Methods 233 (2025) 19–29

Contents lists available at ScienceDirect

Methods

journal homepage: www.elsevier.com/locate/ymeth

Data preprocessing methods for selective sweep detection using convolutional neural networks

Hanqing Zhao, Nikolaos Alachiotis *

University of Twente, Drienerlolaan 5, Enschede, 7522 NB, Overijssel, the Netherlands

ARTICLE INFO

Dataset link: https://figshare.com/articles/dataset/SweepNet-Datasets/22194118, https://github.com/Zhaohq96/Genetic-data-rearrangement

Keywords:
Population genetics
Selective sweep
Convolutional neural networks
Data preprocessing

ABSTRACT

The identification of positive selection has been framed as a classification task, with Convolutional Neural Networks (CNNs) already outperforming summary statistics and likelihood-based approaches in accuracy. Despite the prevalence of CNN-based methods that manipulate the pixels of images representing raw genomic data as a preprocessing step to improve classification accuracy, the efficacy of these pixel-rearrangement techniques remains inadequately examined, particularly in the presence of confounding factors like population bottlenecks, migration and recombination hotspots. We introduce a set of pixel rearrangement algorithms aimed at enhancing CNN classification accuracy in detecting selective sweeps. These algorithms are employed to assess the performance of four CNN models for selective sweep detection. Our findings illustrate that the judicious application of rearrangement algorithms notably enhances the overall classification accuracy of a CNN across various datasets simulating confounding factors. We observed that sorting the columns of the genomic matrices has higher on CNN performance than rearranging the sequences. To some extent, these rearrangement algorithms are more robust to misspecified demographic models compared with the utilization of the default preprocessing algorithm as suggested by the respective authors of each CNN architecture. We provide the data rearrangement algorithms as a distinct package available for download at: https://github.com/Zhaohq96/Genetic-data-rearrangement.

1. Introduction

Positive selection is a crucial catalyst for adaptation, profoundly shaping the evolutionary paths of species. The recognition of positive selection serves as a cornerstone in unraveling evolutionary mechanisms [34] and can help us understand the genetic factors behind diseases [33]. Recent findings have underscored positive selection's role as the driving force behind the early evolution of SARS-CoV-2 [43]. The identification of positive selection holds practical significance in medicine, aiding in the detection of drug-resistant mutations in pathogens [10] and the development of more potent drug treatments [6].

Positive selection leaves a discernible mark on a population by altering its genetic diversity. This imprint is known as a "selective sweep", signifying the rise in frequency of neighboring neutral alleles linked to the selected locus due to genetic hitchhiking [39]. Traditional methods for detecting selective sweeps rely on summary statistics [40,13,4] or likelihood models [35,11,5]. However, these methods are limited by their dependence on a restricted set of features and are often confounded by evolutionary variables such as changes in population size, migration, and recombination hotspots. In recent years, as machine learning (ML) gained prominence in predictive tasks across various domains [21], population geneticists have turned to supervised ML methods for the detection of selective sweeps [38,31,37,24,28]. Approaches such as SURFDAWave [31] and S/HIC [37] leverage machine learning techniques to achieve higher inference accuracy compared to previous methodologies. Timesweeper [44] utilize SLiM [17] and stdpopsim [2] to generate population genomic time-series data as the input for training a 1D CNN to identify selective sweeps. However, these methods still rely on summary statistics, which are susceptible to the influence of various evolutionary factors. Moreover, it remains uncertain whether an extensive array of summary statistics can comprehensively capture all the information embedded within raw genomic data. There is ongoing exploration into whether alternative approaches can overcome these limitations and provide deeper insights into the complex dynamics of selective sweeps.

Recent studies [9,15,42,32,46,47,7] have explored an alternative approach to detect selective sweeps by framing it as an image classification task. These studies employ Convolutional Neural Networks (CNNs)

* Corresponding author.
E-mail addresses: h.zhao-1@utwente.nl (H. Zhao), n.alachiotis@utwente.nl (N. Alachiotis).

https://doi.org/10.1016/j.ymeth.2024.11.003
Received 20 May 2024; Received in revised form 28 October 2024; Accepted 4 November 2024
Available online 15 November 2024
1046-2023/© 2024 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

applied directly to images representing raw genomic data, allowing for the preservation and exploitation of all genetic information. By treating genomic data as images, CNNs can uncover intrinsic features across multiple dimensions without biases or losses associated with user-defined vectors.

Beyond the direct applicability of CNNs to raw genomic data, these investigations underscore the potential of data preprocessing in augmenting the classification capabilities of CNNs. Flagel et al. [15] utilized 1D CNNs with sorting the rows of the alignment matrix based on genetic similarity for detecting recent positive selection. Torada et al. [42] explored how data rearrangements affect the performance of CNNs by sorting the rows and columns based on their frequency of occurrence separately, and adopted a 2D CNN with sorting both rows and columns for detection and quantification of natural selection. Ray et al. [36] applied a variant of the UNet++ architecture [48] with sorting the rows by seriation, which is a statistical method aiming to arrange vectors so that neighboring vectors in the sequence are close to each other [23], to identify introgressed alleles via semantic segmentation. Cecil and Sugden [8] further interpreted the effect of preprocessing on CNN-based models via visualizing ImaGene [42] (a CNN-based method that reorders rows/columns of the genomic matrices for identifying selective sweeps) with SHAP explanations [29], indicating that the largest haplotype blocks in the reordered genomic matrices contribute the most to model decision. In images representing genomic data, each row commonly corresponds to an individual and each column represents a Single Nucleotide Polymorphism (SNP). Unlike traditional pixel-level preprocessing methods such as normalization, padding, and resizing, row/column-aware preprocessing techniques are deemed more effective.

Our previous study [45] evaluated a range of single-element and pairwise preprocessing methods grounded in genetic and information theory for selective sweep detection using recently introduced CNN architectures [15,42,32,46]. In this work, we extend the experimental setup to examine the effect of data rearrangement methods on demographic model misspecification and varying genetic window sizes used for classification. We find that the aforementioned CNN architectures benefit from certain data rearrangement algorithms on overall performance under diverse genetic scenarios. To some extent, the best combinations of algorithms are more tolerant to model misspecification than the original data preprocessing method that each CNN-based method used. Additionally, we find that data preprocessing algorithms yield more robust classification to different window sizes than using the original data preprocessing method that each CNNs used.

2. Background

Evolution is driven by several key forces such as natural selection and recombination. Natural selection is a key factor in the process of evolution [12], whereby organisms possessing advantageous traits for survival and reproduction are predisposed to transmit such traits to subsequent generations. Positive selection denotes an evolutionary force that increases the prevalence of a genetic variant in a population, occurring when an allele is favored by natural selection. Additionally, recombination, occurring during reproductive processes, contributes to offspring variability by reshuffling genetic material, thereby ensuring the maintenance of genetic diversity within the population.

Positive selection leaves its signature, a selective sweep, when acting on a population. Distinct patterns left in genomes after a selective sweep allow for detecting signatures of positive selection within a population. According to the classical selective sweep model outlined by Smith and Haigh [39], these genetic patterns include: a) a shift in the site frequency spectrum (SFS) toward low- and high-frequency derived variants [13], b) a noticeable linkage disequilibrium (LD) pattern [26], and c) decreased genetic diversity in the vicinity of the selected locus [39,22]. SFS represents the frequencies of different genetic variants across the genome of a population. After a selective sweep, the frequency of the beneficial allele increases rapidly, reducing genetic variation nearby and causing a U-shaped Site Frequency Spectrum (SFS) with more rare alleles and fewer intermediate-frequency alleles. LD is a measure of how often two genetic variants occur together on the same chromosome more often than would be expected by chance. When positive selection occurs, high LD occurs on the same side as the favored allele because it drags nearby variants along, reducing genetic diversity in that region. Conversely, there's low LD on the opposite side of the favored allele due to recombination events, restoring genetic diversity and breaking down associations with distant variants. The ω-statistic can be utilized for the detection of selective sweeps by searching for the distinct LD pattern. Given a genomic window with S polymorphic sites split into two groups by the l-th site, the ω-statistic can be calculated by the following function:

$$\omega = \frac{\left(\binom{l}{2} + \binom{S-l}{2}\right)^{-1}\left(\sum_{i,j\in L} r_{ij}^2 + \sum_{i,j\in R} r_{ij}^2\right)}{(1/l(S-l))\sum_{i\in L, j\in R} r_{ij}^2} \text{ [26]}, \quad (1)$$

where L and R denote the left and right set of polymorphic sites, respectively. r^2 represents correlation coefficient, which quantifies the extent of LD between alleles at two loci. r_{ij}^2 is r^2 between the ith and jth sites. When the ω-statistic shows a high value at a particular genomic location and is accompanied by extended contiguous genomic regions with high LD on both sides, while the LD level between these regions remains relatively low, it suggests a potential selective sweep in the region.

