Split-Transformer Impute (STI): Genotype Imputation Using a Transformer-Based Model

Mohammad Erfan Mowlaei[1], Chong Li[1], Junjie Chen[2], Benyamin Jamialahmadi[3], Sudhir Kumar[4,5], Timothy Richard Rebbeck[6,7] and Xinghua Shi[1*]

[1*]Computer & Information Sciences, Temple University, 925 N. 12th Street, Philadelphia, 19122, PA, USA.
[2]Computer Science and Technology, Harbin Institute of Technology, Shenzhen University Town, Shenzhen, 518055, Guangdong, China.
[3]David R. Cheriton School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, N2L 3G1, ON, CA.
[4]Institute for Genomics and Evolutionary Medicine, Temple University, 925 N. 12th Street, Philadelphia, 19122, PA, USA.
[5]Department of Biology, Temple University, 925 N. 12th Street, Philadelphia, 19122, PA, USA.
[6]Division of Population Sciences, Dana-Farber Cancer Institute, 450 Brookline Ave, Boston, 02215, MA, USA.
[7]Department of Epidemiology, Harvard T. H. Chan School of Public Health, 677 Huntington Ave, Boston, 02115, MA, USA.

*Corresponding author(s). E-mail(s): mindyshi@temple.edu;
Contributing authors: mohammad.erfan.mowlaei@temple.edu; chong.li0001@temple.edu; junjiechen@hit.edu.cn; B2jamial@uwaterloo.ca; s.kumar@temple.edu; timothy_rebbeck@dfci.harvard.edu;

Abstract

With recent advances in DNA sequencing technologies, researchers are able to acquire increasingly larger volumes of genomic datasets, enabling the training of powerful models for downstream genomic tasks. However, genome scale dataset often contain many missing values, decreasing the accuracy and power in drawing robust conclusions drawn in genomic analysis. Consequently, imputation of missing information by statistical and machine learning methods has become important. We show that the current state-of-the-art can be advanced significantly by applying a novel variation of the Transformer architecture, called Split-Transformer Impute (STI), coupled with improved preprocessing of data input into deep learning models. We performed extensive experiments to benchmark STI against existing methods using resequencing datasets from human 1000 Genomes Project and yeast genomes. Results establish superior performance of our new methods compared to competing genotype imputation methods in terms of accuracy and imputation quality score in the benchmark datasets.

Keywords: Transformer, Attention, Deep Learning, Machine Learning, Genotype Imputation, Genomics

1 Introduction

Genetic and genomic studies, such as linkage analysis and genome-wide association studies (GWAS), enable us to dissect genetic architecture of complex traits that are the key to understanding the genetic contribution and risks of these traits and diseases. In recent years, whole-genome genotyping platforms and sequencing technologies have advanced greatly and become highly affordable, resulting in the accumulation of large collections of genotypes in growing cohorts awaiting genomic analysis. Although the resolution of genotypes has been improving steadily over time, genotypes still contain many missing values and untyped loci [1]. These missing data may decrease statistical power in disease association studies and causal variants discovery [2–4]. Causes of missing genotypes include the difficulty in sequencing rare alleles [5–7], failure of experimental assays, genotyping calling errors, and differences in densities and properties of genotyping platforms [2].

Consequently there is always a need of reliable imputation of genotypes using computational methods. Imputation is the process of inferring missing values in the data based on the knowledge and distribution in the available datasets. In early studies of association analysis in the pedigree data, genotypes were implicitly imputed based on joint genotype distribution between individuals in the same pedigree [8, 9]. This idea was extended and termed as "in silico genotyping" to refer to computational analysis, instead of laboratory based procedures, for imputing missing genotypes in reference [10]. A comprehensive

discussion about existing imputation methods in general and state-of-the-art imputation models for genomic data is available in the supplementary.

Imputation has applications in a wide range of genomic studies. Some examples of early applications of genotype imputation are in the analysis of Type 2 diabetes [11], and six other complex diseases[12]. In another study [13], genotype imputation was used to evaluate the evidence for additional causal variants based on identified and confirmed Single Nucleotide Polymorphisms (SNPs). In light of the success of these applications, there has been an increase in the practice of genotyping using low-density SNP panels to reduce costs, and imputing the samples to commonly used dense panels [14, 15]. The list of widely used reference panels in GWAS is presented in Table 1. GWAS analysis utilizes a dense set of genotyped genetic variants to increase the power of discovering genetic variants associated with diseases and traits [16–21]. Meta-analysis boosts confidence interval of GWAS analysis by augmenting sample size, through cost-effective computational methods, by fusing data from multiple studies [22, 23]. Since the data across multiple studies is heterogeneous due to sample sizes, platforms utilized, and genetic ancestries surveyed, imputation is utilized as a means to infer values for untyped positions that are not present in any of the datasets [24–26].

Table 1 Reference panels available for imputation. 1kGP is 1000 Genomes Project dataset in the first row.

Name	#Samples	#Sites (Chr1-22)	Variants
1kGP Phase3 V5 [27]	2,504	49,143,605	SNP/INDELs/SVs
UK10K [28]	3,781	45,492,035	SNVs/INDELs
TopMed [29]	97,256	308,107,085	SNPs/SNVs/INDELs/SVs
HRC r1.1 [30]	32,470	39,635,008	SNPs
CAAPA [31]	883	31,163,897	SNPs/SNVs
AFAM [32]	2,269	54,962,430	SNPs/INDELs
WBBC [33]	10,376	81,498,995	SNVs/INDELs
The GenomeAsia Pilot [34]	1,654	21,494,814	SNPs/INDELs

Imputation in genomic data calls for specialized methods since the data are, inherently, different from many other domains, such as vision or natural language processing. First, the data are of high dimension as the number of bases is large, but the number of samples can be orders of magnitude less than the number of bases. Second, there are linear and non-linear interactions between bases whose incorporation into the model may greatly affect the quality of imputation [35]. Third, individuals of shared ancestry are likely to share segments of sequences due to common descent, which can serve as additional prior information to impute the missing data.

Though the overall performance of existing imputation methods for genomic data is relatively high, they do have shortcomings. Reference-based Hidden Markov Models (HMMs), such as Minimac4 [36], have the highest accuracy but are incapable of handling any data without a reference panel.

