A Comparative Analysis in a Clinical Cohort: Multiple Imputation by Chained Equations and a Novel Super Learner-Based Imputation Approach

Tony Zbysinski[1], PhD ; Lezhou Wu[1], PhD; Justin Dale[1], BBA; James Coates[1], MS; Karan Sapiah[1], MS; Jamie Reuben[1], MPH; Frank Markson[1], BA; Ujjwal Kulkarni[1], MS; Nazmul Islam[1], PhD

[1]RefinedScience, Aurora, Colorado

Corresponding author:

Tony Zbysinski III, PhD

RefinedScience, 2115 N Scranton Street, Suite 2-70

Aurora Colorado, 80045, USA

Phone: +17637428523

Email: tony.zbysinski@refinedscience.com

Acknowledgements: We give additional acknowledgement and great gratitude to the internal reviewers of RefinedScience and external reviewers of Flatiron Health.

Declaration of conflicting interests: All authors are affiliated with RefinedScience and receive compensation as part of employment under RefinedScience. The authors declare no other conflicts of interest.

Funding: This research received no specific grant from any funding agency in the public, commercial, or non-profit sectors.

Ethical Approval: This retrospective study was approved by The Colorado Multiple Institutional Review Board (COMIRB) and used a limited dataset with a waiver of consent. The Flatiron Health data that supported the findings of this study were originated by and are the property of Flatiron Health, Inc.

Data Availability: The raw, individual patient data are protected and not available due to data privacy laws. The processed data are available at reasonable requests to the corresponding author. The Flatiron Health data that supported the findings of this study originated by and are the property of Flatiron Health, Inc., which has restrictions prohibiting the authors from making the data set publicly available. Requests for data sharing by license or by permission for the specific purpose of replicating results in this manuscript can be submitted to PublicationsDataAccess@flatiron.com. Lastly, upon publication, the SL-based imputation function will be accompanied by an HTML-based illustration that provides a concise, step-by-step walkthrough that enables users to quickly understand its purpose, inputs, outputs, and performance metrics, and to apply the function to their own datasets.

Abstract

Background: Missing data is a challenge in clinical research, especially in real-world data (RWD), where complete case analysis can bias results and reduce power. Ensemble learning approaches like Super Learner (SL) show strong numerical performance for prediction problems, but their use for missing value imputation (MVI) in oncology datasets is unexplored. We sought to develop and evaluate a novel SL-based imputation function that can impute multiple variables and quantify uncertainty.

Methods: We analyzed two independent cohorts of acute myeloid leukemia patients (n=1641). The SL-based MVI function includes data processing, predictor selection, binary and continuous variable pipelines, and performance measurement. Performance was compared to multiple imputation by chained equations (MICE) using balanced accuracy, F_1-score, root mean square error (RMSE), and visualizations.

Results: In a numerical experiment with 9 clinically important features, the proposed MVI function imputed and achieved higher balanced accuracy than MICE for 7/9 variables (mean balanced accuracy 89.04% vs 80.75%) with comparable performance for other variables. The continuous variable SL ensemble showed comparable RMSE (582) relative to MICE (597).

Conclusions: This study demonstrates that the SL-based imputation function improves accuracy over MICE in high-dimensional RWD while providing novel, observation-level uncertainty quantification.

MeSH keywords: Acute Myeloid Leukemia, Data Science, Machine Learning, Uncertainty, Imputation, Missingness

Introduction

Data missingness is a common occurrence in clinical research and poses a persistent threat to the validity of statistical inference and the strength of predictive models. Missing values can arise for several reasons such as instrument failure, patient dropout, or recording decisions driven by underlying patient characteristics.[1,2] These challenges lead to reduced statistical power, biased effect estimates, and compromised precision, ultimately undermining predictive modeling efforts.[3,4] As real-world data (RWD) sources grow in volume and complexity, particularly in high-dimensional settings such as electronic health records or genomic repositories, the challenge of appropriately handling missingness increases. Traditional approaches like complete-case analysis become unrealistic in many scenarios and can propagate selection bias.[5]

Early work on missing data classified the mechanisms by which values are missing. These include missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). Such mechanisms are one part of understanding which imputation approach is needed.[3] Classical single and multiple imputation techniques, such as mean/median/mode substitution or multiple imputation via chained equations (MICE), provide straightforward and satisfactory solutions under MCAR and MAR assumptions, but can perform poorly when these assumptions are violated or when variable interdependencies are complex.[6,7] Machine learning (ML) methods including k-nearest neighbors (k-NN), random forests, support vector machines, and others have typically demonstrated improved performance over purely statistical methods, particularly for moderate missingness and MAR settings.[7–9] Deep learning approaches, such as generative adversarial imputation networks and

variational autoencoders, offer promise in capturing nonlinear relationships and high-order interactions,[5,10] but comparative studies against well-established algorithms like missForest and MICE in RWD scenarios have shown that conventional methods remain competitive or superior in practice at this time. This is likely due to the volume of tuning parameters and the dataset size requirements of deep learning approaches.[11]

Despite the broad array of available imputation strategies, no single method uniformly or consistently outperforms others across mechanisms, domains, and dimensionalities.[12] In clinical settings, where variable distributions may be skewed and the mechanisms of missingness are frequently a complex mix of MCAR, MAR, and MNAR, ensemble learning techniques hold particular promise by combining multiple base learners to hedge against model-specific weaknesses.[12] The Super Learner (SL) framework implements a stacking ensemble that uses cross-validation (CV) specific risk estimates, risk being a term used for brevity when referring to objective functions top optimally weight ML algorithms (e.g. minimizing mean squared error). In this approach it combines the strengths of multiple semi-parametric or non-parametric models to make a balanced single weighted imputation of missing values,[13,14] unlike MICE which is a more iterative process. It has been applied and tested for missing value imputation (MVI) in previous literature, showing superior performance in at least synthetic data compared to more popular methods such as MICE.[15] Outside of MVI, but within the clinical domain, SL-based approaches have yielded strong results for numerous prediction-related tasks.[16–20] However, SL's adoption for imputation in clinical RWD has been limited despite its ability to outperform traditional methods. This advancement is critical given

46 the potential impact of imputation on downstream tasks, such as treatment effect
47 estimation, risk stratification, and clinical decision-making.
48 This study addresses a key gap in application by evaluating the performance of
49 MVI techniques using a SL-based approach on a high-dimensional dataset. In addition,
50 we build on this approach by introducing a novel function designed to greatly simplify
51 the imputation process with SL and provide measures of uncertainty for other modeling
52 tasks. The evaluation is conducted on clinical data comprising acute myeloid leukemia
53 (AML) patients treated with venetoclax plus azacitidine (Ven/Aza) or cytarabine plus
54 anthracycline (7+3). We experiment with varying missingness ratios and mixed or
55 unknown missingness mechanisms to reflect realistic challenges in clinical data
56 analysis. By benchmarking our SL-based imputation function against a conventional
57 statistical technique and selected ML methods, we aim to determine whether a SL-
58 based approach can deliver more robust and accurate imputations under clinical RWD
59 conditions and provide uncertainty measures for downstream tasks. Our specific
60 objectives are threefold: 1. Develop a novel user-friendly SL-based imputation function
61 capable of iteratively imputing multiple target variables while quantifying uncertainty
62 measures for other modeling purposes; 2. Apply the SL-based imputation function to
63 impute missing diagnostic patient AML data and quantify numerical performance of
64 singular learners of the ensemble; 3. Compare the SL-based imputation function's
65 predictive performance to state-of-the-art MICE performance. Note the proposed model
66 is disease-agnostic and thus can be applied beyond AML without loss of generalizability.
67 While we confine our objectives to imputing baseline features for convenience, the

function can be generalized and extended to imputing values at other clinically relevant time-points.

Methods

Data Description

This study leveraged data from two cohorts (n=1641) of adult patients (aged ≥ 18 years) with newly diagnosed AML. Briefly, these patients were treated with Ven/Aza or 7+3. The University of Colorado (CU) cohort included 546 patients that began treatment in a clinical trial between January 2015 and March 2024 in the UCHealth system. Data is continuously updated over time as follow-up progresses, where follow-up ends at date of death, loss to follow-up, or until the most recent follow-up update. We censor patients only to their last known alive date. The real world cohort (RWC) was derived from the nationwide Flatiron Health electronic health record-derived, de-identified, longitudinal database consisting of patient-level structured and unstructured data curated via technology enabled abstraction and was comprised of 1095 patients from 87 unique sites of care.[21,22] Flatiron Health dataset patients had a minimum of two documented clinical visits between January 2014 and December 2023 and received the Ven/Aza treatment as frontline therapy. The de-identified data is sourced from approximately 280 cancer clinics across the United States, including 800 sites of care primarily within community oncology settings. Inclusion and exclusion criteria were applied to minimize systematic differences between the cohorts, as discussed by Islam et al.[23] Specifically, patients were excluded if they had acute promyelocytic leukemia, biphenotypic leukemia, chronic myeloid leukemia in blast crisis, or central nervous system involvement at diagnosis. A more thorough description of this data can be found

in previous literature.[23] Supplementary Figure S1 describes via a visual summary the genetic alteration patterns of both the RWC and the CU cohorts. Figure 1 illustrates the 5 key steps of our novel SL-based imputation framework, providing an overview of the processes detailed in the following sections on specification and data processing, imputation pipelines, performance assessment, and our application in high-dimensional clinical data.

Specification and Data Processing

Step 1 includes all manual data pre-processing for the purpose of having correct data format and structure for the developed SL-based imputation function. Data pre-processing included, for our analysis, only variables measured at baseline (e.g., cytogenetic testing or laboratory measured values at diagnosis or treatment start date). The function accepts input data containing continuous and binary variable types; multinomial and ordinal features need to be transformed into a design matrix with binary columns before applying the MVI function. In our application, diagnostic clinical variables were harmonized between cohorts by merging on identified common

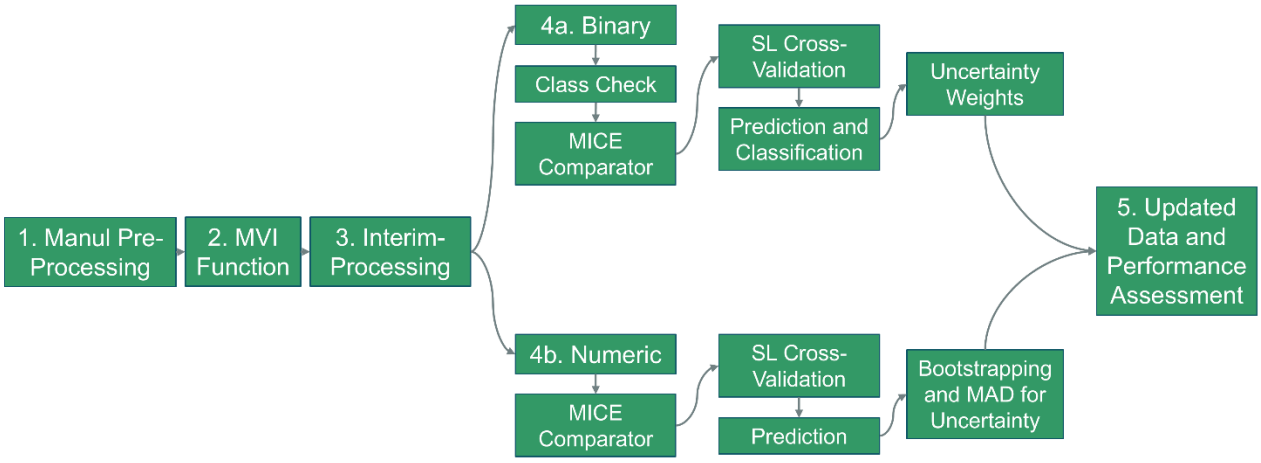

Figure 1. SL-based MVI imputation function methods flow diagram. *Abbreviations:* Median Absolute Deviation (MAD), Multiple Imputation by Chained Equations (MICE), Missing Value Imputation (MVI), Super Learner (SL).

variables. All dichotomous variables were made binary following traditional conventions for reference groups, coding values such as “No” or “Wild Type” as 0. To enhance imputation performance and reduce noise, variables considered irrelevant to modeling, including patient identifiers, upper and lower reference ranges for laboratory values, unknown data flags, and non-binary or non-continuous variables, were excluded during preprocessing, leaving 124 variables.

