Draft: Comparison of missing data handling methods in building variant pathogenicity metapredictors

Mikko Särkkä[1,2], Sami Myöhänen[1], Inka Saarinen[1], Kaloyan Marinov[1], and Jussi Paananen[1,2]

[1]Blueprint Genetics
[2]University of Eastern Finland

1 Introduction

1.1 Variant pathogenicity prediction

- existing methods, how they deal with missingness

1.2 Missingness handling

Missing data is common in real datasets. Consider a matrix of A that represents the unobserved, underlying values that would be obtained by data collection in the absence of any missing data generation mechanisms. The subset of values of A that are observed in data collection is denoted A_{obs}, and the subset of missing values of A is denoted A_{mis}. Of course, the values of A_{mis} will not be known when analysing any real dataset. M is the missingness indicator matrix whose values are 0 when the corresponding value of A is observed, and 1 when the corresponding value of A is missing.

Missing data processes may be classified into *missing completely at random* (MCAR), *missing at random* (MAR) and *missing not at random* (MNAR).[12] In a missing data process with a MCAR mechanism[1], the probability of a value being missing does not depend on any observed or unobserved values. In a missing data process with a MAR mechanism, the probability of a value being missing may depend on observed values. In a missing data process with a MNAR mechanism, the probability of a value being missing may depend on both observed and unobserved values.

[1]Check whether one should only use mechanism, process, or both

1.2.1 Statistical inference[2] vs. prediction

1.2.1.1 Statistical inference Much of missing data literature is focused on treatment of missing data in the context of statistical inference[3].

In such cases, the methods are designed to ensure, in addition to unbiasedness, that the uncertainty introduced by missingness is correctly reflected in the standard errors. This is in contrast to designing methods simply for accurate prediction of the underlying value[4]. Indeed, naïvely imputing data with a single imputation method[5] is misleading as use of even highly accurate single imputation methods will cause underestimation of the standard errors[6][4, subchapter 2.6].

The uncertainty can be properly incorporated via two main avenues: *likelihood-based approaches* and *multiple imputation* (see [12]). The former accounts for missing values by integrating out the parameter representing the missingness generating process, and does not require explicit imputation of missing values.[7][12] A drawback of this method is the restriction to models that can be estimated via maximum likelihood[8]. The latter instead is based on production of multiple complete datasets, on which separate models are fitted and whose estimates are then pooled.[12]

1.2.1.2 Prediction A less commonly studied problem[9] is missingness handling in the prediction, as opposed to explanation. See [2] and [21] for discussion of the differences of predictive and explanatory statistics[10].

An important observation regarding the distinction is that the *theoretically correct model* may not be the *best model* with regards to prediction accuracy[21].

1.2.1.3 Missing values at estimation time Most studies introducing imputation methods focus on problems where missingness may occur in the model estimation phase, but where data is assumed to be complete at prediction time. In this context, it suffices to design methods which facilitate model estimation[11] in a way that maximizes generalization performance.

[2] Use word "explanation" like Shmueli, or "statistical inference"; or "estimation", like Sarle?

[3] Cite examples; difficult statement to quantify. Stated in a different format [21] on page 296.

[4] Statement needs a lot more precision, and citations

[5] Define single imputation

[6] Might use quote about machine learning imputation being the even more dangerous version of regression imputation

[7] The book does note a similarity between expectation maximization and imputation, but this is not used in all situations.

[8] Elaborate, examples of models both MLE and not MLE

[9] can we to quantify this?

[10] Reread these and possibly elaborate

[11] Should I use "model estimation" as a term at all, and just refer to model training? There are two perspectives here, the machine learning and the statistics perspective. Machine learning might be considered a subset of predictive statistics.

1.2.1.4 Missing values both at estimation and prediction time

Sarle notes that "The usual characterizations of missing values as *missing at random* or *missing completely at random* are important for estimation but not prediction."[19][12] Ding & Simonoff [6] provide real-world evidence[13] in support of this statement in the use of classification trees[14]. In a predictive context, the presence of *informative missingness* (as defined by Sarle[19]) in the data, i.e. missingness being dependent on the response variable conditional on X_{obs}, may actually lead to improved predictive accuracy compared to complete data[19][6].

