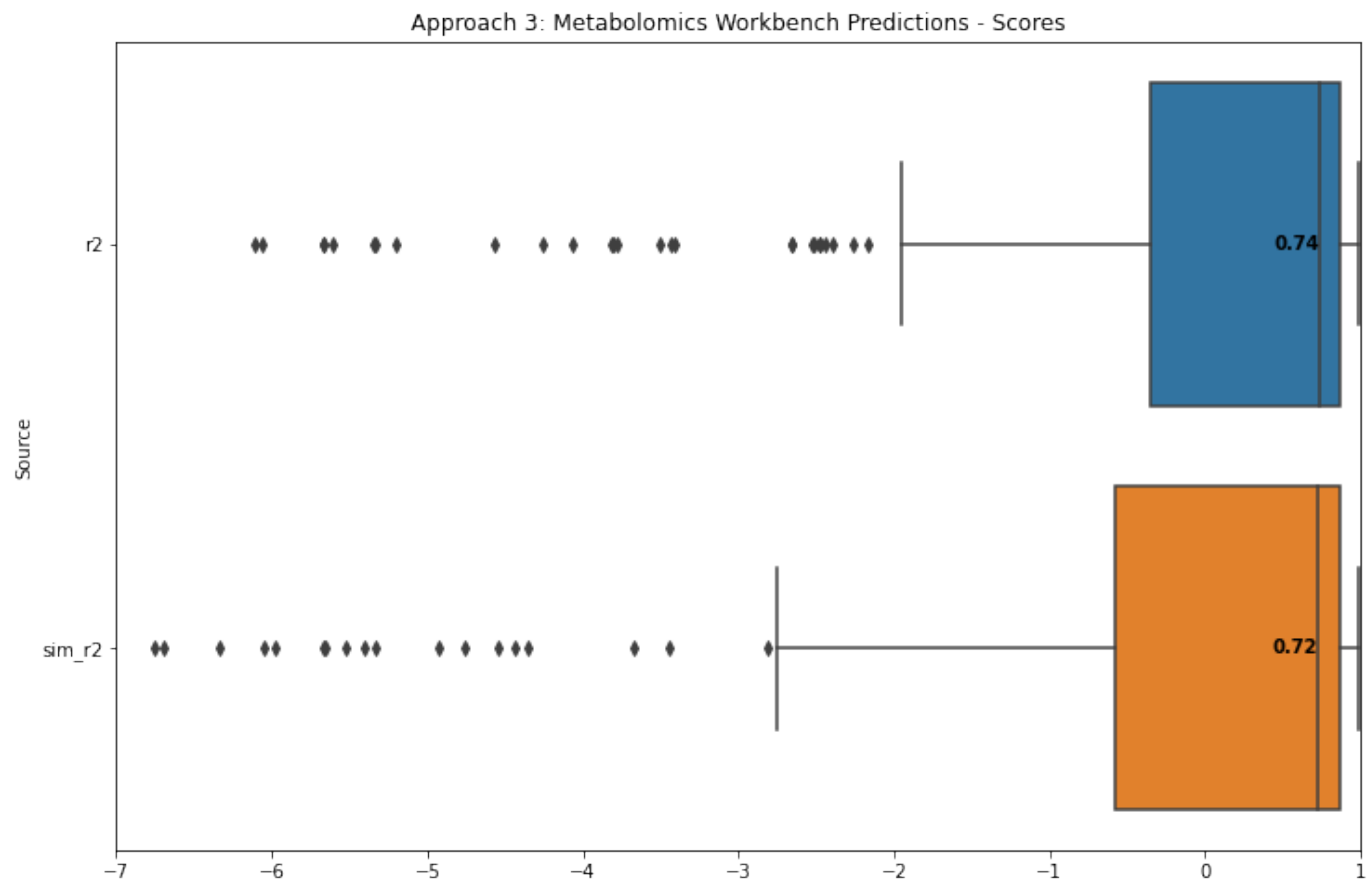


Fig. S6: Average scores of *Model-guided Agglomerative Merging* on Metabolomics Workbench studies.