Step 2 specifies the SL-based imputation function and its interim preprocessing steps. This function accepts 4 parameters. The target_vars parameter takes a vector of binary or continuous variables for imputation. We recommend ordering these variables from least to most missingness, or from least to most importance, as the function’s iterative imputation scheme can leverage earlier imputations to improve subsequent estimates. The learners and screeners parameters are user-defined character vectors listing the base learners and screening wrappers to be incorporated into the SL calls. A comprehensive list of available wrappers can be retrieved using the listWrappers function in the SuperLearner package.[13] The final parameter, df_input, specifies the input dataset, formatted according to the structure described in step 1.

In step 3, interim processing, the SL-based imputation function constructs the SL library, removes variables produced by earlier imputations that are irrelevant for subsequent targets, identifies the top predictors for each target variable, handles missingness in those predictors, and identifies the target variable as binary or continuous. An embedded utility function generates all possible combinations of user-specified base learners and screening wrappers. For example, if conditional forest and neural network learners are selected alongside correlation- and t-test-based screeners,

each learner is paired with each screener to create a SL library of four unique algorithm inputs. It is important to note that the function has a built-in method of identifying and ranking top potential predictors based on missingness and correlation, reducing the demand for screener specification. To enhance interpretability in downstream outputs, a mapping of learner names is also established. If a user does not specify any base learning or screening wrappers, a default library of 19 learners is used with no screeners. These learners are listed with a brief description and reference in Supplementary Table S1. While users may incorporate most available base learner wrappers and screeners within the SL framework, doing so may quickly become computationally infeasible, particularly in high-dimensional settings. Predictor selection is performed by first excluding variables generated during previous iterations (e.g., row ID trackers, uncertainty weights, confidence interval outputs) that do not inform estimation. For the remaining candidate predictors, a predictor score is computed as the product of (i) the absolute Pearson correlation with the target variable and (ii) the proportion of observed (non-missing) data, elevating predictors that are both informative and well-represented in the dataset. By default, the top 10 predictors ranked by this score are selected, keeping the next steps informed, parsimonious, and reducing overfitting. The number of predictors to use in modeling is a customizable parameter. Since the SL framework requires complete predictor data and missingness is prevalent in RWD, simple internal imputation is applied, median imputation for continuous variables and mode imputation for binary variables. This simple imputation step is used solely within the function and is not returned in the final output. Target variables with

>80% missingness are flagged, and then the top predictor's and target variable's data is subset for subsequent SL-based methods to reduce dimensionality.

SL-based Binary Imputation Pipeline

Step 4a defines the binary target variable pipeline, encompassing several key processes: (i) class balance checks, (ii) performance benchmarking via MICE, (iii) the core SL prediction process with CV, (iv) optimal threshold selection for classification, (v) resampling for accuracy variability, (vi) imputation of missing data, and (vii) generation of uncertainty weights. A class balance check identifies severely imbalanced binary variables. If one class comprises more than 92.5% of observations, or a class has equal to or less than 20 observations, a flag is triggered, and mode imputation is applied instead of SL-based modeling, as extreme imbalance can cause k-fold cross-validation to fail. In this case, a simple weight is assigned equal to the observed class proportion to reflect reduced uncertainty. For performance benchmarking of MICE, 30% of observed values are randomly masked, and MICE, using 5 imputations, is applied using a predictor matrix built from the input data. This provides a reference for comparison against the SL-based approach. The SL process then performs nested modeling and validation, where each base learner is trained and validated within cross-validation folds to generate predictions. These predictions are then combined by the meta-learner to optimize ensemble performance and minimize error.[13] Within this pipeline, each learner undergoes 10-fold CV to estimate accuracy, and an outer CV loop (via `CV.SuperLearner`) evaluates the entire SL estimation process to provide performance estimates. The model uses the top 10 predictors identified in step 3, with simple internal imputation applied to handle missingness, and optimizes predictions for the area under the ROC

curve (AUC). We next find the optimal threshold for classifying predicted probabilities by maximizing AUC when class imbalance is below 90%. In scenarios of 90-92.5% class imbalance, the F_1-score was prioritized over AUC as it better reflects the trade-off between precision and recall and provides a more informative measure of performance for the minority class. Estimated probabilities obtained from the CV process are resampled 1000 times with replacement. This resampling enables calculation of confidence intervals for balanced accuracy, quantifying between subject variability in classification performance by applying the same threshold to each bootstrap replicate. Additionally, by resampling from the predicted distribution, computational feasibility is increased while accounting for uncertainty to an extent. The `CV.SuperLearner` object cannot directly estimate unseen data, we therefore rerun the SL process using `snowSuperLearner` with identical arguments to generate imputations for truly missing values.

SL-based Binary Imputation Uncertainty

A similar 1000 resampling scheme is applied and the 2.5th and 97.5th percentiles of these samples form 95% observation-specific uncertainty intervals, reflecting variability in potential imputed predicted probabilities. The previously determined optimal threshold is applied to classify the original predicted probability and the lower 2.5th and upper 97.5th predicted probabilities. Finally, uncertainty weights are assigned to each imputed observation's predicted probabilities to reflect classification confidence. These weights are calculated, as seen in Equation 1, by taking the absolute difference between the predicted probability of each observation, $\hat{p}_i$ and the optimal threshold τ, scaled by the maximum possible distance to that threshold.

$$\text{weight}_i = |\hat{p}_i - \tau| \,/\, \max(\tau, 1-\tau) \quad (1)$$

Predicted probabilities closer to the threshold receive lower weights, reflecting higher uncertainty, while those further away approach a weight of 1, normalizing uncertainty values between 0 and 1 while observed data is assigned a weight of 1.

SL-based Continuous Imputation Pipeline

Step 4b describes the continuous target variable pipeline, which mirrors the binary pipeline but is adapted for continuous outcomes. After selecting the top 10 predictors and applying simple internal imputations to handle missingness among these predictors, a MICE-based comparison is conducted with specifications consistent with Step 4a. The SL CV process is then performed using 10-fold CV, a Gaussian family, and a non-negative least squares (NNLS) loss function to estimate performance for each base learner and the ensemble. The final SL ensemble model is subsequently trained on all observed data for the selected predictors and used to impute missing values in the target variable, producing direct value imputations rather than probabilities.

SL-based Continuous Imputation Uncertainty

Imputation uncertainty for each imputed observation is quantified using a residual-based bootstrap approach, generating 1000 resampled imputations by adding resampled residuals to model-predicted values. To account for heteroskedasticity, the bootstrap resampling process selects the 50 residuals closest to each predicted value based on Euclidean distance from the observed target values, ensuring that resampled errors are drawn from observations with similar outcome magnitudes. The 2.5th and 97.5th percentiles of these samples form 95% observation-specific uncertainty intervals, reflecting variability in potential imputed values. Observation-specific uncertainty

weights for continuous variables are shown in equation 2, calculated as the inverse of 1 plus the median absolute deviation (MAD) of the bootstrap sample distribution per observation, B_i, normalized by the mean predicted value of the observation-specific bootstrap distribution $\bar{p}_i$, thereby assigning higher weights to imputed values with lower relative variability.

$$\text{weight}_i = 1 \,/\, (1 + (\text{MAD}(B_i) \,/\, \bar{p}_i)) \tag{2}$$

These outputs provide both a point estimate and observation-specific uncertainty intervals, enabling other analyses to account for uncertainty and down-weight less certain imputations.

Performance Assessment and Application

Step 5 involves data output, iterative imputation, and comprehensive performance assessment. The SL-based imputation function produces an interim dataset in which each target variable is sequentially replaced by its imputed counterpart, supplemented with observation-specific uncertainty weights. The output additionally includes observation-specific 95% estimation uncertainty intervals that capture model uncertainty, enabling downstream analyses to consider a plausible range of imputed values rather than relying on single-point estimates. This augmented dataset is iteratively updated as each variable in the input vector is imputed, ultimately yielding a complete dataset supported with uncertainty-aware outputs. In addition to the final imputed data, the proposed imputation function returns a suite of diagnostic metrics and visualizations, providing strong transparency into the imputation process over standard methods like MICE. These include visualizations of top predictor scores for each target variable, flags for variables with >80% missingness, benchmark MICE performance

243 metrics (confusion matrices and F_1-scores for binary variables, RMSE, root mean
244 square error, for continuous variables), and illustration of average CV risk estimates for
245 every base learner in the SL ensemble (risk being RMSE for continuous variables, AUC
246 for binary variable). The function also reports ensemble-level performance measures,
247 balanced accuracy with bootstrap-based confidence intervals for binary targets, optimal
248 thresholds for classification, and visualization of average ensemble weights based on
249 nested CV folds illustrating each learner's contribution to the SL model. Predicted
250 values or probabilities are visualized along with resampling distributions. Furthermore,
251 computing time is provided for both the independent MICE run and the final SL
252 ensemble that is trained for imputation. Collectively, these outputs deliver not only a fully
253 imputed dataset but also a novel, uncertainty-aware, and performance-benchmarked
254 framework, offering users enhanced interpretability and awareness in the resulting
255 imputations compared to conventional MVI approaches.

256 We summarize general descriptives of our merged 1641 patient cohort via counts
257 for categorical variables and median and IQR for continuous variables. Additionally we
258 describe missingness of the entire cohort leveraging the `naniar` R package.[25]
259 Missingness descriptives include missingness prevalence and visualizations to aid in
260 identification of missingness patterns. In our initial testing application of this novel SL-
261 based imputation function, we selected variables that have been previously identified in
262 analyses of our data as high priority for upfront risk stratification and clinical relevance.[23]
263 These binary flag variables include NPM1 mutation, IDH1 mutation, IDH2 mutation,
264 good risk cytogenetics as based on SWOG,[26] inv(3) abnormality, TP53 mutation, -17
265 abnormality, t(9;11) abnormality, -5 abnormality, complex cytogenetics, and FLT3-ITD

mutation. Given that these variables do not include a continuous data type, we additionally selected the laboratory-measured lactate dehydrogenase (LDH) value variable for continuous variable evaluation due to its prognostic importance.[27] We ordered all selected variables from least to most missingness and supplied this vector to the MVI function, reporting summarizations of the previously mentioned performance metrics. In our application, we used the default 19 learner algorithms library. This diverse selection of learners ensures that the SL library balances simple benchmarks, linear and nonlinear models, tree-based approaches, penalized regression, and modern machine learning methods, enhancing flexibility and robustness in the estimation procedure and accounting for algorithmic uncertainty. To enable deeper insight into the SL-based imputation's behavior and output data, we further isolated TP53 mutation and LDH to conduct a more detailed evaluation of SL predictive performance. This deeper comparison examined the performance of individual base learners, highlighting one algorithm that emerged as the top-performing single learner, referred to as the discrete learner. For SL processes, parallelization was configured using the `parallel` R package, and for resampling parallelization was configured using the `future` R package,[24] both leveraging maximum cores available. All statistical analyses were performed in R version 4.4.1.[28]

Results

Descriptive Results

General cohort and target variable descriptive statistics are available in Table 1. Most of the 1641 patient cohort were male (703, 57.16%), White (1211, 73.8%), not Hispanic or Latino (1302, 79.34%), deceased at follow-up (1042, 63.5%), on the

289 **Table 1.** Merged cohort and target variable descriptive results and missingness.