Fig. 1 shows how positive selection affects the frequency of mutations (both beneficial and linked) in a population at different time points. Each row represents an individual, with white circles indicating neutral mutations and black circles indicating advantageous mutations. An advantageous mutation appears in an individual (Fig. 1a), which then spreads throughout the population over generations (Fig. 1b). When the frequency of the advantageous mutation reaches 1.0, meaning all individuals carry it, the mutation is considered fixed (Fig. 1c). Genetic hitchhiking causes nearby linked alleles' frequencies to change. High LD is observed on each side of the selection target, while low LD is observed between loci on different sides.

Identifying positive selection in a population involves recognizing specific patterns it leaves in the genome, essentially, distinguishing them from neutral regions. Genomic sequences, represented as character sequences, can be converted into images by assigning colorful pixels to mutations. The implementation of converting genomic data into image pixels sets the stage for using Convolutional Neural Networks (CNNs), a type of deep learning method widely applied for image classification, to detect positive selection. Several recent researches [15,42,32,46,47] have already utilized CNNs to identify selective sweeps, achieving comparable performance with summary-statistic-based methods. Most of the aforementioned methods apply data preprocessing methods to enhance classification ability of CNNs. However, most of their preprocessing methods were evaluated under certain demographic models, questioning the universality of how the data preprocessing methods perform under extreme cases, e.g., confounding factors, evolutionary model misspecification. In consideration that input data is one of the important factors determining the CNNs' outcomes, careful assessment and application are essential in data preprocessing.

3. Material and methods

3.1. Setup

We extended the experimental setup of our previous work [45]. We conducted simulations based on three distinct demographic models that introduce complexities in the identification of selective sweeps: population bottlenecks, migration events, and regions with high rates of recombination (recombination hotspots). Our simulations assumed a present-day population size of 50,000 haploid genomes, with a sample size of 128 individuals. Selective sweeps were simulated at the central loci of the genomes under study, from which we extracted matrices

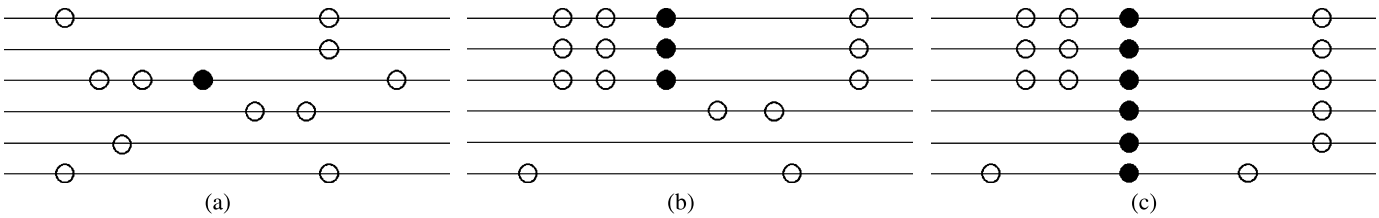

Fig. 1. A selective sweep. a. An advantageous mutation (black circle) appears in the population, where the white circle denotes neutral mutation. b. The frequency of the chromosome that carries the advantageous mutation increases. c. The advantageous mutation appears in all individuals of the population (fixed beneficial mutation).

Table 1
Simulated datasets for evaluation (source from [45]). "s", "bt", "d", "jt" and "ri" represent severity, begin time, duration, population join time and recombination intensity, respectively.

Dataset	Confounding factor	Simulation software	Model parameters
D1	Mild bottleneck	ms and mssel	s = 0.5, bt = 0.1, d = 0.01
D2	Severe bottleneck	ms and mssel	s = 0.005, bt = 0.1, d = 0.004
D3	Recent migration	ms and mssel	jt = 0.003
D4	Old migration	ms and mssel	jt = 3
D5	Low intensity recombination	msHOT and mbs	ri = 2
D6	High intensity recombination	msHOT and mbs	ri = 20

Table 2
Details of CNN architectures for 128 × 128 images size (source from [45]).

Authors	Convolution type	Kernel size	Depth	Mnemonic
Flagel et al. [15]	1D convolution	64	5	Net-1
Torada et al. [42]	2D convolution	3×3	4	ImaGene
Nguembang F et al. [32]	2D convolution	10×10 & 5×5	4	Net-2
Zhao et al. [46]	2D convolution	2×2	5	SweepNet

consisting of 128 Single Nucleotide Polymorphisms (SNPs) from each simulation. These extracted matrices underwent preprocessing and were converted into binary grayscale images sized 128 × 128 pixels. Here, we assigned '0' (representing the ancestral state) as white and '1' (representing the derived state) as black. The simulated datasets are detailed in Table 1 and are accessible at the provided link https://figshare.com/articles/dataset/SweepNet-Datasets/22194118. Each dataset comprises three subsets per class: a training set containing 800 images, a validation set comprising 200 images, and a test set consisting of 1,000 images. All datasets are balanced, i.e., each class contains an equal number of samples in both the training, validation, and testing sets. For classification purposes, we deployed four distinct Convolutional Neural Networks (CNNs), the specifics of which are outlined in Table 2, tailored for processing images sized 128 × 128 pixels. The original data preprocessing methods for the CNNs are 1) sorting the sequences by hamming distance and sorting the SNPs by derived allele frequency for SweepNet, 2) sorting both sequences and SNPs by the frequency of the occurrence for ImaGene, 3) sorting the sequences by genetic similarity for Net-1, and 4) no data preprocessing for Net-2. The data rearrangement algorithms were implemented in the C programming language. A fixed number of SNPs were extracted from the center of each simulation, where the selective sweeps were introduced, and the sorted genomic matrices were saved as text files. Finally, a Python script was used to convert the genomic matrices into a designated image format, which was then provided as input to the CNN models.

3.2. Mathematical concepts

Hamming weight The hamming weight [41], referred to as the bit count or population count, is the total number of non-zero bits in a binary sequence:

$$\mathrm{hw}(x) = \sum_{i=1}^{n} [x_i = 1], \tag{2}$$

where x is the binary string of length n, and x_i is the i-th bit of the string (0 or 1). The hamming weight of a binary string x is equivalent to the count of the number of 1's present in x.

Hamming distance Hamming distance, a metric used to measure the difference between two strings of equal length, is defined as the number of positions at which the corresponding symbols in the two strings are different:

$$d_H(x, y) = \sum_{i=1}^{n} [x_i \neq y_i], \tag{3}$$

where x and y are two strings of equal length n, and $[x_i \neq y_i]$ is an indicator function that equals 1 when the i-th symbol of x is different from the i-th symbol of y, and 0 otherwise.

Correlation coefficient The correlation coefficient is a statistical measure of the strength and direction of the linear relationship between two variables.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}, \tag{4}$$

where n is the number of observations while x_i and y_i represent observations of x and y, and $\bar{x}$ and $\bar{y}$ are the means of x and y, respectively.

3.3. Sequence-based preprocessing methods

Hamming weight The sequences are rearranged in descending order based on their hamming weights.

Numerical representation Every sequence is encoded as an integer value, and these values are sorted in descending order to determine the sequence arrangement.

Table 3
Preprocessing methods (adapted from [45]).