Moreover, they are incapable of imputing multi-allelic events. On the other hand, existing Deep Learning (DL) models do not have a mechanism to capture pairwise correlations among markers, such as the presence of Linkage Disequilibrium (LD), and result in lower performance compared to reference-based models. An effective solution to this problem, capable of capturing pairwise interactions, is the attention mechanism in transformer architecture [37].

Attention mechanism in DL mimics the visual attention to focus on specific parts of pictures [38] in order to generate an output [39], by calculating pairwise importance scores over subsets of data. Attention can capture global interactions amongst the markers at the cost of quadratic memory consumption, making it a suitable candidate to capture LD structures. The memory cost becomes important in genomic analysis since the number of bases in a sequence is normally in thousands. In genotypes, the majority of interactions are local [40]. Therefore, it is of great importance to limit the scope of attention to save computational resources. Additionally, genotype imputation methods, to the best of our knowledge, are either designed to solely tackle bi-allelic events [36, 41, 42] or are not evaluated for multi-allelic variants [2, 43, 44]. This, in turn, prevents confident imputation of multi-allelic events or complex genetic variants such copy number variants, duplications, and insertions.

In this paper, we present a novel genotype imputation model based on the attention mechanisms in a transformer framework. Our model utilizes attention to capture correlations among the SNPs/SNVs in the data. It achieves high imputation accuracy at a modest memory consumption cost by dividing the data into chunks, enabling efficient application to long sequences. To summarize, our contributions in this study are fourfold. First, we propose an improvement for the training process of DL models for imputation. Second, we present Split-Transformer Impute (STI), a novel DL transformer model, designed to specifically address the genotype imputation problem. STI performs comparable to competing imputation model, Minimac4, while it does not need to be trained each time prior to inference and can be applied to genotyped datasets without reference panels. Third, we extend language embedding in order to apply it to any categorical data, including, but not limited to, genotype data. We term this embedding as Cat-Embedding and through experiments show the contribution of Cat-Embedding in enhancing the performance of STI. Lastly, we perform experiments using multi-allelic datasets in order to benchmark STI against available methods for such data and investigate how their performance is affected compared to bi-allelic events.

2 Results

2.1 Overview of the study

An overview of our proposed model, STI, is presented in Figure 1. STI uses Cat-Embedding layer in order to capture allele information per SNP, in addition to the information of missing values. This, in conjunction with multi-headed attention layers, enables STI to model correlations among SNPs to impute

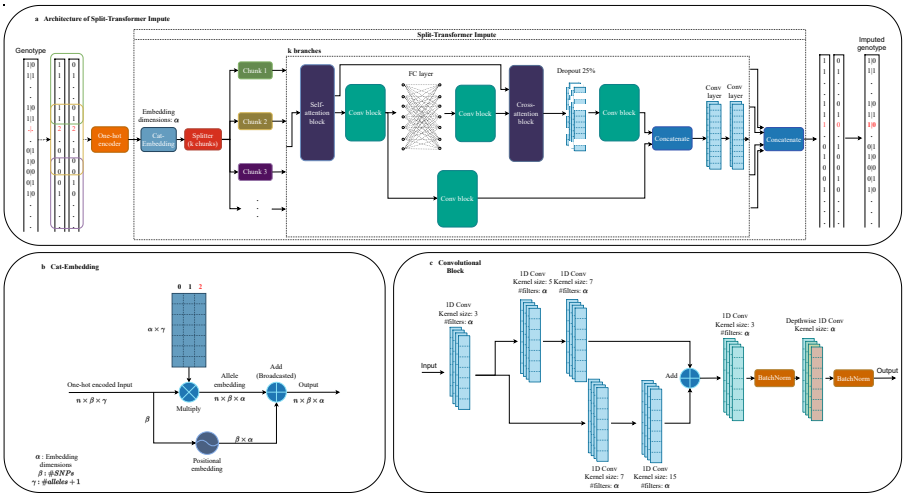

Fig. 1 a. Overall pipeline of the proposed framework: the data is separated into paternal and maternal haplotypes in case of the HLA, and it remains the same in case of the Yeast. While the figure shows phased genotypes, STI can handle unphased data as well since it is only a matter of encoding the data in pre-processing. Then the data is one-hot encoded and fed into our Cat-Embedding layer, followed by splitting the data vertically into k windows. The windows have overlap in order to capture information for the SNPs at the edge of windows. Each branch passes through a unique set of attention, convolution, and fully connected layers, and the cross-attention block shrinks the number of SNPs to the intended window size. Finally, the results of all branches are assembled to generate the final sequence. **b. Workflow of proposed Cat-Embedding:** we consider a unique vector space for each unique categorical value in each SNP/feature. To save computational resources, instead of pre-allocating these vectors, we use addition of positional embedding and categorical value embeddings in order to generate unique embedding vectors for each categorical value in each SNP/feature. We consider missing value as another categorical value (allele) in our model. Here, 2 (highlighted as red) represents the missing value. **c. Convolution block details:** after the initial *1D Conv* layer, the data is passed down to two parallel branches of *1D Conv* layers with differing kernel sizes, and the results of these branches are fused via addition. After another convolution, we used *Depthwise 1D Conv* layer which considerably improves the quality of imputation. Using *BatchNorm* layers in this block proved to perform superior to using *LayerNorm*.

missing values based on known and missing values per position. By vertical partitioning (splitting) of the data and passing resulted SNP windows through separate branches, STI saves memory in attention layers. We use four datasets in order to benchmark STI, namely yeast [45] and three datasets extracted from phase 3 of 1000 Genomes Project [27]: Human Leukocyte Antigen (HLA), deletions in chromosome 22, and all Single Nucleotide Variants (SNVs) in chromosome 22.