Variable and Type	Count (%, n=1641)	Missingness Count and Ratio
Categorical		
Sex		0%
Female	703 (42.84)	
Male	938 (57.16)	
Race		201 (12.25)
Asian	17 (1.04)	
Black	76 (4.63)	
Other	136 (8.29)	
White	1211 (73.80)	
Ethnicity		256 (15.60)
Hispanic or Latino	78 (4.75)	
Not Hispanic or Latino	1302 (79.34)	
Not Reported	5 (0.30)	
Secondary AML		0%
No	1076 (65.57)	
Yes	565 (34.43)	
Vital Status		0%
Censored To Last Known Alive Date	599 (36.50)	
Deceased	1042 (63.50)	
Treatment		0%
7+3	187 (11.40)	
Ven/Aza	1454 (88.60)	
Data Source		0%
CU Cohort	546 (33.27)	
Real World Cohort	1095 (66.73)	
Good Risk Cytogenetics	88 (5.36)	3 (0.18)
Complex Cytogenetics	269 (16.39)	38 (2.32)
Inv(3) abnormality	21 (1.28)	362 (22.06)
T(9;11) abnormality	17 (1.04)	362 (22.06)
-5 abnormality	182 (11.09)	367 (22.36)
-17 abnormality	109 (6.64)	368 (22.43)
IDH2 Mutation	182 (11.09)	584 (35.59)
IDH1 Mutation	104 (6.34)	592 (36.08)
NPM1 Mutation	193 (11.76)	617 (37.60)
FLT3-ITD Mutation	123 (7.50)	700 (40.27)
TP53 Mutation	249 (15.17)	744 (45.34)
Continuous	**Median (IQR)**	
Age at Enrollment	74 (67-79)	0%
LDH	285 (190-487)	889 (54.17%)

Abbreviations: Acute Myeloid Leukemia (AML), University of Colorado (CU), Lactate Dehydrogenase (LDH).

290 Ven/Aza treatment regime (1454, 88.60%), and de novo AML (1076, 65.57%). Median
291 age at enrollment was 74 (IQR 67-79). The target variables had missingness ratios that
292 ranged from as low as 0.18% for good risk cytogenetics (3/1641) to high missingness of
293 54.17% for LDH (889/1641). Supplementary Figure S2 displays missingness ratios
294 across all variables in the merged dataset, missingness across all variables can be
295 summarized as highly diverse. Supplementary Table S2 shows all variables categorized
296 as either binary or numeric and then sorted by missingness thresholds for a more
297 comprehensive description of missingness. Broadly, most variables were binary and
298 had 40-80% missingness. Supplementary Figures S3-5 demonstrate missingness
299 feature categories of cytogenetics, mutations, and lab values, respectively. Taken
300 together, these missingness figures indicate primarily arbitrary and block patterns of
301 missingness.

302 *Primary Results*

303 Table 2 details the performance for nearly all binary target variables of interest,
304 comparing the performance of the SL-based ensemble approach to MICE. Variables
305 inv(3) and t(9;11) abnormalities are not shown as they had high class imbalance and as
306 a result were defaulted to the simple imputation pipeline. Performance measures
307 included balanced accuracy, F_1-score, and computational time. The SL-based ensemble
308 approach had a greater mean balanced accuracy, 89.04%, than MICE, 80.75% and had
309 superior balanced accuracy for 7 out of 9 variables, with 2 out of 9 variables having
310 comparable (<1% difference) balanced accuracy. The mean F_1-score for MICE, 94.34%,
311 and the SL-based ensemble approach, 96.25%, were more mixed and slightly favoring
312 the SL-based ensemble approach. Lastly, computational time was on average longer for

313 **Table 2.** Performance of all binary target variables of interest compared between MICE
314 and the SL-based imputation function approach.

Variable	MICE Balanced Accuracy (%)	SL Balanced Accuracy (95% CI[a], %)	MICE F_1-score (%)	SL F_1-score (%)	MICE Time (seconds)	SL-based Time[b] (seconds)
Good Risk Cytogenetics	92.68	96.93 (94.11-99.15)	98.91	99.61	0.79	137.67
Complex Cytogenetics	95.04	98.88 (98.45-99.26)	98.08	98.86	0.72	203.89
-5 abnormality	63.1	82.69 (79.89-85.36)	90.47	87.82	0.42	190.88
-17 abnormality	58.03	57.58 (54.38-61.16)	92.53	95.98	0.55	116.01
IDH2 Mutation	69.63	94.93 (92.87-96.63)	88.93	97.52	0.51	131.08
IDH1 Mutation	95.99	95.19 (92.05-97.69)	98.57	99.47	0.46	178.63
NPM1 Mutation	86.69	93.65 (91.54-95.71)	95.95	97.22	0.48	123.61
TP53 Mutation	91.78	95.27	94.68	98.02	0.44	127.77
FLT3-ITD Mutation	87.22	95.67 (93.12-97.81)	90.91	91.71	0.53	95.65

[a] 95% confidence intervals represent the SL CV loop's 1000 resamples with classified via the optimal threshold for measures of precision for balanced accuracy.
[b] SL-based time refers to only the SL model trained for prediction and not the CV loop for performance measures.
Abbreviations: Multiple Imputation by Chained Equations (MICE), Super Learner (SL), Cross-Validation (CV).

315 the SL ensemble trained for imputation, 145 seconds, compared to MICE, 0.5 seconds.
316 In our specific performance analysis of the SL-based ensemble and MICE approaches
317 of TP53 mutation and LDH imputation, we found that the SL-based approach was
318 mostly superior to MICE. For TP53 mutation imputation, MICE had a balanced accuracy
319 of 91.8%, sensitivity 93.7%, specificity 89.9%, and an F_1-score of 94.7% while the SL-
320 based ensemble approach had respective values of 95.3%, 99.4%, 91.2%, and 98.0%.
321 Compared to the SL-based ensemble approach, the discrete learner of the ensemble
322 had comparable performance (balanced accuracy 95.3%; sensitivity 99.4%; specificity
323 91.2%; F_1-score 97.9%) and another select algorithm for comparison, GLM interaction,

had worse performance (balanced accuracy 91.7%; sensitivity 96.6%; specificity 87.8%; F_1-score 95.8%). Supplementary Figure S6 shows the top 10 predictors of the TP53 mutation variable that were used in the SL-based approach, indicating that several previous target variables that were imputed supported the imputation of TP53 mutation. Figure 2 shows the CV-specific base learner risks for TP53 mutation, suggesting that most base learners performed well (>0.9 AUC) with conditional forest having the absolute minimum risk observed. Figure 3 demonstrates the subsequent average assigned weights of the SL-based ensemble's learners where many of the learners can be perceived to receive comparable weights as the SL technique aims to reduce overfitting. Lastly, for TP53 mutation, Figure 4 shows the predicted probability and corresponding resampled distribution of predicted probabilities based on 1000 resamples. With a calculated optimal threshold of 0.38 and the observed predicted

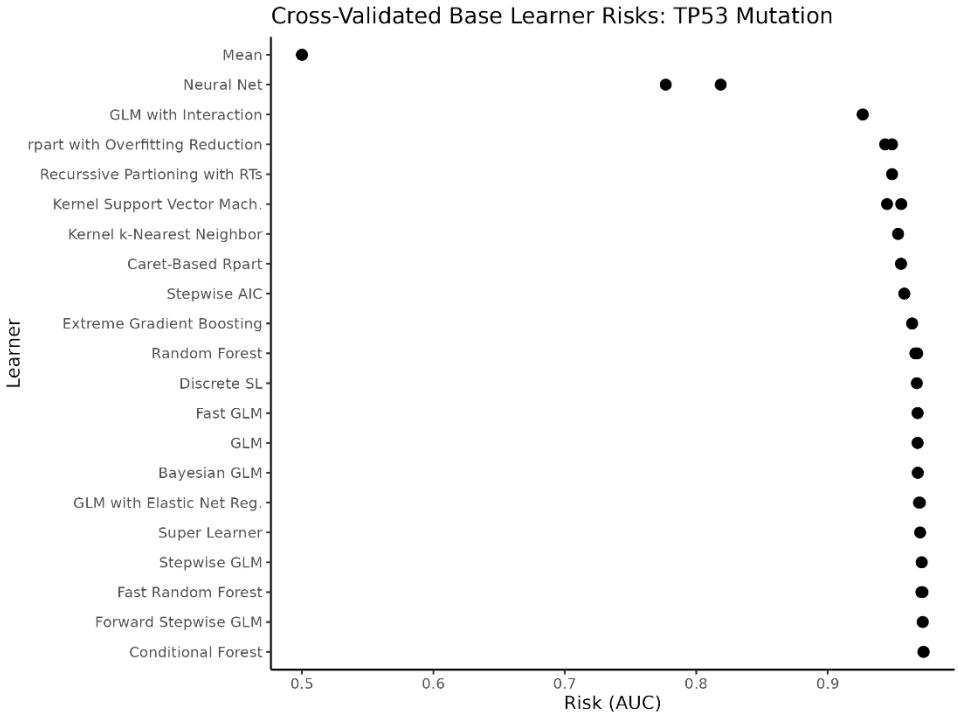

Figure 2. Cross-validated base learner risks for binary TP53 mutation variable prediction, risk measured as AUC. *Note:* multiple points may appear because of multiple default hyperparameter settings. *Abbreviations:* Area Under the Curve (AUC).

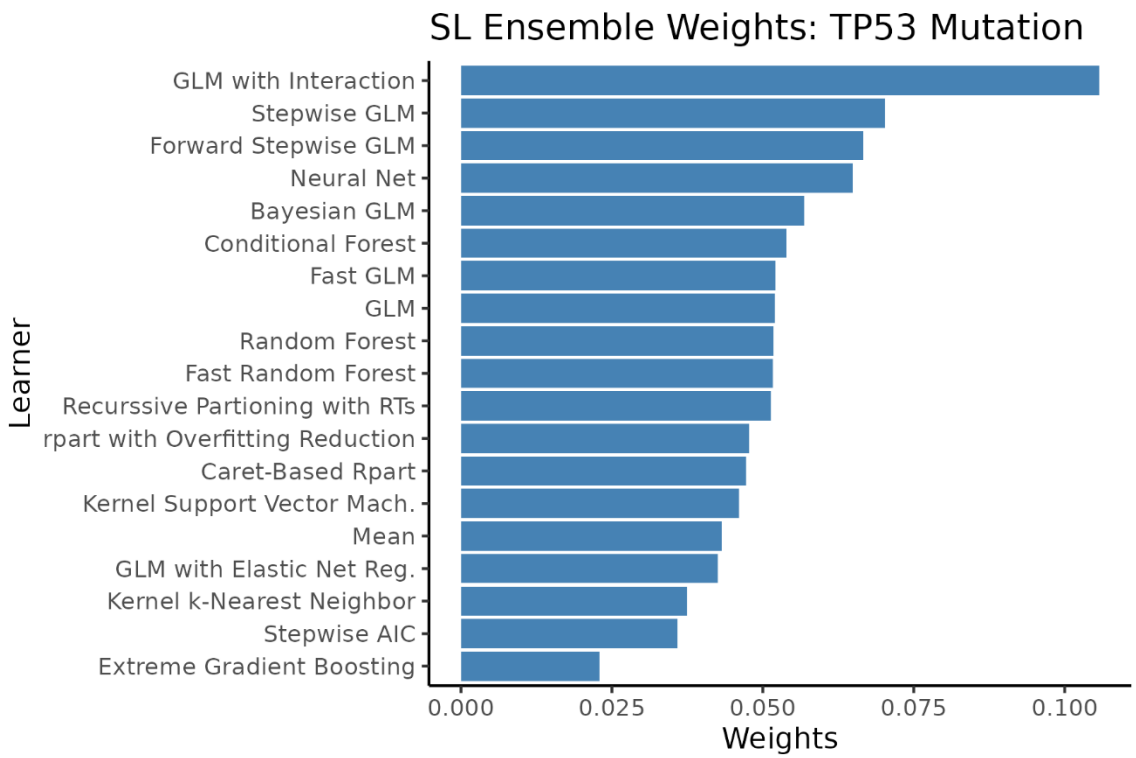

Figure 3. Cross-validated SL ensemble weight averages for binary variable TP53 mutation. *Abbreviations:* Super Learner (SL).