One additional challenge that arises in this context is that most imputation methods are not implemented in a way that easily allows reuse of learned parameters. That is, it is difficult to first estimate imputation method parameters on the training set, and then impute the test set using those same parameters. This leads to diminished prediction accuracy, as the distributions of imputed data differ in the training and test sets. Even worse, since the parameters for test set imputation are estimated from the test set, the content and size of the test set itself may affect prediction accuracy.

- Saar-Tsechansky & Provost
  - Predictive value imputation vs. distribution-based imputation

2 Materials and methods

2.1 Data

2.1.1 ClinGen dataset

The dataset consists of ClinGen[17] expert-reviewed single-nucleotide variants from ClinVar[11, 10, 9].[15]

We annotated the variants using the Ensembl Variant Effect Predictor (VEP)[16] version 96 and dbNSFP3.5[13, 14]. In addition, we incorporated annotations used by CADD[18], matching by VEP-annotated Ensembl transcripts. [16]

2.1.2 Features

The initial feature set was defined manually[17]. We excluded any metapredictors from the feature set.

- Caveat: lack of explicit variable selection may disadvantage logistic regression
  - It is not clear whether this affects the relative performances of imputation methods

[12] Check correct academic formatting for this sort of inline quote
[13] Can I use this expression?
[14] Check this
[15] include date obtained?
[16] add variant number
[17] add inclusion rationale

- Described observed missingness patterns
- Caveat: missing values in covariance matrices
  - This can be understood as a consequence of monotonous missingness (though the missingness is not fully monotonous)
  - Monotonous missingness should not be a problem in itself, even though we don't specifically use the information of its presence.

2.2 Preprocessing

- Caveat: centering and scaling not done, except for BPCA, which requires it.
  - Neither scaling nor centering should affect random forest, logistic regression (intercept takes care of centering) and at least Van Buuren does not mention either centering or scaling to matter in MICE (which makes sense, since one wouldn't want to lose the original scaling when doing inference).

For each variant, we chose transcript specific values from dbNSFP to match the Ensembl canonical transcript annotated by VEP. Variants whose canonical VEP-annotated transcript ID did not match that from dbNSFP were discarded.

Missing values were replaced by default values for features where the missingness implied the default value *a priori* (e.g. a prediction of effect of amino acid substitution for a protein may be imputed with the neutral value (usually 0) when a variant is intronic) .[18]

Feature name	Feature interpretation	Default value
`motifECount`		0
`motifEScoreChng`		0
`motifEHIPos`		`FALSE`
`tOverlapMotifs`		0
`motifDist`		0
`gnomAD_exomes_AF`	gnomAD allele frequency from exomes	0

Categorical variables were processed to dummy variables.

- Caveat: No imputation methods designed specifically for categorical variables used
- Dummy variable coding is not full rank, but this is fixed by the removal of highly correlated variables.
- Caveat: Setting all missing categorical values to zeros in dummies; no imputation method does anything for categorical variables

[18] Add table here or in supp. materials

– Though very few categorical variables (3)

We formed a binary outcome vector by defining that variants classified as pathogenic or likely pathogenic as belonging to the positive class, and variants classified as benign or likely benign as belonging to the negative class. [19]

The final feature vectors of some sets of variants[20] may be equal (i.e. duplicated). In such cases, we kept only one variant [21].

Use of categorical variables with high class imbalance within certain levels may obfuscate the performance of the imputation methods due to allowing the classifier to learn to classify all variants with that level into either the positive or the negative class, and therefore ignoring all other features upon which imputation may have been performed. VEP-predicted variant consequence is one such feature. For this reason, we removed variants with consequences for which either class had less than 5% of overall variants of that consequence[22][23].

To match an ordinary machine-learning process and to avoid issues with certain imputation methods[24], we removed features with fewer than 1% unique values.[25]

For feature pairs with high correlation, we kept only one of the features[26].