The datasets from 1000 Genomes Project are accompanied with a reference panel that enables reference-based methods, such as Minimac4, to impute missing values for that data. In contrast, the yeast dataset has no such reference, making reference-free approaches, such as STI, the only candidates able to impute the data. For the other three datasets we calculated either min allele

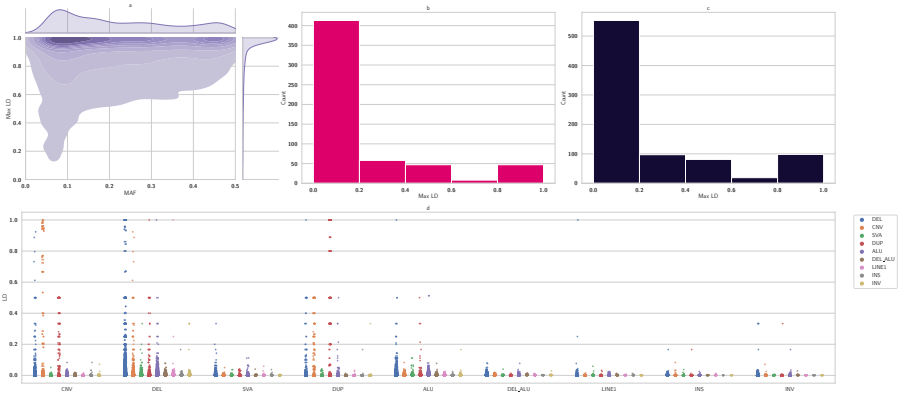

Fig. 2 MAF and LD distributions of three benchmark datasets from 1000 Genomes Project: **a.** Kernel density estimation plot for MAF and maximum LD distributions of SNPs in HLA dataset. Darker shades of color represents higher density in this plot. The closer MAF is to 0.5, the harder it will be for the model to achieve a high accuracy, since the probability of bi-allelic events becomes equal. **b.** Maximum LD distribution for deletions in chromosome 22. **c.** Maximum LD distribution for all SNVs in chromosome 22. We can observe that the number of positions falling in [0.6, 0.8] bin, in **b** and **c**, is less compared to the other bins, and we can expect to see a drop in accuracy for these positions despite the high LD. **d.** LD among different SNV types in chromosome 22. This plot shows other structural events are commonly correlated with deletions. Furthermore, deletion, copy number variation, and duplication events appear in different ranges of LD while the rest of the events are limited to $LD \leq 0.1$. Lastly, the majority of correlated SNVs to deletions are of the same event, making deletions a good separate dataset for our experiment.

frequency (MAF) or LD, as shown in Figure 2, and selected the SNPs/SNVs for the test set proportional to those.

We compare STI to state-of-the-art imputation models: SCDA [43], AE [2], HLA*DEEP [44], and Minimac4 [36]. Additionally, in order to asses the contribution of Cat-Embedding, we replaced it with a convolution layer in STI, named the resulting model STI*WE, fine-tuned it, and applied it to the benchmark datasets. Lastly, we train SCDA, in addition to HLA*DEEP and STI, using our proposed pre-processing and training procedure, and compare it to AE. Since AE and original SCDA are the same and only differ in training process, we believe that this comparison can show the effectiveness of our proposed pre-processing and training procedure.

2.2 Experimental settings

We implemented STI and other DL models using Tensorflow framework [46] in Python. In order to train the models, we used tensor processing units (TPU) provided by Google Colaboratory platform. Learning rate scheduler and early stopping are employed in order to reduce the loss and training duration, as much as possible, on the training set for all DL models.

The input to all DL models is one-hot encoded. STI can handle diplotypes but the best performance, according to our experiments which were inspired by [44], is achieved when the inputs of the DL models are haplotypes. Therefore,

for the HLA dataset and chromosome 22 datasets, we break each diplotype into maternal and paternal haplotypes, feed them into the model, and recombine the resulting predictions for HLA*DEEP [44], SCDA [43], and STI. We keep using diplotypes as inputs for AE [2] since it is an improved version of SCDA in which the training process was modified and we wanted to keep it intact. By doing so, we also compare the improvement in AE to our implementation of SCDA, called SCDA+, in which we use proposed pre-processing in conjunction with the changes to training process as a contribution. The yeast dataset contains haplotypes, so there is no need for the aforementioned extra steps.

In this study, accuracy and imputation quality score (IQS) [47] are used in order to evaluate the imputation power of the models. Accuracy is calculated only for missing positions. IQS adjusts the concordance between predicted and the ground truth SNPs for chance, and is defined for bi-allelic events. Therefore, IQS cannot be calculated for all SNVs in chromosome 22. More on these metrics is discussed in the Metrics section of Supplementary material.

2.3 Baseline models

In order to benchmark our model, we selected four genotype imputation models: reference-based Minimac4 [36] and reference-free deep learning models SCDA [43], AE [2], and DEEP*HLA [44]. In [43], experimental results indicate superior performance of SCDA to ML models for genotype imputation and as such, we do not repeat the same in this study.

For fine-tuning, we use a grid-search and obtain optimal hyper-parameters for each fold using the validation data. Then we use the average of these results to select one best set of hyper-parameter for each model per dataset. The details of the hyper-parameter tuning for each model is discussed in Hyper-parameter tuning section of the supplementary. The upper limit for the hyper-parameters was the resource limit of Google Colaboratory using batch sizes as small as 16. Minimac4 does not require fine tuning for the experiments we are running.

2.4 Experimental results

For each dataset, we performed a 3-fold cross validation. In all of the experiments, missing positions in the test set for all models are identical. The overall results for accuracy and IQS metrics are presented in Figures 3 and 4, respectively. We used maximum LD bins (Figure 2 b & c) to distribute missing positions in deletions and SNVs from chromosome 22 of 1000 Genomes Project. Since some bins have too few positions to be selected at a 0.01 missing rate, we excluded this missing rate for the experiments related to these datasets.

Yeast dataset: Missing positions in samples are selected completely randomly but LD analysis shows that maximum LD for all the SNPs is within $[0.8, 1.0]$ range. As mentioned, Minimac4 cannot be applied to this dataset due to lack of a reference-panel. We can observe that STI performs considerably better compared to the other methods with a minimum average imputation

accuracy of 0.9986 (Figure 3.a). Furthermore, SCDA+ and AE are performing similarly. Conversely, when it comes to IQS (Figure 4.a), for missing rates of 0.01 and 0.05, AE outperforms the other models while for higher missing rates, STI is in the lead in terms of performance.

HLA dataset: Missing positions are selected according the the MAF distribution, which is fairly distributed across different MAF bins according to Figure 2.a. Minimac4 outperforms other methods in terms of accuracy (Figure 3.b), while it falls short in terms of IQS (Figure 4.b), especially with an increase in the missing rate. Interestingly, in case of IQS, STI*WE performs the best with missing rate of 0.01, while STI delivers the best results for higher missing rates. For this dataset, when comparing AE and SCDA+, the former has the best IQS while the latest has the best accuracy. More detailed on accuracy over MAF is presented in **Table 6** of supplementary material, where we can see that accuracy is increasing with the increase in MAF, but there is a considerable drop for $[0.3, 0.4]$ bin. The reason for this drop can be explained using Figure 2.a, in which we observe that the density of SNPs having a high LD for this bin is less than other bins.