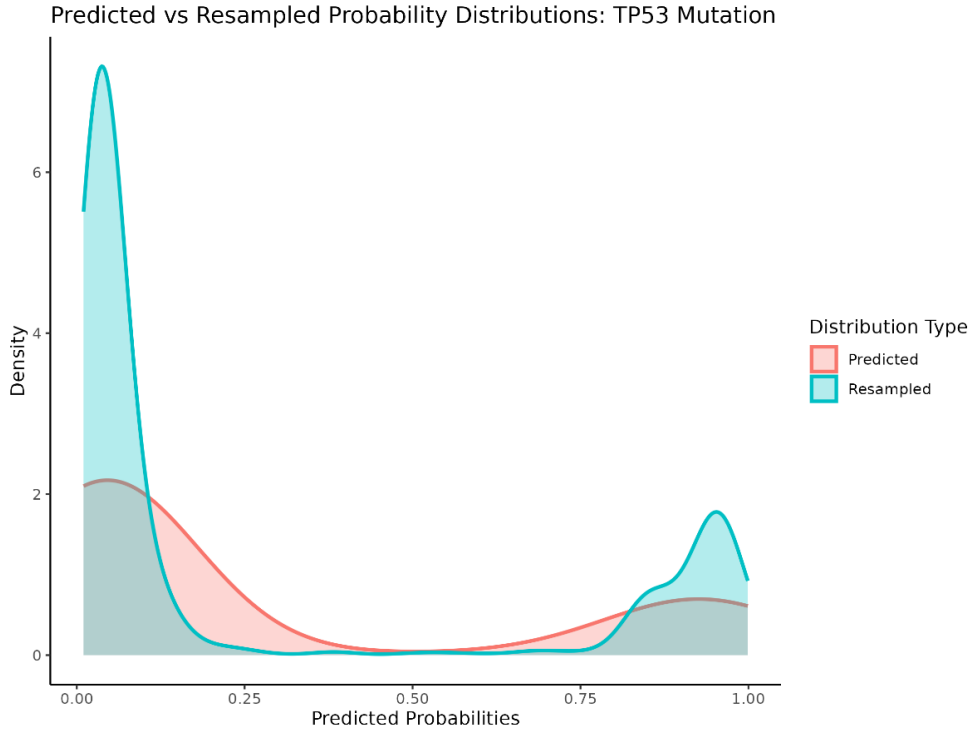

Figure 4. SL-based ensemble's predicted probability density distribution for TP53 mutation binary variable data with overlapping 1000 resampled density distribution. *Abbreviations:* Super Learner (SL).

probabilities, we would estimate that most classifications were 0 and were likely assigned moderate uncertainty weights. This is validated when analyzing the distribution of the TP53 mutation uncertainty weights (median 0.52, IQR 0.45-0.53). With respect to LDH, results were more mixed. MICE had a RMSE of 597, higher than that of the SL-based ensemble's RMSE of 582. Furthermore, the discrete learner of the ensemble performed best with 557 RMSE, and other algorithms including neural network and all GLM algorithms, performed far worse with >5000 RMSE. Supplementary Figure S7 displays the top 10 predictors of the LDH variable that were used in the SL-based approach. Figure 5 shows the average CV-specific base learner risks for LDH. The SL-based ensemble's learner's average weights are shown in Figure 6, supporting a narrower range of base learners unequally. Figure 7 displays the SL

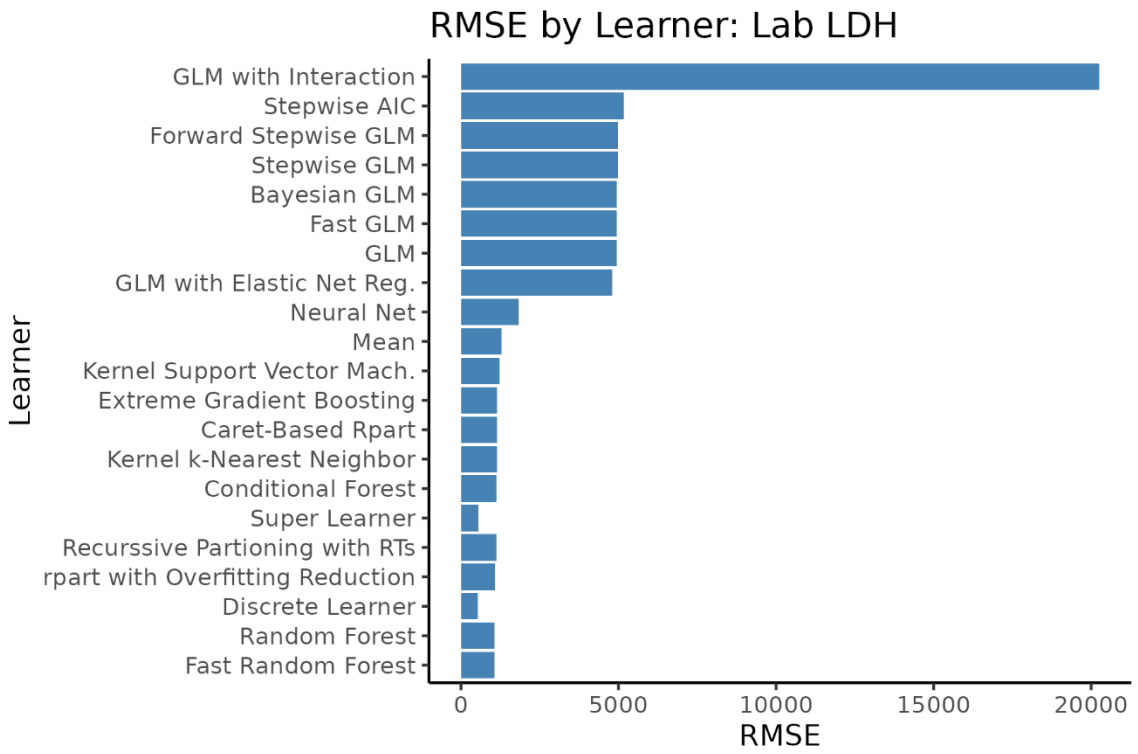

Figure 5. Cross-validated base learner average risks for continuous variable LDH, risk measured as RMSE. *Abbreviations:* Root Mean Square Error (RMSE), Lactate Dehydrogenase (LDH).

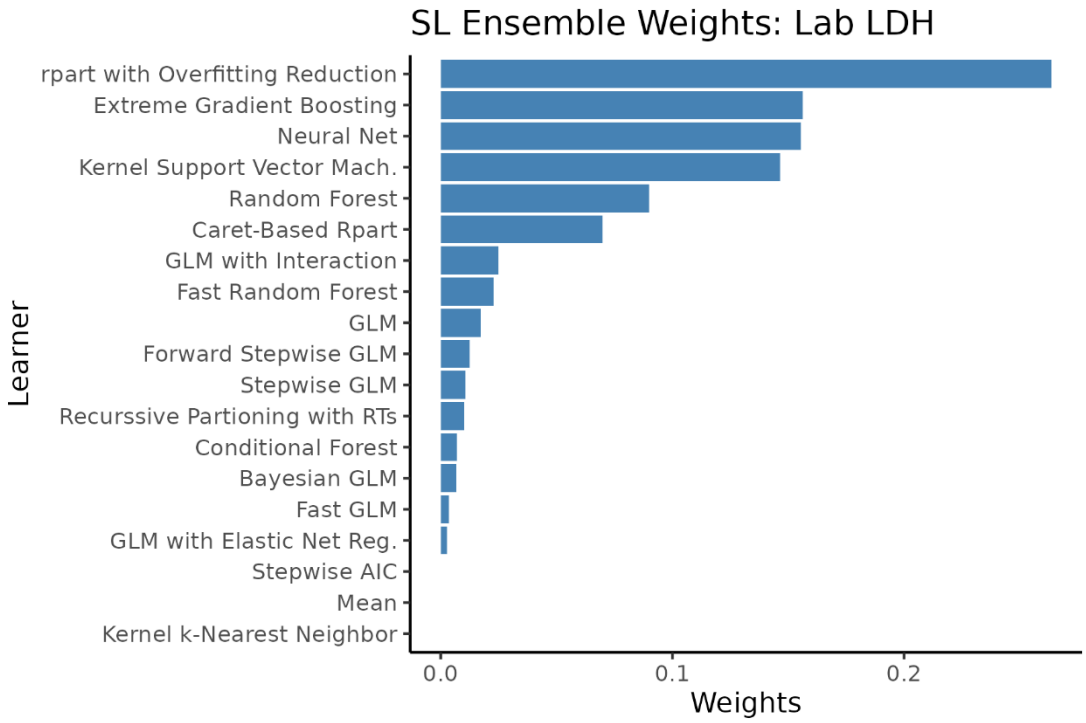

Figure 6. Cross-validated SL-based ensemble average weights for continuous variable LDH. *Abbreviations:* Super Learner (SL), Lactate Dehydrogenase (LDH).

ensemble's predicted values density distribution for LDH and the corresponding density distribution of resampled values based on 1000 resamples, suggesting introduced variation. Subsequent weights indicate moderate to high certainty and relatively small variability of imputations across resamples compared to the observed data's variation (median 0.81, IQR 0.80-0.83). Our computing setup with 48 cores and 188 GB of RAM, with parallelization and a 19-learner library, processed 11 binary targets and a continuous target in approximately 2.5 hours.