Method (Abbreviation)	Applicable for	Attribute
Hamming weight (HW)	sequences/SNPs	single-element
Numerical representation (NUM)	sequences	single-element
Minor allele frequency (MAF)	SNPs	single-element
Bit-flipping minor allele frequency (MAFB)	SNPs	single-element
Occurrence frequency (FRQ)	sequences/SNPs	single-element
Correlation coefficient (COR)	sequences/SNPs	pairwise
Hamming distance (HD)	sequences/SNPs	pairwise
Hamming distance with hamming weight (HDW)	sequences	pairwise
Bipartite correlation (CORR)	SNPs	pairwise
Sub-regions bipartite correlation (CORS)	SNPs	pairwise

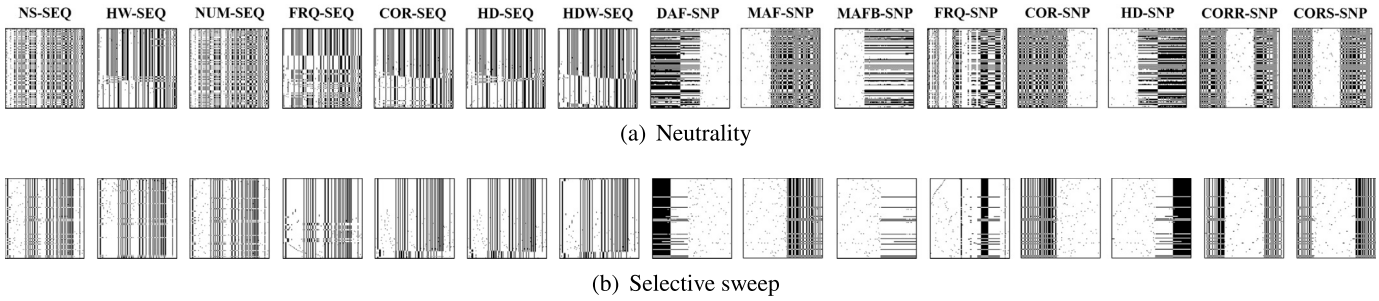

Fig. 2. Rearranged images of a severe bottleneck (source from [45]). "SEQ" and "SNP" represent sorting sequences and SNPs, respectively. Henceforth, "NS" denotes no sorting, while "DAF" refers to sort SNPs based on Hamming weight. Images within the same column underwent identical rearrangement procedures.

Correlation coefficient The sequence exhibiting the highest average correlation with all other sequences serves as the reference and is placed at the top. Subsequently, the remaining sequences are sorted in descending order according to their correlation with the reference.

Hamming distance The sequence displaying the lowest average hamming distance with all other sequences is positioned at the top. Subsequent sequences are sorted in descending order based on their hamming distance from this reference.

Hamming distance based on hamming weight The sequence with the highest hamming weight is positioned at the top, followed by the remaining sequences arranged in descending order based on their hamming distance from the initial sequence.

3.4. SNP-based preprocessing methods

Derived allele frequency (mathematically, hamming weight) The SNPs are rearranged in descending order from the right to the left based on their derived allele frequency.

Minor allele frequency The SNPs are rearranged in ascending order according to their minor allele frequency.

Minor allele frequency with bit-flipping The ancestral and derived states are represented by '0' and '1', respectively. After sorting the SNPs, any SNP where the lowest-frequency state differs from the respective lowest-frequency state in first SNP is considered as having a different minor allele state than the first SNP in the final analysis. SNPs are initially arranged in increasing order based on minor allele frequency from right to left. Then, the SNPs with a different minor allele than the first SNP are flipped.

Correlation coefficient The SNP with the highest average correlation to others is placed first (leftmost SNP), and subsequent SNPs are sorted in descending order based on their correlation with the first.

Hamming distance The SNP with the lowest average Hamming distance to others is placed first (leftmost SNP), and subsequent SNPs are sorted based on their Hamming distance to the reference in descending order.

Bipartite correlation The pair of SNPs with the lowest correlation are placed first, one at the leftmost and the other at the rightmost position. Subsequent SNPs are reordered toward the center based on decreasing correlation with the first two.

Sub-regions bipartite correlation Considering the LD patterns expected on both sides of the favored allele, we evaluate LD based on the correlation coefficient and limit column rearrangement in distinct regions left and right of the favored allele (assumed at the center of the image). SNPs are divided into left and right regions. For each region, we calculate the correlation coefficient of each SNP with all others in the same region. The pair in each region with the lowest correlation is selected as reference for the region. The references are positioned at the edges of their respective regions. SNPs are then reordered based on their correlation with the corresponding reference SNP, moving towards the center. Fig. 2 provides examples of 128×128 images of neutrality and a selective sweep for each preprocessing method. See Table 3 for the abbreviations.

3.5. Evaluation metrics

We assessed the performance of CNNs using accuracy, which indicates the rate of accurate predictions for the highest-probability class in a given task. To gauge the impact of preprocessing methods on CNN classification performance across various scenarios, we introduced a modified variation of variance. This variation is centered on the highest observed accuracy rather than the mean observation for a particular dataset. We henceforth refer to this definition as "variance", which is computed as:

$$Var(X) = \frac{1}{n}\sum_{i=1}^{n}(Acc_{top-1_{x_i}} - Acc_{top-1_{highest_i}})^2, \quad (5)$$

where n is the total number of datasets, $Acc_{top-1_{x_i}}$ is the accuracy for a specific preprocessing method for the i-th dataset and $Acc_{top-1_{highest_i}}$

is the highest accuracy observed over all methods for the i-th dataset. A preprocessing method's overall performance improves as the variance decreases.

To quantify the overall performance of each preprocessing method for different genetic scenarios, we use the equation:

$$Nor_{score}(x_i) = \log_{Var_{lowest}} Var_{x_i}, \quad (6)$$

where Var_{x_i} is the variance of the preprocessing method x_i while Var_{lowest} is the preprocessing method with the overall best performance for all datasets. A higher score indicates better algorithm performance, while the maximum normalized score of 1 indicates the overall best preprocessing method.

4. Results and discussion

4.1. Accuracy

The CNN-based methods have achieved promising performance on identifying selective sweeps. Most of the existing methods reorder pixels of images converted from raw genomic data and improve the classification accuracy. However, most of the aforementioned published methods with data preprocessing were only evaluated on a certain demographic model, which may limit their effectiveness under different genetic scenarios. In consideration of the complexity and variety of demographic models, we thoroughly evaluated how different data rearrangement methods implemented with four established CNN architectures perform on distinct genetic scenarios. In our previous work [45], we trained all CNN architectures with different data preprocessing methods for 10 epochs. Due to random initialization of training weights, some CNN architectures could not be trained effectively within 10 epochs for certain cases. In this study, we initially trained all architectures to understand their performance under various data preprocessing methods. To make sure all models being effectively trained, we trained each CNN model for 1000 epochs. To avoid overfitting, we adopted an early-stopping criterion to terminate the training process when the validation accuracy does not increase for 30 epochs. We also fixed the initial weights by setting a seed for the random number generation in TensorFlow [1], making the results reproducible. Finally, the model with the highest validation accuracy was stored and used for classification.

Fig. 3 shows the accuracy results of various CNNs using different preprocessing methods under various genetic scenarios. Under a mild population bottleneck, SweepNet (Fig. 3a) achieved accuracy of 0.9875 or higher across different preprocessing methods. The following six combinations of preprocessing methods for sorting sequences and SNPs achieved the same highest accuracy of 0.9995: 1) HD and DAF, 2) COR and COR, 3) FRQ and COR, 4) NS and COR, 5) COR and MAF, and 6) COR and FRQ. For a severe bottleneck, SweepNet exhibited accuracy ranging from 0.8985 to 0.9385. Sorting the sequences by FRQ and not sorting the SNPs achieved the best performance. In the case of recent migration, SweepNet achieved the highest accuracy of 0.9945 out of all methods by sorting the sequences based on COR and sorting the SNPs by CORR. Under old migration, the performance of SweepNet was affected significantly by data preprocessing, ranging from 0.836 to 0.9075. The best performance was achieved by sorting the sequences with FRQ and sorting the SNPs using DAF. For recombination hotspots, SweepNet achieved the highest accuracy of 0.89 (sorting the sequences by HDW and the SNPs by DAF/FRQ) in the case of a low intensity recombination hotspot, and 0.9975 (sorting the sequences by HD and SNPs by FRQ) under the high intensity event.

Fig. 3b displays the accuracy results for ImaGene. Under certain cases, ImaGene could not distinguish selective sweeps from neutral regions. For a mild bottleneck, sorting the sequences by FRQ while sorting the SNPs by HD achieved the highest accuracy of 0.999. Under a severe bottleneck, ImaGene achieved the highest accuracy of 0.935 with sorting the sequences by NUM while not sorting the SNPs. For recent migration, the best performance of 0.9885 accuracy was achieved with sorting the sequences by HDW and sorting the SNPs by MAFB. In the case of old migration, ImaGene displayed the highest accuracy of 0.828 with rearranging the sequences by FRQ and sorting the SNPs by DAF. ImaGene exhibited the highest accuracies of 0.845 (sequences by COR and SNPs by HD) and 0.995 (sequences by either HW or NUM and SNPs by FRQ) for low and high intensity recombination hotspots, respectively.