The final processed dataset contains n [27] variants characterized by m[28] features.

- Note that features removal leads to "largest feature set that works with all methods"; more elaborate analysis might find larger sets for some of them and thus give an advantage that is now lost

2.2.1 Data split

The data was randomly split into training and test subsets, with 70% ($N = n$[29]) of variants in the training set and 30%[30] ($N = m$[31]) of variants in the test set.

[19] How many in each?
[20] How many?
[21] With the lowest chromosomal position; write this in supplementary notes
[22] Removing how many?
[23] Explain that this would not be done in ordinary training practice.
[24] e.g. in PMM one often got failures due to singular matrices before application of these steps
[25] List these
[26] List removed features
[27] How many?
[28] How many?
[29] How many?
[30] Check exact final percentages and add them
[31] How many?

2.3 Training

2.3.1 Classifiers

Both logistic regression and random forest classifiers are trained using the `caret` package[8]. Logistic regression is trained with the base R `glm`, and random forest as implemented in the package `randomForest`.

2.3.1.1 Classifier hyperparameter search The random forest was trained using the out-of-bag (OOB) performance for model selection.

- Caveat: we use OOB for model selection of RF, but MCC for selection of imputation hyperparameters.

`glm` does not offer any tuning parameters.

2.3.2 Imputation hyperparameter search

In order to maximize the performance of each imputation method for fair comparison, hyperparameter grids were defined for each method for which different hyperparameters[32] could be passed.

2.3.3 Multiple datasets from probabilistic[33] methods

Multiple imputation methods are naturally designed to draw multiple datasets from the estimated imputation model[34]. We utilize the multiple datasets to estimate the performance variability that arises from randomness in the imputation of both training and test sets.

For each completed dataset from a probabilistic imputation method, we train a separate classifier (performing its usual hyperparameter search and model selection procedure separately on each dataset, see section Classifier hyperparameter search). The training set performances of each classifier trained on a completed dataset produced by the same imputation model and the same imputation hyperparameters are averaged. The hyperparameter configuration with the highest mean classifier performance is selected for each imputation model[35]. Each classifier trained on datasets produced by the winning hyperparameter configuration is retained for test set performance evaluation.

Another use of multiply imputed datasets would be that one can in principle train a classifier on each. In prediction, one would average results, hypothetically

[32] Are these usually called hyperparameters in imputation methods? Technically I think they are hyperparameters even if not called that, but it might be better to conform to common terminology.

[33] Are probabilistic and stochastic equivalent in this context?

[34] Does MICE actually estimate a model and draw from it m times, or just estimate a model m times? IIRC, the model is represented across iterations as simply the previous completed datasets. This makes it sound like it would be the latter.

[35] Model, or model type?

leading to better overall performance[36]. However, this is even more resource intensive than the usual setting.
The benefits of this approach are not explored in this paper.

One can also simply run any probabilistic single imputation methods multiple times with different seeds.

- Caveat: I do not monitor RF convergence.
- Caveat: I optimize (out-of-bag) accuracy, not MCC in RF training and HP selection
- Caveat: RF hp grid is not big. However IMO it doesn't need to be.

2.4 Imputation methods

2.4.1 Univariate imputation

The simplest imputation methods impute every missing value within a feature with the same value, which may either be a constant or a statistic estimated from the observed values of the feature.

Simple imputation methods	Value
Zero imputation	0
Maximum imputation	Maximum observed value within feature
Minimum imputation	Minimum observed value within feature
Median imputation	Median observed value within feature
Mean imputation	Mean observed value within feature
Unique-value[37] imputation	For observed values F_{obs} of feature F, $\lvert\max(F_{obs}) - \min(F_{obs})\rvert \cdot 10$
Missingness indicator	For each feature, perform zero imputation and create a binary feature[38] indicating original missing values

- Caveat: should missingness indicators + logistic regression have interactions between variable + its indicator?

2.4.2 Multiple imputation by chained equations

- Caveat: I do not explore *joint modeling* (JM) models, presented as the canonical alternative to MICE for performing multivariate multiple imputation.