Discussion

In this study we evaluated a SL-based ensemble imputation approach for handling missing diagnostic data in a dataset comprised of 1641 newly diagnosed AML patients treated with Ven/Aza or 7+3. Leveraging the SL framework, we demonstrated consistently improved imputation performance over a commonly used statistical

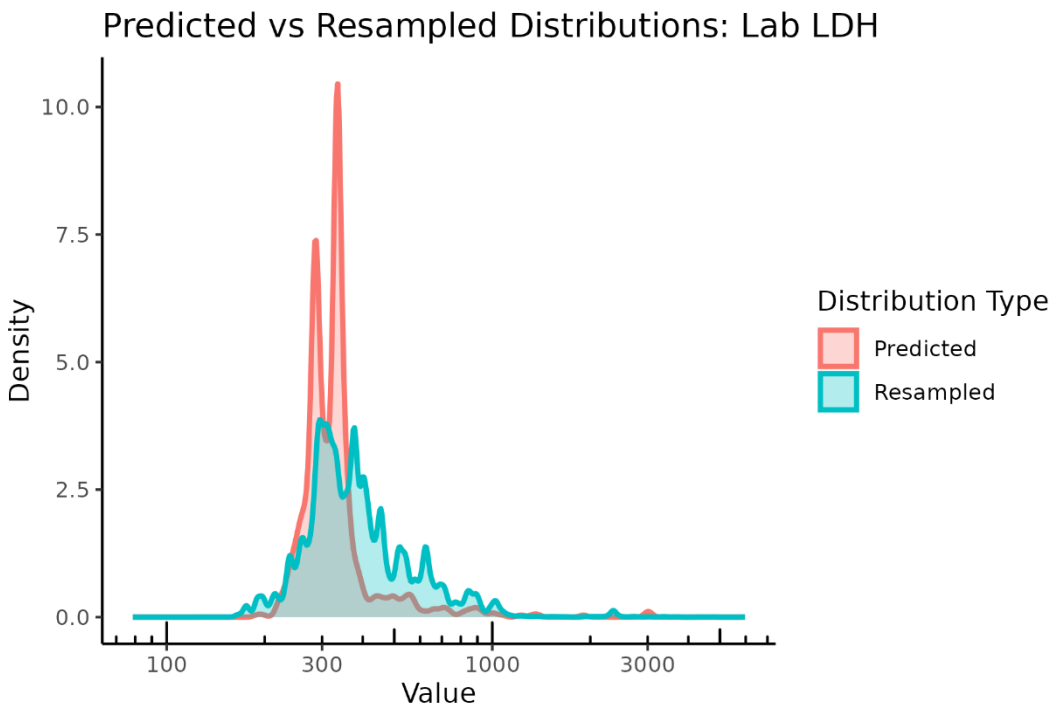

Figure 7. SL-based ensemble's predicted values density distribution for missing LDH continuous variable data, overlayed with the 1000 resampled density distribution. Figure was log transformed after running the function for interpretability. *Abbreviations:* Lactate Dehydrogenase (LDH).

method, MICE, specifically for binary genetic variables, even when they had substantial missingness. On average, the SL-based ensemble achieved higher balanced accuracy (89.0% vs 80.8%) and F1 score (96.3% vs 94.3%) compared to MICE across 9 binary genetic features. In focused evaluations of TP53 mutation and LDH laboratory values, the SL-based ensemble approach led to superior or comparable performance to singular algorithms and uncertainty weights. These results underscore the potential of SL-based methods to enhance data integrity in clinical prediction pipelines, especially in high-dimensional electronic health record derived datasets where conventional single-model approaches may struggle. To our knowledge, this study represents one of the first applications of the SL-based ensemble for missing value imputation in a clinical oncology context. Furthermore, we demonstrate the method's utility by achieving strong

372 performance while introducing novel, observation-level uncertainty estimates which
373 quantify variation, while being delivered through a streamlined, user-friendly function
374 that automates imputation, performance evaluation, and data output.

375 Prior studies into missing value imputation in clinical cohorts have repeatedly
376 shown that standard approaches such as MICE can yield reasonable accuracy when
377 data are moderately complete and missingness mechanisms approximate MAR, but
378 their performance often deteriorates in high-dimensional or block-missing contexts.[7,29]
379 MICE has strong advantages such as being well-documented and easily applied.[30]
380 These strengths are evident by the notable abundance of MICE's application in health
381 research literature, with MICE serving as either the primary imputation method or the
382 benchmark comparison. An earlier 2015 review by Rezvan et al. with narrow search
383 criteria showed the dominance of MICE in imputation methodology in medical research,
384 with 53% (20/38) of research articles reporting methods of imputation using MICE.[31]
385 ML-based methods like random forest and k-NN have improved on MICE's limitations in
386 several ways such as by capturing nonlinear feature relationships.[5,8] In the systematic
387 review of 94 ML performance assessing research articles by Alabadla et al. it was found
388 that across most studies k-NN, random forest, and support vector machine methods
389 performed best, these results would agree with our rankings of ML algorithms by AUC
390 and RMSE. This study also noted though the lack of, and need for, computational time
391 reporting and consistency across performance measures to best evaluate overall
392 performance.[7] Another study that analyzed high dimensional genetic data compared
393 ML-based approaches to MICE and simple mean imputation, also finding that k-NN and
394 random forest typically performed far better than both of these approaches.[32] Such

395 popularity for k-NN, and other algorithms, is explained by a systematic review of 117
396 research articles by Thomas and Rajabi. The review clarifies that these algorithm's easy
397 implementation across knowledge domains make them an attractive option. However,
398 these authors also note that despite such popularity there continues to be no single
399 universal solution to the missing data problem.[33] While ML-based methods have grown
400 immensely in popularity and application frequency since 2015,[7] in part due to
401 improvements over MICE and simple imputation, as previously mentioned these
402 methods remain relatively constrained by the reliance on a single learner that may have
403 varying performance for different missingness scenarios. Ensemble learning
404 approaches may prove an effective next step to overcome these challenges. While
405 using a more typical ensemble strategy compared to our SL-based strategy, our results
406 would concur with those found by Batra et al. who found that a stacked ensemble
407 strategy proved superior to simple imputation and several singular ML algorithms.[12] In a
408 different context than imputation, a 2023 review of 45 articles evaluated ensemble
409 learning approaches in highly researched disease prediction. They concluded that
410 amongst these studies, those that employed the stacking method of ensemble learning
411 performed best compared to other ensemble methods. This work gives us greater
412 confidence in our findings using a SL-based approach.[14] Prior literature using super
413 learning for missing data imputation is sparse, however, one study showed that multiple
414 imputation by super learning yielded superior results compared to traditional methods in
415 a comparative analysis. Albeit, this application was limited by synthetic data, not having
416 quantified uncertainty, and a process that did not automatically generate performance
417 measures.[15] Lastly, many imputation strategies do not automatically provide

observation-level uncertainty estimates. Our results extend prior work by demonstrating that a CV-specific stacking ensemble can more effectively integrate varying base learners to handle diverse variable missingness contexts, and when incorporated into a function, can also output measures of uncertainty.

Despite the promising performance of our SL-based approach, our study has several limitations. First, we employed default SL hyperparameters for all base learners rather than performing extensive tuning. While this decision simplified the pipeline and demonstrated robust gains over MICE by default, it may have missed additional opportunities to further improve performance. Second, the computational burden of training the CV-specific ensemble represents the greatest practical hurdle to routine deployment. However, this cost is justified in settings where imputation accuracy and uncertainty quantification materially impact other modeling tasks in high stakes clinical applications such as risk stratification for patient-enrichment or treatment maintenance decisions related to aggressive diseases like AML. Regarding the computational time of the whole process, it was also longer than MICE but to quantify uncertainty from MICE using traditional approaches of bootstrapping would likely incur even more computational costs. Lastly, prior to ensemble fitting we perform simple imputation on the top 10 predictors for each target, which may propagate initial error and contribute to cumulative uncertainty. We partially mitigate this through our resampling-based uncertainty weights, observations with high variance can be down-weighted in downstream models. However, such bootstrapping-based uncertainty does not capture within-subject variability and may be susceptible to the amount of missing data and representation of the true missing data pattern in the bootstrap samples. More rigorous

initial imputation of predictors, such as using MICE,[34] and incorporation of initial subject-specific uncertainty measures in imputation of target variables would further reduce this concern. Conversely, our SL-based imputation function offers several key strengths. By vectorizing imputation across diverse base learners, it provides a nearly fully automated, scalable framework that consistently outperforms single-method approaches in our application in high-dimensional clinical data. Additionally, while we applied the function to AML data for evaluation, there stands no reason it cannot be applied to other datasets. The model's ability to generate observation-specific uncertainty weights and uncertainty intervals allow end-users to conduct a variety of sensitivity analyses using these values across different settings. The flexibility of the SL framework allows inclusion of new algorithms as they emerge, ensuring that the pipeline can adapt to evolving patterns of data complexity and missingness. Finally, in line with identified gaps in performance assessment in the literature[35] and combined with descriptive statistics of missingness, the SL-based imputation function's performance output allows a user to well understand key aspects of MVI including computational time and common performance measures.

In summary, our work demonstrates that a SL-based imputation function can significantly improve the accuracy and robustness of missing-value estimation in a high-dimensional AML real-world dataset, outperforming the standard MICE approach for most of our target variables while providing observation-level uncertainty metrics. These findings support our central hypothesis that ensemble strategies yield more reliable imputation in complex clinical data. In future deployments we hope to add automated and customized hyperparameter optimization, integrate automatic support for

multinomial and ordinal targets, and further validate its utility in diverse missing data formats and contexts. Future research would benefit from evaluating this function among diverse disease areas, across a greater range of variables, and its impact on optimizing risk stratification strategies. Ultimately, by combining improved performance over traditional methods and flexibility, ensemble learning-derived imputation can enhance downstream modeling by managing missingness' threat to validity.

Abbreviations

AML	Acute myeloid leukemia
AUC	Area under the curve
CU	University of Colorado
CYT	Cytogenetics
IRB	Institutional review board
k-NN	k-Nearest Neighbor
LAB	Laboratory values
LDH	Lactate dehydrogenase
MAR	Missing at random
MAD	Median absolute deviation
MCAR	Missing completely at random
MICE	Multiple imputation by chained equations
ML	Machine learning
MVI	Missing value imputation
NNLS	Non-negative least squares
RMSE	Root mean square error
RWC	Real world cohort
RWD	Real world data
SL	Super Learner
Ven/Aza	Venetoclax plus azacitidine
$\hat{p}_i$	Predicted probability
τ	Optimal threshold
B_i	Bootstrap samples
$\bar{p}_i$	Mean predicted value

References

1. RUBIN DB. Inference and missing data. *Biometrika* 1976; 63: 581–592.

2. Schafer JL, Graham JW. Missing data: our view of the state of the art. *Psychol Methods* 2002; 7: 147–177.

3. Roderick J.A. Little and Donald B. Rubin (2002) Statistical Analysis with Missing Data, 2nd ed. John Wiley & Sons | RTI, https://www.rti.org/publication/roderick-j-little-donald-b-rubin-2002-statistical-analysis-missing-data-2nd-ed-john-wiley-amp-sons-b (accessed 3 July 2025).

4. Petersen I, Welch CA, Nazareth I, et al. Health indicator recording in UK primary care electronic health records: key implications for handling missing data. *Clin Epidemiol* 2019; 11: 157–167.

5. Liu M, Li S, Yuan H, et al. Handling missing values in healthcare data: A systematic review of deep learning-based imputation techniques. *Artificial Intelligence in Medicine* 2023; 142: 102587.

6. Buuren S van, Groothuis-Oudshoorn K. mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software* 2011; 45: 1–67.

7. Alabadla M, Sidi F, Ishak I, et al. Systematic Review of Using Machine Learning in Imputing Missing Values. *IEEE Access* 2022; 10: 44483–44502.

8. Stekhoven DJ, Bühlmann P. MissForest--non-parametric missing value imputation for mixed-type data. *Bioinformatics* 2012; 28: 112–118.

9. Ismail AR, Abidin NZ, Maen MK. Systematic Review on Missing Data Imputation Techniques with Machine Learning Algorithms for Healthcare. *Journal of Robotics and Control (JRC)* 2022; 3: 143–152.

10. Afkanpour M, Hosseinzadeh E, Tabesh H. Identify the most appropriate imputation method for handling missing values in clinical structured datasets: a systematic review. *BMC Med Res Methodol* 2024; 24: 188.

11. Sun Y, Li J, Xu Y, et al. Deep learning versus conventional methods for missing data imputation: A review and comparative study. *Expert Systems with Applications* 2023; 227: 120201.

12. Batra S, Khurana R, Khan MZ, et al. A Pragmatic Ensemble Strategy for Missing Values Imputation in Health Records. *Entropy* 2022; 24: 533.