As shown in Fig. 3c, Net-1 achieved the highest accuracy of 0.9855 (sequences sorted by HW while SNPs sorted by MAFB) for a mild bottleneck and 0.9165 (sequences sorted by NS/FRQ while SNPs sorted by CORS) for a severe bottleneck. The best accuracy for recent and old migration was 0.981 (sequences sorted by HW while SNPs sorted by DAF) and 0.796 (sequences sorted by NUM while SNPs sorted by MAFB), respectively. In the case of low intensity recombination, the highest accuracy was 0.8005, achieved by sorting the sequences based on COR and sorting the SNPs based on DAF. For a high intensity recombination hotspot, Net-1 attained the highest accuracy of 0.9925 with sorting the sequences by NUM and the SNPs by FRQ.

Fig. 3d exhibits the results of Net-2. Net-2 achieved the best accuracy of 0.997 (sorting the sequences by COR and sorting the SNPs by DAF) for a mild bottleneck and 0.9295 (only sorting the sequences by HW) for a severe bottleneck. Under recent migration, the highest accuracy is 0.9875, attained by sorting the sequences based on HDW and sorting the SNPs by FRQ. For the rest of the genetic scenarios, most of the data rearrangement methods were able to distinguish selective sweeps. The best accuracy values for the rest of the genetic scenarios were 0.802 (sequences sorted by FRQ, SNPs sorted by MAFB) for old migration, 0.8535 (only sequences sorted by NUM) for a low intensity recombination, and 0.9965 (sequences sorted by NUM, SNPs sorted by FRQ) for a high intensity recombination.

The results demonstrate that all CNNs employing appropriate rearrangement algorithms achieved higher accuracy compared to when using their original data preprocessing methods across various genetic scenarios. This underscores the effectiveness of employing tailored data rearrangement algorithms in enhancing the classification capabilities of CNNs trained on raw genetic data for identifying selective sweeps. Notably, we observed significant differences in the performance of a specific CNN when employing different algorithms within a particular scenario. This discrepancy arises from the dimensional redistribution induced by preprocessing methods in the original genetic matrix. As individual units of the genetic matrix, such as sequences or SNPs, are relocated to different regions based on the rearrangement criteria, the CNN architecture produces markedly varied outcomes on the reordered genetic matrix due to divergent configurations. Furthermore, we noted that specific rearrangement algorithms exerted varying influences on distinct CNNs within the same scenario. As different CNN architectures process the reordered matrix, they generate diverse feature maps owing to their distinct perceptive fields (kernel sizes). As the feature maps pass through the layers, the gap between them widens, resulting in significantly disparate outcomes.

We utilized ANOVA [14], a statistical test for analyzing the differences among means, to identify which dimension (when sorted) is more influential on CNN performance. We measured p-values of the sequence/SNP-based data preprocessing methods for each CNN under a certain scenario. The chosen significance threshold is 0.05. The p-value indicates a statistically significant difference between groups if it is below the threshold, otherwise, representing no statistically significant difference observed between groups. Table 4 shows the details.

For SweepNet, the sequence-based methods demonstrated a statistically significant difference only under the scenario of an old migration, whereas the SNP-based methods exhibited significant differences across all genetic scenarios except for a recent migration. ImaGene showed a significant difference in response to a mild bottleneck when using the sequence-based methods, and achieved p-values below 0.05 with the SNP-based methods in cases of a severe bottleneck and recombination hotspots. Net-1, when applied with the sequence-based methods, did

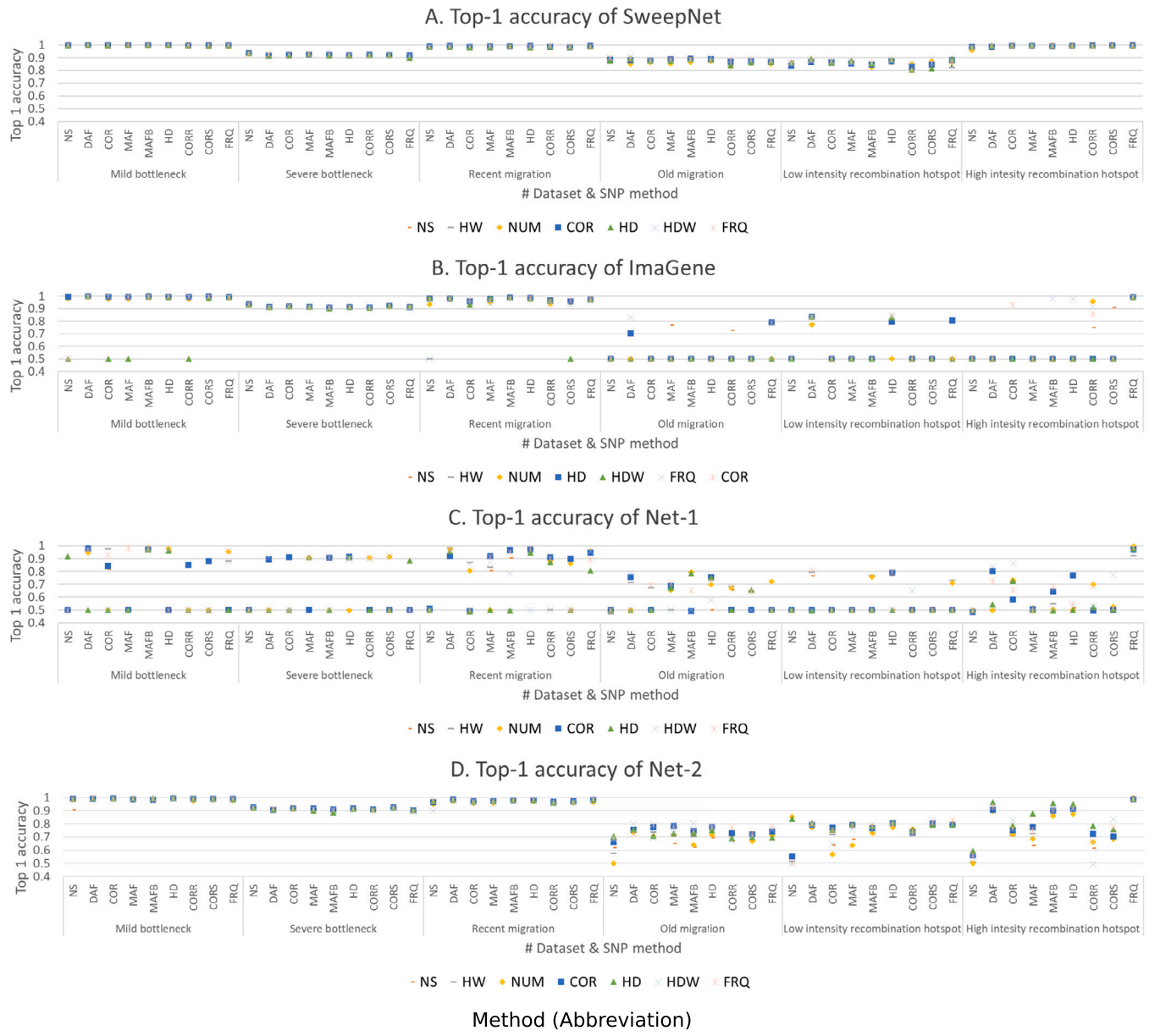

Fig. 3. Accuracy of various CNNs using different preprocessing methods under various genetic scenarios. A. SweepNet. B. ImaGene. C. Net-1. D.Net-2.

Table 4
Comparison of p-values (ANOVA test) of preprocessing methods for each CNN under different datasets.