Deletions in chromosome 22: For this dataset, we selected missing positions proportional to the maximum-LD distribution Figure 2.b. Since the total number of positions falling in $[0.6, 0.8]$ is considerably lower compared to other bins (19 SNPs only), we expected to see a lower accuracy for the predictions of those SNPs, while with the increase in maximum LD, the accuracy is expected to increase. In terms of overall results, STI outperforms other method in both metrics (Figures 3.c & 4.c). Furthermore, SCDA+ is better in both metrics compared to AE, with an average score difference of 0.12, 0.06, 0.24 for missing rates of 0.05, 0.1, and 0.2, respectively. Looking at **Table 7** in supplementary material, We can see the trend for accuracy based on different maximum LD values for this dataset. Based on these observations, we can observe that Minimac4 is less accurate for SNPs with less maximum LD compared to HLA*DEEP, SCDA+, and STI, and since most of the positions fall within the $[0, 0.2]$ bin, the overall performance of Minimac4 is lower in comparison. As for why Minimac4 has lower accuracy for lower LD values, we presume that the answer lies within the HMM mechanism. Since HMMs rely on transition probabilities, they are ought to perform weakly when the correlation between the events (states) are low. On the other hand, with an increase in LD, we can see that Minimac4 easily outperforms other models.

All SNVs in chromosome 22: For this dataset, similar to the previous dataset, missing positions are distributed among SNVs based on maximum LD (Figure 2.c). Despite having a reference-panel, Minimac4 cannot handle this dataset because it can only handle up to bi-allelic events. Furthermore, IQS is not defined for these events and we can only rely on accuracy for comparison. According to Figure 3.d, STI outperforms all other methods on average accuracy. Additionally, SCDA+ again outperforms AE, indicating the effectiveness of our proposed training procedure. More details of accuracy over different maximum LDs for this dataset can be found in **Table 8** of supplementary

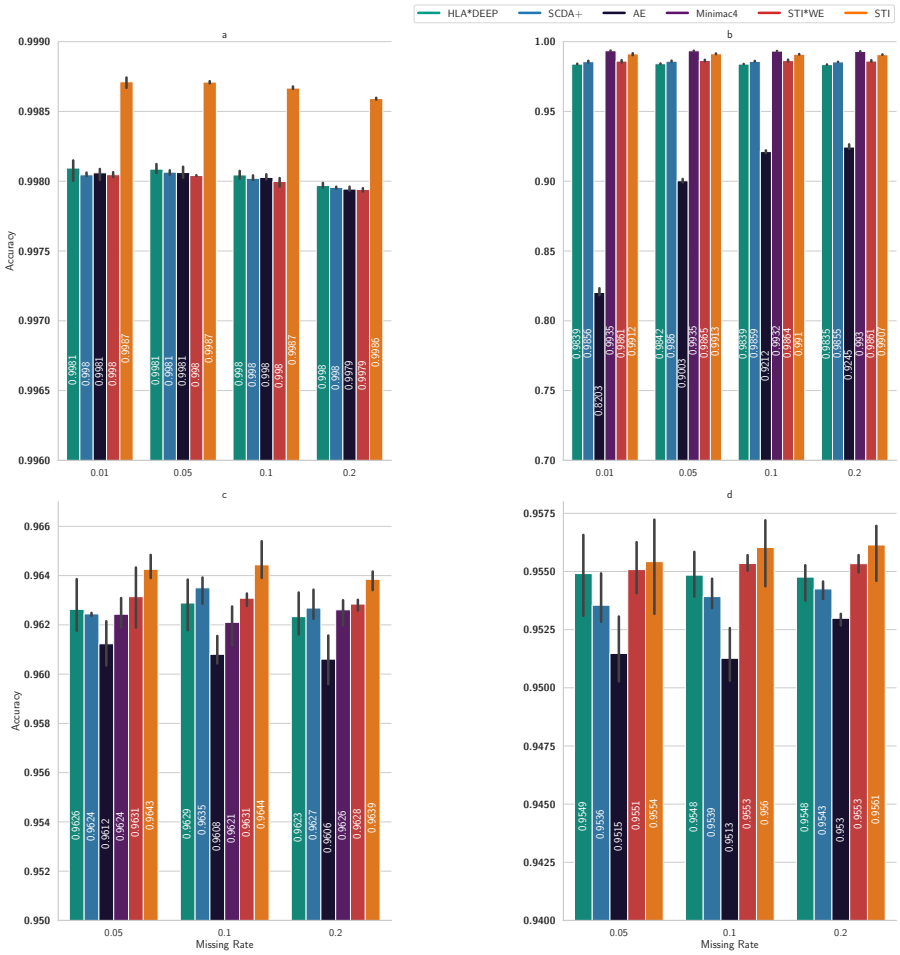

Fig. 3 Accuracy of different models on benchmark datasets over different missing rates (MR) using 3-fold cross validation. While bars indicate average accuracy, black lines indicate highest and lowest accuracy over 3 folds. **a.** Yeast dataset, **b.** HLA dataset, **c.** A dataset composed of Deletions in chromosome 22 of 1000 Genomes Project, **d.** A dataset composed of All events in chromosome 22 of 1000 Genomes Project. Results of b and c, considering respective maximum LD distributions in Figure 2 a & b, suggest that Minimac4 outperforms DL methods when there is a strong LD between SNPs and it will fall short otherwise. In the majority of the results, SCDA+ which benefits from our proposed training pipeline outperforms AE. Lastly, for every dataset, STI is either producing the best or second best results, and outperforms STI*WE, highlighting the effectiveness of Cat-Embedding.

material. Moreover, a breakdown of accuracy over mostly used SNVs is presented in Figure 5, while Figure 3 of supplementary represents the complete breakdown of SNVs in this experiment. According to Figure 5 SCDA+ and STI are performing the best in predicting multi-allelic events