13. ‘Super Learner In Prediction’ by Eric C. Polley and Mark J. van der Laan, https://biostats.bepress.com/ucbbiostat/paper266/ (accessed 25 June 2025).

14. Laan MJ van der, Polley EC, Hubbard AE. Super Learner. *Statistical Applications in Genetics and Molecular Biology*; 6. Epub ahead of print 16 September 2007. DOI: 10.2202/1544-6115.1309.

15. Carpenito T, Manjourides J. MISL: Multiple imputation by super learning. *Statistical Methods in Medical Research*. Epub ahead of print 5 June 2022. DOI: 10.1177/09622802221104238.

16. Murugesan S, Bhuvaneswaran RS, Nehemiah HK, et al. Feature Selection and Classification of Clinical Datasets Using Bioinspired Algorithms and Super Learner. *Computational and Mathematical Methods in Medicine*. Epub ahead of print 18 May 2021. DOI: 10.1155/2021/6662420.

17. Li S, Zhang Y, Fu K, et al. Machine learning ensemble meets clinical practice: developing a real-world risk prediction model for metabolic syndrome using super learner and scorecard approaches. *Journal of Advanced Research*. Epub ahead of print 27 June 2025. DOI: 10.1016/j.jare.2025.06.072.

18. Petersen ML, LeDell E, Schwab J, et al. Super Learner Analysis of Electronic Adherence Data Improves Viral Prediction and May Provide Strategies for Selective HIV RNA Monitoring. *J Acquir Immune Defic Syndr* 2015; 69: 109–118.

19. Jin W, Hao W, Shi X, et al. Using Multi-Modal Electronic Health Record Data for the Development and Validation of Risk Prediction Models for Long COVID Using the Super Learner Algorithm. *Journal of Clinical Medicine* 2023; 12: 7313.

20. Zhu X, Zhang P, Jiang H, et al. Using the Super Learner algorithm to predict risk of major adverse cardiovascular events after percutaneous coronary intervention in patients with myocardial infarction. *BMC Med Res Methodol* 2024; 24: 59.

21. Ma X, Long L, Moon S, et al. Comparison of Population Characteristics in Real-World Clinical Oncology Databases in the US: Flatiron Health, SEER, and NPCR. 2023; 2020.03.16.20037143.

22. Birnbaum B, Nussbaum N, Seidl-Rathkopf K, et al. Model-assisted cohort selection with bias analysis for generating large-scale cohorts from the EHR for oncology research. Epub ahead of print 13 January 2020. DOI: 10.48550/arXiv.2001.09765.

23. Islam N, Reuben JS, Dale JL, et al. A simplified risk model for pretreatment stratification of newly diagnosed acute myeloid leukemia patients treated with venetoclax and azacitidine. 2024; 2024.12.02.24318344.

24. Bengtsson H. future: Unified Parallel and Distributed Processing in R for Everyone, https://cran.r-project.org/web/packages/future/index.html (2025, accessed 29 July 2025).

25. Tierney N, Cook D, McBain M, et al. naniar: Data Structures, Summaries, and Visualisations for Missing Data, https://cran.r-project.org/web/packages/naniar/index.html (2024, accessed 30 July 2025).

543 26. Slovak ML, Kopecky KJ, Cassileth PA, et al. Karyotypic analysis predicts outcome of
544 preremission and postremission therapy in adult acute myeloid leukemia: a Southwest
545 Oncology Group/Eastern Cooperative Oncology Group study. *Blood* 2000; 96: 4075–4083.

546 27. Islam N, Reuben JS, Dale J, et al. Predictive Models for Long Term Survival of AML
547 Patients Treated with Venetoclax and Azacitidine or 7+3 Based on Post Treatment Events
548 and Responses: Retrospective Cohort Study. *JMIR Cancer* 2024; 10: e54740.

549 28. R Development Core Team. R: A Language and Environment for Statistical Computing,
550 https://cran.r-project.org/ (2025).

551 29. Jerez JM, Molina I, García-Laencina PJ, et al. Missing data imputation using statistical and
552 machine learning methods in a real breast cancer problem. *Artificial Intelligence in*
553 *Medicine* 2010; 50: 105–115.

554 30. Zhang Z. Multiple imputation with multivariate imputation by chained equation (MICE)
555 package. *Ann Transl Med* 2016; 4: 30.

556 31. Hayati Rezvan P, Lee KJ, Simpson JA. The rise of multiple imputation: a review of the
557 reporting and implementation of the method in medical research. *BMC Med Res Methodol*
558 2015; 15: 30.

559 32. Petrazzini BO, Naya H, Lopez-Bello F, et al. Evaluation of different approaches for missing
560 data imputation on features associated to genomic data. *BioData Mining* 2021; 14: 44.

561 33. Thomas T, Rajabi E. A systematic review of machine learning-based missing value
562 imputation techniques. *Data Technologies and Applications* 2021; 55: 558–585.

563 34. Mera-Gaona M, Neumann U, Vargas-Canas R, et al. Evaluating the impact of multivariate
564 imputation by MICE in feature selection. *PLOS ONE* 2021; 16: e0254720.

565 35. Lin W-C, Tsai C-F. Missing value imputation: a review and analysis of the literature
566 (2006–2017). *Artif Intell Rev* 2020; 53: 1487–1509.

567 36. Gelman A, Su Y-S. arm: Data Analysis Using Regression and Multilevel/Hierarchical
568 Models. 2007; 1.14-4.

569 37. Kuhn M. caret: Classification and Regression Training. 2007; 7.0–1.

570 38. Strobl C, Boulesteix A-L, Zeileis A, et al. Bias in random forest variable importance
571 measures: Illustrations, sources and a solution. *BMC Bioinformatics* 2007; 8: 25.

572 39. R Development Core Team. R: A Language and Environment for Statistical Computing,
573 https://cran.r-project.org/ (2010).

574 40. Regularization Paths for Generalized Linear Models via Coordinate Descent | Journal of
575 Statistical Software, https://www.jstatsoft.org/article/view/v033i01 (accessed 29 July 2025).

41. Friedman J, Hastie T, Tibshirani R, et al. glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models. 2008; 4.1-10.

42. Mouselimis L, Contributor) MP (Github: G. KernelKnn: Kernel k Nearest Neighbors, https://cran.r-project.org/web/packages/KernelKnn/index.html (2023, accessed 29 July 2025).

43. Karatzoglou A, Smola A, Hornik K, et al. kernlab - An S4 Package for Kernel Methods in R. *Journal of Statistical Software* 2004; 11: 1–20.

44. Venables WN, Ripley BD. *Modern Applied Statistics with S*. New York, NY: Springer. Epub ahead of print 2002. DOI: 10.1007/978-0-387-21706-2.

45. Breiman L. Random Forests. *Machine Learning* 2001; 45: 5–32.

46. original) LB (Fortran, original) AC (Fortran, port) AL (R, et al. randomForest: Breiman and Cutlers Random Forests for Classification and Regression, https://cran.r-project.org/web/packages/randomForest/index.html (2024, accessed 29 July 2025).

47. Wright MN, Ziegler A. ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. *Journal of Statistical Software* 2017; 77: 1–17.

48. Therneau T, Atkinson B, port BR (producer of the initial R, et al. rpart: Recursive Partitioning and Regression Trees, https://cran.r-project.org/web/packages/rpart/index.html (2025, accessed 29 July 2025).

49. Enea M, LTD) RM (on behalf of DmwA, LTD) TK (on behalf of DmwA. speedglm: Fitting Linear and Generalized Linear Models to Large Data Sets, https://cran.r-project.org/web/packages/speedglm/index.html (2023, accessed 29 July 2025).

50. Ripley B, Venables B, Bates DM, et al. MASS: Support Functions and Datasets for Venables and Ripley's MASS, https://cran.r-project.org/web/packages/MASS/index.html (2025, accessed 29 July 2025).

51. Chen T, He T, Benesty M, et al. xgboost: Extreme Gradient Boosting, https://cran.r-project.org/web/packages/xgboost/index.html (2025, accessed 29 July 2025).

Supplementary Material

Table S1. Default, and used, SL algorithm library with descriptions.

Learner	Description
SL.bayesglm	Bayesian generalized linear modeling for estimating linear relationships between predictors and outcomes, incorporating prior distributions to stabilize estimates.[36]
SL.caret.rpart	Recursive partitioning via the *caret* package, building decision trees by recursively splitting data into homogeneous groups for prediction.[37]
SL.cforest	Conditional inference forests from the *party* package, combining multiple conditional inference trees with resampling to reduce variable selection bias.[38]
SL.glm	Standard generalized linear model from base R.[39]
SL.glm.interaction	Generalized linear model including specified interaction terms to capture variable interplay.[39]
SL.glmnet	Penalized regression with elastic net regularization, balancing LASSO and ridge penalties for improved prediction and feature selection in high-dimensional settings.[40,41]
SL.kernelKnn	Kernelized k-nearest neighbors using flexible distance metrics to identify local patterns in predictor space [42].
SL.ksvm	Support vector machines from the *kernlab* package, capable of handling nonlinear classification or regression through hyperplane-based separation.[43]
SL.mean	Simple mean imputation serving as a baseline for comparison with more complex learners.
SL.nnet	Feed-forward neural network modeling complex nonlinear relationships between predictors and outcomes.[44]
SL.randomForest	Classic random forest implementation that builds ensembles of classification or regression trees using random subsets of data and predictors.[45,46]
SL.ranger	High-speed random forest implementation optimized for large, high-dimensional datasets.[47]
SL.rpart	Recursive partitioning and regression trees for interpretable, hierarchical data modeling.
SL.rpartPrune	Modified version of *rpart* that applies pruning to reduce overfitting and improve generalizability.[48]
SL.speedglm	Optimized generalized linear modeling for large datasets, improving computational performance over standard GLMs [49].
SL.step	Stepwise regression for variable selection using both forward and backward selection.[44,50]
SL.step.forward	Stepwise regression using forward selection only [44,50].
SL.stepAIC	Stepwise regression based on Akaike's Information Criterion via the *MASS* package.[44,50]
SL.xgboost	Gradient boosting machine implementation that sequentially builds trees to correct previous errors, often achieving strong predictive performance in tabular data.[51]

608 **Table S2.** Merged cohort common variable's missingness ratio categories.

Missingness Ratio Category	Binary Variables	Continuous Variables
0%	20	3
<5%	5	0
5-20%	0	0
20-40%	25	10
40-80%	41	10
>80%	9	1

609 *Note:* Variables with >80% missingness include CD36 flow cytometry, FLT3 other PCR, IDH2
610 other PCR, FISH DEK rearrangement, IDH1 other PCR, FISH BCRABL rearrangement, peripheral blood blasts, NMP1 PCR, CEBPA PCR, and JAK2 other PCR. *Abbreviations:* University of Colorado (CU), Lactate Dehydrogenase (LDH), Polymerase Chain Reaction (PCR), Fluorescence In Situ Hybridization (FISH).