Mnemonic	Dimension	p-value					
		A mild bottleneck	A severe bottleneck	Recent bottleneck	Old bottleneck	A low intensity recombination hotspot	A high intensity recombination hotspot
SweepNet	Sequence	0.118	0.733	0.753	0.00234	0.559	0.879
	SNP	0.00471	3.03E-12	0.885	8.26E-05	2.63E-05	3.65E-07
ImaGene	Sequence	0.00759	0.978	0.643	0.511	0.824	0.601
	SNP	0.08	2.88E-20	0.0544	0.257	1.73E-12	1.91E-08
Net-1	Sequence	0.968	0.36	0.629	0.513	0.19	0.884
	SNP	1.31E-05	0.106	1.763E-05	0.0158	0.00765	2.65E-14
Net-2	Sequence	0.0476	0.812	0.137	0.0128	0.644	0.92
	SNP	0.229	6.5E-14	9.01E-07	0.000182	7.51E-06	4.22E-24

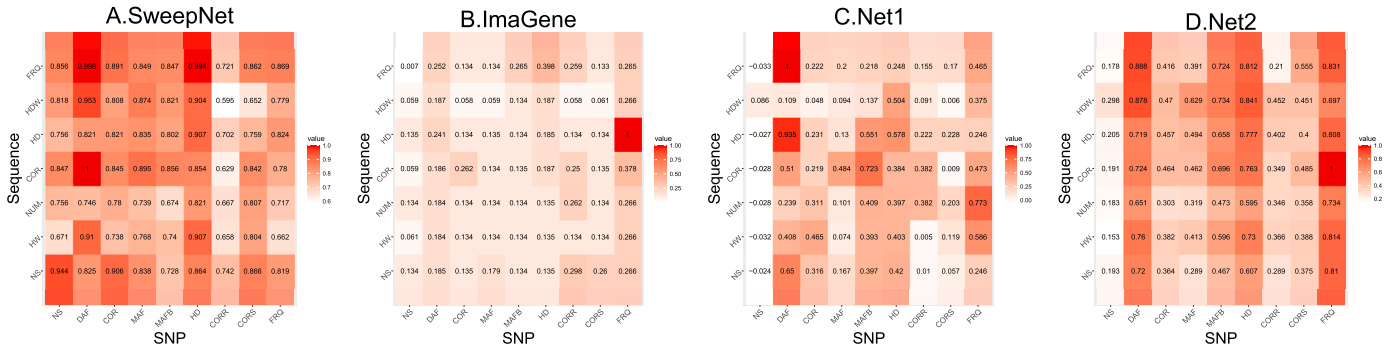

Fig. 4. Normalized results of different preprocessing methods under various genetic scenarios. A. SweepNet. B. ImaGene. C. Net-1. D.Net-2. The datasets from "D1" to "D6" represent the genetic scenario of a mild bottleneck, a severe bottleneck, recent migration, old migration, a low intensity recombination hotspot and a high intensity recombination hotspot respectively.

Table 5
Comparison of the normalized scores (variances) of preprocessing methods for each CNN with all datasets.

Mnemonic	Without data preprocessing			The original combination			The best combination		
	Sequence	SNP	Normalization (Variance)	Sequence	SNP	Normalization (Variance)	Sequence	SNP	Normalization (Variance)
SweepNet	NS	NS	0.944 (0.000955)	HD	DAF	0.953 (0.000896)	COR	DAF	1 (0.000634)
ImaGene	NS	NS	0.134 (0.472)	FRQ	FRQ	0.265 (0.228)	HD	FRQ	1 (0.00379)
Net-1	NS	NS	-0.0245 (1.05)	HD	NS	0.0864 (0.829)	FRQ	DAF	1 (0.116)
Net-2	NS	NS	0.193 (0.322)	NS	NS	0.193 (0.322)	COR	FRQ	1 (0.00278)

not show any significant differences across the evaluated datasets but did reveal significant differences across all scenarios except for a severe bottleneck when using the SNP-based methods. Finally, for Net-2, the sequence-based methods yielded p-values below 0.05 in the context of a mild bottleneck and an old migration, while the SNP-based methods demonstrated statistically significant differences in all scenarios except for a mild bottleneck.

The results clearly indicate that for all CNN architectures, sorting the SNPs has a greater impact on classification performance than sorting the sequences. Compared with the sequences of a genetic matrix, which are placed in genetic datasets in arbitrary order, the order of SNPs represents more specific information. Since a selective sweep influences allele frequency and linkage disequilibrium, rearranging SNPs has higher effect on the traits of the sweep compared with reordering sequences.

4.2. Variance

Given the complexity of real demographic scenarios, we evaluate the collective performance of preprocessing methods across diverse datasets. We quantified the algorithms' performance across all datasets by normalizing variance scores. As shown by Fig. 4, we observed the combination of data rearrangement methods with the best overall performance for each CNN architecture is: a) sequences sorted by COR while SNPs sorted by DAF (SweepNet, Fig. 4a), b) sequences sorted by HD while SNPs sorted by FRQ (ImaGene, Fig. 4b), c) sequences sorted by FRQ while SNPs sorted by DAF (Net-1, Fig. 4c) and d) sequences sorted by COR while SNPs sorted by FRQ (Net-2, Fig. 4d). To assess the effectiveness of data rearrangement algorithms, we listed the scores of each CNN architecture tested with: 1) no preprocessing, 2) the original data preprocessing method, and 3) the best data rearrangement method combination obtained in this work. Table 5 shows this comparison.

For the four best combinations of data preprocessing methods, we observed three different rearrangement methods from each dimension (sequence/SNPs). This result shows that the performance of a certain data rearrangement method is significantly dependent on the CNN architecture. Although no single data preprocessing method consistently outperforms all others across various CNN architectures and genetic scenarios, it is evident that all CNNs benefit from employing data preprocessing methods. Particularly, Net-1 and Net-2 demonstrate noteworthy overall enhancements in terms of variance.

4.3. Evolutionary model misspecification

To assess how data rearrangement methods perform on demographic model misspecification, we utilized CNNs to infer each genetic scenario based on a misspecified evolutionary model for training.

Fig. 5 shows how each CNN architecture performs with the original data preprocessing method under model misspecification. SweepNet (Fig. 5a) demonstrated better resilience between a mild bottleneck and recent migration, with both models achieving at least 0.981 accuracy when tested on each other. However, when trained on a dataset representing a high intensity recombination hotspot, all other inferences only reached around 0.5 accuracy. ImaGene (Fig. 5b) also shows comparable performance on demographic models misspecified as a mild bottleneck or recent migration. For the rest of the effectively trained models, ImaGene is highly sensitive to model misspecification. For instance, although ImaGene achieved 0.992 accuracy when training and testing on a high intensity recombination hotspot, it achieved 0.498 testing on recent migration, which is a relatively distinguishable genetic scenario. Since Net-1 (Fig. 5c) wasn't able to distinguish selective sweeps with its original data preprocessing, it exhibited unsatisfactory performance when the model was misspecified. Although Net-1 achieved a higher accuracy (0.5645) when tested on a mild bottleneck, despite being trained on a dataset representing a recent migration, compared to when it was trained on the correct model (0.5135), it was still not able to practically distinguish between sweeps and neutral regions. Net-2 (Fig. 5d) displayed a similar pattern to SweepNet and ImaGene regarding a mild bottleneck and recent migration. However, for a high intensity recombination hotspot, all models achieved very low accuracy, with the highest being 0.523 when trained with the correct dataset. Interestingly, the model trained on old migration achieved an accuracy of 0.749 even for a severe bottleneck, surpassing the accuracy of 0.62 when tested on the correct genetic scenario (as discussed in the last subsection of this chapter).

Fig. 6 illustrates the performance of each CNN architecture using the overall best combination of data preprocessing methods that comes out

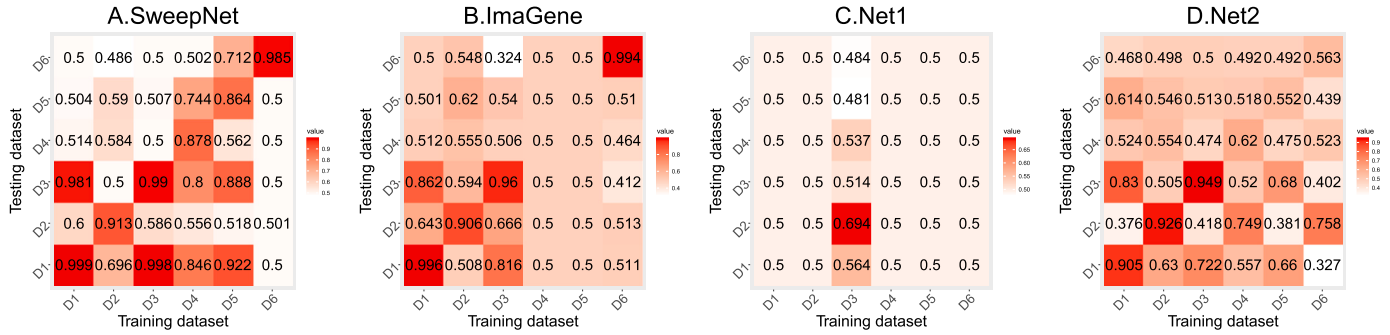

Fig. 5. Classification accuracy of CNNs using the original data preprocessing methods on demographic misspecification. A. SweepNet. B. ImaGene. C. Net-1. D.Net-2. The datasets from "D1" to "D6" represent the genetic scenario of a mild bottleneck, a severe bottleneck, recent migration, old migration, a low intensity recombination hotspot and a high intensity recombination hotspot respectively.