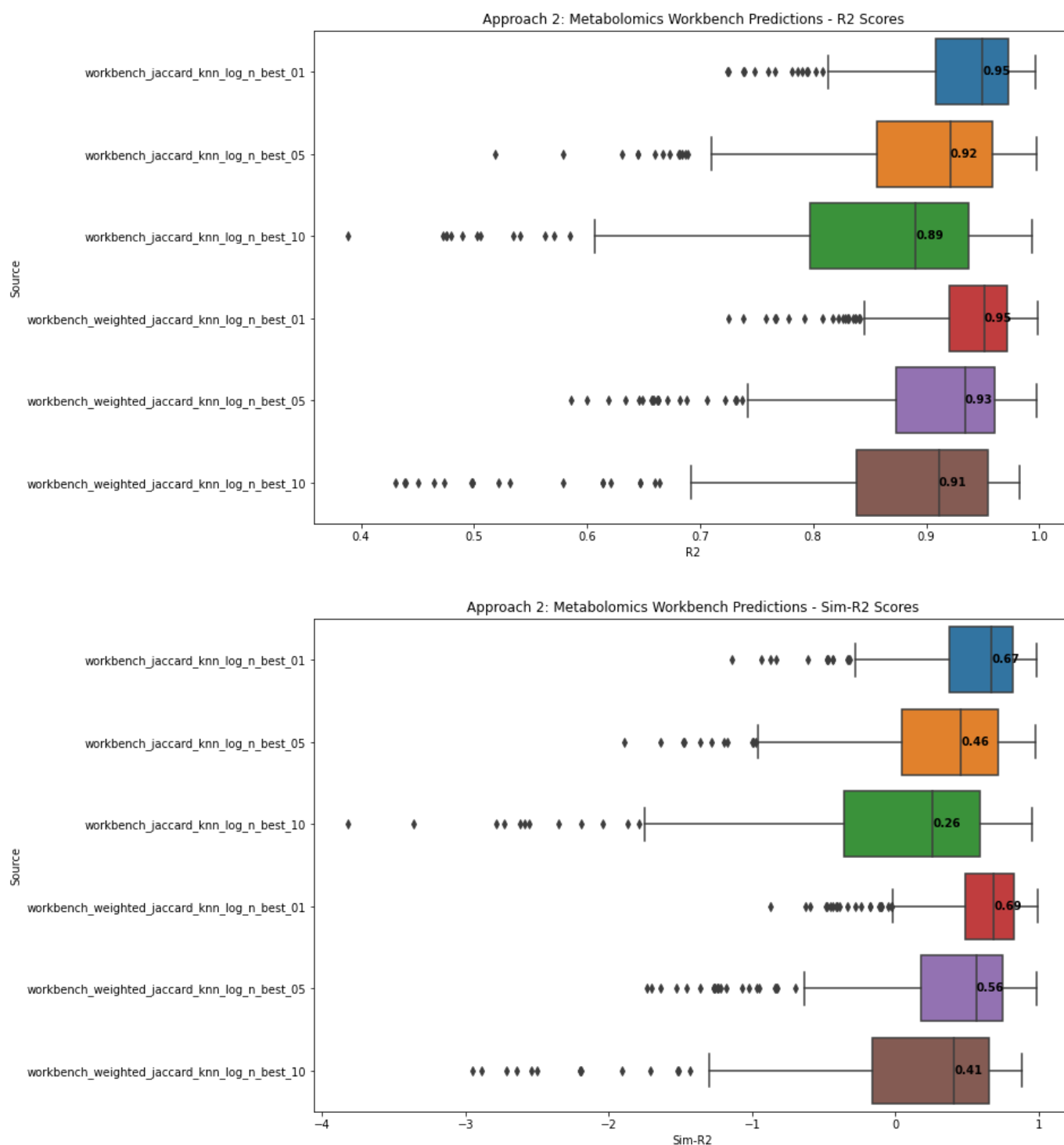


Fig. S7: Average test scores of *Iterative Similarity-based Merging* with different similarity metrics on Metabolomic Workbench studies. R^2 at the top and, $Sim-R^2$ at the bottom.

- [12] Z. Zhang, H. Sun, R. Mariappan, X. Chen, X. Chen, M. S. Jain, M. Efremova, S. A. Teichmann, V. Rajan, and X. Zhang, "scmomat jointly performs single cell mosaic integration and multi-modal bio-marker detection," *Nature Communications*, vol. 14, no. 1, p. 384, 2023.