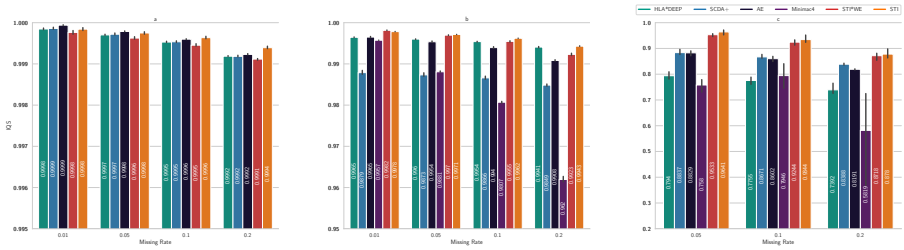

Fig. 4 IQS of different models on benchmark datasets over different missing rates (MR) using 3-fold cross validation. While bars indicate average IQS, black lines indicate highest and lowest IQS over 3 folds. **a.** Yeast dataset, **b.** HLA dataset, **c.** A dataset composed of Deletions in chromosome 22 of 1000 Genomes Project. IQS assesses imputation quality with a focus on rare variants, meaning that the lower the MAF of a SNP is, the higher the penalty for incorrect prediction of that SNP would be. Results indicate that in the majority of experiments STI outperforms competing methods. Additionally, SCDA and AE are performing roughly similar. Furthermore in the yeast and HLA datasets, with an increase in MR, STI*WE starts to fall behind STI, highlighting the effectiveness of Cat-Embedding in more challenging settings.

3 Discussion

Genotype imputation can improve the performance of downstream GWAS studies. One of applications of imputation is predicting missing values in genotyped samples, which is the focus of this study. To address this problem, we propose STI, a DL model utilizing transformer architecture, capable of capturing correlations among SNPs/SNVs, such as LD structures, which can impute multi-allelic events. Through experiments, we compared our proposed imputation model, STI, to various imputation models for genotypes. Additionally, we propose a novel embedding for the task that can be used for any other data consisting of categorical values. Our last contribution is proposing changes to the training process of DL imputation models.

Experimental results of our experiments show that STI considerably outperforms other DL models in majority of cases and delivers comparable and sometimes better performance compared to gold-standard Minimac4 model, while harbors less limitations. The results also indicate that our proposed Cat-Embedding and training process substantially improves the performance of the models.

We also observed some interesting patterns. For chromosome 22 datasets, in Tables 7 & 8 of supplementary materials, we observe that there is an unexpected drop in accuracy of every model for the highest maximum LD block ($[0.8, 1]$ bin). This was surprising since, generally, the performance of models is expected to increase with an increase in maximum LD of the SNPs/SNVs. Furthermore, $[0.8, 1]$ bin does not contain too few SNPs/SNVs in both cases, the phenomenon which could explain the slight drop of accuracy for $[0.6, 0.8]$ bin. While this incident requires deeper analysis, one possible cause could be that the average correlation of indirectly correlated SNPs/SNVs to these

Fig. 5 Breakdown of most common SNV types in missing positions for imputation of all SNVs in chromosome 22 of 1000 Genomes Project dataset. On the left (**a**, **c**, and **e**), charts represent bi-allelic events while on the right (**b**, **d**, and **f**), multi-alleic events are shown. In each block from top to bottom, first plot shows total number of missing positions per SNV type in 3 folds combined, and second plot shows breakdown of missing positions based on LD bins for the same. The last plot shows average accuracy of benchmark models per SNV type. For bi-allelic events, in presence of strong LD structure (e.g., DUP events) and in all cases, we hardly see any performance change with an increase in MR. For multi-allelic events, STI and SCDA+ outperform the rest while we can observe a consistent drop in accuracy with an increase in MR.

events drops rapidly compared to other bins, resulting in reduced accuracy for positions within $[0.8, 1]$ bin.

To our knowledge, this is the first use of transformer architecture to address imputation problem in genomics, which can be extended by integrating privacy-preserving mechanism (through homomorphic encryption and available Tensorflow compatible libraries such as https://github.com/tf-encrypted/tf-encrypted). Since STI divides the sequence into chunks and

makes an isolated network to impute each, it is feasible to implement distributed STI using a native message passing interface, or distributed capabilities in widely used DL libraries such as Tensorflow or PyTorch. This will enable STI to scale for larger sequences. Furthermore, DL models are known to be data intensive. With release of new large panels, such as TopMed, we expect that STI and other DL models produce better results compared to reference-based models.

4 Methods

In this section, we introduce the datasets we used in this study and discuss their characteristics. Additionally, the architectural design of Split-Transformer Impute is put forward, in addition to the loss function used to train the model.

4.1 Data

In this study, we used four datasets from two well-known sequencing projects in order to benchmark STI against baselines. All datasets contain real-world samples. We used Scikit-allel package [48] to compute LD and MAF for the datasets. The characteristics of the datasets is as follows:

Yeast dataset: The first dataset is the comprehensively assayed yeast dataset [45], representing simple genetic background and high correlation among genotypes. This dataset contains 4390 genotyped profiles for 28220 genetic variants. The samples were obtained by sequencing crosses between two strains of yeast, namely an isolate from a vineyard (RM) and a popular laboratory strain (BY). In the original dataset, the data is encoded as -1/1 for BY/RM, which are mapped to 0/1 in our code, respectively, before one-hot encoding.

HLA dataset: This dataset contains human leukocyte antigen genotypes, covering a 3 Mbp region at chromosome 6p21.31, and sitting at major histocompatibility complex (MHC) region. HLA region is in charge of regulation of the immune system in humans [49]. This region is highly polymorphic and heterogeneous among individuals, meaning that it harbors various alleles, enabling the adaptive immune system to be fine-tuned [50]. In this study, we used the genotypes of this region, obtained from the phase 3 of 1000 Genomes Project [27], containing 7161 unique genetic variants for 2504 individuals from five super-populations across the world, namely American (AMR), East Asian (EAS), European (EUR), South Asian (SAS), and African (AFR). All SNPs in this dataset have a maximum LD value in range of $[0.4, 0.5]$. In the preprocessing step, we split HLA samples into paternal and maternal sequences and feed them to the model. In post-processing step, pairs of consecutive sequences are put together to reconstruct the genotypes.