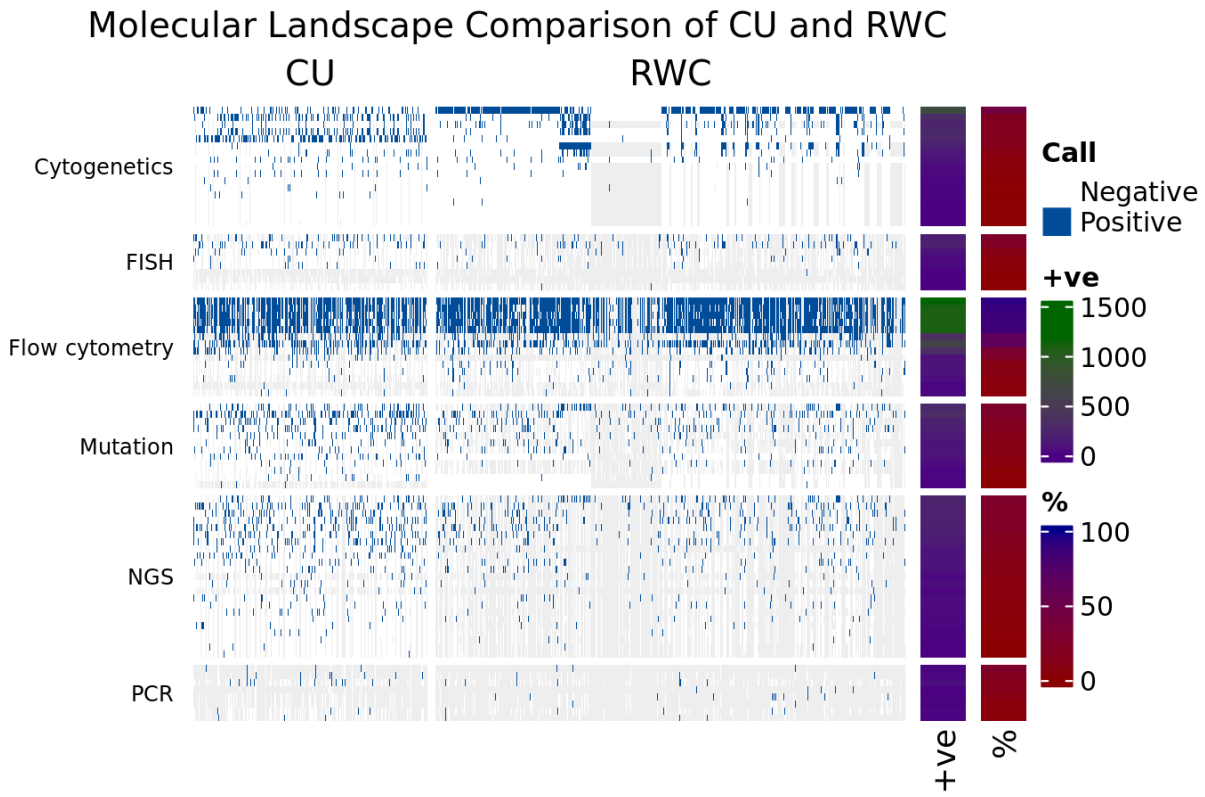

Figure S1. CU and RWC merged cohort's OncoPlot, representing AML phenotypic and genotypic features per-patient. +ve refers to the number of patients with positive features and the % refers to that positive feature proportion. Light grey regions represent missing data. *Abbreviations:* University of Colorado (CU), Real World Cohort (RWC).

611

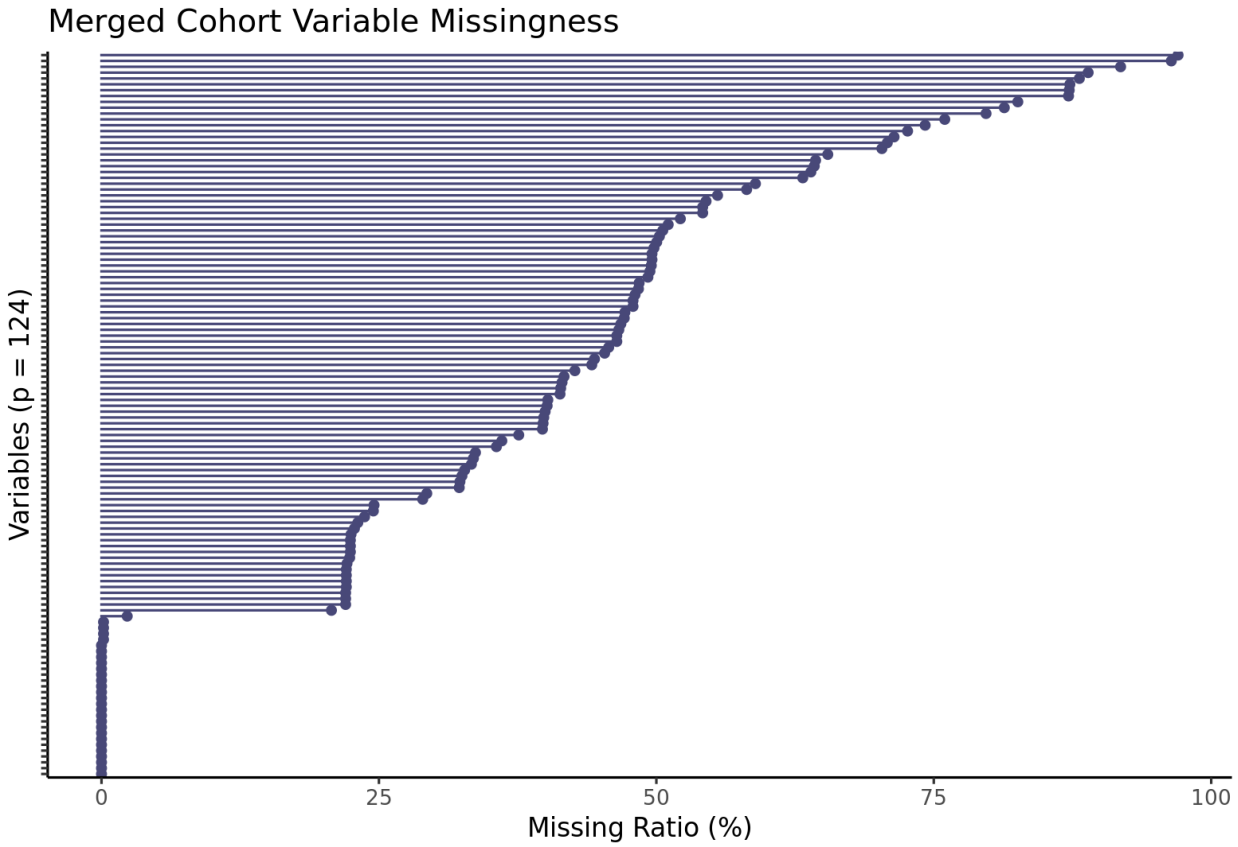

Figure S2. CU and RWC merged cohort's variable missingness ratios. *Abbreviations:* University of Colorado (CU), Real World Cohort (RWC).

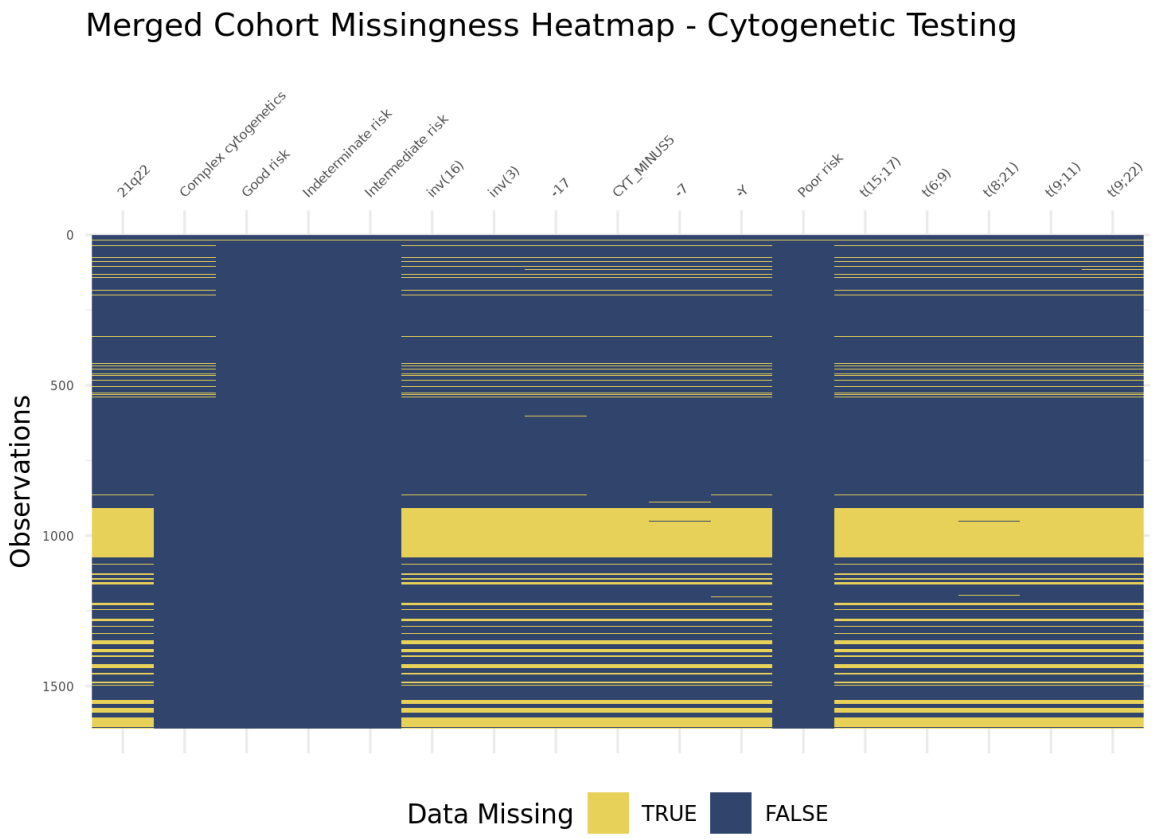

Figure S3. CU and RWC merged cohort's missingness heatmap for all observations and binary cytogenetics variables. *Abbreviations:* University of Colorado (CU), Real World Cohort (RWC), Cytogenetics (CYT).

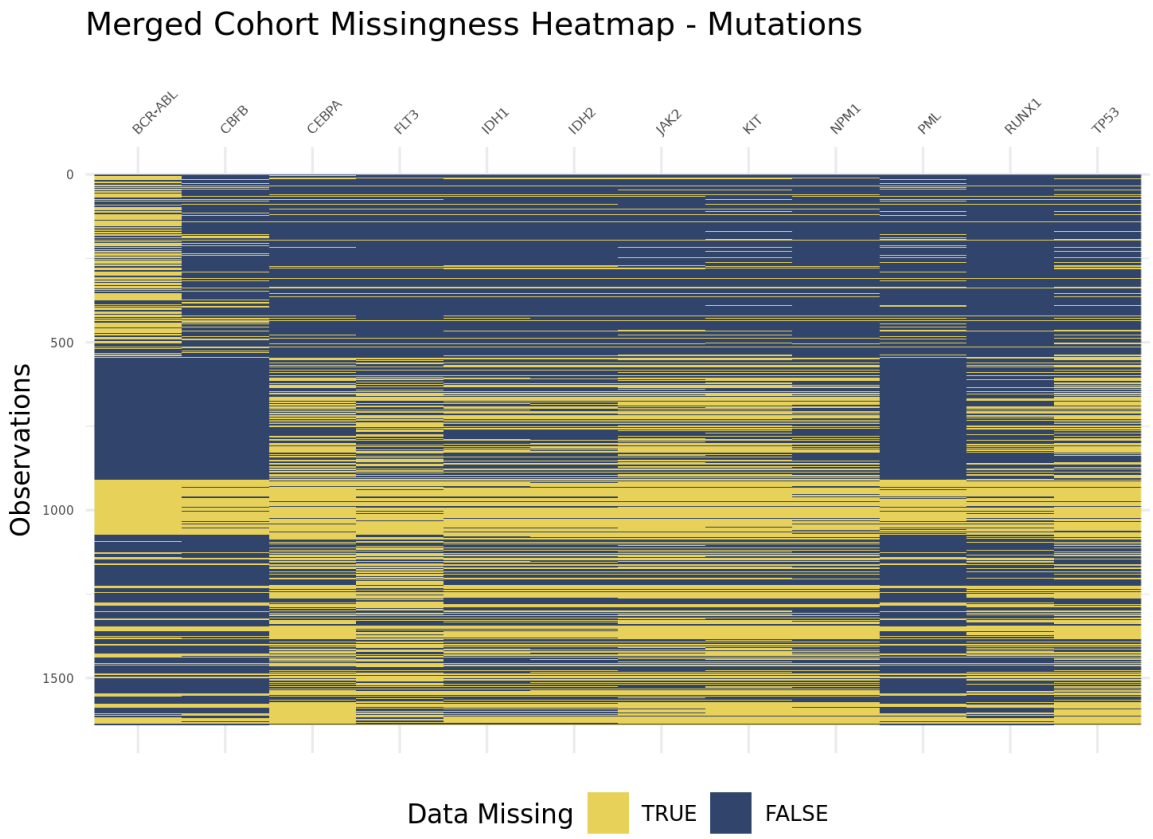

Figure S4. CU and RWC merged cohort's missingness heatmap for all observations and binary mutation variables. *Abbreviations:* University of Colorado (CU), Real World Cohort (RWC).