- [13] J.-H. Du, Z. Cai, and K. Roeder, "Robust probabilistic modeling for single-cell multimodal mosaic integration and imputation via scvaeit," *Proceedings of the National Academy of Sciences*, vol. 119, no. 49, p. e2214414119, 2022.
- [14] H. De los Santos, K. P. Bennett, and J. M. Hurley, "Mosaic: a joint modeling methodology for combined circadian and non-circadian analysis of multi-omics data," *Bioinformatics*, vol. 37, no. 6, pp. 767–774, 2021.
- [15] A. Fouché, L. Chadoutaud, O. Delattre, and A. Zinovyev, "Transmorph: a unifying computational framework for modular single-cell rna-seq data integration," *NAR Genomics and Bioinformatics*, vol. 5, no. 3, p. lqad069, 2023.
- [16] E. M. Mirkes, J. Bac, A. Fouché, S. V. Stasenko, A. Zinovyev, and A. N. Gorban, "Domain adaptation principal component analysis: base linear method for learning with out-of-distribution data," *Entropy*, vol. 25, no. 1, p. 33, 2022.
- [17] D. Sun, X. Guan, A. E. Moran, L.-Y. Wu, D. Z. Qian, P. Schedin, M.-S. Dai, A. V. Danilov, J. J. Alunkal, A. C. Adey *et al.*, "Identifying phenotype-associated subpopulations by integrating bulk and single-cell sequencing data," *Nature biotechnology*, vol. 40, no. 4, pp. 527–538, 2022.
- [18] P. Cui, H. Wang, and Z. Bai, "Integrated single-cell and bulk rna-seq analysis identifies a prognostic t-cell signature in colorectal cancer," *Scientific Reports*, vol. 14, no. 1, p. 20177, 2024.
- [19] M. Loza, S. Teraguchi, D. M. Standley, and D. Diez, "Unbiased integration of single cell transcriptome replicates," *NAR genomics and bioinformatics*, vol. 4, no. 1, p. lqac022, 2022.
- [20] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022.
- [21] A. Moayyeri, C. J. Hammond, D. J. Hart, and T. D. Spector, "The UK adult twin registry (TwinsUK resource)," *Twin Res Hum Genet*, vol. 16, no. 1, pp. 144–149, Oct. 2012.
- [22] A. Mackiewicz and W. Ratajczak, "Principal components analysis (pca)," *Computers and Geosciences*, vol. 19, no. 3, pp. 303–342, 1993. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/009830049390090R>
- [23] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *Artificial Neural Networks — ICANN'97*, W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 583–588.
- [24] T.-L. Yang, H. Shen, A. Liu, S.-S. Dong, L. Zhang, F.-Y. Deng, Q. Zhao, and H.-W. Deng, "A road map for understanding molecular and genetic determinants of osteoporosis," *Nature Reviews Endocrinology*, vol. 16, no. 2, pp. 91–103, Feb 2020. [Online]. Available: <https://doi.org/10.1038/s41574-019-0282-7>
- [25] S. Zheng and N. Charoenphakdee, "Diffusion models for missing value imputation in tabular data," 2023.
- [26] L. Gondara and K. Wang, "Mida: Multiple imputation using denoising autoencoders," 2018.
- [27] J. Yoon, J. Jordon, and M. van der Schaar, "Gain: Missing data imputation using generative adversarial nets," 2018.
- [28] M. Sud, E. Fahy, D. Cotter, K. Azam, I. Vadelu, C. Burant, A. Edison, O. Fiehn, R. Higashi, K. S. Nair, S. Sumner, and S. Subramaniam, "Metabolomics workbench: An international repository for metabolomics data and metadata, metabolite standards, protocols, tutorials and training, and analysis tools," *Nucleic Acids Res*, vol. 44, no. D1, pp. D463–70, Oct. 2015.
- [29] E. Fahy and S. Subramaniam, "Refmet: a reference nomenclature for metabolomics," *Nature Methods*, vol. 17, no. 12, pp. 1173–1174, Dec 2020. [Online]. Available: <https://doi.org/10.1038/s41592-020-01009-y>
- [30] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 06 2001. [Online]. Available: <https://doi.org/10.1093/bioinformatics/17.6.520>