Chromosome 22 datasets: We used SNV data from 1000 Genomes Project in two settings. In the first one, we only selected deletions, excluding ALU deletions, among all SNVs. This resulted in 573 positions harboring biallelic events in the dataset. In the second one, a total of 848 SNVs including,

but not limited to deletion, insertion, duplication, and copy number variations in chromosome 22 are selected. As shown in Figure 2 b & c, the majority of SNVs in chromosome 22 have a low LD, making these datasets challenging for imputation. According to Figure 2.d, deletions cover a wide range of LD among them and other SNVs, making them a good target for a separate bi-allelic dataset. A summary of all available SNVs in chromosome 22 can be found in Table 9 in supplementary. While CNVs and DUPs show a higher internal average maximum LD, we selected DEL events for a separate dataset since the number of SNVs for this event is higher compared to the rest.

4.2 Proposed training procedure

In [2, 43] studies, the training data is filled with missing values using different percentages, e.g., 0.1, 0.2, etc,. In our experiments, we observed that when 50 percent of the SNPs/SNVs in the training data are randomly replaced with missing values in each iteration, the overall performance of the model is improved, in addition to saving time in the training process. The reason for improvement is straightforward: when the missing rate is low, the model is less likely to learn patterns for predicting missing values at every position.

Another improvement that we observed was when instead of feeding one-hot encoded diplotypes to the models, we break them down to haploids first, and then perform one-hot encoding. This idea is proposed in [44] but there is no discussion about the merits of this procedure. Presumably, since SNVs in paternal and maternal haploids are independent, predicting haploids would be easier for the models compared to predicting diploids.

4.3 Split-Transformer Impute architecture.

Split-Transformer Impute is an extended transformer model [37] especially tailored for genotype imputation. STI is a reference-free model, unlike Minimac4. This enables STI to be applied to a wider range of datasets with less effort and fewer preparations. An overview of STI is presented in Figure 1.

Cat-Embedding: One important contribution of this study is introducing categorical embedding, termed as Cat-Embedding, which enables STI to learn embedding representation per allele in each position. Cat-Embedding is basically an extension of positional encoding used in natural language processing and vision-related transformer models. For the imputation task, we consider missing value as another allele. According to Figure 1.b, the vector for each allele is added to the respective positional (SNP) embedding vector to generate final embedding. The alternative design for Cat-Embedding was to consider an embedding vector for each allele in each position, leading to a $\alpha \times \beta \times \gamma$ embedding space. We decided to discard the initial design since compared to our current design it uses more parameters and results in the same performance.

Splitting: To take advantage of existing LD in the data, we split the SNPs/SNVs into windows, in order to limit the scope of attention in the model,

leading to computational memory savings. Additionally, each window passes through a dedicated branch inside the model, leading to increased imputation quality. In a vanilla transformer, the cost of computing a global attention is quadratic with respect to the number of SNPs (m^2); however, the amount is lowered to $(m/w) \times (w+o)^2 = mw$ in STI, considering the overlaps of windows are negligible. For instance, for $m = 10^4$ and a window size of 10^3, STI uses 10 times less memory for attention computations compared to an attention in a vanilla transformer.

Branching: As mentioned above, STI uses a separate branch per window, meaning that a separate network is built for each group of adjacent SNPs. In order to prevent loss of imputation accuracy at window borders, we include additional SNPs from neighboring windows in each split, and shrink the window at the end of each branch. Average LD block size in the dataset can be used to decide the size of overlap.

Attention: The attention blocks are implemented similar to those of other transformers, such as self-attention blocks in Vision Transformer (ViT) [51]. There is a difference between first and second attention blocks in the branches. The first block is a self-attention block, meaning that query, key, and value of the attention layer are the same, query excludes overlapping SNPs from neighbouring windows. This way, we shrink the window to the intended size after applying multi-headed attention. In the second block, query is the output of the previous layer, while key and value are the outputs of the first self-attention block. This skip connection considerably affects the overall performance of the model.

Convolutional blocks: Convolutional blocks are another important components of STI, as illustrated in Figure 1.c. Through empirical studies, we found out that using exactly two parallel convolutional branches, similar to Inception module [52], is the best trade-off between accuracy gain and increase in number of model parameters, compared to using a single branch or more than two branches. Furthermore, Depth-wise convolutional layer at the end of the block helps STI extract local information without mixing channel information, and substantially improves imputation accuracy.

Assembly: Finally, the outputs of all branches are concatenated to form the output, that is either maternal or paternal haplotype in case of 1000 Genomes Project datasets, or the genotypes in case of yeast. For the former, by assembling maternal and paternal haplotypes, we obtain imputed genotypes and the latter needs no further post-processing. Since genetic variations in parents are independent, directly encoding and imputing the genotypes in diploid life-forms results in lower imputation accuracy compared to imputing their haplotypes. Hence we go through extra steps in pre-processing and post-processing for the HLA dataset.

Loss function: For the loss function, we used a combination of Kullback–Leibler divergence (D_{KL}) and categorical cross entropy (CCE), similar to loss function of variational autoencoder [53], as follows:

$$Loss(y, \hat{y}) = (\theta)CCE(y, \hat{y}) + (1 - \theta)D(y\|\hat{y}), \tag{1}$$

where θ is the weight parameter, and the first term, representing categorical cross entropy, and the second term, representing Kullback–Leibler divergence loss, are calculated as follows:

$$CCE(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} \log\left(p(y_{ij})\right) \tag{2}$$

$$D_{KL}(y\|\hat{y}) = \sum_{i=1}^{N} p(y_i)\frac{p(y_i)}{p(\hat{y}_i)} \tag{3}$$

We set θ to 0.5, meaning that STI minimizes Equations 2, 3 equally. CCE captures reconstruction error between the input and the output, while D_{KL} measures asymmetric distance, with y as the base, between their probability distributions. In our experiments, omitting any of these losses resulted in a reduced model performance.

Data availability

All data used in this study are publicly available. The yeast dataset can be found as the *Supplementary Data 5* at https://www.nature.com/articles/ncomms9712 and the rest of datasets are extracted from the 1000 Genomes Project phase 3 dataset available at http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/.

Code availability

The source code of STI is publicly available on GitHub (https://github.com/shilab/STI)

Acknowledgments

This work is partially supported by the US National Science Foundation (Award Number: 1750632).

Contributions

M.E.M. developed the method with the help from J.C., B.J., and X.S. and M.E.M. implemented the code. C.L. prepared the datasets. M.E.M, C.L., and X.S. conducted the data analysis. M.E.M, C.L., S.K, T.R.R., and X.S. wrote the manuscript. All the authors read and approved the submitted manuscript.