Multiple imputation by chained equations (MICE), or *fully conditional specification* (FCS)[**vanbuuren2007**][39], refers to iteratively imputing single variables

[36] IMO this fits one of the formulae in Sarle. Check
[37] or "outlier"
[38] Make sure to note that duplicated indicator vectors added here are removed.
[39] Does it suffice to cite only MICE here? The paper makes this statement.

conditional on other variables. In short, a fully conditional specification algorithm

1. uses univariate imputation to sequentially impute each variable[40] conditional on the observed values of other variables
2. reimputes each variable conditional on the imputed data from the previous iteration

Step 2 is repeated until some maximum number of iterations or some measure of convergence is reached.

We used the R `mice` package [5] to perform the imputation. The following univariate imputation methods provided by `mice` were used[41]:

- Caveat: I use a "low" number of iterations and completions (10 for both). They are considered good numbers in a statistical inference context, but it is not fully clear whether the same applies in predictive contexts.

Method	`mice` model
Predictive mean matching	`pmm`
Random forest	`rf`
Linear regression	`norm.predict`
Bayesian linear regression	`norm`

- Caveat: convergence issues with mice::RF
- Caveat: norm.predict particularly vulnerable to re-estimation parameters from test set

2.4.3 Non-MICE imputation

We used several popular models falling outside the multiple imputation framework.

Method	R package
BPCA	`pcaMethods`
k-NN	`DMwR`
MissForest	`missForest`

- Caveat: in k-NN, you must have enough complete cases to even start imputation. The number depends on k. In our case, the largest k that could be used was 3[42].
- Caveat: BPCA required scaling of the dataset, unlike all other methods.

[40] Try to use only feature or variable? It seems more natural to use variable here, due to the more statistical context.

[41] Should I also present the hyperparameters that were searched over?

[42] Check max k

Somehow reuse of scaling parameters from training set made the results still worse, so there is likely something wrong with how I am using the model.

2.4.4 Built-in method in random forest

Random forest offers a method for handling missing data by iteratively making use of the proximities of observations [3]. This is only available for the training phase. [43]

2.5 Simulation

2.5.1 Main idea

In addition to the main experiment, we wished to gain some understanding into whether the rankings of the imputation methods would be affected by different percentages of missingness. A common strategy for studying this is simulation of missing values either on fully simulated data, or on the complete subsets of real datasets. In our context of application, the number of complete cases in the dataset is very low and thus cannot be used as the basis for simulation.

Instead, we chose to take the full, incomplete dataset as the basis of simulations.

1. Create many simulated datasets based on the full dataset with additional missing values using `ampute`[20] on the dataset while varying mechanism and percentage
2. Impute each simulated dataset with each imputation method (using a downsampled hyperparameter grid compared to main experiment)
3. Compute RMSE for each simulated dataset with respect to values that were observed in the full dataset but missing in the simulated dataset
4. Train a classifier on each completed dataset, and evaluate performance on completed test set

2.5.2 Amputing additional missing values

The simulation of additional missing values was performed using `ampute` implemented in `mice`, which allows simulation of each of MCAR, MAR and MNAR missingness mechanisms. This allows us to assess the change in performance with increasing levels of MAR and MNAR, common in real data, instead of just MCAR, which is usually the only mechanism generated in articles presenting new imputation methods. The input matrix of `ampute` is required to be complete, so we partitioned the data according to missingness patterns. This forms a set of matrices for which every feature is either fully observed or fully missing. We then used each of the fully observed submatrices as inputs to `ampute`, and combined the resulting output back to form a matrix of the original size.

[43] The other way described on the page amounts to mean and mode imputation.

- Caveat: MAR and MNAR mechanisms may depend on different variables in different original missingness patterns
- Might want a diagram here

2.5.3 "Lean" experiment

Many imputation studies are specifically built to assess an imputation methods capability to predict the original values from the observed values of a dataset with missing data. Thus, they use the RMSE between the original dataset and an imputed dataset as a metric of performance of the imputation method. However, a model with the lowest RMSE is not in general the best one in the statistical inference context [4, chapter 2.6]. In short, the best model wrt. RMSE is linear regression estimated via least squares, and the deterministic nature of a regression prediction necessarily ignores uncertainty due to the missing data. The same argument cannot be applied to the predictive context, but the same end result may apply when missingness is informative.