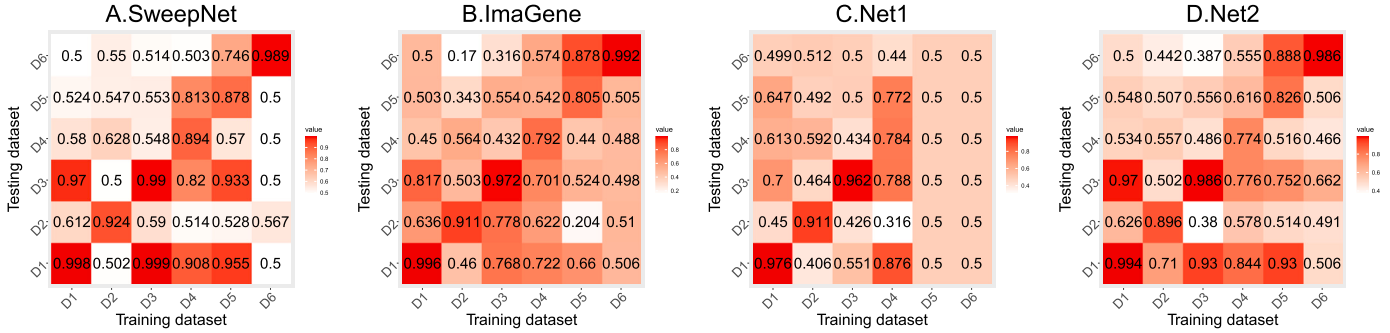

Fig. 6. Classification accuracy of CNNs using the best combination of data preprocessing methods (corresponding to the tile with value '1' (normalized) for each CNN-based method in Fig. 4) on demographic misspecification. A. SweepNet. B. ImaGene. C. Net-1. D.Net-2. The datasets from "D1" to "D6" represent the genetic scenario of a mild bottleneck, a severe bottleneck, recent migration, old migration, a low intensity recombination hotspot and a high intensity recombination hotspot respectively.

from this work in response to model misspecification. SweepNet (illustrated in Fig. 6a) achieved slightly higher accuracy when using the best combination of data preprocessing methods compared to the original method, except in the case of a mild bottleneck (0.9985 versus 0.999). In 10 out of 30 misspecified cases, the best combination yielded up to 0.194 lower accuracy than the original combination. On the other hand, in 20 misspecified cases, the best combination achieved up to 0.069 higher accuracy than the original combination, with the model trained on an old migration and tested on a low intensity recombination hotspot being an example. These results indicate that SweepNet still struggled with demographic misspecification, whether using the original data preprocessing or the best combination. Fig. 6b shows how ImaGene performed on model misspecification using the best recombination of data preprocessing methods. ImaGene achieved up to 0.378 (trained on a low intensity recombination and tested on a high intensity recombination) higher accuracy with the best combination than the original data preprocessing method in 14 misspecified cases. The best combination showed considerable tolerance for misspecification when trained in the case of old migration, even achieving 0.722 (the third best performance) testing accuracy with a mild bottleneck. In 10 out of 20 misspecified cases (the Net-1 with either the original preprocessing method or the best rearrangement method couldn't identify selective sweeps in 10 certain cases), Net-1 (shown by Fig. 6c) achieved up to 0.3765 (training on an old migration and testing on a mild bottleneck) higher accuracy with the best combination of data preprocessing than the original. When the model was trained on an old migration, it obtained 0.8765 testing accuracy on a mild bottleneck, which was even higher than test on the correct scenario (0.7835). As shown by Fig. 6d, Net-2 obtained up to 0.4225 (a high intensity recombination hotspot) accuracy with the best combination of data rearrangement methods higher than with the original data preprocessing on correct model inferences. Although Net-2 could gain increases on accuracy for most misspecified cases, the fact that effective discrimination on identifying selective sweeps remains elusive has yet to change.

In general, most CNNs exhibited similar behavior when facing model misspecification. For instance, they attained relatively high accuracy when a mild bottleneck and recent migration were misspecified for each other, but encountered challenges when a high-intensity recombination hotspot was misspecified as something else. While the best rearrangement algorithm could enhance a CNN's performance for certain misspecified models compared to the original data preprocessing method, it also led to a noticeable decrease in accuracy in some cases. Since primary simulation parameters such as the mutation rate, recombination rate and parameters that model demography are estimated from the data or obtained from literature [20,25,2,27], to some degree, misspecification during the simulation is unavoidable [30]. Mo and Siepel [30] applied an established domain-adaptation technique to mitigate the effects of simulation misspecification, integrating a gradient reversal layer (GRL) [16] originally introduced for image classification into two state-of-the-art deep-learning population genetic methods respectively, 1) SIA [19], identifying positive selection from features of the ancestral recombination graph (ARG), and 2) ReLERNN [3], inferring recombination rates from genotype matrices. Although domain-adaptive SIA and ReLERNN outperform the standard models, their performance deteriorates when the demographic models become complex. In this work, we evaluated how data preprocessing algorithms for CNN-based methods perform when the genetic model is misspecified, observing that preprocessing algorithms can overall increase robustness when the model is heavily misspecified. However, all CNNs faced challenges under specific scenarios of model misspecification. For instance, with the exception of Net-1, all CNNs achieved an accuracy of at least 0.9855 for high-intensity recombination hotspot (Fig. 6) when the genetic model is correctly specified, but none of them performed well when trained with high-intensity recombination hotspot data and tested on other de-

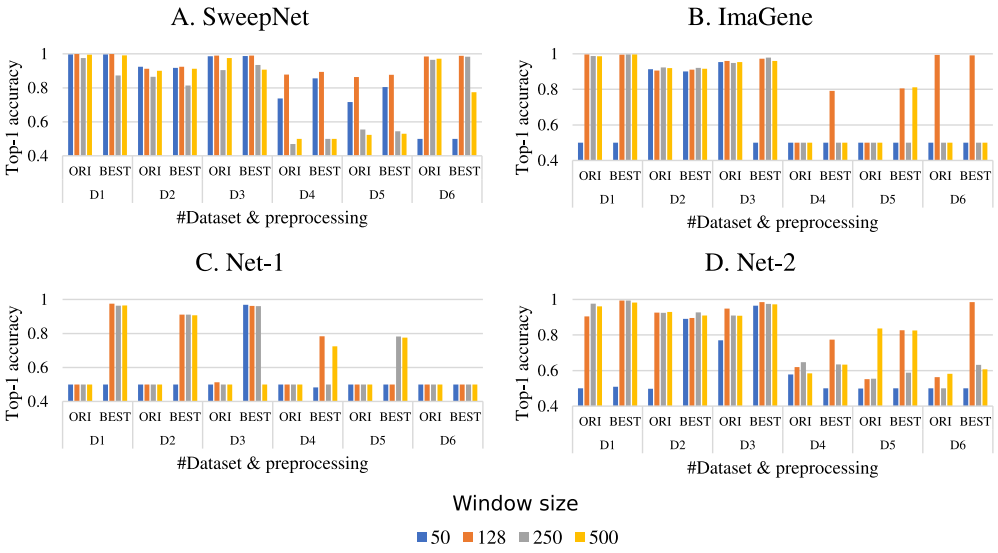

Fig. 7. Results of CNNs using the original data preprocessing methods on varing genomic window sizes. "ORI" and "BEST" represent the original and the best data preprocessing methods, respectively. A. SweepNet. B. ImaGene. C. Net-1. D.Net-2. The datasets from "D1" to "D6" represent the genetic scenario of a mild bottleneck, a severe bottleneck, recent migration, old migration, a low intensity recombination hotspot and a high intensity recombination hotspot respectively.

mographic models. The future exploration on CNN-based data preprocessing methods for this specific problem can be 1) to combine feature vectors that capture deep information from the spatial skews to help CNNs amend misdirection caused by model misspecification, and 2) to train CNN models on datasets with wide range of evolutionary parameters encapsulating the actual parameter values.

4.4. Effect of varying genomic window size

To assess how data rearrangement methods perform on windows of different sizes, we utilized each CNN with the original data processing method and the best combination for inference for window widths of 50, 128, 250 and 500 SNPs.