Ethics declarations

Competing interests. The authors declare that they have no competing interests.

Declarations

Not applicable.

References

[1] Torkamaneh, D., Belzile, F.: Accurate imputation of untyped variants from deep sequencing data. Deep Sequencing Data Analysis, 271–281 (2021)

[2] Song, M., Greenbaum, J., Luttrell IV, J., Zhou, W., Wu, C., Luo, Z., Qiu, C., Zhao, L.J., Su, K.-J., Tian, Q., et al.: An autoencoder-based deep learning method for genotype imputation. Frontiers in Artificial Intelligence **5** (2022)

[3] Das, S., Abecasis, G.R., Browning, B.L.: Genotype imputation from large reference panels. Annu Rev Genomics Hum Genet **19**(1), 73–96 (2018)

[4] Graffelman, J., Nelson, S., Gogarten, S., Weir, B.: Exact inference for hardy-weinberg proportions with missing genotypes: Single and multiple imputation. G3: Genes, Genomes, Genetics **5**(11), 2365–2373 (2015)

[5] Wigginton, J.E., Cutler, D.J., Abecasis, G.R.: A note on exact tests of hardy-weinberg equilibrium. The American Journal of Human Genetics **76**(5), 887–893 (2005)

[6] Pei, Y.-F., Li, J., Zhang, L., Papasian, C.J., Deng, H.-W.: Analyses and comparison of accuracy of different genotype imputation methods. PloS one **3**(10), 3551 (2008)

[7] Auer, P.L., Wang, G., Project, N.E.S., Leal, S.M.: Testing for rare variant associations in the presence of missing data. Genetic epidemiology **37**(6), 529–538 (2013)

[8] Keavney, B., McKenzie, C.A., Connell, J.M., Julier, C., Ratcliffe, P.J., Sobel, E., Lathrop, M., Farrall, M.: Measured haplotype analysis of the angiotensin-i converting enzyme gene. Human Molecular Genetics **7**(11), 1745–1751 (1998)

[9] George, V.T., Elston, R.C., Rao, D.: Testing the association between polymorphic markers and quantitative traits in pedigrees. Genetic epidemiology **4**(3), 193–201 (1987)

[10] Burdick, J.T., Chen, W.-M., Abecasis, G.R., Cheung, V.G.: In silico method for inferring genotypes in pedigrees. Nature genetics **38**(9), 1002–1004 (2006)

[11] Scott, L.J., Mohlke, K.L., Bonnycastle, L.L., Willer, C.J., Li, Y., Duren, W.L., Erdos, M.R., Stringham, H.M., Chines, P.S., Jackson, A.U., *et al.*: A genome-wide association study of type 2 diabetes in finns detects multiple susceptibility variants. science **316**(5829), 1341–1345 (2007)

[12] Burton, P., Clayton, D., Cardon, L., Craddock, N., Deloukas, P., Duncanson, A., Kwiatkowski, D., McCarthy, M., Ouwehand, W., Samani, N., *et al.*: Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. Nature **447**(7145), 661–678 (2007)

[13] Orho-Melander, M., Melander, O., Guiducci, C., Perez-Martinez, P., Corella, D., Roos, C., Tewhey, R., Rieder, M.J., Hall, J., Abecasis, G., *et al.*: Common missense variant in the glucokinase regulatory protein gene is associated with increased plasma triglyceride and c-reactive protein but lower fasting glucose concentrations. Diabetes **57**(11), 3112–3121 (2008)

[14] Deng, T., Zhang, P., Garrick, D., Gao, H., Wang, L., Zhao, F.: Comparison of genotype imputation for snp array and low-coverage whole-genome sequencing data. Frontiers in genetics **12** (2021)

[15] Calus, M., Bouwman, A., Hickey, J., Veerkamp, R., Mulder, H.: Evaluation of measures of correctness of genotype imputation in the context of genomic prediction: a review of livestock applications. animal **8**(11), 1743–1753 (2014)

[16] Hirschhorn, J.N., Daly, M.J.: Genome-wide association studies for common diseases and complex traits. Nature reviews genetics **6**(2), 95–108 (2005)

[17] Hugot, J.-P., Chamaillard, M., Zouali, H., Lesage, S., Cézard, J.-P., Belaiche, J., Almer, S., Tysk, C., O'Morain, C.A., Gassull, M., *et al.*: Association of nod2 leucine-rich repeat variants with susceptibility to crohn's disease. Nature **411**(6837), 599–603 (2001)

[18] Ogura, Y., Bonen, D.K., Inohara, N., Nicolae, D.L., Chen, F.F., Ramos, R., Britton, H., Moran, T., Karaliuskas, R., Duerr, R.H., *et al.*: A frameshift mutation in nod2 associated with susceptibility to crohn's disease. Nature **411**(6837), 603–606 (2001)

[19] Rioux, J.D., Daly, M.J., Silverberg, M.S., Lindblad, K., Steinhart, H., Cohen, Z., Delmonte, T., Kocher, K., Miller, K., Guschwan, S., *et al.*: Genetic variation in the 5q31 cytokine gene cluster confers susceptibility to crohn disease. Nature genetics **29**(2), 223–228 (2001)

[20] Stefansson, H., Petursson, H., Sigurdsson, E., Steinthorsdottir, V., Bjornsdottir, S., Sigmundsson, T., Ghosh, S., Brynjolfsson, J., Gunnarsdottir, S., Ivarsson, O., *et al.*: Neuregulin 1 and susceptibility to schizophrenia.