Consider a situation where missingness is highly informative. Then a perfect imputation method (with respect to RMSE; i.e. one where RMSE would equal 0) would impute the dataset with the original values. However, the informativeness in the missing values would be lost. The loss of information could, in principle, be avoided by adding missingness indicators before imputing, but this comes with the increase in dimensionality (doubling the number of features in the worst case) and thus cannot be seen as a universal solution.

Since RMSE cannot be used as a universal metric of imputation method performance, we perform a leaner version of the main experiment on each simulated dataset. That is, we impute each simulated dataset using a smaller hyperparameter grid (downsampled separately for each simulated dataset) and producing only one dataset each when using probabilistic methods. As in the main experiment, a classifier of each type is trained on the imputed simulated datasets, the best performing imputation hyperconfiguration is chosen by the highest performing classifier trained on a dataset imputed via that configuration, and performance is estimated on the test set imputed with the winning configuration. We can thus investigate whether low RMSE on the training set implies high downstream classifier performance on the test set.

- Note that different simulations may lead to different feature sets due to preprocessing

2.6 Performance evaluation

We use two main metrics to evaluate the performance of resulting classifiers.

The *Matthews' correlation coefficient* (MCC) is defined as

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}},$$

- Why MCC? Because has less issues e.g. w/ overoptimistic estimates in problems with high class imbalance, unlike accuracy or F_1

The *area under ROC curve* (AUC-ROC, or just AUC) is defined as the area under the receiver operating characteristic curve.

The root-mean-square error (RMSE) is defined as

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$$

where y_i is the ith true value, $\hat{y}_i$ is the ith prediction (in this case, the imputed value), N is the number of predictions (in this case, the number of imputed values).

2.6.1 Ranging over multiple imputation datasets

Due to the production of multiple completed versions of both training and test datasets, we can observe the variability in performance of classifiers downstream of probabilistic imputation methods due to both the variation in the imputations of the training set and the variation in the imputations of the test set.

2.6.2 Issues with studying imputation methods developed for statistical inference in a predictive context

Due to the differences in intended usage between statistical inference and prediction, imputation methods developed for the former have some additional difficulties in their use for the latter. The first and main issue is that out-of-the-box implementations often do not provide an easy way to reuse parameters from an earlier run. This makes it difficult to use parameters from the training set on the test set.

Some ways to deal with this are

1. Reimplementing the method in a way that allows reuse of parameters
2. Ignoring the issue, and imputing the test set allowing the imputation method to re-estimate its parameters
3. For imputing the test set, concatenate the training set and test set, impute the combined dataset and then remove rows belonging to the training set
4. 3., but concatenating the test set with an imputed training set
5. 3., but concatenating only one observation from the test set to the training set, and repeating this until the full test set is imputed

Each of these options lead to different advantages and disadvantages.

1. is work-intensive, and requires deep understanding of each imputation method. Fully solves the problem.

2. leads to diminished classifier performance on the test set, as the distributions of imputed values may differ between training and test sets[44]
3. This reduces the difference between the distributions compared to 2.
4. Faster than 3., but probably introduces biases
5. Avoids the issue where it the size of the test set affects the performance on the test set, and makes it impossible to predict on a single observation at a time. Really slow.

Options 1. and 5. are the only ones that allow imputation of test observations independently from each other; options based on 2. necessarily perform imputation with parameters estimated from the other observations that are being predicted on at the same time. This is in general undesirable.

Imputation method / family	Implementation	Out-of-the-box parameter reuse
MICE	`mice`	No
BPCA	`pcaMethods`	No
k-NN	`DMwR`	Yes
Simple methods	custom	Yes
MissForest	`missForest`	No

The package `mlr`[1] offers wrapper functionality that allows use of any prediction method offered by the package also for univariate imputation, along with functionality for correct reimputing data with previously learned parameters, but we did not explore this possibility in this work. Investigation of the imputation performance of methods originally intended for prediction could be an opportunity for future work.