Fig. 7a displays the results for SweepNet. Generally, for population bottlenecks and recent migration, SweepNet's performance remained consistent across different window sizes with both data rearrangement methods. However, under old migration and a low-intensity recombination hotspot, compared to a window width of 128, all preprocessing methods showed a noticeable decrease in accuracy with other window sizes, particularly for 250 and 500 SNPs, with up to 0.4085 lower accuracy observed. In the case of a high intensity recombination, except that neither the original method nor the best combination can identify selective sweeps for a window size of 50, both rearrangement methods achieved similar accuracy for window sizes of 128, 250, and 500, while observing accuracy of 0.775 with the best combination for a window size of 500. For ImaGene, as depicted in Fig. 7b, similar performance was observed for different window sizes using both the best combination of data preprocessing methods and the original methods across all genetic scenarios. However, in the case of a low intensity recombination hotspot, both the best combination and the original method struggled with window sizes of 50 and 128. Fig. 7c shows the results for Net-1. With the original preprocessing methods, Net-1 struggled to effectively under selective sweeps under most genetic scenarios across various window sizes. Specifically, for a mild bottleneck, Net-1 achieved 0.9645 accuracy with the best preprocessing method for a window size of 500. In the case of a severe bottleneck, Net-1 obtained similar accuracy across all window sizes except for a size of 50. For recent migration, Net-1 achieved at least 0.907 accuracy with different window sizes except for a window size of 50. Regarding recombination hotspots, Net-1 with the best preprocessing methods showed differences of less than 0.776 accuracy for window sizes of 250 and 500. However, it was not effectively trained for window sizes of 50 or 128. Fig. 7d illustrates the performance of Net-2. The best combination of data rearrangement methods showed slightly higher accuracy than the original method for window widths of 128, 250, and 500 under a mild bottleneck, whereas the original method could not identify selective sweeps for a size of 50. In the case of a severe bottleneck, both data preprocessing methods exhibited similar accuracy for all window sizes, with the exception of the original preprocessing method for a window size of 50. For the remaining genetic scenarios, the best combination outperformed the original method in most cases. Specifically, the best combination surpassed the original method for a window size of 128 in the case of a high-intensity recombination hotspot, while the original method encountered significant challenges for all genetic window sizes.

We observed that all the CNN architectures with the best data preprocessing methods generally achieved higher accuracy than using their original preprocessing methods, representing more robustness to the influence that window size exerts. In this work, selective sweeps were simulated at the center of the entire simulated genomic region. The total number of SNPs differs according to the demographic models. For instance, the simulations of a mild bottleneck have around 6,000 SNPs while some simulations of old migration only have hundreds of SNPs. With the window size increasing, it is inevitable that the genomic matrix extracted from datasets of sweeps involved neutral region, contaminating the input data. The classification ability of a CNN is limited when the window size is small, as it contains fewer SNPs, providing only a partial signature of selective sweeps. Moreover, selective sweeps often involve multiple SNPs within a genomic region; due to the effect of genetic hitchhiking, a selective sweep will affect a number of SNPs near the site of the beneficial mutation. Since the extent of genetic hitchhiking is not known, the exact number of SNPs affected by a selective sweep is unknown too. Narrow regions may lack the necessary contextual information for the CNN to accurately identify these patterns. Indeed, data rearrangement methods reorder the sequences/SNPs in a lucid way to reveal the information embodied in genomic matrix, helping CNN detect selective sweeps, even the input data are biased by window sizes. For a large window size that contain neutral region, a proper data rearrangement method is able to dissociate feature of selective sweeps and neutrality by gathering the relevant elements according to a specific criterion. Similarly, it can also highlight the pertinent information hidden in the narrow region. Therefore, the best data preprocessing methods were more robust to different window sizes. An alternative approach for detecting selective sweeps in large genomic regions could be object detection methods. Hamid et al. [18] recently developed a deep learning

object detection method for localizing post-admixture positive selection in human genomes. Surprisingly, the deep learning object detection model also performed stably under demographic misspecification.

5. Conclusion

We introduced a number of data preprocessing methods with the goal of enhancing the accuracy of existing CNNs in identifying selective sweeps across diverse genetic scenarios. All CNNs demonstrate practical benefits from these preprocessing methods, gaining up to 0.449 accuracy improvement when adopting an appreciate algorithm (or a combination of rearrangement algorithms). Additionally, the best combination of algorithms were robust to various window sizes, mitigating the negative affect caused by unproper window sizes (large window sizes may contaminate region of sweeps by introducing neutrality and small sizes contain limited information of sweeps) by reordering the elements of genomic matrix in a lucid way. By analyzing the classification accuracy using the ANOVA test, we observed that sorting the SNPs has more impact on CNNs' performance than sorting the sequences. Although the algorithms (or combinations of rearrangement algorithms) with the best overall performance for CNNs were more robust than the original data preprocessing methods for specific misspecified demography, in general, model misspecification still poses a challenge for raw-data-based CNN methods on detecting selective sweeps. To enable deeper exploration for users to apply rearrangement algorithms to genomic data processing, the standalone package of our implementation can be accessed at https://github.com/Zhaohq96/Genetic-data-rearrangement.

We evaluated data preprocessing methods that treat a sequence or a SNP as a basic unit, keeping the integrity of genomic information as much as possible. The data preprocessing algorithms are based on the theory of genetics or informatics in order to help several existing CNN-based methods identify selective sweeps in the genetic data. For example, columns were sorted by derived allele frequency to cluster SNPs with higher numbers of derived states, based on the theory that the SFS of regions experiencing selective sweeps exhibits a U-shaped distribution, indicating an elevated presence of derived alleles at both high and low frequencies. In this way, the rearrangement algorithms facilitate the classification of selective sweeps by presenting implicit data information more clearly. However, as the alleles in a selective sweep region can be correlated, rearrangement inevitably alters the intrinsic features of the original genomic matrix, thereby distorting signatures of selective sweeps to some extent. Similarly, confounding factors can also generate similar patterns or disrupt the patterns to selective sweeps. For instance, population bottlenecks can reduce genetic diversity, creating patterns that resemble selective sweeps, while recombination hotspots can increase genetic diversity and disrupt the LD patterns typical of selective sweeps. Thus, applying the same algorithm across different datasets may yield markedly different outcomes. Since a selective sweep influences allele frequencies and linkage disequilibrium, rearranging SNPs has more effect on traits of the sweep compared with reordering sequences. Although distortion caused by data rearrangement is inevitable, the proper rearrangement can improve CNN classification by exposing implicit information as well. For instance, Net-2 only achieved accuracy of 0.563 without any data rearrangement under a high intensity recombination hotspot, while it obtained 0.916 accuracy just by sorting the SNPs using DAF (shown on Fig. 3d). A specific data rearrangement algorithm may also yield inconsistent performance across different CNN architectures on the same datasets. This inconsistency arises because CNNs with varying kernel sizes (i.e., receptive fields) applied to the same input image produce different feature maps. As these feature maps are processed through subsequent convolutional layers, the initial differences are amplified, ultimately resulting in divergent outputs. Moreover, the performance of a certain data rearrangement method is not only affected by distinct CNN architectures under the same genomic scenario but also by diverse datasets tested with a specific CNN. For instance, Net-2 (Fig. 3d) achieved the fourth highest accuracy of 0.989 with sorting the sequences based on NUM and sorting the SNPs based on MAFB for a mild bottleneck (the highest is 0.997), while it obtained the fifth lowest accuracy of 0.6395 using the same method for old migration (the highest is 0.802), out of 63 possible combinations of data rearrangement methods. Overall, although data rearrangement distorts the signature of a selective sweep, CNNs can benefit from this distortion when coupled with an effective pixel rearrangement strategy to improve accuracy. Since it is not straightforward how to quantify the extent of this distortion, because the demographic history affects the signature of a selective sweep as well, pixel rearrangement as data preprocessing for CNNs used for the classification of genetic matrices requires meticulous attention on design and assessment.

Declaration of competing interest

Declarations of interest: none.

Acknowledgements

This research was supported by the China Scholarship Council (CSC) (grant No. 202106290021).

Data availability

The datasets are available at https://figshare.com/articles/dataset/SweepNet-Datasets/22194118 and the code can be accessed at https://github.com/Zhaohq96/Genetic-data-rearrangement.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: large-scale machine learning on heterogeneous systems, https://www.tensorflow.org/, 2015, Software available from tensorflow.org.

[2] J.R. Adrion, C.B. Cole, N. Dukler, J.G. Galloway, A.L. Gladstein, G. Gower, C.C. Kyriazis, A.P. Ragsdale, G. Tsambos, F. Baumdicker, et al., A community-maintained standard library of population genetic models, eLife 9 (2020) e54967.

[3] J.R. Adrion, J.G. Galloway, A.D. Kern, Predicting the landscape of recombination using deep learning, Mol. Biol. Evol. 37 (2020) 1790–1808.

[4] N. Alachiotis, P. Pavlidis, RAiSD detects positive selection based on multiple signatures of a selective sweep and SNP vectors, Commun. Biol. 1 (2018) 1–11.