The American Journal of Human Genetics **71**(4), 877–892 (2002)

[21] Nisticò, L., Buzzetti, R., Pritchard, L.E., Van der Auwera, B., Giovannini, C., Bosi, E., Martinez Larrad, M.T., Serrano Rios, M., Chow, C., Cockram, C.S., *et al.*: The ctla-4 gene region of chromosome 2q33 is linked to, and associated with, type 1 diabetes. Human molecular genetics **5**(7), 1075–1080 (1996)

[22] Li, X., Quick, C., Zhou, H., Gaynor, S.M., Liu, Y., Chen, H., Selvaraj, M.S., Sun, R., Dey, R., Arnett, D.K., et al.: Powerful, scalable and resource-efficient meta-analysis of rare variant associations in large whole genome sequencing studies. Nature Genetics, 1–11 (2022)

[23] Zeggini, E., Scott, L.J., Saxena, R., Voight, B.F., Marchini, J.L., Hu, T., de Bakker, P.I., Abecasis, G.R., Almgren, P., Andersen, G., *et al.*: Meta-analysis of genome-wide association data and large-scale replication identifies additional susceptibility loci for type 2 diabetes. Nature genetics **40**(5), 638–645 (2008)

[24] Panagiotou, O.A., Willer, C.J., Hirschhorn, J.N., Ioannidis, J.P.: The power of meta-analysis in genome-wide association studies. Annual review of genomics and human genetics **14**, 441–465 (2013)

[25] Skol, A.D., Scott, L.J., Abecasis, G.R., Boehnke, M.: Optimal designs for two-stage genome-wide association studies. Genetic Epidemiology: The Official Publication of the International Genetic Epidemiology Society **31**(7), 776–788 (2007)

[26] Evangelou, E., Maraganore, D.M., Ioannidis, J.P.: Meta-analysis in genome-wide association datasets: strategies and application in parkinson disease. PLoS One **2**(2), 196 (2007)

[27] Consortium, .G.P., *et al.*: A global reference for human genetic variation. Nature **526**(7571), 68 (2015)

[28] Huang, J., Howie, B., McCarthy, S., Memari, Y., Walter, K., Min, J.L., Danecek, P., Malerba, G., Trabetti, E., Zheng, H.-F., *et al.*: Improved imputation of low-frequency and rare variants using the uk10k haplotype reference panel. Nature communications **6**(1), 8111 (2015)

[29] Taliun, D., Harris, D.N., Kessler, M.D., Carlson, J., Szpiech, Z.A., Torres, R., Taliun, S.A.G., Corvelo, A., Gogarten, S.M., Kang, H.M., *et al.*: Sequencing of 53,831 diverse genomes from the nhlbi topmed program. Nature **590**(7845), 290–299 (2021)

[30] A reference panel of 64,976 haplotypes for genotype imputation. Nature genetics **48**(10), 1279–1283 (2016)

[31] Mathias, R.A., Taub, M.A., Gignoux, C.R., Fu, W., Musharoff, S., O'Connor, T.D., Vergara, C., Torgerson, D.G., Pino-Yanes, M., Shringarpure, S.S., *et al.*: A continuum of admixture in the western hemisphere revealed by the african diaspora genome. Nature communications **7**(1), 12522 (2016)

[32] O'Connell, J., Yun, T., Moreno, M., Li, H., Litterman, N., Kolesnikov, A., Noblin, E., Chang, P.-C., Shastri, A., Dorfman, E.H., *et al.*: A population-specific reference panel for improved genotype imputation in african americans. Communications biology **4**(1), 1–9 (2021)

[33] Cong, P.-K., Bai, W.-Y., Li, J.-C., Yang, M.-Y., Khederzadeh, S., Gai, S.-R., Li, N., Liu, Y.-H., Yu, S.-H., Zhao, W.-W., *et al.*: Genomic analyses of 10,376 individuals in the westlake biobank for chinese (wbbc) pilot project. Nature Communications **13**(1), 1–15 (2022)

[34] The genomeasia 100k project enables genetic discoveries across asia. Nature **576**(7785), 106–111 (2019)

[35] Bartlett, J.W., Seaman, S.R., White, I.R., Carpenter, J.R., Initiative*, A.D.N.: Multiple imputation of covariates by fully conditional specification: accommodating the substantive model. Statistical methods in medical research **24**(4), 462–487 (2015)

[36] Das, S., Forer, L., Schönherr, S., Sidore, C., Locke, A.E., Kwong, A., Vrieze, S.I., Chew, E.Y., Levy, S., McGue, M., *et al.*: Next-generation genotype imputation service and methods. Nature genetics **48**(10), 1284–1287 (2016)

[37] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

[38] Desimone, R., Duncan, J.: Neural mechanisms of selective visual attention. Annual review of neuroscience **18**(1), 193–222 (1995)

[39] Cho, K., Courville, A., Bengio, Y.: Describing multimedia content using attention-based encoder-decoder networks. IEEE Transactions on Multimedia **17**(11), 1875–1886 (2015)

[40] Weir, B.: Linkage disequilibrium and association mapping. Annual review of genomics and human genetics **9**(1), 129–142 (2008)

[41] Browning, B.L., Zhou, Y., Browning, S.R.: A one-penny imputed genome from next-generation reference panels. The American Journal of Human Genetics **103**(3), 338–348 (2018)

[42] Howie, B.N., Donnelly, P., Marchini, J.: A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. PLoS genetics **5**(6), 1000529 (2009)

[43] Chen, J., Shi, X.: Sparse convolutional denoising autoencoders for genotype imputation. Genes **10**(9), 652 (2019)

[44] Naito, T., Suzuki, K., Hirata, J., Kamatani, Y., Matsuda, K., Toda, T., Okada, Y.: A deep learning method for hla imputation and trans-ethnic mhc fine-mapping of type 1 diabetes. Nature communications **12**(1), 1–14 (2021)

[45] Bloom, J.S., Kotenko, I., Sadhu, M.J., Treusch, S., Albert, F.W., Kruglyak, L.: Genetic interactions contribute less than additive effects to quantitative trait variation in yeast. Nature communications **6**(1), 1–6 (2015)

[46] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)

[47] Lin, P., Hartz, S.M., Zhang, Z., Saccone, S.F., Wang, J., Tischfield, J.A., Edenberg, H.J., Kramer, J.R., M. Goate, A., Bierut, L.J., *et al.*: A new statistic to evaluate imputation reliability. PloS one **5**(3), 9697 (2010)

[48] Miles, A., pyup.io bot, R., M., Ralph, P., Harding, N., Pisupati, R., Rae, S., Millar, T.: Cggh/scikit-allel: V1.3.3. https://doi.org/10.5281/zenodo.4759368. https://doi.org/10.5281/zenodo.4759368

[49] Hillert, J.: Human leukocyte antigen studies in multiple sclerosis. Annals of Neurology: Official Journal of the American Neurological Association and the Child Neurology Society **36**(S1), 15–17 (1994)

[50] Terasaki, P.I., Cai, J.: Human leukocyte antigen antibodies and chronic rejection: from association to causation. Transplantation **86**(3), 377–383 (2008)

[51] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)

[52] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)

[53] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)