We choose to implement option 2. due to its simplicity when out-of-the-box parameter reuse is not available.

- Caveat: this may bias the comparison in favor of k-NN & the simple methods.

2.6.3 Circularity in variant databases

Grimm et al.[7] described several biasing factors in variant effect predictor training and performance evaluation using data from commonly used variant databases, e.g. the tendency for variants within the same gene being classified as all pathogenic or all neutral, or simply due to difficulty of finding datasets completely disjoint with the training set. Mahmood et al. [15] further analysed existing variant effect predictors[45] using datasets generated from functional assays, and found drastically lower performance compared to earlier reported estimates.

[44] I am not fully convinced of this yet.

[45] Make sure to distinguish between variant effect prediction and variant pathogenicity prediction

Our approach is not immune to these biases, and we expect that any reported performance metrics will be overoptimistic. However, we expect that the main result of the study, the relative performance rankings of imputation methods, will not be affected by the biases. The classifiers built described in this work are not intended to outperform earlier approaches or be used for variant effect prediction.

3 Results

4 Discussion

discussion[46]

References

[1] Bernd Bischl et al. "mlr: Machine Learning in R". In: *Journal of Machine Learning Research* 17.170 (2016), pp. 1–5. URL: http://jmlr.org/papers/v17/15-066.html.

[2] Leo Breiman. "Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author)". In: *Statistical Science* 16 (Aug. 2001), pp. 199–231. DOI: 10.1214/ss/1009213726.

[3] Leo Breiman and Adele Cutler. *Random forests - classification description.* URL: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm (visited on 01/27/2020).

[4] S. van Buuren. *Flexible Imputation of Missing Data.* 2nd ed. Chapman & Hall/CRC Interdisciplinary Statistics. CRC Press LLC, 2018. ISBN: 9781138588318. URL: https://www.crcpress.com/Flexible-Imputation-of-Missing-Data-Second-Edition/Buuren/p/book/9781138588318.

[5] Stef van Buuren and Karin Groothuis-Oudshoorn. "mice: Multivariate Imputation by Chained Equations in R". In: *Journal of Statistical Software, Articles* 45.3 (2011), pp. 1–67. ISSN: 1548-7660. DOI: 10.18637/jss.v045.i03. URL: https://www.jstatsoft.org/v045/i03.

[6] Yufeng Ding and Jeffrey S. Simonoff. "An Investigation of Missing Data Methods for Classification Trees Applied to Binary Response Data". In: *J. Mach. Learn. Res.* 11 (Mar. 2010), pp. 131–170. ISSN: 1532-4435.

[7] Dominik G. Grimm et al. "The Evaluation of Tools Used to Predict the Impact of Missense Variants Is Hindered by Two Types of Circularity". In: *Human Mutation* 36.5 (2015), pp. 513–523. DOI: 10.1002/humu.22768. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/humu.22768. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/humu.22768.

[8] Max Kuhn. "Building Predictive Models in R Using the caret Package". In: *Journal of Statistical Software, Articles* 28.5 (2008), pp. 1–26. ISSN: 1548-7660. DOI: 10.18637/jss.v028.i05. URL: https://www.jstatsoft.org/v028/i05.

[46] Commit to either "missing data handling" or "missingness handling" over the whole thing

[9] Melissa J. Landrum et al. "ClinVar: improving access to variant interpretations and supporting evidence". In: *Nucleic Acids Research* 46.D1 (Nov. 2017), pp. D1062–D1067. ISSN: 0305-1048. DOI: 10.1093/nar/gkx1153. eprint: http://oup.prod.sis.lan/nar/article-pdf/46/D1/D1062/23162472/gkx1153.pdf. URL: https://doi.org/10.1093/nar/gkx1153.