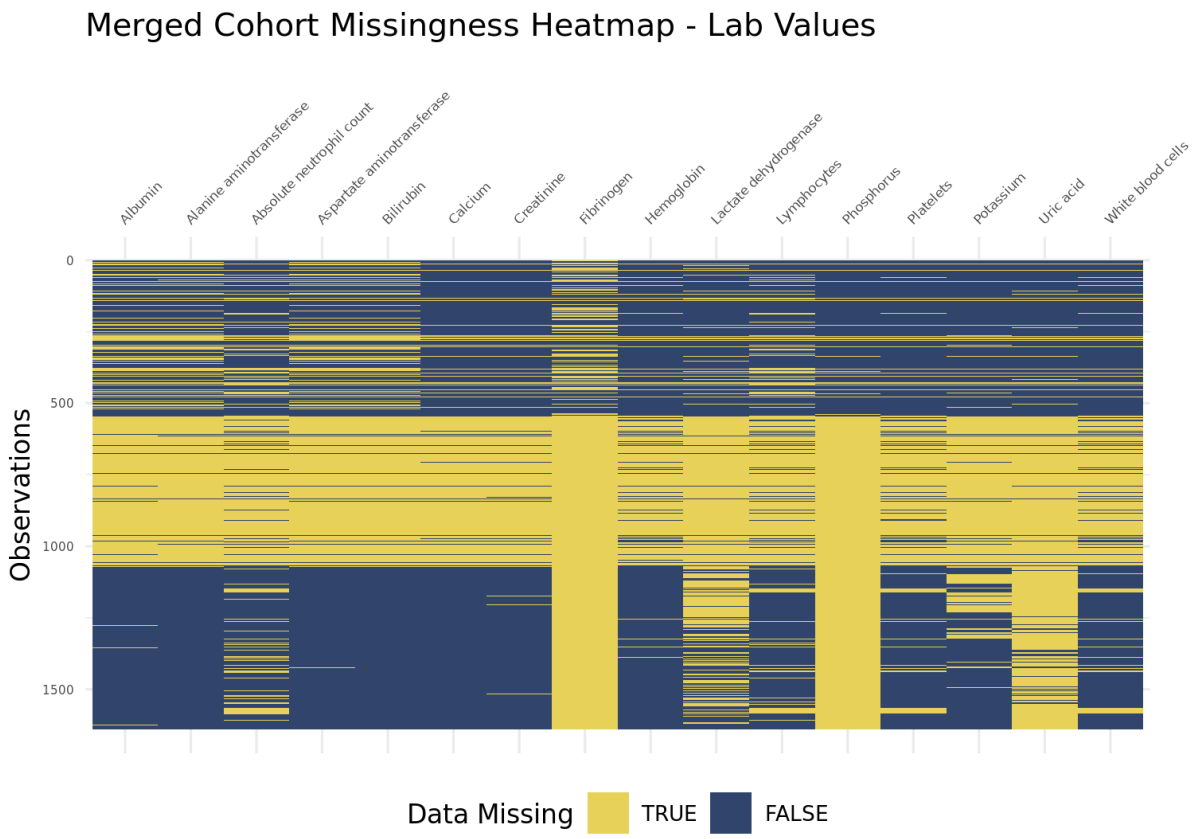

Figure S5. CU and RWC merged cohort's missingness heatmap for all observations and continuous laboratory values variables. *Abbreviations:* University of Colorado (CU), Real World Cohort (RWC), Laboratory Values (LAB).

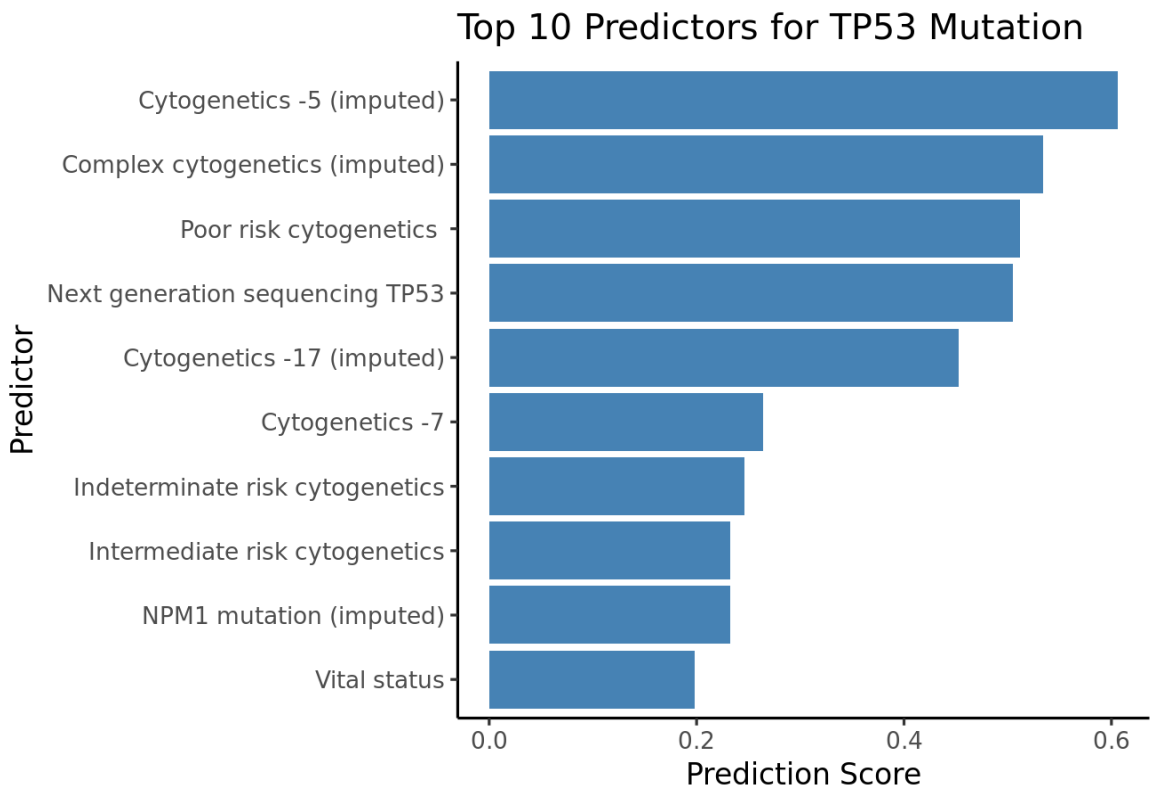

Figure S6. Top 10 predictors for the TP53 mutation binary variable from the merged CU and RWC cohorts. *Abbreviations:* University of Colorado (CU), Real World Cohort (RWC).

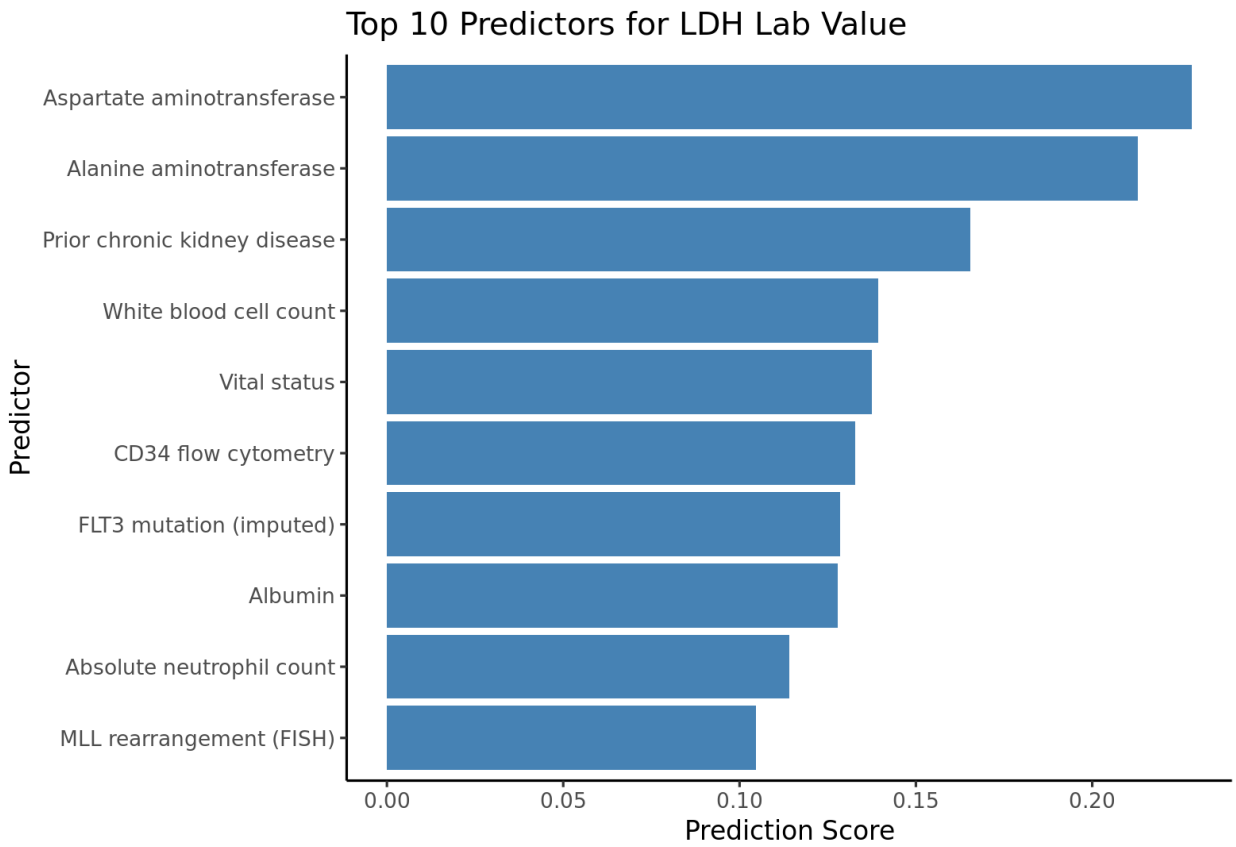

Figure S7. Top 10 predictors for the LDH continuous variable from the merged CU cohort and RWC. *Abbreviations:* Lactate Dehydrogenase (LDH), University of Colorado (CU), Real World Cohort (RWC).

624