[5] N. Alachiotis, A. Stamatakis, P. Pavlidis, OmegaPlus: a scalable tool for rapid detection of selective sweeps in whole-genome datasets, Bioinformatics 28 (2012) 2274–2275.

[6] M.T. Alam, D.K. De Souza, S. Vinayak, S.M. Griffing, A.C. Poe, N.O. Duah, A. Ghansah, et al., Selective sweeps and genetic lineages of Plasmodium falciparum drug-resistant alleles in Ghana, J. Infect. Dis. 203 (2011) 220–227.

[7] S. van den Belt, H. Zhao, N. Alachiotis, Scalable CNN-based classification of selective sweeps using derived allele frequencies, Bioinformatics 40 (2024) ii29–ii36.

[8] R.M. Cecil, L.A. Sugden, On convolutional neural networks for selection inference: revealing the effect of preprocessing on model learning and the capacity to discover novel patterns, PLoS Comput. Biol. 19 (2023) e1010979.

[9] J. Chan, V. Perrone, J. Spence, P. Jenkins, S. Mathieson, Y. Song, A likelihood-free inference framework for population genetic data using exchangeable neural networks, Adv. Neural Inf. Process. Syst. 31 (2018).

[10] N.G. De Groot, R.E. Bontrop, The HIV-1 pandemic: does the selective sweep in chimpanzees mirror humankind's future? Retrovirology 10 (2013) 1–15.

[11] M. DeGiorgio, C.D. Huber, M.J. Hubisz, I. Hellmann, R. Nielsen, Sweepfinder2: increased sensitivity, robustness and flexibility, Bioinformatics 32 (2016) 1895–1897.

[12] J.A. Endler, Natural Selection in the Wild, vol. 21, Princeton University Press, 1986.

[13] J.C. Fay, C.I. Wu, Hitchhiking under positive Darwinian selection, Genetics 155 (2000) 1405–1413.

[14] R.A. Fisher, Statistical methods for research workers, in: Breakthroughs in Statistics: Methodology and Distribution, Springer, 1970, pp. 66–70.

[15] L. Flagel, Y. Brandvain, D.R. Schrider, The unreasonable effectiveness of convolutional neural networks in population genetic inference, Mol. Biol. Evol. 36 (2019) 220–238.

[16] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: International Conference on Machine Learning, PMLR, 2015, pp. 1180–1189.

[17] B.C. Haller, P.W. Messer, Slim 3: forward genetic simulations beyond the Wright–Fisher model, Mol. Biol. Evol. 36 (2019) 632–637.
[18] I. Hamid, K.L. Korunes, D.R. Schrider, A. Goldberg, Localizing post-admixture adaptive variants with object detection on ancestry-painted chromosomes, Mol. Biol. Evol. 40 (2023) msad074.
[19] H.A. Hejase, Z. Mo, L. Campagna, A. Siepel, A deep-learning approach for inference of selective sweeps from the ancestral recombination graph, Mol. Biol. Evol. 39 (2022) msab332.
[20] C.D. Huber, B.Y. Kim, C.D. Marsden, K.E. Lohmueller, Determining the factors driving selective effects of new nonsynonymous mutations, Proc. Natl. Acad. Sci. 114 (2017) 4465–4470.
[21] M.I. Jordan, T.M. Mitchell, Machine learning: trends, perspectives, and prospects, Science 349 (2015) 255–260.
[22] N.L. Kaplan, R.R. Hudson, C.H. Langley, The "hitchhiking effect" revisited, Genetics 123 (1989) 887–899.
[23] D.G. Kendall, A mathematical approach to seriation, Philos. Trans. R. Soc. Lond. Ser. A, Math. Phys. Sci. 269 (1970) 125–134.
[24] A.D. Kern, D.R. Schrider, diploS/HIC: an updated approach to classifying selective sweeps, G3 Genes Genomes Genet. 8 (2018) 1959–1970.
[25] B.Y. Kim, C.D. Huber, K.E. Lohmueller, Inference of the distribution of selection coefficients for new nonsynonymous mutations using large samples, Genetics 206 (2017) 345–361.
[26] Y. Kim, R. Nielsen, Linkage disequilibrium as a signature of selective sweeps, Genetics 167 (2004) 1513–1524.
[27] M.E. Lauterbur, M.I.A. Cavassim, A.L. Gladstein, G. Gower, N.S. Pope, G. Tsambos, J. Adrion, S. Belsare, A. Biddanda, V. Caudill, et al., Expanding the stdpopsim species catalog, and lessons learned for realistic genome simulations, eLife 12 (2023) RP84874.
[28] M.E. Lauterbur, K. Munch, D. Enard, Versatile detection of diverse selective sweeps with flex-sweep, Mol. Biol. Evol. (2023) msad139.
[29] S.M. Lundberg, S.I. Lee, A unified approach to interpreting model predictions, Adv. Neural Inf. Process. Syst. 30 (2017).
[30] Z. Mo, A. Siepel, Domain-adaptive neural networks improve supervised machine learning based on simulated population genetic data, PLoS Genet. 19 (2023) e1011032.
[31] M.R. Mughal, H. Koch, J. Huang, F. Chiaromonte, M. DeGiorgio, Learning the properties of adaptive regions with functional data analysis, PLoS Genet. 16 (2020) e1008896.
[32] F.A. Nguembang, F. Riguzzi, G. Bertorelle, E. Trucchi, Identification of natural selection in genomic data with deep convolutional neural network, BioData Min. 14 (2021) 1–18.
[33] R. Nielsen, S. Williamson, Y. Kim, M.J. Hubisz, A.G. Clark, C. Bustamante, Genomic scans for selective sweeps using snp data, Genome Res. 15 (2005) 1566–1575.
[34] T. Ohta, The neutral theory is dead. The current significance and standing of neutral and nearly neutral theories, BioEssays 18 (1996) 673–677.
[35] P. Pavlidis, D. Živković, A. Stamatakis, N. Alachiotis, SweeD: likelihood-based detection of selective sweeps in thousands of genomes, Mol. Biol. Evol. 30 (2013) 2224–2234.
[36] D.D. Ray, L. Flagel, D.R. Schrider, Introunet: identifying introgressed alleles via semantic segmentation, PLoS Genet. 20 (2024) e1010657.
[37] D.R. Schrider, A.D. Kern, S/HIC: robust identification of soft and hard sweeps using machine learning, PLoS Genet. 12 (2016) e1005928.
[38] D.R. Schrider, A.D. Kern, Supervised machine learning for population genetics: a new paradigm, Trends Genet. 34 (2018) 301–312.
[39] J.M. Smith, J. Haigh, The hitch-hiking effect of a favourable gene, Genet. Res. 23 (1974) 23–35.
[40] F. Tajima, Statistical method for testing the neutral mutation hypothesis by dna polymorphism, Genetics 123 (1989) 585–595.
[41] T.M. Thompson, From Error-Correcting Codes Through Sphere Packings to Simple Groups, vol. 21, Mathematical Association of America, 1983.
[42] L. Torada, L. Lorenzon, A. Beddis, U. Isildak, L. Pattini, S. Mathieson, M. Fumagalli, ImaGene: a convolutional neural network to quantify natural selection from genomic data, BMC Bioinform. 20 (2019) 1–12.
[43] L. Velazquez-Salinas, S. Zarate, S. Eberl, D.P. Gladue, et al., Positive selection of orf1ab, orf3a, and orf8 genes drives the early evolutionary trends of sars-cov-2 during the 2020 covid-19 pandemic, Front. Microbiol. 11 (2020) 550674.
[44] L.S. Whitehouse, D.R. Schrider, Timesweeper: accurately identifying selective sweeps using population genomic time series, Genetics 224 (2023) iyad084.
[45] H. Zhao, N. Alachiotis, Effective data preprocessing techniques for cnn-based selective sweep detection, in: 2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2023, pp. 793–800.
[46] H. Zhao, P. Pavlidis, N. Alachiotis, SweepNet: a lightweight CNN architecture for the classification of adaptive genomic regions, in: Proceedings of the Platform for Advanced Scientific Computing Conference, 2023, pp. 1–10.
[47] H. Zhao, M. Souilljee, P. Pavlidis, N. Alachiotis, Genome-wide scans for selective sweeps using convolutional neural networks, Bioinformatics 39 (2023) i94–i203.
[48] Z. Zhou, M.M. Rahman Siddiquee, N. Tajbakhsh, J. Liang, Unet++: a nested u-net architecture for medical image segmentation, in: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, in: Proceedings 4, Springer, 2018, pp. 3–11.