[10] Melissa J. Landrum et al. "ClinVar: public archive of interpretations of clinically relevant variants". In: *Nucleic Acids Research* 44.D1 (Nov. 2015), pp. D862–D868. ISSN: 0305-1048. DOI: 10.1093/nar/gkv1222. eprint: http://oup.prod.sis.lan/nar/article-pdf/44/D1/D862/9483060/gkv1222.pdf. URL: https://doi.org/10.1093/nar/gkv1222.

[11] Melissa J. Landrum et al. "ClinVar: public archive of relationships among sequence variation and human phenotype". In: *Nucleic Acids Research* 42.D1 (Nov. 2013), pp. D980–D985. ISSN: 0305-1048. DOI: 10.1093/nar/gkt1113. eprint: http://oup.prod.sis.lan/nar/article-pdf/42/D1/D980/3584314/gkt1113.pdf. URL: https://doi.org/10.1093/nar/gkt1113.

[12] Roderick J. A. Little and Donald B. Rubin. *Statistical analysis with missing data*. eng. 2nd ed. Hoboken, New Jersey: John Wiley & Sons, Inc., 2002. ISBN: 9781118625866.

[13] Xiaoming Liu, Xueqiu Jian, and Eric Boerwinkle. "dbNSFP: A lightweight database of human nonsynonymous SNPs and their functional predictions". In: *Human Mutation* 32.8 (2011), pp. 894–899. DOI: 10.1002/humu.21517. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/humu.21517. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/humu.21517.

[14] Xiaoming Liu et al. "dbNSFP v3.0: A One-Stop Database of Functional Predictions and Annotations for Human Nonsynonymous and Splice-Site SNVs". In: *Human Mutation* 37.3 (2016), pp. 235–241. DOI: 10.1002/humu.22932. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/humu.22932. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/humu.22932.

[15] Khalid Mahmood et al. "Variant effect prediction tools assessed using independent, functional assay-based datasets: implications for discovery and diagnostics". In: *Human Genomics* 11.1 (2017), p. 10. DOI: 10.1186/s40246-017-0104-8. URL: https://doi.org/10.1186/s40246-017-0104-8.

[16] William McLaren et al. "The Ensembl Variant Effect Predictor". In: *Genome Biology* 17.122 (2016). DOI: 10.1186/s13059-016-0974-4.

[17] Heidi L. Rehm et al. "ClinGen — The Clinical Genome Resource". In: *New England Journal of Medicine* 372.23 (2015). PMID: 26014595, pp. 2235–2242. DOI: 10.1056/NEJMsr1406261. eprint: https://doi.org/10.1056/NEJMsr1406261. URL: https://doi.org/10.1056/NEJMsr1406261.

[18] Philipp Rentzsch et al. "CADD: predicting the deleteriousness of variants throughout the human genome". In: *Nucleic Acids Research* 47.D1 (Oct. 2018), pp. D886–D894. ISSN: 0305-1048. DOI: 10.1093/nar/gky1016. eprint: http://oup.prod.sis.lan/nar/article-pdf/47/D1/D886/27436395/gky1016.pdf. URL: https://doi.org/10.1093/nar/gky1016.

[19] Warren S. Sarle. "Prediction with missing inputs". In: *3rd, International conference on computational intelligence and neurosciences*. JCIS '98 pro-

ceedings (Research Triangle Park, NC). Ed. by Paul P. Wang. Vol. 2. Association for Intelligent Machinery, 1998, pp. 399–402.

[20] Rianne Margaretha Schouten, Peter Lugtig, and Gerko Vink. "Generating missing values for simulation purposes: a multivariate amputation procedure". In: *Journal of Statistical Computation and Simulation* 88.15 (2018), pp. 2909–2930. DOI: 10.1080/00949655.2018.1491577. eprint: https://doi.org/10.1080/00949655.2018.1491577. URL: https://doi.org/10.1080/00949655.2018.1491577.

[21] Galit Shmueli. "To Explain or to Predict?" In: *Statist. Sci.* 25.3 (Aug. 2010), pp. 289–310. DOI: 10.1214/10-STS330. URL: https://doi.org/10.1214/10-STS330.